Learning Topology and Dynamics of Large Recurrent Neural Networks

Yiyuan She, Yuejia He, and Dapeng Wu, *Fellow, IEEE*

Abstract—**Large-scale recurrent networks have drawn increasing attention recently because of their capabilities in modeling a large variety of real-world phenomena and physical mechanisms. This paper studies how to identify all authentic connections and estimate system parameters of a recurrent network, given a sequence of node observations. This task becomes extremely challenging in modern network applications, because the available observations are usually very noisy and limited, and the associated dynamical system is strongly nonlinear. By formulating the problem as multivariate sparse sigmoidal regression, we develop simple-to-implement network learning algorithms, with rigorous convergence guarantee in theory, for a variety of sparsity-promoting penalty forms. A quantile variant of progressive recurrent network screening is proposed for efficient computation and allows for direct cardinality control of network topology in estimation. Moreover, we investigate recurrent network stability conditions in Lyapunov's sense, and integrate such stability constraints into sparse network learning. Experiments show excellent performance of the proposed algorithms in network topology identification and forecasting.**

Index Terms—**Recurrent networks, topology learning, shrinkage estimation, variable selection, dynamical systems, Lyapunov stability.**

I. INTRODUCTION

There has been an increasing interest in identifying network dynamics and topologies in the emerging scientific discipline of network science. In a dynamical network, the evolution of a node is controlled not only by itself, but also by other nodes. For example, in gene regulatory networks [1], the expression levels of genes influence each other, following some dynamic rules, such that the genes are connected together to form a dynamical system. If the topology and evolution rules of the network are known, we can analyze the regulation between genes or detect unusual behaviors to help diagnose and cure genetic diseases. Similarly, the modeling and estimation of dynamical networks are of great importance in various domains including stock market, brain network and social network [2, 3, 4]. To accurately identify the topology and dynamics underlying those networks, scientists are devoted to developing appropriate mathematical models and corresponding estimation methods.

In the literature, linear dynamical models are commonly used. For example, the human brain connectivity network [5] can be characterized by a set of linear differential equations, where the rate of change of activation/observation of any node is a weighted sum of the activations/observations of its neighbors: $\mathrm{d}x_i/\mathrm{d}t = \sum_{j \neq i} \alpha_{ij} x_j - d_i x_i, 1 \leq i \leq n$. Here α_{ij} provide the connection weights and d_i is the decay rate. Nevertheless, a lot of complex dynamical networks clearly demonstrate *nonlinear* relationships between the nodes. For instance, the strength of influence is unbounded in the previous simple linear combination, but the so-called "**saturation**" effect widely exists in physical/biological systems (neurons, genes, and stocks)—the external influence on a node, no matter how strong the total input activation is, cannot go beyond a certain threshold. To capture the mechanism, nonlinearity must be introduced into the network system: $\mathrm{d}x_i/\mathrm{d}t = l_i\pi(\sum_{j \neq i} \alpha_{ij} x_j + u_i) - d_i x_i + c_i$, where π denotes a nonlinear activation function typically taken to be the sigmoidal function $\pi(\theta) = 1/(1 + e^{-\theta})$. It has a proper shape to resemble many real-world mechanisms and behaviors.

The model description is associated with a *continuous-time recurrent neural network*. The existing feedback loops allow the network to exhibit interesting dynamic temporal behaviors to capture many kinds of relationships. It is also biologically realistic in say modeling the effect of an input spike train. Recurrent networks have been successfully applied to a wide range of problems in bioinformatics, financial market forecast, electric circuits, computer vision, and robotics; see, e.g., [6, 7, 8, 9, 10] among many others.

In practical applications, it is often necessary to include noise contamination: $\mathrm{d}x_i = (l_i\pi(\sum_{j \neq i} \alpha_{ij} x_j + u_i) - d_i x_i + c_i)\,\mathrm{d}t + \sigma\,\mathrm{d}\mathcal{B}_t, 1 \leq i \leq n$, where $\mathcal{B}_t$ stands for an n-dimensional Brownian motion. Among the very many unknown parameters, α_{ij} might be the most important: the zero-nonzero pattern of α_{ij} indicates if there exists a (direct) connection from node j to node i. Collecting all such connections results in a directed graph to describe the node interaction structure.

A fundamental question naturally arises: *Given a sequence of node observations (possibly at very few time points), can one identify all existing connections and estimate all system parameters of a recurrent network?*

This task becomes extremely challenging in modern big network applications, because the available observations are usually very noisy and only available at a relatively small number of time points (say T), due to budget or equipment limitations. One frequently faces applications with n^2 much larger than T. In addition, in this continuous time setting, no analytical formula of the likelihood exists for the stochastic

Yiyuan She is with Department of Statistics, Florida State University, Tallahassee, FL 32306. Yuejia He and Dapeng Wu are with Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611. Correspondence author: Prof. Dapeng Wu, wu@ece.ufl.edu, http://www.wu.ece.ufl.edu.

model, which increases the estimation difficulty even in large samples [11]. Instead of considering multi-step *ad-hoc* procedures, this paper aims at learning the network system *as a whole*. Multivariate statistical techniques will be developed for identifying complete topology and recovering all dynamical parameters. To the best of our knowledge, automatic topology and dynamics learning in large-scale recurrent networks has not been studied before.

In this work, we are interested in networks that are **sparse** in topology. First, many real-world complex dynamical networks indeed have sparse or approximately sparse structures. For example, in regulatory networks, a gene is only regulated by a handful of others [12]. Second, when the number of nodes is large or very large compared with the number of observations, the sparsity assumption reduces the number of model parameters so that the system is estimable. Third, from a philosophical point of view, a sparse network modeling is consistent with the principle of Occam's razor.

Not surprisingly, there is a surge of interest of using compressive sensing techniques for parsimonious network topology learning and dynamics prediction. However, relying on sparsity alone seems to have only limited power in addressing the difficulties of large-scale network learning from big data. To add more prior knowledge and to further reduce the number of effective unknowns, we propose to study how to incorporate structural properties of the network system into learning and estimation, in addition to sparsity. In fact, real-life networks of interest usually demonstrate **asymptotic stability**. This is one of the main reasons why practitioners only perform limited number of measurements of the system, which again provides a type of parsimony or shrinkage in network learning.

In this paper we develop sparse sigmoidal network learning algorithms, with rigorous convergence guarantee in theory, for a variety of sparsity-promoting penalty forms. A quantile variant, the progressive recurrent network screening, is proposed for efficient computation and allows for direct cardinality control of network topology in estimation. Moreover, we investigate recurrent network stability conditions in *Lyapunov*'s sense, and incorporate such stability constraints into sparse network learning. The remaining of this paper is organized as follows. Section II introduces the sigmoidal recurrent network model, and formulates a multivariate regularized problem based on the discrete-time approximate likelihood. Section III proposes a class of sparse network learning algorithms based on the study of sparse sigmoidal regressions. A novel and efficient recurrent network screening (RNS) with theoretical guarantee of convergence is advocated for topology identification in ultra-high dimensions. Section IV investigates asymptotic stability conditions in recurrent systems, resulting in a stable-sparse sigmoidal ($\mathbf{S}^3$) network learning. In Section V, synthetic data experiments and real applications are given. All proof details are left to the Appendices.

II. MODEL FORMULATION

To describe the evolving state of a continuous-time recurrent neural network, one usually defines an associated dynamical system. Ideally, without any randomness, the network behavior can be specified by a set of ordinary differential equations:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = l_i \pi (\sum_{j \neq i} \alpha_{ij} x_j + u_i) - d_i x_i + c_i, \quad i = 1, \cdots, n, \tag{1}$$

where x_i, short for $x_i(t)$, denotes the dynamic process of node i. Throughout the paper, π is the *sigmoidal* activation function

$$\pi(\theta) = \frac{1}{1 + e^{-\theta}}, \tag{2}$$

which is most frequently used in recurrent networks. This function is smooth and strictly increasing. It has a proper shape to resemble many real-world mechanisms and behaviors. Due to noise contamination, a stochastic differential equation model is more realistic

$$\mathrm{d}x_i = (l_i \pi (\sum_{j \neq i} \alpha_{ij} x_j + u_i) - d_i x_i + c_i) \, \mathrm{d}t + \sigma \, \mathrm{d}\mathcal{B}_t, \tag{3}$$

where $\mathcal{B}_t$ is a standard Brownian motion and reflects the stochastic nature of the system. Typically $l_i > 0, d_i > 0$, and in some applications $\alpha_{ii} = 0$ (no self-regulation) is required.

In the sigmoidal recurrent network model, the coefficients α_{ij} characterize the between-node interactions. In particular, $\alpha_{ij} = 0$ indicates that node j does not *directly* influence node i; otherwise, node j regulates node i, and is referred to as a regulator of node i in gene regulatory networks. Such a regulation relationship can be either excitatory (if α_{ij} is positive) or inhibitory (if α_{ij} is negative). In this way, $\mathbf{A} = [\alpha_{ij}]$ is associated with a directed graph that captures the Granger causal relationships between all nodes [13], and the topology of the recurrent network is revealed by the zero-nonzero pattern of the matrix. Therefore, to identify all significant regulation links, it is of great interest to estimate $\mathbf{A}$, given a sequence of node observations (snapshots of the network).

Denote the system state at time t by $\mathbf{x}(t) = [x_1(t) \cdots x_n(t)]^{\mathsf{T}}$ or $\mathbf{x}_t$ (or simply $\mathbf{x}$ when there is no ambiguity.) Define $\mathbf{l} = [l_1, \cdots l_n]^{\mathsf{T}}$, $\mathbf{u} = [u_1, \cdots, u_n]^{\mathsf{T}}$, $\mathbf{c} = [c_1, \cdots, c_n]^{\mathsf{T}}$, $\mathbf{D} = \mathrm{diag}\{d_i\}$, $\mathbf{L} = \mathrm{diag}\{l_i\}$, and $\mathbf{A} = [\alpha_{ij}] = [\boldsymbol{\alpha}_1 \cdots \boldsymbol{\alpha}_n]^{\mathsf{T}} \in \mathbb{R}^{n \times n}$. Then (3) can be represented in a *multivariate* form

$$\mathrm{d}\mathbf{x}_t = (\mathbf{L}\boldsymbol{\pi}(\mathbf{A}\mathbf{x} + \mathbf{u}) - \mathbf{D}\mathbf{x} + \mathbf{c}) \, \mathrm{d}t + \sigma \, \mathrm{d}\boldsymbol{\mathcal{B}}_t, \tag{4}$$

where $\boldsymbol{\mathcal{B}_t}$ is an n-dimensional standard Brownian motion. While this model is specified in continuous time, in practice, the observations are always collected at discrete time points. Estimating the parameters of an SDE model from few discrete observations is very challenging. There rarely exists an analytical expression of the exact likelihood. A common treatment is to discretize (4) and use an approximate likelihood instead. We use the Euler discretization scheme (see, e.g., [14]):

$$\Delta\mathbf{x} = (\mathbf{L}\boldsymbol{\pi}(\mathbf{A}\mathbf{x} + \mathbf{u}) - \mathbf{D}\mathbf{x} + \mathbf{c})\Delta t + \sigma\Delta\boldsymbol{\mathcal{B}}_t.$$

Suppose the system (4) is observed at $T + 1$ time points $t_1, \cdots, t_{T+1}$. Let $\mathbf{x}_s = [x_1(t_s), \cdots, x_n(t_s)]^{\mathsf{T}} \in \mathbb{R}^n$ be the observed values of all n nodes at t_s. Define $\Delta\mathbf{x}_s = (\mathbf{x}_{s+1} - \mathbf{x}_s)$ and $\Delta t_s = (t_{s+1} - t_s)$, $1 \leq s \leq T$. Because $\Delta\boldsymbol{\mathcal{B}_t} \sim \mathcal{N}(\mathbf{0}, \Delta t\mathbf{I})$, the negative conditional log-likelihood

for the discretized model is given by

$$\ell(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{T+1}|\boldsymbol{x}_1) = -\log P(\Delta\boldsymbol{x}_1, \cdots, \Delta\boldsymbol{x}_T|\boldsymbol{x}_1)$$
$$= -\log P(\Delta\boldsymbol{x}_1|\boldsymbol{x}_1) \cdots P(\Delta\boldsymbol{x}_T|\boldsymbol{x}_T)$$
$$= \sum_{s=1}^{T} \|\Delta\boldsymbol{x}_s/\Delta t_s - (\boldsymbol{L}\boldsymbol{\pi}(\boldsymbol{A}\boldsymbol{x}_s + \boldsymbol{u}) - \boldsymbol{D}\boldsymbol{x}_s + \boldsymbol{c})\|_2^2 \Delta t_s/(2\sigma^2)$$
$$+ C(\sigma^2) =: f(\boldsymbol{A}, \boldsymbol{u}, \boldsymbol{l}, \boldsymbol{d}, \boldsymbol{c})/\sigma^2 + C(\sigma^2).$$

$C(\sigma^2$ is a function that depends on σ^2 only. The fitting criterion f is separable in $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_n$. To see this, let $x_{i,s} = x_i(t_s)$ and $\Delta x_{i,s} = x_i(t_{s+1}) - x_i(t_s)$. Then, it is easy to verify that

$$f(\boldsymbol{A}, \boldsymbol{u}, \boldsymbol{l}, \boldsymbol{d}, \boldsymbol{c})$$
$$= \frac{1}{2} \sum_{i=1}^{n} \sum_{s=1}^{T} \left(\frac{\Delta x_{i,s}}{\Delta t_s} - (l_i\pi(\boldsymbol{\alpha}_i^\mathsf{T}\boldsymbol{x}_s + u_i) - d_i x_{i,s} + c_i) \right)^2 \Delta t_s. \tag{5}$$

Conventionally, the unknown parameters can then be estimated by minimizing f. In modern applications, however, the number of available observations (T) is often much smaller than the number of variables to be estimated ($n^2 + 4n$), due to, for example, equipment/budget limitations. Classical MLE methods do not apply well in this high-dimensional setting.

Fortunately, the networks of interest in reality often possess topology sparsity. For example, a stock price may not be directly influenced by all the other stocks in the stock market. A parsimonious network with only significant regulation links is much more interpretable. Statistically speaking, the sparsity in $\boldsymbol{A}$ suggests the necessity of **shrinkage estimation** [15] which can be done by adding penalties and/or constraints to the loss function. The general penalized maximum likelihood problem is

$$\min_{\boldsymbol{A}, \boldsymbol{u}, \boldsymbol{l}, \boldsymbol{d}, \boldsymbol{c}} f(\boldsymbol{A}, \boldsymbol{u}, \boldsymbol{l}, \boldsymbol{d}, \boldsymbol{c}) + \sum_{i,j} P(\alpha_{ij}; \lambda_{ji}) \tag{6}$$

where P is a penalty promoting sparsity and λ_{ji} are regularization parameters. Among the very many possible choices of P, the ℓ_1 penalty is perhaps the popular nowadays to enforce sparsity:

$$P(t; \lambda) = \lambda|t|. \tag{7}$$

It provides a convex relaxation of the ℓ_0 penalty

$$P(t; \lambda) = \frac{\lambda^2}{2} 1_{t \neq 0}. \tag{8}$$

Taking both topology identification and dynamics prediction into consideration, we are particularly interested in the $\ell_0 + \ell_2$ penalty [16]

$$P(t; \lambda, \eta) = \frac{1}{2} \frac{\lambda^2}{1 + \eta} 1_{t \neq 0} + \frac{\eta}{2} t^2, \tag{9}$$

where the ℓ_2 penalty or Tikhonov regularization can effectively deal with large noise and collinearity [17, 18] to enhance estimation accuracy.

The shrinkage estimation problem (6) is however nontrivial. The loss f is nonconvex, π is nonlinear, and the penalty P may be nonconvex or even discrete, let alone the high-dimensionality challenge. Indeed, in many practical networks,

the available observations are usually quite limited and noisy. Effective and efficient learning algorithms are in great need to meet the modern big data challenge.

III. Sparse Sigmoidal Regression for Recurrent Network Learning

A. Univariate-response sigmoidal regression

As analyzed previously, to solve (6), it is sufficient to study a univariate-response learning problem

$$\min_{(\boldsymbol{\beta}, l, \boldsymbol{\gamma})} \frac{1}{2} \sum_{s=1}^{T} w_s \left(y_s - l\pi(\tilde{\boldsymbol{x}}_s^\mathsf{T}\boldsymbol{\beta}) - \boldsymbol{z}_s^\mathsf{T}\boldsymbol{\gamma} \right)^2 + \sum_{k=1}^{n} P(\beta_k, \lambda_k)$$
$$\triangleq F(\boldsymbol{\beta}, l, \boldsymbol{\gamma}) \tag{10}$$

where $\tilde{\boldsymbol{x}}_s, \boldsymbol{z}_s, y_s, w_s$ are given, and $l, \boldsymbol{\gamma}, \boldsymbol{\beta}$ are unknown with $\boldsymbol{\beta}$ desired to be sparse. (10) is the recurrent network learning problem for node i when we set $\tilde{\boldsymbol{x}}_s = [1, \boldsymbol{x}_s]^\mathsf{T}$, $y_s = \Delta x_{i,s}/\Delta t_s$, $\boldsymbol{z}_s = [1, x_{i,s}]^\mathsf{T}$, and $w_s = \Delta t_s$ ($1 \leq s \leq T$) (note that the intercept β_1 is usually subject to no penalization corresponding to $\lambda_1 = 0$). For notational ease, define $\tilde{\boldsymbol{X}} = [\tilde{\boldsymbol{x}}_1, \cdots, \tilde{\boldsymbol{x}}_T]^\mathsf{T}$, $\boldsymbol{y} = [y_1, \cdots, y_T]^\mathsf{T}$, $\boldsymbol{Z} = [\boldsymbol{z}_1, \cdots, \boldsymbol{z}_T]^\mathsf{T}$, $\boldsymbol{\lambda} = [\lambda_k] \in \mathbb{R}^n$, and $\boldsymbol{W} = \text{diag}\{\boldsymbol{w}\} = \text{diag}\{w_1, \cdots, w_s\}$ (to be used in this subsection only, unless otherwise specified).

We propose a simple-to-implement and efficient algorithm to solve the general optimization problem in the form of (10). First define two useful auxiliary functions

$$\xi(\theta, y) = \pi(\theta)(1 - \pi(\theta))(\pi(\theta) - y), \tag{11}$$
$$k_0(\boldsymbol{y}, \boldsymbol{w}) = \max_{1 \leq s \leq T} \frac{w_s}{16} \left(1 + \frac{(1 - 2y_s)^2}{2} \right). \tag{12}$$

The vector versions of π and ξ are defined componentwise: $\boldsymbol{\pi}(\boldsymbol{\theta}) = [\pi(\theta_1), \cdots, \pi(\theta_T)]^\mathsf{T}$ and $\boldsymbol{\xi}(\boldsymbol{\theta}, \boldsymbol{y}) = [\xi(\theta_1, y_1), \cdots, \xi(\theta_T, y_T)]^\mathsf{T}$, and the matrix versions are defined similarly. A prototype algorithm is described as follows, starting with an initial estimate $\boldsymbol{\beta}^{(0)}$, a thresholding rule Θ (an odd, shrinking and nondecreasing function, cf. [19]), and $j = 0$.

repeat
  0) $j \leftarrow j + 1$
  1) $\boldsymbol{\mu}^{(j)} \leftarrow \boldsymbol{\pi}(\tilde{\boldsymbol{X}}\boldsymbol{\beta}^{(j-1)})$
  2) Fit a weighted least-squares model

$$\min_{l, \boldsymbol{\gamma}} \|\boldsymbol{W}^{1/2}(\boldsymbol{y} - [\boldsymbol{\mu}^{(j)} \ \boldsymbol{Z}][l \ \boldsymbol{\gamma}^\mathsf{T}]^\mathsf{T})\|_2^2, \tag{13}$$

  with the corresponding solution denoted by $(l^{(j)}, \boldsymbol{\gamma}^{(j)})$.
  3) Construct $\tilde{\boldsymbol{y}}^{(j)} \leftarrow (\boldsymbol{y} - \boldsymbol{Z}\boldsymbol{\gamma}^{(j)})/l^{(j)}$, $\tilde{\boldsymbol{w}}^{(j)} \leftarrow (l^{(j)})^2\boldsymbol{w}$, $\tilde{\boldsymbol{W}}^{(j)} = \text{diag}\{\tilde{\boldsymbol{w}}^{(j)}\}$. Let $K^{(j)}$ be any constant no less than $k_0(\tilde{\boldsymbol{y}}^{(j)}, \tilde{\boldsymbol{w}}^{(j)})\|\tilde{\boldsymbol{X}}\|_2^2$, where $\|\cdot\|_2$ is the spectral norm.
  4) Update $\boldsymbol{\beta}$ via thresholding:

$$\boldsymbol{\beta}^{(j)} = \Theta(\boldsymbol{\beta}^{(j-1)} - \frac{1}{K^{(j)}} \tilde{\boldsymbol{X}}^\mathsf{T} \tilde{\boldsymbol{W}}^{(j)} \boldsymbol{\xi}(\tilde{\boldsymbol{X}}\boldsymbol{\beta}^{(j-1)}, \tilde{\boldsymbol{y}}^{(j)}); \boldsymbol{\lambda}^{(j)}), \tag{14}$$

  where $\boldsymbol{\lambda}^{(j)} = [\lambda_k^{(j)}]_{n \times 1}$ is a scaled version of $\boldsymbol{\lambda}$ satisfying $P(t; \lambda_k)/K^{(j)} = P(t; \lambda_k^{(j)})$ for any $t \in \mathbb{R}$, $1 \leq k \leq n$.
until convergence

Before proceeding, we give some examples of Θ and $\boldsymbol{\lambda}^{(j)}$. The specific form of the thresholding function Θ in (14) is related to the penalty P through the following formula [16]:

$$P(t; \lambda) - P(0; \lambda)$$
$$= \int_0^{|t|} (\sup\{s : \Theta(s; \lambda) \leq u\} - u)\, \mathrm{d}u + q(t; \lambda) \quad (15)$$

with $q(\cdot; \lambda)$ nonnegative and $q(\Theta(s; \lambda)) = 0$ for all s. The regularization parameter(s) are rescaled at each iteration according to the form of P. Examples include: (i) the ℓ_1 penalty (7), and its associated *soft-thresholding* $\Theta_S(t; \lambda) = \mathrm{sgn}(t)(|t| - \lambda)1_{|t|>\lambda}$, in which case $P(t; \lambda)/K^{(j)} = P(t; \lambda^{(j)}), \forall t$ implies $\lambda^{(j)} = \lambda/K^{(j)}$, (ii) the ℓ_0 penalty (8) and the *hard-thresholding* $\Theta_H(t) = t1_{|t|>\lambda}$, which determines $\lambda^{(j)} = \lambda/\sqrt{K^{(j)}}$, and (iii) the $\ell_0 + \ell_2$ penalty (9) and its associated *hard-ridge thresholding* $\Theta_{HR}(t; \lambda, \eta) = \frac{t}{1+\eta}1_{|t|>\lambda}$, where $\lambda^{(j)} = \frac{\lambda}{K^{(j)}}\sqrt{(\eta + K^{(j)})/(\eta + 1)}, \eta^{(j)} = \eta/K^{(j)}$. The ℓ_p ($0 < p < 1$) penalties, the elastic net penalty, and others [16] are also instances of this framework.

Theorem 1. *Given the objective function in* (10), *suppose Θ and P satisfy* (15) *for some nonnegative $q(t; \lambda)$ with $q(\Theta(t; \lambda)) = 0$ for all t and λ. Then given any initial point $\boldsymbol{\beta}^{(0)}$, with probability 1 the sequence of iterates $(\boldsymbol{\beta}^{(j)}, l^{(j)}, \boldsymbol{\gamma}^{(j)})$ generated by the prototype algorithm satisfies*

$$F(\boldsymbol{\beta}^{(j-1)}, l^{(j-1)}, \boldsymbol{\gamma}^{(j-1)}) \geq F(\boldsymbol{\beta}^{(j)}, l^{(j)}, \boldsymbol{\gamma}^{(j)}). \quad (16)$$

See Appendix A for the proof.

Normalization is usually necessary to make all predictors equally span in the space before penalizing their coefficients using a single regularization parameter λ. We can center and scale all predictor columns but the intercept before calling the algorithm. Alternatively, it is sometimes more convenient to specify λ_k componentwise—e.g., in the ℓ_1 penalty $\sum \lambda_k|\beta_k|$ set $\lambda_k = \lambda \cdot \|\boldsymbol{X}[:, k]\|_2$ for non-intercept coefficients.

B. Cardinality constrained sigmoidal network learning

In the recurrent network setting, one can directly apply the prototype algorithm in Section III-A to solve (6), by updating the columns in $\boldsymbol{B}$ one at a time. On the other hand, a multivariate update form is usually more efficient and convenient in implementation. Moreover, it facilitates integrating stability into network learning (cf. Section IV).

To formulate the loss in a multivariate form, we introduce

$$\boldsymbol{Y} = [y_{i,s}] = [\Delta x_{i,s}/\Delta t_s] \in \mathbb{R}^{T \times n},$$
$$\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T]^\mathsf{T} \in \mathbb{R}^{T \times n},$$
$$\boldsymbol{B} = \boldsymbol{A}^\mathsf{T} = [\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_n] \in \mathbb{R}^{n \times n},$$
$$\boldsymbol{W} = \mathrm{diag}\{\boldsymbol{w}\} = \mathrm{diag}\{\Delta t_1, \cdots, \Delta t_T\}.$$

Then f in (5) can be rewritten as

$$f(\boldsymbol{B}, \boldsymbol{u}, \boldsymbol{l}, \boldsymbol{d}, \boldsymbol{c})$$
$$= \frac{1}{2}\|\boldsymbol{W}^{1/2}\{\boldsymbol{Y} - [\boldsymbol{\pi}(\boldsymbol{XB} + \boldsymbol{1u}^\mathsf{T})\boldsymbol{L} - \boldsymbol{XD} + \boldsymbol{1c}^\mathsf{T}]\}\|_F^2, \quad (17)$$

with $\|\cdot\|_F$ denoting the Frobenius norm, and the objective function to minimize becomes $\|\boldsymbol{W}^{1/2}\{\boldsymbol{Y} - [\boldsymbol{\pi}(\boldsymbol{XB} +$

$\boldsymbol{1u}^\mathsf{T})\boldsymbol{L} - \boldsymbol{XD} + \boldsymbol{1c}^\mathsf{T}]\}\|_F^2 + P(\boldsymbol{B}, \boldsymbol{\Lambda})$.

One of the main issues is to choose a proper penalty form for sparse network learning. Popular sparsity-promoting penalties include ℓ_1, SCAD, ℓ_0, among others. The ℓ_0 penalty (8) is ideal in pursuing a parsimonious solution. However, the matter of parameter tuning cannot be ignored. Most tuning strategies (such as K-fold cross-validation) require computing a solution path for a grid of values of λ, which is quite time-consuming in large network estimation. Rather than applying the ℓ_0 penalty, we propose an ℓ_0 **constrained** sparse network learning

$$\|\boldsymbol{B}\|_0 \leq m, \quad (18)$$

where $\|\cdot\|_0$ denotes the number of nonzero components in a matrix. In contrast to the penalty parameter λ in (8), m is more meaningful and customizable. One can directly specify its value based on prior knowledge or availability of computational resources to have control of the network connection cardinality. To account for collinearity and large noise contamination, we add a further ℓ_2 penalty in $\boldsymbol{B}$, resulting in a new '$\ell_0 + \ell_2$' regularized criterion

$$\min_{\boldsymbol{B}, \boldsymbol{L}, \boldsymbol{D}, \boldsymbol{u}, \boldsymbol{c}} \frac{1}{2}\|\boldsymbol{W}^{1/2}\{\boldsymbol{Y} - [\boldsymbol{\pi}(\boldsymbol{XB} + \boldsymbol{1u}^\mathsf{T})\boldsymbol{L} - \boldsymbol{XD} + \boldsymbol{1c}^\mathsf{T}]\}\|_F^2$$
$$+ \frac{\eta}{2}\|\boldsymbol{B}\|_F^2 =: F_0, \text{ s.t. } \|\boldsymbol{B}\|_0 \leq m. \quad (19)$$

Not only is the cardinality bound m convenient to set, because of the sparsity assumption, but the ℓ_2 shrinkage parameter η is easy to tune. Indeed, η is usually not a very sensitive parameter and does not require a large grid of candidate values. Many researchers simply fix it at a small value (say, $1e - 3$) which can effectively reduce the prediction error (e.g., [18]). Similarly, to handle the possible collinearity between $\boldsymbol{\pi}$ and $\boldsymbol{1}$, we recommend adding mild ridge penalties $\frac{\eta_l}{2}\|\boldsymbol{l}\|_2^2 + \frac{\eta_c}{2}\|\boldsymbol{c}\|_2^2$ in (19) (say, $\eta_l = 1e - 4$ and $\eta_c = 1e - 2$).

The constrained optimization of (19) does not apparently fall into the penalized framework proposed in Section III-A. However, we can adapt the technique to handle it through a *quantile* thresholding operator (as a variant of the hard-ridge thresholding (9)). The detailed recurrent network screening (**RNS**) algorithm is described as follows, where for notational simplicity $\tilde{\boldsymbol{X}} := [\boldsymbol{1} \ \boldsymbol{X}]$, $\tilde{\boldsymbol{B}} := [\boldsymbol{u} \ \boldsymbol{B}^\mathsf{T}]^\mathsf{T}$, and we denote by $\boldsymbol{A}[I, J]$ the submatrix of $\boldsymbol{A}$ consisting of the rows and columns indexed by I and J, respectively.

Theorem 2. *Given any initial point $\tilde{\boldsymbol{B}}^{(0)}$, Algorithm 1 converges in the sense that with probability 1, the function value decreasing property holds:*

$$F_0(\boldsymbol{B}^{(j-1)}, \boldsymbol{L}^{(j-1)}, \boldsymbol{D}^{(j-1)}, \boldsymbol{u}^{(j-1)}, \boldsymbol{c}^{(j-1)})$$
$$\geq F_0(\boldsymbol{B}^{(j)}, \boldsymbol{L}^{(j)}, \boldsymbol{D}^{(j)}, \boldsymbol{u}^{(j)}, \boldsymbol{c}^{(j)}),$$

and all $\boldsymbol{B}^{(j)}$ satisfy $\|\boldsymbol{B}^{(j)}\|_0 \leq m, \ \forall j \geq 1$.

See Appendix B for the proof.

Step 2 can be achieved by solving n weighted least squares. Or, one can re-formulate it as a single-response problem to obtain the solution $(\boldsymbol{l}^{(j)}, \boldsymbol{d}^{(j)}, \boldsymbol{c}^{(j)})$ in one step. The latter way is usually more efficient. When there are ridge penalties imposed on $\boldsymbol{l}$ and $\boldsymbol{c}$, the computation is similar.

Algorithm 1 Recurrent Network Screening (**RNS**)

---

given $\tilde{\boldsymbol{B}}^{(0)}$ (initial estimate), m (cardinality bound), η (ℓ_2 shrinkage parameter).
$j \leftarrow 0$
repeat
   0) $j \leftarrow j + 1$
   1) $\boldsymbol{\mu}^{(j)} \leftarrow \pi(\tilde{\boldsymbol{X}}\tilde{\boldsymbol{B}}^{(j-1)})$
   2) Update $\boldsymbol{L} = \mathrm{diag}\{\boldsymbol{l}\}, \boldsymbol{D} = \mathrm{diag}\{\boldsymbol{d}\}$, and $\boldsymbol{c}$ by fitting a weighted vector least-squares model:

$$\min_{\boldsymbol{L},\boldsymbol{D},\boldsymbol{c}} \|\boldsymbol{W}^{1/2}(\boldsymbol{Y} - [\boldsymbol{\mu}^{(j)}\boldsymbol{L} - \boldsymbol{X}\boldsymbol{D} + \boldsymbol{1}\boldsymbol{c}^{\mathsf{T}}])\|_F^2, \quad (20)$$

with the solution denoted by $\boldsymbol{L}^{(j)} = \mathrm{diag}\{\boldsymbol{l}^{(j)}\} = \mathrm{diag}\{l_1^{(j)}, \cdots, l_n^{(j)}\}$, $\boldsymbol{D}^{(j)} = \mathrm{diag}\{d_1^{(j)}, \cdots, d_n^{(j)}\}$, and $\boldsymbol{c}^{(j)}$. (This amounts to solving n separate weighted least squares problems.)
   3) Construct $\tilde{\boldsymbol{Y}} \leftarrow (\boldsymbol{Y} + \boldsymbol{X}\boldsymbol{D}^{(j)} - \boldsymbol{1}(\boldsymbol{c}^{(j)})^{\mathsf{T}})(\boldsymbol{L}^{(j)})^-$, $\tilde{\boldsymbol{W}} \leftarrow \boldsymbol{w} \cdot (\boldsymbol{l}^{(j)} \circ \boldsymbol{l}^{(j)})^{\mathsf{T}}$, where $^-$ denotes the Moore-Penrose pseudoinverse and $\circ$ is the Hadamard product. Let $K_i^{(j)}$ be any constant no less than $k_0(\tilde{\boldsymbol{Y}}[:,i], \tilde{\boldsymbol{W}}[:,i]])\|\tilde{\boldsymbol{X}}\|_2^2$, and $\boldsymbol{K}^{(j)} \leftarrow \mathrm{diag}\{K_1^{(j)}, \cdots, K_n^{(j)}\}$.
   4) Update $\boldsymbol{B}$ and $\boldsymbol{u}$:
   4.1) $\tilde{\boldsymbol{\Xi}} \leftarrow \tilde{\boldsymbol{B}}^{(j-1)} - \tilde{\boldsymbol{X}}^{\mathsf{T}}\{\tilde{\boldsymbol{W}} \circ \boldsymbol{\xi}(\tilde{\boldsymbol{X}}\tilde{\boldsymbol{B}}^{(j-1)}, \tilde{\boldsymbol{Y}})\}(\boldsymbol{K}^{(j)})^-$, $\boldsymbol{u}^{(j)} \leftarrow (\tilde{\boldsymbol{\Xi}}[1,:])^{\mathsf{T}}$, $\boldsymbol{\Xi} \leftarrow \tilde{\boldsymbol{\Xi}}[2:\text{end},:]$ (the submatrix of $\tilde{\boldsymbol{\Xi}}$ without the first row), $\boldsymbol{\eta}^{(j)} = \eta \cdot (\boldsymbol{1}\boldsymbol{1}^{\mathsf{T}})(\boldsymbol{K}^{(j)})^-$.
   4.2) Perform the hard-ridge thresholding $\boldsymbol{B}^{(j)} \leftarrow \Theta_{HR}(\boldsymbol{\Xi}; \boldsymbol{\zeta}(\boldsymbol{\Xi}), \boldsymbol{\eta}^{(j)})$ (cf. Section III-A for the definition of Θ_{HR}) with an adaptive threshold matrix $\boldsymbol{\zeta}(\boldsymbol{\Xi})$. The entries of $\boldsymbol{\zeta}(\boldsymbol{\Xi})$ are all set to the medium of the mth and the $(m+1)$th largest components of $\mathrm{vec}(|\boldsymbol{\Xi}|)$. See (21) for other variants when certain links must be maintained or forbidden.
   4.3) $\tilde{\boldsymbol{B}}^{(j)} \leftarrow [\boldsymbol{u}^{(j)} \ (\boldsymbol{B}^{(j)})^{\mathsf{T}}]^{\mathsf{T}}$
until the decrease in function value is small
deliver $\hat{\boldsymbol{B}} = \boldsymbol{B}^{(j)}, \hat{\boldsymbol{u}} = \boldsymbol{u}^{(j)}, \hat{\boldsymbol{L}} = \boldsymbol{L}^{(j)}, \hat{\boldsymbol{D}} = \boldsymbol{D}^{(j)}, \hat{\boldsymbol{c}} = \boldsymbol{c}^{(j)}$.

In certain applications self-regulations are not allowed. Then $\boldsymbol{\zeta}(\boldsymbol{\Xi})$ should be the medium of the mth and the $(m+1)$th largest elements of $|\boldsymbol{\Xi} - \boldsymbol{\Xi} \circ \boldsymbol{I}|$ ($\boldsymbol{\Xi}$ in absolute value after excluding its diagonal entries).[1] Other prohibited links can be similarly treated in determining the dynamic threshold. In general, given $\mathcal{T}$ the index set of the links that must be maintained and $\mathcal{F}$ the index set of the links that are forbidden, the threshold is constructed as follows

$$\boldsymbol{\Xi}[i] \leftarrow 0, \forall i \in \mathcal{F}$$
$$\boldsymbol{\zeta}[i] \leftarrow \begin{cases} 0, & \text{if } i \in \mathcal{T} \\ |\boldsymbol{\Xi}[\mathcal{T}^c]|_{(m+1)}, & \text{otherwise,} \end{cases} \quad (21)$$

where $|\boldsymbol{\Xi}[\mathcal{T}^c]|_{(m+1)}$ is the $(m+1)$th largest element (in absolution value) in $\boldsymbol{\Xi}[\mathcal{T}^c]$ (all entries of $\boldsymbol{\Xi}$ except those indexed by $\mathcal{T}$). It is easy to show that the convergence result still holds based on the argument in Appendix B.

---

[1] Throughout the paper $|\boldsymbol{A}|$ is the absolute value of the elements of $\boldsymbol{A}$. That is, for $\boldsymbol{A} = [a_{ij}], |\boldsymbol{A}| = [|a_{ij}|]$.

In implementation, we advocate reducing the network cardinality in a **progressive** manner to lessen greediness: at the jth step, m is replaced by $m(j)$, where $\{m(j)\}$ is a monotone sequence of integers decreasing from $n(n-1)$ (assuming no self-regulation) to m. Empirical study shows that the sigmoidal decay cooling schedule $m(j) = \lceil 2n(n-1)/(1+e^{\alpha j}) \rceil$ with $\alpha = 0.01$, works well.

RNS involves no costly operations at each iteration, and is simple to implement. It runs efficiently for large networks. The RNS estimate can be directly used for analyzing the network topology. One can also use it for screening (in which case a relatively large value of m is specified), followed by a fine network learning algorithm restricted on the screened connections. In either case RNS substantially reduces the search space of candidate links.

IV. STABLE-SPARSE SIGMOIDAL NETWORK LEARNING

For a general dynamical system possibly nonlinear, stability is one of the most fundamental issues [20, 21]. If a system's equilibrium point is *asymptotically stable*, then the perturbed system must approach the equilibrium point as t increases. Moreover, one of the main reasons many real network applications only have limited number of observations measured after perturbation is that the associated dynamical systems stabilize fast (e.g., exponentially fast). This offers another important type of parsimony/shrinkage in network parameter estimation.

To design a new type of regularization, we first investigate stability conditions of sigmoidal recurrent networks in Lyapunov's sense [22, 23]. Then, we develop a stable sparse sigmoidal (S^3) network learning approach.

A. Conditions for asymptotic stability

Recall the multivariate representation of (1)

$$\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}t} = \boldsymbol{L}\boldsymbol{\pi}(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{u}) - \boldsymbol{D}\boldsymbol{x} + \boldsymbol{c}. \quad (22)$$

Because $\boldsymbol{A}$ is sparse and typically singular and the degradation rates d_i are not necessarily identical, in general (22) can not be treated as an instance of the Cohen-Grossberg neural networks [24]. We must first derive its own stability conditions to be considered in network estimation.

Hereinafter, $\boldsymbol{A} \succeq \boldsymbol{A}'$ and $\boldsymbol{A} \succ \boldsymbol{A}'$ stand for the positive semi-definiteness and positive definiteness of $\boldsymbol{A} - \boldsymbol{A}'$, respectively, and the set of eigenvalues of $\boldsymbol{A}$ is $\mathrm{spec}(\boldsymbol{A})$. Our conditions are stated below:

$$\boldsymbol{D} \succ \boldsymbol{0}, \quad (\text{A}1)$$
$$\boldsymbol{L} \succeq \boldsymbol{0}, \quad (\text{A}2)$$
$$\mathrm{Re}(\lambda) < 0 \text{ for any } \lambda \in \mathrm{Spec}(\boldsymbol{L}\boldsymbol{A} - 4\boldsymbol{D}), \quad (\text{A}3a)$$
$$\boldsymbol{L}\boldsymbol{A}/2 + \boldsymbol{A}^{\mathsf{T}}\boldsymbol{L}/2 \prec 4\boldsymbol{D}, \quad (\text{A}3b)$$

Theorem 3. *Suppose* (A1) & (A2) *hold. Then* (A3a) *guarantees the network defined by* (22) *has a unique equilibrium point* $\boldsymbol{x}^*$ *and is globally exponentially stable in the sense that* $\|\boldsymbol{x}(t) - \boldsymbol{x}^*\|_2^2 \leq e^{-\varepsilon t}\|\boldsymbol{x}(0) - \boldsymbol{x}^*\|_2^2$ *for any solution* $\boldsymbol{x}(t)$. *The same conclusion holds if* (A3a) *is replaced by* (A3b).

See Appendix C for the detailed proof.
Figure 1 shows an example of stochastic processes gener-

ated from a stable recurrent network and an unstable recurrent network respectively. In the upper panel, the recurrent network system parameters satisfy the stability condition (A3a), while those in the lower panel violate (A3a). (In both situations, the number of nodes is 10 and the diffusion parameter σ is fixed at 0.5.) The differences between the two models are obvious.

In reality, asymptotically stable systems are commonly observed. The stability conditions reflect structural characteristics. For example, when all l_i are equal and $d_i > 0$, then the *skew-symmetry* of A, i.e., $A = -A^\mathsf{T}$, guarantees asymptotic stability. The information provided by the constrains can assist topology learning. This motivates the design of sparse recurrent network learning with stability.
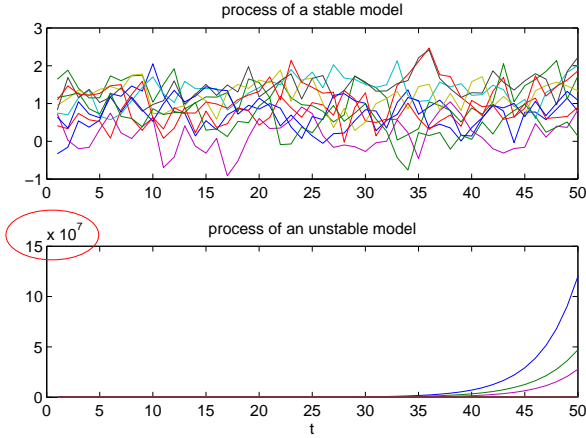

Figure 1: Stochastic processes generated from a stable recurrent network model and an unstable recurrent network model.

B. S^3 network estimation

(A3a) is less restrictive than (A3b). In optimization, imposing (A3a) seems however difficult. We propose

$$\min_{B,L,D,u,c} \frac{1}{2}\|W^{1/2}\{Y - [\pi(XB + 1u^\mathsf{T})L - XD + 1c^\mathsf{T}]\}\|_F^2$$
$$+ P(B, \Lambda) =: F_1,$$
$$\text{s.t. } L \succeq 0, D \succeq 0, (LB^\mathsf{T} + BL)/2 \preceq 4D,$$
(23)

referred to as the Stable-Sparse Sigmoidal (S^3) network learning. The stability constraints are now imposed on B as the transpose of the raw coefficient matrix A. Similar to the discussion of (19), in implementation, we add mild ℓ_2 penalties on l_i and c_i to deal with possible collinearity, and it is common to replace 0 by ϵI with ϵ extremely small.

In this part, we focus on the ℓ_1 penalty $P(B, \Lambda) = \|\Lambda \circ B\|_1$ where matrix Λ usually has the form $\Lambda[k, k'] = \|X[:, k]\|_2$ for any $k, k' \leq n$ (other options are possible.) Introducing the componentwise regularization matrix is helpful when one wants to maintain or forbid certain links based on prior knowledge or preliminary screening. For example, if no self-regulation is allowed, then all diagonal entries of Λ ought to be $+\infty$.

It turns out that one can modify Step 2 and Step 4 of the RNS algorithm to solve (23).

First, the non-negativity of l_i and d_i can be directly incorporated, if the weighted least-squares problem in Step 2 is replaced by the following quadratic programming with non-negativity constraints:

Step 2) Solve $\min_{L,D,c} \|W^{1/2}(Y - [\mu^{(j)}L - XD + 1c^\mathsf{T}])\|_F^2$

$$\text{s.t. } L = \text{diag}\{l_i\} \succeq 0, D = \text{diag}\{d_i\} \succeq 0. \quad (24)$$

A variety of algorithms and packages can be used [25].

Integrating the spectral constraint on B into network learning is much trickier. We modify Step 4 of Algorithm 1 as follows.

Step 4) Update B and u:

4.1) $\tilde{\Xi} \leftarrow \tilde{B}^{(j-1)} - \tilde{X}^\mathsf{T}\{\tilde{W} \circ \xi(\tilde{X}\tilde{B}^{(j-1)}, \tilde{Y})\}(K^{(j)})^-$, $u^{(j)} \leftarrow (\tilde{\Xi}[1,:])^\mathsf{T}$, $\Xi \leftarrow \tilde{\Xi}[2:\text{end},:]$, and $\Lambda^{(j)} = \Lambda \cdot (K^{(j)})^-$.

4.2) Perform the inner loop iterations, starting with $B_3 \leftarrow \Xi$, $C_3 \leftarrow (L^{(j)}B_3^\mathsf{T} + B_3 L^{(j)})/2$, $P = 0$, $Q_B = 0$, $Q_C = 0$, $R = 0$, and the operators $\mathcal{P}^1, \mathcal{P}^2, \mathcal{P}^3$ defined in Lemmas 4-6:

repeat

  i) $B_1 \leftarrow \mathcal{P}^1(B_3 + P; \Lambda^{(j)})$, $C_1 \leftarrow C_3$, $P \leftarrow P + B_3 - B_1$.

  ii) $[B_2, C_2] \leftarrow \mathcal{P}^2(B_1 + Q_B, C_1 + Q_C; L^{(j)})$, $Q_B \leftarrow Q_B + B_1 - B_2$, $Q_C \leftarrow Q_C + C_1 - C_2$.

  iii) $B_3 \leftarrow B_2$, $C_3 \leftarrow \mathcal{P}^3(C_2 + R; D^{(j)})$, $R \leftarrow R + C_2 - C_3$.

until convergence

$B^{(j)} \leftarrow B_3$

4.3) $\tilde{B}^{(j)} \leftarrow [u^{(j)} \ (B^{(j)})^\mathsf{T}]^\mathsf{T}$

Algorithm 1 with such modifications in Step 2 and Step 4 is referred to as the S^3 estimation algorithm.

Theorem 4. *Given any initial point $\tilde{B}^{(0)}$, the S^3 algorithm converges in the sense that the function value decreasing property holds, and furthermore, $B^{(j)}$, $L^{(j)}$, and $D^{(j)}$ satisfy $L^{(j)} \succeq 0, D^{(j)} \succeq 0, (L^{(j)}(B^{(j)})^\mathsf{T} + B^{(j)}L^{(j)})/2 \preceq 4D^{(j)}$ for any $j \geq 1$.*

The proof is given in Appendix D.

We observe that practically, the inner loop converges fast (usually within 100 steps). Moreover, to guarantee the functional value is decreasing, one does not have to run the inner loop till convergence. Although it is possible to apply the stable-sparse estimation directly, we recommend running the screening algorithm (RNS) first, followed by the fine S^3 network learning.

V. EXPERIMENTS

A. Simulation Studies

In this subsection, we conduct synthetic data experiments to demonstrate the performance of the proposed learning framework in recurrent network analysis. An Erdős-Rényi-like scheme of generating system parameters, including a sparse regulation matrix A, is described as follows. Given any node i, the number of its regulators is drawn from a binomial distribution $\mathcal{B}(n-1, \frac{1}{2n})$. The regulator set S_i is

chosen randomly from the rest $(n-1)$ nodes (excluding node i itself). If $j \notin S_i$, $a_{ij} = 0$. Otherwise, a_{ij} follows a mixture of two Gaussians $\mathcal{N}(1.5, 0.1^2)$ ad $\mathcal{N}(-1.5, 0.1^2)$ with probability $1/2$ for each. Then draw random $\boldsymbol{l}, \boldsymbol{u}, \boldsymbol{c}$ from Gaussian distributions (independently) $l_i \sim \mathcal{N}(1.5, 0.1^2)$, $u_i \sim \mathcal{N}(0, 0.1^2)$, $c_i \sim \mathcal{N}(0, 0.1^2)$. Finally $\boldsymbol{d}$ is generated so that the system satisfies the stability condition (A3a).

Topology identification. First, we test the performance of RNS in recurrent network topology identification. We compare it with TSNI [26] and QTIS [27]. TSNI is a popular network learning approach and applies principle component analysis for dimensionality reduction. QTIS is a network screening algorithm based on sparsity-inducing techniques. To avoid the ad-hoc issue of parameter tuning, ROC curves will be used to summarize link detection results in a comprehensive way, in terms of true positive rate (TPR) and false positive rate (FPR).

We simulate two networks according to the scheme introduced early. In the first example we set $n = 10, T = 100$, in the second example $n = 200, T = 500$, and in the third $n = 100, T = 1000$. Given all system parameters, we can call Matlab functions SDE and SDE.SIMULATE to generate continuous-time stochastic processes according to (4). The discrete-time observations are sampled from the stochastic processes with sampling period $\Delta T = 1$.

In this experiment, the number of unknowns in either case is larger than the sample size, especially in Ex.2 which has about **41K** variables but only 500 observations. Given any algorithm, we vary the target cardinality parameter m from 1 to $n(n-1)$, collect all estimates, and compute their associated TPRs and FPRs. The experiment is repeated for 50 times and the averaged rates are shown in the ROC curves in Figure 2. RNS beats TSNI and QTIS by a large margin. In fact, the ROC curve of RNS is *everywhere* above the TSNI curve and the QTIS curve.

System stability. Next, we show the necessity of stable learning in network dynamics forecasting. For simplicity, the ℓ_1 penalization is used. We compare the $\mathbf{S}^3$ network estimation with the approach based on sparse sigmoidal regression in Section III-A (with no stability guarantee), denoted by SigSpar.

We use two network examples (Ex.4 and Ex.5) with $n = 20, 40$ respectively. In each setting, we generate $T = 20$ samples for training, and 200 validation samples for parameter tuning. In this experiment, forecast error at a future time point is the major concern. Suppose $\boldsymbol{x}(T)$, the network snapshot at T, is given. With system parameter estimates obtained, one can simulate a stochastic process $\hat{\boldsymbol{x}}(t)$ $(t \geq T)$ starting with $\boldsymbol{x}(T)$ based on model (4). The forecast error at time point $T + h$ is defined as FE $= \|\boldsymbol{x}(T + h) - \hat{\boldsymbol{x}}(T + h)\|_2^2/n$. Long-term forecasting corresponds to large values of h. We repeat the experiment for 50 times and show the average FE in Table I. The error of SigSpar becomes extremely large as h increases, because there is no stability guarantee of its network estimate, while $\mathbf{S}^3$ has excellent performance even in long-term forecasting.

B. Real data

Yeast gene regulatory network. We use RNS to study the transcriptional regulatory network in the yeast cell cycle. The dataset is publicly available and a detailed description of the microarray experiment is in [28]. Following [29], we focus on the 20 genes whose expression patterns fluctuate in a periodic manner. The dataset contains their expression levels recorded at 18 time points during a cell cycle. In this regulatory (sub)network, five genes have been identified as transcription factors, namely SWI4, HCM1, NDD1, SWI5, ASH1, and 19 regulatory connections from them to target genes have been reported with direct evidence in the literature (cf. the YEASTRACT database at http://yeastract.com/). [29] found 55 connections from the transcription factors to the target genes, of which 14 have biological evidence (and so the true positive rate is $14/19 = 73.7\%$). For a fair comparison, we also let RNS detect 55 connections from the transcription factors to the target genes, and achieved a higher true positive rate 89.5%. A detailed comparison of the identified regulatory connections is shown in Table II.

fMRI data. The resting state fMRI data provided by the ADHD-200 consortium [30] have been preprocessed by the NIAK interface [31]. The dataset we are using has 954 ROIs (regions of interest) and 257 time points. In the experiment, the first 200 observations are for training (and so $T = 200$ and $n = 954$), and the following 57 observations are for testing. We applied RNS to get a network pattern, followed by the $\mathbf{S}^3$ network estimation for stability correction. Table III shows the results averaged over 10 randomly chosen subjects. Our learning algorithm can achieve much lower error than TSNI, and its performance is pretty robust to the choice of m.

APPENDIX A
PROOF OF THEOREM 1

For notation simplicity, we introduce $\mu_s(\boldsymbol{\beta}) := \pi(\tilde{\boldsymbol{x}}_s^\mathsf{T}\boldsymbol{\beta})$. Then $\partial \mu_s(\boldsymbol{\beta})/\partial \boldsymbol{\beta} = \mu_s(\boldsymbol{\beta})(1 - \mu_s(\boldsymbol{\beta}))\tilde{\boldsymbol{x}}_s^\mathsf{T}$, from which it follows that

$$\nabla(\sum_{s=1}^T w_s(y_s - \mu_s(\boldsymbol{\beta}))^2/2) = \tilde{\boldsymbol{X}}^\mathsf{T} \boldsymbol{W} \boldsymbol{\xi}(\tilde{\boldsymbol{X}}\boldsymbol{\beta}, \boldsymbol{y}),$$

where $\boldsymbol{W} = \text{diag}\{w_s\}_{s=1}^T$ and $\boldsymbol{\xi}$ is the function defined in (11) applied componentwise. Moreover, we can compute its Hessian (details omitted)

$$\begin{aligned}
\boldsymbol{H} &= D(\nabla(\sum w_s(y_s - \mu_s(\boldsymbol{\beta}))^2/2)) \\
&= \tilde{\boldsymbol{X}}^\mathsf{T} \boldsymbol{W} \text{diag}\{\mu_s(1 - \mu_s) \\
&\quad [\mu_s(1 - \mu_s) + (\mu_s - y_s)(1 - 2\mu_s)]\}\tilde{\boldsymbol{X}} \\
&\equiv \tilde{\boldsymbol{X}}^\mathsf{T} \boldsymbol{W} \boldsymbol{\Sigma}(\boldsymbol{\beta})\tilde{\boldsymbol{X}}.
\end{aligned}$$

Note that $\boldsymbol{H}$ is not necessarily positive semi-definite.

Let $F_0(\boldsymbol{\beta}) = \frac{1}{2}\sum w_s(y_s - \mu_s(\boldsymbol{\beta}))^2 + \sum P(\beta_k; \lambda)$. Define a surrogate function as $G(\boldsymbol{\beta}, \boldsymbol{\beta}') = \frac{1}{2}\sum w_s(y_s - \mu_s(\boldsymbol{\beta}'))^2 + \sum P(\beta_k'; \lambda) + \frac{K}{2}\|\boldsymbol{\beta} - \boldsymbol{\beta}'\|_2^2 + \frac{1}{2}\sum w_s((y_s - \mu_s(\boldsymbol{\beta}))^2 - (y_s - \mu_s(\boldsymbol{\beta}'))^2) + \sum w_s\xi(\boldsymbol{x}_s^\mathsf{T}\boldsymbol{\beta}, y_s)(\boldsymbol{x}_s^\mathsf{T}\boldsymbol{\beta}' - \boldsymbol{x}_s^\mathsf{T}\boldsymbol{\beta})$. Based on the

(a) Example 1.　　　　(b) Example 2.　　　　(c) Example 3.
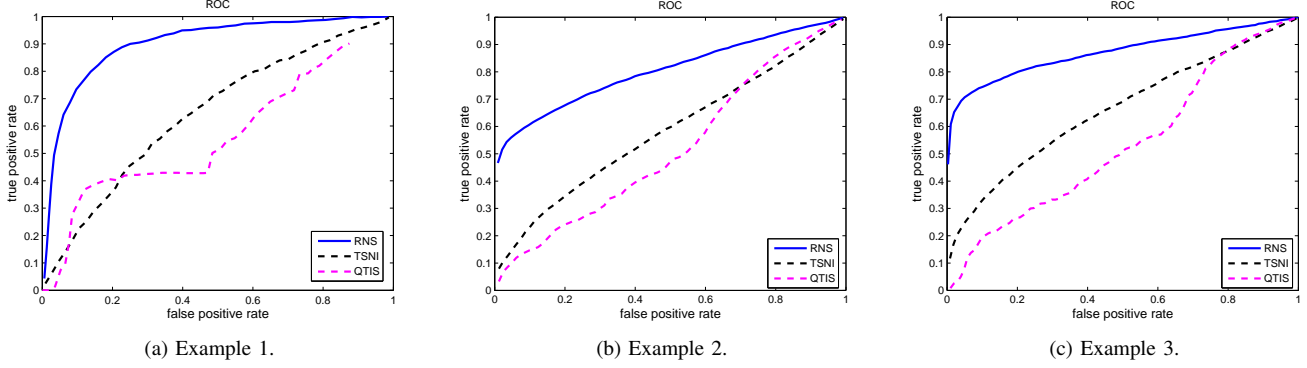
Figure 2: Comparison of ROC curves.

Table I: Forecasting error comparison.

	h	1	5	10	15	20
Ex.4	SigSpar	0.08	3.01	41.95	450.54	4.4×10^3
	$\mathbf{S}^3$	0.04	0.48	1.01	1.39	1.76
Ex.5	SigSpar	0.35	23.76	41.95	7.3×10^3	1.0×10^5
	$\mathbf{S}^3$	0.06	0.70	1.75	2.80	4.04

Table II: Regulatory connections of the Yeast cell cycle subnetwork. '$\times$' stands for the regulatory connections identified by RNS, '_' stands for those by [29], and '$\square$' stands for the confirmed regulatory connections with published evidence in the literature.

	SWI4	HCM1	NDD1	SWI5	ASH1
SWI4					
HCM1	⊠				
NDD1	⊠				
SWI5					
ASH1				⊠	
CLN2	⊠				
SVS1	⊠				
SWE1	⊠				
MNN1	⊠				
CLB6	⊠				
HTA1	⊠				
HTB1	⊠				
HHT1					
CLB4				⊠	
CLB2	⊠		⊠		
CDC20					
SPO12			$\square$		
SIC1				⊠	
CLN3	⊠	⊠		⊠	$\square$
CDC46					

Table III: Forecast error for the fMRI data.

m/n^2	0.5	0.6	0.7	0.8	0.9
TSNI	0.981	1.021	1.044	1.056	1.060
$\mathbf{S}^3$	0.194	0.194	0.194	0.194	0.194

previous calculation, we have

$$\frac{1}{2}\sum w_s((y_s - \mu_s(\boldsymbol{\beta}))^2 - (y_s - \mu_s(\boldsymbol{\beta}'))^2)$$

$$=(\tilde{\boldsymbol{X}}^\mathsf{T}\boldsymbol{W}\boldsymbol{\xi}(\tilde{\boldsymbol{X}}\boldsymbol{\beta},\boldsymbol{y}))^\mathsf{T}(\boldsymbol{\beta} - \boldsymbol{\beta}')$$
$$\quad - \frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}')^\mathsf{T}(\tilde{\boldsymbol{X}}^\mathsf{T}\boldsymbol{W}\boldsymbol{\Sigma}(\theta\boldsymbol{\beta} + (1-\theta)\boldsymbol{\beta}')\tilde{\boldsymbol{X}})(\boldsymbol{\beta} - \boldsymbol{\beta}'),$$

for some $\theta \in [0,1]$. Let $\boldsymbol{\zeta} = \theta\boldsymbol{\beta} + (1-\theta)\boldsymbol{\beta}'$. It follows that

$$\frac{K}{2}\|\boldsymbol{\beta} - \boldsymbol{\beta}'\|_2^2 + \frac{1}{2}\sum w_s((y_s - \mu_s(\boldsymbol{\beta}))^2 - (y_s - \mu_s(\boldsymbol{\beta}'))^2)$$
$$\quad + \sum w_s\xi(\tilde{\boldsymbol{x}}_s^\mathsf{T}\boldsymbol{\beta}, y_s)(\tilde{\boldsymbol{x}}_s^\mathsf{T}\boldsymbol{\beta}' - \tilde{\boldsymbol{x}}_s^\mathsf{T}\boldsymbol{\beta})$$
$$=\frac{1}{2}(\boldsymbol{\beta}' - \boldsymbol{\beta})^\mathsf{T}(K\boldsymbol{I} - \tilde{\boldsymbol{X}}^\mathsf{T}\boldsymbol{\Sigma}(\boldsymbol{\zeta})\boldsymbol{W}\tilde{\boldsymbol{X}})(\boldsymbol{\beta}' - \boldsymbol{\beta})$$
$$\geq \frac{K - \|\tilde{\boldsymbol{X}}\|_2^2\|\boldsymbol{\Sigma}(\boldsymbol{\zeta})\boldsymbol{W}\|_2}{2}\|\boldsymbol{\beta} - \boldsymbol{\beta}'\|_2^2.$$

Because

$$w_s\mu_s(1 - \mu_s)((1 - 2\mu_s)(\mu_s - y_s) + \mu_s(1 - \mu_s))$$

$$\leq \frac{w_s}{4}\left(\frac{1}{2}\left(\frac{1 - 2\mu_s + 2\mu_s - 2y_s}{2}\right)^2 + \frac{1}{4}\right)$$

$$=\frac{w_s}{4}\left(\frac{(1 - 2y_s)^2}{8} + \frac{1}{4}\right),$$

the diagonal entries of $\boldsymbol{W}\boldsymbol{\Sigma}(\boldsymbol{\zeta})$ are uniformly bounded by

$$\max_s \frac{w_s}{4}\left(\frac{(1 - 2y_s)^2}{8} + \frac{1}{4}\right)$$

or $k_0(\boldsymbol{y}, \boldsymbol{w})$ (see (12)). Therefore, choosing $K \geq k_0(\boldsymbol{y}, \boldsymbol{w})\|\tilde{\boldsymbol{X}}\|_2^2$, we have $G(\boldsymbol{\beta}, \boldsymbol{\beta}') = \frac{1}{2}\sum w_s(y_s - \mu_s(\boldsymbol{\beta}'))^2 + \sum P(\beta_k'; \lambda) + \frac{1}{2}(K - \|\tilde{\boldsymbol{X}}\|_2^2\|\boldsymbol{W}\boldsymbol{\Sigma}\|_2^2) \geq F_0(\boldsymbol{\beta})$ for any $\boldsymbol{\beta}, \boldsymbol{\beta}'$.

On the other hand, based on the definition, it is easy to show (details omitted) that given $\boldsymbol{\beta}$, $\min_{\boldsymbol{\beta}'} G(\boldsymbol{\beta}, \boldsymbol{\beta}')$ is equivalent to:

$$\min_{\boldsymbol{\beta}'} \frac{K}{2}\|\boldsymbol{\beta}' - \boldsymbol{\beta} + \frac{1}{K}\tilde{\boldsymbol{X}}^\mathsf{T}\boldsymbol{W}\boldsymbol{\xi}(\tilde{\boldsymbol{X}}\boldsymbol{\beta}, \boldsymbol{y})\|_2^2 + \sum P(\beta_k'; \lambda). \tag{25}$$

Applying Lemma 1 in [16] without requiring uniqueness, there exists a globally optimal solution

$$\boldsymbol{\beta}_o'(\boldsymbol{\beta}) = \Theta(\boldsymbol{\beta} - \tilde{\boldsymbol{X}}^\mathsf{T}\boldsymbol{W}\boldsymbol{\xi}(\tilde{\boldsymbol{X}}\boldsymbol{\beta}, \boldsymbol{y})/K; \lambda'), \tag{26}$$

provided that $P(t; \lambda') = P(t; \lambda)/K$ for any t. In summary, we obtain $F_0(\boldsymbol{\beta}) = G(\boldsymbol{\beta}, \boldsymbol{\beta}) \geq G(\boldsymbol{\beta}, \boldsymbol{\beta}_o'(\boldsymbol{\beta})) \geq F_0(\boldsymbol{\beta}_o'(\boldsymbol{\beta}))$.

We are now in a position to prove (16). In fact, given $\boldsymbol{\gamma}$ and l, the optimization problem

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\sum w_s(y_s - \boldsymbol{z}_s^\mathsf{T}\boldsymbol{\gamma} - l\pi(\tilde{\boldsymbol{x}}_s^\mathsf{T}\boldsymbol{\beta}))^2 + \sum P(\beta_k, \lambda),$$

is equivalent to $\min_{\boldsymbol{\beta}} \frac{1}{2}\sum w_s'(\tilde{y}_s - \pi(\tilde{\boldsymbol{x}}_s^\mathsf{T}\boldsymbol{\beta}))^2 + \sum P(\beta_k, \lambda)$ with $\tilde{y}_s = (y_s - \boldsymbol{z}_s^\mathsf{T}\boldsymbol{\gamma})/l$, $w_s' = l^2 w_s$. (Note that the l obtained from WLS is nonzero with probability 1.) Therefore, the function value decreasing property always holds during the iteration.

APPENDIX B
PROOF OF THEOREM 2

Define a quantile thresholding rule $\Theta^\#(\cdot; m, \eta)$ as a variant of the hard-ridge thresholding rule (9). Given $1 \leq m \leq np$:

$\boldsymbol{A} \in \mathcal{R}^{n\times p} \to \boldsymbol{B} \in \mathcal{R}^{n\times p}$ is defined as follows: $b_{ij} = a_{ij}/(1 + \eta)$ if $|a_{ij}|$ is among the m largest in the set of $\{|a_{ij}| : 1 \leq i \leq n, 1 \leq j \leq p\}$, and $b_{ij} = 0$ otherwise.

Lemma 1. $\hat{\boldsymbol{B}} = \Theta^\#(\boldsymbol{A}; m, \eta)$ *is a globally optimal solution to*

$$\min_{\boldsymbol{B}} f_0(\boldsymbol{A}) = \frac{1}{2}\|\boldsymbol{A} - \boldsymbol{B}\|_F^2 + \frac{\eta}{2}\|\boldsymbol{B}\|_F^2$$
$$s.t. \ \|\boldsymbol{B}\|_0 \leq m.$$

Proof: Let $I \subset \{(i,j)|1 \leq i \leq n, 1 \leq j \leq p\}$ with $|I| = m$. Assuming $\boldsymbol{B}_{I^c} = \boldsymbol{0}$, we get the optimal solution $\hat{\boldsymbol{B}}$ with $\hat{\boldsymbol{B}} = \frac{1}{1+\eta}\boldsymbol{A}_I$. It follows that $f_0(\hat{\boldsymbol{B}}) = \frac{1}{2}\|\boldsymbol{A}\|_F^2 - \frac{1}{2}\sum_{i,j\in I}a_{ij}^2$. Therefore, the quantile thresholding $\Theta^\#(\boldsymbol{A}; m, \eta)$ yields a global minimizer. ∎

Using Lemma 1, we can prove the function value decreasing property; the remaining part follows similar lines of Theorem 1 because of the separability of F.

APPENDIX C
PROOF OF THEOREM 3

Let $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{L}\boldsymbol{\pi}(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{u}) - \boldsymbol{D}\boldsymbol{x} + \boldsymbol{c}$, where $\boldsymbol{x}$ is short for $\boldsymbol{x}(t)$. First, we prove the existence of an equilibrium. It suffices to show that there is a solution to $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{0}$ or $\boldsymbol{x} = \boldsymbol{D}^{-1}\boldsymbol{L}\boldsymbol{\pi}(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{u}) + \boldsymbol{D}^{-1}\boldsymbol{c} =: \varphi(\boldsymbol{x})$. Obviously, the mapping φ is continuous and bounded (say $\|\varphi\|_\infty \leq M$), Brouwer's fixed point theorem [32] indicates the existence of at least one equilibrium in $[-M, M]^n$.

Let $\boldsymbol{x}^*$ be an equilibrium point, i.e., $\boldsymbol{f}(\boldsymbol{x}^*) = \boldsymbol{0}$. Construct a Lyapunov function candidate $V(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^*)^\mathsf{T}\boldsymbol{P}(\boldsymbol{x} - \boldsymbol{x}^*)$ with $\boldsymbol{P}$ positive definite and to be determined. Then

$$\frac{\mathrm{d}V(\boldsymbol{x})}{\mathrm{d}t} = V'(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x})$$
$$=(\boldsymbol{x} - \boldsymbol{x}^*)^\mathsf{T}\boldsymbol{P}\boldsymbol{f}(\boldsymbol{x})$$
$$=(\boldsymbol{x} - \boldsymbol{x}^*)^\mathsf{T}\boldsymbol{P}(\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}^*))$$
$$=-(\boldsymbol{x} - \boldsymbol{x}^*)^\mathsf{T}\boldsymbol{P}\boldsymbol{D}(\boldsymbol{x} - \boldsymbol{x}^*)$$
$$\quad +(\boldsymbol{x} - \boldsymbol{x}^*)^\mathsf{T}\boldsymbol{P}\boldsymbol{L}(\boldsymbol{\pi}(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{u}) - \boldsymbol{\pi}(\boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{u}))$$
$$=-(\boldsymbol{x} - \boldsymbol{x}^*)^\mathsf{T}(\boldsymbol{P}\boldsymbol{D} - \boldsymbol{P}\boldsymbol{L}\boldsymbol{G}\boldsymbol{A})(\boldsymbol{x} - \boldsymbol{x}^*)$$
$$=-(\boldsymbol{x} - \boldsymbol{x}^*)^\mathsf{T}\left(\frac{\boldsymbol{P}\boldsymbol{D} + \boldsymbol{D}\boldsymbol{P}^\mathsf{T}}{2} - \frac{\boldsymbol{P}\boldsymbol{L}\boldsymbol{G}\boldsymbol{A} + \boldsymbol{A}^\mathsf{T}\boldsymbol{G}\boldsymbol{L}\boldsymbol{P}^\mathsf{T}}{2}\right)$$
$$\quad \cdot (\boldsymbol{x} - \boldsymbol{x}^*),$$

where

$$\boldsymbol{G} = \mathrm{diag}\left\{\frac{\pi(\boldsymbol{\alpha}_i^\mathsf{T}\boldsymbol{x} + \boldsymbol{u}) - \pi(\boldsymbol{\alpha}_i^\mathsf{T}\boldsymbol{x}^* + \boldsymbol{u})}{\boldsymbol{\alpha}_i^\mathsf{T}x_i - \boldsymbol{\alpha}_i^\mathsf{T}x_i^*}\right\}.$$

It is easy to verify that $\boldsymbol{G} = \mathrm{diag}\{\pi'(\boldsymbol{\xi})\} \preceq \boldsymbol{I}/4$, and thus $\boldsymbol{P}\boldsymbol{L}\boldsymbol{G}\boldsymbol{A} + \boldsymbol{A}^\mathsf{T}\boldsymbol{G}\boldsymbol{L}\boldsymbol{P}^\mathsf{T} \preceq (\boldsymbol{P}\boldsymbol{L}\boldsymbol{A} + \boldsymbol{A}^\mathsf{T}\boldsymbol{L}\boldsymbol{P}^\mathsf{T})/4$. It is well known [21] that under (A3a), the Lyapunov equation

$$\boldsymbol{P}\left(\boldsymbol{D} - \frac{\boldsymbol{L}\boldsymbol{A}}{4}\right) + \left(\boldsymbol{D} - \frac{\boldsymbol{L}\boldsymbol{A}}{4}\right)^\mathsf{T}\boldsymbol{P}^\mathsf{T} = -\boldsymbol{R} \tag{27}$$

is solvable and uniquely determines a positive definite $\boldsymbol{P}$ for any positive definite $\boldsymbol{R}$. Therefore, V is indeed a Lyapunov function for the nonlinear dynamical system (22). Moreover,

(A3a) implies

$$\frac{dV(\boldsymbol{x})}{dt} \leq -\varepsilon_0 \|\boldsymbol{x} - \boldsymbol{x}^*\|_2^2 \leq -\varepsilon V(\boldsymbol{x})$$

for some $\varepsilon_0, \varepsilon > 0$. By the Lyapunov stability theory—see, e.g., [33] (Chapter 3), (22) must be globally exponentially stable. The uniqueness of the equilibrium is implied by the global exponential stability.

The second result can be shown by setting $\boldsymbol{P} = \boldsymbol{I}$ in (27); details are omitted.

APPENDIX D
PROOF OF THEOREM 4

Based on the proof of Theorem 1, the modified Step 2 does not affect the convergence of function value because at each iteration a global optimum of (24) is obtained. It remains to show that $\tilde{\boldsymbol{B}}^{(j)}$ generated by the modified Step 4 improves $\tilde{\boldsymbol{B}}^{(j-1)}$ in terms of reducing the objective function value, and $\boldsymbol{B}^{(j)}$ obeys the stability condition.

Let $\tilde{\boldsymbol{\Xi}} = \tilde{\boldsymbol{B}}^{(j-1)} - \tilde{\boldsymbol{X}}^{\mathsf{T}} \boldsymbol{W} \boldsymbol{\xi}(\tilde{\boldsymbol{X}} \tilde{\boldsymbol{B}}^{(j-1)}, \tilde{\boldsymbol{Y}})(\boldsymbol{K}^{(j)})^-$, $\boldsymbol{\Lambda}^{(j)} = \boldsymbol{\Lambda} \cdot (\boldsymbol{K}^{(j)})^-$. (The modified Step 2 may result in zeros in $\tilde{\boldsymbol{\Xi}}$ to make the associated activation terms in $\boldsymbol{\pi}(\tilde{\boldsymbol{X}} \tilde{\boldsymbol{B}})$ vanish; using the pseudoinverse can handle the issue and maintain the decreasing property.) Based on the argument of Appendix A, the problem in Step 4 reduces to

$$\min_{\tilde{\boldsymbol{B}} = [\boldsymbol{u}, \boldsymbol{B}^{\mathsf{T}}]^{\mathsf{T}}} \frac{1}{2} \|\tilde{\boldsymbol{B}} - \tilde{\boldsymbol{\Xi}}\|_F^2 + \|\boldsymbol{\Lambda}^{(j)} \circ \boldsymbol{B}\|_1, \tag{28}$$
$$\text{s.t. } (\boldsymbol{L}^{(j)} \boldsymbol{B}^{\mathsf{T}} + \boldsymbol{B} \boldsymbol{L}^{(j)})/2 \preceq 4 \boldsymbol{D}^{(j)}.$$

Therefore it suffices to prove Lemma 2 for any given $\boldsymbol{\Xi}(= \tilde{\boldsymbol{\Xi}}[2 : \text{end}, :])$, $\boldsymbol{\Lambda}^{(j)}$, $\boldsymbol{L}^{(j)} \succeq \boldsymbol{0}$, $\boldsymbol{D}^{(j)} \succeq \boldsymbol{0}$, and $\boldsymbol{B}^{(j-1)}$. (In fact, the lemma holds given any initialization of $\boldsymbol{C}_3$.)

Lemma 2. *For any $j \geq 1$, , $\frac{1}{2} \|\boldsymbol{B}^{(j)} - \boldsymbol{\Xi}\|_F^2 + \|\boldsymbol{\Lambda}^{(j)} \circ \boldsymbol{B}^{(j)}\|_1 \leq \frac{1}{2} \|\boldsymbol{B}^{(j-1)} - \boldsymbol{\Xi}\|_F^2 + \|\boldsymbol{\Lambda}^{(j)} \circ \boldsymbol{B}^{(j-1)}\|_1$, and $(\boldsymbol{L}^{(j)} \boldsymbol{B}^{(j)\mathsf{T}} + \boldsymbol{B}^{(j)} \boldsymbol{L}^{(j)})/2 \preceq 4 \boldsymbol{D}^{(j)}$.*

Proof: Define $f(\boldsymbol{B}) = \frac{1}{2} \|\boldsymbol{B} - \boldsymbol{\Xi}\|_F^2 + \|\boldsymbol{\Lambda} \circ \boldsymbol{B}\|_1$, and $g(\boldsymbol{B}', \boldsymbol{C}', \boldsymbol{B}, \boldsymbol{C}) = \frac{1}{2} \|\boldsymbol{B}' - \boldsymbol{\Xi}\|_F^2 + \|\boldsymbol{\Lambda} \circ \boldsymbol{B}\|_1 + \frac{1}{2} \|\boldsymbol{B} - \boldsymbol{B}'\|_F^2 + \frac{1}{2} \|\boldsymbol{C} - \boldsymbol{C}'\|_F^2 + \langle \boldsymbol{B}' - \boldsymbol{\Xi}, \boldsymbol{B} - \boldsymbol{B}' \rangle$. Then $f(\boldsymbol{B}') = g(\boldsymbol{B}', \boldsymbol{C}', \boldsymbol{B}', \boldsymbol{C}')$ and $g(\boldsymbol{B}', \boldsymbol{C}', \boldsymbol{B}, \boldsymbol{C}) \geq f(\boldsymbol{B})$.

On the other hand, given $(\boldsymbol{B}', \boldsymbol{C}')$, we can write g as a function of $(\boldsymbol{B}, \boldsymbol{C})$: $\frac{1}{2} \|[\boldsymbol{\Xi}, \boldsymbol{C}'] - [\boldsymbol{B}, \boldsymbol{C}]\|_F^2 + \|\boldsymbol{\Lambda} \circ \boldsymbol{B}\|_1$ (up to additive functions of $\boldsymbol{B}'$ and $\boldsymbol{C}'$). Based on Lemma 3, with $\boldsymbol{\Xi}_C = \boldsymbol{C}'$, $g(\boldsymbol{B}', \boldsymbol{C}', \boldsymbol{B}', \boldsymbol{C}') \geq g(\boldsymbol{B}', \boldsymbol{C}', \boldsymbol{B}^o, \boldsymbol{C}^o) \geq f(\boldsymbol{B}^o)$ and $(\boldsymbol{L} \boldsymbol{B}^{o\mathsf{T}} + \boldsymbol{B}^o \boldsymbol{L})/2 = \boldsymbol{C}^o \preceq 4 \boldsymbol{D}$.

Applying the result to the modified Step 4 in Section IV-B yields the desired result. ∎

Lemma 3. *Consider the sequence of $(\boldsymbol{B}_3, \boldsymbol{C}_3)$ generated by the following procedure, with the operators $\mathcal{P}^1$, $\mathcal{P}^2$, $\mathcal{P}^3$ defined in Lemma 4, Lemma 5 and Lemma 6, respectively:*

0) $\boldsymbol{B}_3 \leftarrow \boldsymbol{\Xi}_B$, $\boldsymbol{C}_3 \leftarrow \boldsymbol{\Xi}_C$, $\boldsymbol{P} = \boldsymbol{0}$, $\boldsymbol{Q}_B = \boldsymbol{0}$, $\boldsymbol{Q}_C = \boldsymbol{0}$, $\boldsymbol{R} = \boldsymbol{0}$
   repeat
   1) $\boldsymbol{B}_1 \leftarrow \mathcal{P}^1(\boldsymbol{B}_3 + \boldsymbol{P}; \boldsymbol{\Lambda})$, $\boldsymbol{C}_1 \leftarrow \boldsymbol{C}_3$, $\boldsymbol{P} \leftarrow \boldsymbol{P} + \boldsymbol{B}_3 - \boldsymbol{B}_1$

2) $[\boldsymbol{B}_2, \boldsymbol{C}_2] \leftarrow \mathcal{P}^2(\boldsymbol{B}_1 + \boldsymbol{Q}_B, \boldsymbol{C}_1 + \boldsymbol{Q}_C; \boldsymbol{L})$, $[\boldsymbol{Q}_B, \boldsymbol{Q}_C] \leftarrow [\boldsymbol{Q}_B, \boldsymbol{Q}_C] + [\boldsymbol{B}_1, \boldsymbol{C}_1] - [\boldsymbol{B}_2, \boldsymbol{C}_2]$
   3) $\boldsymbol{B}_3 \leftarrow \boldsymbol{B}_2$, $\boldsymbol{C}_3 \leftarrow \mathcal{P}^3(\boldsymbol{C}_2 + \boldsymbol{R}; \boldsymbol{D})$, $\boldsymbol{R} \leftarrow \boldsymbol{R} + \boldsymbol{C}_2 - \boldsymbol{C}_3$.
   ***until** convergence*

Then, the sequence of iterates converges to a globally optimal solution $(\boldsymbol{B}^o, \boldsymbol{C}^o)$ to

$$\min_{\boldsymbol{B}, \boldsymbol{C}} \frac{1}{2} \|[\boldsymbol{\Xi}_B, \boldsymbol{\Xi}_C] - [\boldsymbol{B}, \boldsymbol{C}]\|_F^2 + \|\boldsymbol{\Lambda} \circ \boldsymbol{B}\|_1 \tag{29}$$
$$\text{s.t. } \boldsymbol{C} = (\boldsymbol{L} \boldsymbol{B}^{\mathsf{T}} + \boldsymbol{B} \boldsymbol{L})/2, \boldsymbol{C} \preceq 4 \boldsymbol{D}.$$

Proof: With the following three lemmas, applying Theorem 3.2 and Theorem 3.3 in [34] yields the strict convergence of the iterates and the global optimality of the limit point. ∎

Lemma 4. *Let $\mathcal{P}^1(\boldsymbol{\Phi})$ be the optimal solution to*

$$\min_{\boldsymbol{B}} \frac{1}{2} \|\boldsymbol{B} - \boldsymbol{\Phi}\|_F^2 + \|\boldsymbol{\Lambda} \circ \boldsymbol{B}\|_1. \tag{30}$$

Then $\mathcal{P}^1(\boldsymbol{\Phi}; \boldsymbol{\Lambda}) = \Theta_S(\boldsymbol{\Phi}; \boldsymbol{\Lambda})$ where Θ_S, applied componentwise on $\boldsymbol{\Phi}$, is the soft-thresholding rule given in Section III-A.

Proof: Apply Lemma 1 in [16]. ∎

Lemma 5. *The optimal solution to*

$$\min_{\boldsymbol{B}, \boldsymbol{C}} \frac{1}{2} \|[\boldsymbol{B} \ \boldsymbol{C}] - [\boldsymbol{\Phi}_B \ \boldsymbol{\Phi}_C]\|_F^2 \ \text{s.t.} \ \boldsymbol{C} = (\boldsymbol{L} \boldsymbol{B}^{\mathsf{T}} + \boldsymbol{B} \boldsymbol{L})/2. \tag{31}$$

is given by $\mathcal{P}^2(\boldsymbol{\Phi}_B, \boldsymbol{\Phi}_C; \boldsymbol{L}) = [\boldsymbol{B}^o, \boldsymbol{C}^o]$ with $\boldsymbol{B}^o = [b_{i,j}^o]$, $b_{i,j}^o = \psi_{i,j} \frac{2 + l_i^2}{2 + l_i^2 + l_j^2} - \psi_{j,i} \frac{l_i l_j}{2 + l_i^2 + l_j^2}$, and $\boldsymbol{C}^o = (\boldsymbol{L} \boldsymbol{B}^{o\mathsf{T}} + \boldsymbol{B}^{o\mathsf{T}} \boldsymbol{L})/2$, where $\boldsymbol{\Psi} = [\psi_{i,j}] = \boldsymbol{\Phi}_B + (\boldsymbol{\Phi}_C + \boldsymbol{\Phi}_C^{\mathsf{T}}) \boldsymbol{L}/2$.

Proof: Let $f(\boldsymbol{B}) = \|\boldsymbol{B} - \boldsymbol{\Phi}_B\|_F^2 / 2 + \|(\boldsymbol{L} \boldsymbol{B}^{\mathsf{T}} + \boldsymbol{B} \boldsymbol{L})/2 - \boldsymbol{\Phi}_C\|_F^2 / 2$. It is not difficult to obtain the gradient (details omitted): $\boldsymbol{B} - \boldsymbol{\Phi}_B + (\boldsymbol{L} \boldsymbol{B}^{\mathsf{T}} \boldsymbol{L} + \boldsymbol{B} \boldsymbol{L}^2)/2 - (\boldsymbol{\Phi}_C + \boldsymbol{\Phi}_C^{\mathsf{T}}) \boldsymbol{L}/2$. The optimal $\boldsymbol{B}$ and $\boldsymbol{C}$ can be evaluated accordingly. ∎

Lemma 6. *Let $\mathcal{P}^3(\boldsymbol{\Phi}; \boldsymbol{D})$ be the optimal solution to*

$$\min_{\boldsymbol{C}} \frac{1}{2} \|\boldsymbol{C} - \boldsymbol{\Phi}\|_F^2 \ \text{s.t.} \ \boldsymbol{C} \ \text{is symmetric and satisfies} \ \boldsymbol{C} \preceq 4 \boldsymbol{D}. \tag{32}$$

Then it is given by $\boldsymbol{U} diag\{\min(s_i, 0)\} \boldsymbol{U}^{\mathsf{T}} + 4 \boldsymbol{D}$, where $\boldsymbol{S} = diag\{s_1, \cdots, s_n\}$ and $\boldsymbol{U}$ are from the spectral decomposition of $(\boldsymbol{\Phi} + \boldsymbol{\Phi}^{\mathsf{T}})/2 - 4 \boldsymbol{D} = \boldsymbol{U} \boldsymbol{S} \boldsymbol{U}^{\mathsf{T}}$.

Proof: Because $\boldsymbol{C}$ is symmetric (but $\boldsymbol{\Phi}$ may not be), we have

$$\|\boldsymbol{C} - \boldsymbol{\Phi}\|_F^2$$
$$= \sum_{1 \leq i < j \leq p} [(c_{ij} - \phi_{ij})^2 + (c_{ij} - \phi_{ji})^2] + \sum_{i=1}^p (c_{ii} - \phi_{ii})^2$$
$$= \sum_{1 \leq i < j \leq p} 2(c_{ij} - \frac{\phi_{ij} + \phi_{ji}}{2})^2 + \sum_{i=1}^p (c_{ii} - \phi_{ii})^2 + const(\boldsymbol{\Phi})$$
$$= \|\boldsymbol{C} - \frac{\boldsymbol{\Phi} + \boldsymbol{\Phi}^{\mathsf{T}}}{2}\|_F^2 + const(\boldsymbol{\Phi}), \tag{33}$$

where $const(\boldsymbol{\Phi})$ is a term that does not depend on $\boldsymbol{C}$.

Therefore, problem (32) is equivalent to

$$\min_{C} \frac{1}{2}\|C - \frac{\Phi + \Phi^{\top}}{2}\|_F^2, \text{ s.t. } C - 4D \preceq 0. \quad (34)$$

The optimality of $\mathcal{P}^3(\Phi; D)$ can then be argued by von Neumann's trace inequality [35, 36]. ∎

REFERENCES

[1] J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins, and T. S. Gardner, "Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles," *PLoS Biol*, vol. 5, no. 1, p. e8, Jan. 2007.

[2] T. Mills and R. Markellos, *The econometric modelling of financial time series*. Cambridge University Press, 2008.

[3] E. Bullmore and O. Sporns, "Complex brain networks: graph theoretical analysis of structural and functional systems," *Nature Reviews Neuroscience*, vol. 10, pp. 186–198, Mar. 2009.

[4] S. Hanneke, W. Fu, and E. P. Xing, "Discrete temporal models of social networks," *Electronic Journal of Statistics*, vol. 4, pp. 585–605, 2010.

[5] A. Roebroeck, E. Formisano, and R. Goebel, "Mapping directed influence over the brain using granger causality and fmri," *Neuroimage*, vol. 25, no. 1, pp. 230–242, 2005.

[6] R. D. Beer, "The dynamics of adaptive behavior: A research program," *Robotics and Autonomous Systems*, vol. 20, no. 2, pp. 257–289, 1997.

[7] J. C. Gallacher and J. M. Fiore, "Continuous time recurrent neural networks: a paradigm for evolvable analog controller circuits," in *National Aerospace and Electronics Conference 2000, Proceedings of the IEEE 2000*. IEEE, 2000, pp. 299–304.

[8] H. Mayer, F. Gomez, D. Wierstra, I. Nagy, A. Knoll, and J. Schmidhuber, "A system for robotic heart surgery that learns to tie knots using recurrent neural networks," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006, pp. 543–548.

[9] R. Xu, D. Wunsch II, and R. Frank, "Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 4, no. 4, pp. 681–692, 2007.

[10] T. T. Vu and J. Vohradsky, "Nonlinear differential equation model for quantification of transcriptional regulation applied to microarray data of saccharomyces cerevisiae," *Nucleic acids research*, vol. 35, no. 1, pp. 279–287, 2007.

[11] S. C. Kou, B. P. Olding, M. Lysy, and J. S. Liu, "A multiresolution method for parameter estimation of diffusion processes," *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1558–1574, 2012.

[12] A. Fujita, J. Sato, H. Garay-Malpartida, R. Yamaguchi, S. Miyano, M. Sogayar, and C. Ferreira, "Modeling gene expression regulatory networks with the sparse vector autoregressive model," *BMC Systems Biology*, vol. 1, no. 1, 2007.

[13] C. W. J. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica*, vol. 37, no. 3, pp. 424–438, 1969.

[14] V. Bally and D. Talay, "The law of the Euler scheme for stochastic differential equations," *Probability theory and related fields*, vol. 104, no. 1, pp. 43–60, 1996.

[15] W. James and C. Stein, "Estimation with quadratic loss," in *Proceedings of the fourth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 1961, 1961, pp. 361–379.

[16] Y. She, "An iterative algorithm for fitting nonconvex penalized generalized linear models with grouped predictors," *Computational Statistics and Data Analysis*, vol. 9, pp. 2976–2990, 2012.

[17] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[18] M. Y. Park and T. Hastie, "L1-regularization path algorithm for generalized linear models," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 4, pp. 659–677, 2007.

[19] Y. She and A. Owen, "Outlier detection using nonconvex penalized regression," *Journal of the American Statistical Association*, vol. 106, no. 494, pp. 626–639, 2011.

[20] S. Sastry, *Nonlinear systems: analysis, stability, and control*. Springer New York, 1999, vol. 10.

[21] J. P. LaSalle and Z. Artstein, *The stability of dynamical systems*. SIAM, 1976, vol. 25.

[22] A. Lyapunov, *The General Problem of the Stability Of Motion*. Kharkov Mathematical Society, 1892.

[23] ——, *The General Problem of the Stability Of Motion*, ser. Control Theory and Applications Series. Taylor & Francis, 1992.

[24] M. A. Cohen and S. Grossberg, "Absolute stability of global pattern formation and parallel memory storage by competitive neural networks," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. SMC-13, no. 5, pp. 815–826, 1983.

[25] P. Gill, W. Murray, and M. Wright, *Practical optimization*. Academic Press, 1981.

[26] M. Bansal, G. Della Gatta, and D. Di Bernardo, "Inference of gene regulatory networks and compound mode of action from time course gene expression profiles," *Bioinformatics*, vol. 22, no. 7, pp. 815–822, 2006.

[27] Y. He, Y. She, and D. Wu, "Stationary-sparse causality network learning," *Journal of Machine Learning Research*, vol. 14, pp. 3073–3104, 2013.

[28] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher, "Comprehensive identification of cell cycle–regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization," *Molecular biology of the cell*,

vol. 9, no. 12, pp. 3273–3297, 1998.

[29] H.-C. Chen, H.-C. Lee, T.-Y. Lin, W.-H. Li, and B.-S. Chen, "Quantitative characterization of the transcriptional regulatory network in the yeast cell cycle," *Bioinformatics*, vol. 20, no. 12, pp. 1914–1927, 2004.

[30] M. P. Milham, D. Fair, M. Mennes, and S. H. Mostofsky, "The adhd-200 consortium: a model to advance the translational potential of neuroimaging in clinical neuroscience," *Frontiers in Systems Neuroscience*, vol. 6, p. 62, 2012.

[31] S. Lavoie-Courchesne, P. Rioux, F. Chouinard-Decorte, T. Sherif, M.-E. Rousseau, S. Das, R. Adalat, J. Doyon, C. Craddock, D. Margulies *et al.*, "Integration of a neuroimaging processing pipeline into a pan-canadian computing grid," in *Journal of Physics: Conference Series*, vol. 341, no. 1.   IOP Publishing, 2012, p. 012032.

[32] L. E. J. Brouwer, "Über abbildung von mannigfaltigkeiten," *Mathematische Annalen*, vol. 71, no. 1, pp. 97–115, 1911.

[33] W. M. Haddad and V. Chellaboina, *Nonlinear dynamical systems and control: a Lyapunov-based approach*. Princeton University Press, 2008.

[34] H. H. Bauschke and P. L. Combettes, "A Dykstra-like algorithm for two monotone operators," *Pacific Journal of Optimization*, vol. 4, no. 3, pp. 383–391, Sep. 2008.

[35] J. von Neumann, "Some matrix inequalities and metrization of matrix space," *Tomsk. Univ. Rev.*, vol. 1, pp. 153–167, 1937.

[36] Y. She, "Reduced rank vector generalized linear models for feature extraction," *Statistics and Its Interface*, vol. 6, pp. 197–209, 2013.

Dapeng Wu (S'98–M'04–SM'06–F'13) received Ph.D. in Electrical and Computer Engineering from Carnegie Mellon University, Pittsburgh, PA, in 2003. He is a professor at the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL. His research interests are in the areas of networking, communications, signal processing, computer vision, machine learning, smart grid, and information and network security.
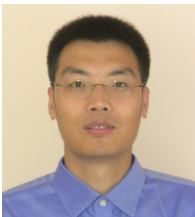
Yiyuan She received B.S. in Mathematics and M.S. in Computer Science from Peking University in 2000 and 2003, respectively, and received Ph.D. in Statistics from Stanford University in 2008. He is currently an Associate Professor in the Department of Statistics at Florida State University. His research interests include high-dimensional statistics, machine learning, multivariate statistics, robust statistics, statistics computing, and network science.

Yuejia He received Ph.D. degree in Electrical and Computer Engineering from University of Florida, Gainesville, FL, in 2013. Her research interest is mainly in large scale network learning.