Robust and efficient network reconstruction in complex system via adaptive signal lasso

Lei Shi,[1,2,*] Jie Hu,[3] Libin Jin,[2] Chen Shen,[4] Huaiyu Tan,[1] and Dalei Yu[5]

[1] *School of Statistics and Mathematics, Yunnan University of Finance and Economics, Kunming 650221, China*
[2] *Interdisciplinary Research Institute of Data Science, Shanghai Lixin University of Accounting and Finance, Shanghai 201209, China*
[3] *School of Management, University of Science and Technology in China, Hefei 230026, China*
[4] *Faculty of Engineering Sciences, Kyushu University, Kasuga-koen, Kasuga-shi, Fukuoka 816-8580, Japan*
[5] *Department of Statistics, School of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an 710049, China*

(Received 24 March 2023; accepted 16 November 2023; published 5 December 2023)

Network reconstruction is a crucial task in understanding and controlling the collective dynamics of complex systems. Most real-world networks exhibit sparse properties, and the connection parameter is a binary signal (0 or 1). Traditional shrinkage methods, such as lasso or compressed sensing (CS), are not suitable for revealing this property. Recently, the signal lasso method was introduced to solve the network reconstruction problem, which was found to be more effective than lasso and CS methods. However, the signal lasso method has a limitation: it cannot accurately classify estimated coefficients that fall between 0 and 1. To address this issue, this paper proposes a method called adaptive signal lasso, which can accurately estimate the signal parameter and uncover the network topology in complex networks with a small number of observations. Our proposed method has at least three advantages: first, it is highly effective in uncovering the network topology and can completely shrink the signal parameter to either 0 or 1, eliminating the unclassified portion in network reconstruction; second, it performs well in both sparse and nonsparse signal scenarios and is robust to noise contamination; third, it only requires the selection of one tuning parameter, reducing computational cost and making it easy to apply. Theoretical properties of this method have been studied, and numerical simulations from linear regression, evolutionary game, and the Kuramoto model are deeply explored. Finally, two real-world examples from human behavioral experiments and the world trade web are used for illustration. It is expected that our proposed method will establish a reliable and uniform framework for estimating signal parameters in complex systems.

DOI: 10.1103/PhysRevResearch.5.043200

I. INTRODUCTION

Complex networks have wide applications and have seen much progress [1–4]. In a complex network, the pattern of node-to-node interaction or network topology is unknown, and uncovering of the network topology based on a series of observable quantities obtained from experiments or observations is important and may play a role in the understanding and controlling of collective dynamics of complex systems [5–7]. Network reconstruction as an inverse problem in network science has been paid much attention recently, such as in the reconstruction of gene networks using expression data [8,9], extraction of various functional networks in the human brain from activation data [10,11], and detection of organizational networks in social science and trade networks in economics [12]. Evolutionary-game-based dynamics have been used to study network reconstruction, where it is possible to observe a series of a small number of discrete quantities [5,6,13,14], in which case the problem can be transformed to a statistical linear model with sparse and high-dimensional properties.

We use two typical examples to illustrate how such signal parameters appear in practice. The first example is a dynamic equation governing the evolution state in a general complex system, which can be written as differential equations [15,16],

$$\dot{\mathbf{x}}_i(t) = \psi_i(\mathbf{x}_i(t), \nu_i) + \sum_{i=1}^{N} a_{ij}\phi_{ij}(\mathbf{x}_i(t), \mathbf{x}_j(t)) + \epsilon_i(t), \quad (1)$$

where $\mathbf{x}_i(t)$ denotes an m-dimensional internal state variable of a system consisting of N dynamic units at time t, where $\psi_i \in R^m$ and $\phi_{ij} \in R^m$, respectively, define the intrinsic and interaction dynamics of the units; $\epsilon_i(t)$ is a dynamic noise term; ν_i is a set of dynamic parameters; and a_{ij} defines the interaction topology and is called an adjacency matrix such that $a_{ij} = 1$ if there is a direct physical interaction from unit j to i, and $a_{ij} = 0$ otherwise. The matrix $A = [a_{ij}]$ completely defines a network with size N, i.e., an abstraction used to model a system that contains discrete interconnected elements. The elements are represented by nodes (also called vertices) and connections by edges. In general, $\mathbf{x}_i(t)$ can be observed as time-series data, but a_{ij} for $i = 1, \ldots, N$ are unknown and must be estimated. It is clear that Eq. (1) can be rewritten as a linear regression model if the functional forms of ψ_i and ϕ_{ij} are known. This model includes synchronization models, oscillator networks, and spreading networks [15].

The second example comes from the evolutionary game on structured populations, where a node represents a player,

[*] lshi@ynufe.edu.cn

and a link indicates that two players have a game relationship. The prisoner's dilemma game (PDG), snowdrift game (SDG), or spatial ultimatum game (SUG) can be used for network reconstruction [5–7,14]. We use the PDG, with temptation to defect T, reward for mutual cooperation R, punishment for mutual defection P, and sucker's payoff S. Thus the payoff matrices can be defined as

$$\mathbf{M_{PDG}} = \begin{pmatrix} R & S \\ T & P \end{pmatrix}, \tag{2}$$

and $T > R > P > S$, where mutual defections are the equilibrium solutions [17–20]. For clarity, this manuscript employs a weak prisoner's dilemma game, where the parameters are adjusted to $T = b$, $P = S = 0$, and $R = 1$. This scaling with $1 \leqslant b < 2$ guarantees the correct payoff hierarchy ($T > R > P > S$) and captures the essential social dilemma [18,21]. In some experimental driving studies, each player can interact with other players by choosing either a cooperator (C) or defector (D) to obtain their payoff, and the procedure is continued for a predetermined number of rounds [22–25]. In a theoretical study, some updating mechanism can be used to generate theoretical data on three types of topologies [26]. Suppose that each player i is either a cooperator (C) or defector (D) with equal probability, which can be written as $s_i = (1, 0)$ or $s_i = (0, 1)$. In a spatial PDG game, player i, say the focal player, acquires its fitness (total payoff) F_i by playing the game with all its connected neighbors,

$$F_i = \sum_{j\in\Omega_i} s_i M_{PDG} s'_j = \sum_{j=1, j\neq i}^{N} a_{ij} P_{ij}, \tag{3}$$

where Ω_i is the set of all connected neighbors of player i, $P_{ij} = s_i M_{PDG} s'_j$, and the prime denotes the transposition of a vector or matrix. Equation (3) can be converted to a linear model, with elements a_{ij} of the adjacency matrix for a network. If $a_{ij} = 1$, then players i and j are connected, and if $a_{ij} = 0$, then they are not. The process can produce time-series data. In each step players can update their strategies using a rule or determine them themselves. Suppose L accessible time is available. Then the model containing the time-series data can be rewritten as

$$Y_i = \Phi_i \tilde{X}_i + e_i, \tag{4}$$

where $Y_i = [F_i(t_1), F_i(t_2), \ldots, F_i(t_L)]'$, $\Phi_i = [P_{ij}(t)] \in R^{L\times(N-1)}$, and $\tilde{X}_i = (a_{i1}, \ldots, a_{i,i-1}, a_{i,i+1}, \ldots, a_{iN})'$, in which the ith connection with itself is removed. The introduction of e_i is due to noises or missing nodes in real applications. Therefore the aim of Eq. (4) is to estimate the elements a_{ij} of the connectivity matrix, which is important for uncovering network structures, such as a possible social network in the social science or an intrinsic scientific relationship in gene-regulatory network reconstruction from the expression data in systems biology.

In the areas of complex systems and applied physics, the compressed sensing (CS) or lasso methods are techniques to estimate a_{ij} and achieve the purpose of network reconstruction [5,6,13], and the lasso has been found to be robust against noises in the reconstruction of sparse signals. Player i and player j are predicted to have a game relationship (connection) if $|\hat{a}_{ij} - 1| \leqslant 0.1$ and no relationship if $|\hat{a}_{ij}| \leqslant 0.1$, where $\hat{a}_{ij}$ is an estimator of a_{ij}. Otherwise, the relationship is not identifiable. Although the CS or lasso method can shrink parameter estimates toward zero under natural sparsity in complex networks, links between nodes cannot be shrunk to a true value of 1, which will decrease estimation accuracy in most cases. For this reason, Shi *et al.* (2021) [14] proposed the signal lasso method to solve the network reconstruction problem and found that it performed better than the lasso and CS methods. However values of $\hat{a}_{ij}$ that fall in the interval (0.1,0.9) cannot be placed in the correct class and leave an unclassified portion in network reconstruction.

In this paper we propose a method called adaptive signal lasso to estimate the signal parameter and uncover the network topology in complex networks with a small number of observations. Our method utilizes a weight on the penalty terms of the traditional signal lasso method. The results show that our method can shrink the parameter to either 0 or 1 completely, greatly improving estimation accuracy. Additionally, our method only requires tuning one parameter within a small range, significantly reducing computational time. The method is also robust to noise and missing nodes due to the inclusion of a least-squares error control term. We conduct simulation and comparison studies using a linear regression model with signal parameters, comparing our method with six existing shrinkage methods. We also validate our reconstruction framework using evolutionary game and synchronization models, and consider three different topological structures, including random (ER), small-world, and scale-free networks. All results demonstrate that our method can achieve high prediction accuracy, remove unclassified subjects, and decrease computational cost compared to the traditional signal lasso method. We also use two real-world examples to illustrate the effectiveness of our method, particularly in detecting signals in nonsparse or dense networks. Our method has potential applications in various fields such as social, economic, physical, biological systems and machine learning research, where recovering the signal is an important task.

II. MOTIVATION AND RELATED WORKS

Consider the general linear regression model,

$$Y = \Phi X + \epsilon, \tag{5}$$

where ϵ is a noise or random error with mean zero and finite variance, $\Phi = [\phi_{ij}]$ is an $n \times p$ matrix, $Y = [y_i]$ is an $n \times 1$ vector, and $X = [X_i]$ is a $p \times 1$ unknown vector. To eliminate the intercept from (5), throughout this paper we center the response and predictor variables so that the mean of the response is zero. We assume the parameter X has a signal property, e.g., the true values of X_j, $j = 1, \ldots, p$, are either 0 or 1. This kind of problem is common in the reconstruction of complex networks to identify a signal as either connected or not [5,6].

The signal lasso method minimizes [14]

$$\frac{1}{2}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p}\phi_{ij}X_j\right)^2 + \lambda_1\sum_{j=1}^{p}|X_j| + \lambda_2\sum_{j=1}^{p}|X_j - 1|, \tag{6}$$

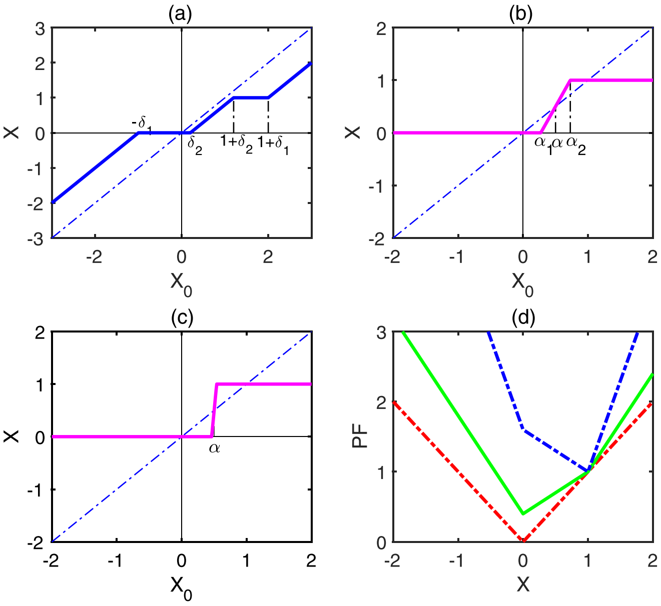

FIG. 1. Solution of X under an orthogonal design in signal lasso and adaptive signal lasso. (a) Signal lasso with $\lambda_1 = 0.6$ and $\lambda_2 = 0.4$, where $\delta_1 = \lambda_1 + \lambda_2$, $\delta_2 = \lambda_1 - \lambda_2$. (b) Adaptive signal lasso with $\lambda_1 = 0.6$ and $\lambda_2 = 1.2$, where $\alpha = \lambda_1/\lambda_2$, $\alpha_1 = \lambda_1/(1 + \lambda_2)$, $\alpha_2 = (1 + \lambda_1)/(1 + \lambda_2)$. (c) Adaptive signal lasso with $\lambda_1 = 6$ and $\lambda_2 = 12$. (d) Penalty function of adaptive signal lasso vs X for different OLS of X when $\lambda_1 = 0.6$ and $\lambda_2 = 1.2$, where the blue line is for $X_{10} = 0.1, X_{20} = 0.9$, the green line for $X_{10} = 0.9, X_{20} = 0.1$, and the red line is a penalty function of the lasso.

where $\lambda_1, \lambda_2 > 0$ are two tuning parameters. The term $\sum_{j=1}^{p} |X_j - 1|$ is added in the penalty term because some elements of X should be 1. When $\lambda_2 = 0$, it reduces to the lasso method [27]. The tuning parameters λ_1 and $\lambda_2 > 0$ must be determined by the dataset through cross-validation. This is a compromise between shrinking terms to 0 and 1, and we expect some elements of X will be close to 0, and others to 1.

If the columns of Φ are orthogonal to each other and $p < n$, denoting the ordinary least-squares estimate by $\hat{X}_0 = \Phi' Y$, the estimator of X in signal lasso is given by (see [14] for detail)

$$\hat{X}_k = \begin{cases} (\hat{X}_{k0} + \delta_1)_-, & \hat{X}_{k0} \leqslant 0, \\ (\hat{X}_{k0} - \delta_2)_+, & 0 < \hat{X}_{k0} \leqslant 1 + \delta_2, \\ \max\{1, \hat{X}_{k0} - \delta_1\}, & \hat{X}_{k0} > 1 + \delta_2, \end{cases} \tag{7}$$

for $k = 1, \ldots, p$, where $\delta_1 = \lambda_1 + \lambda_2$ and $\delta_2 = \lambda_1 - \lambda_2$, and $\hat{X}_{k0}$ and $\hat{X}_k$ are the kth element of $\hat{X}_0$ and $\hat{X}$, respectively; B_+ denotes the positive part of B and means that $B_+ = B$ if $B \geqslant 0$ and 0 otherwise. B_- is similarly defined as the negative part of B.

Figure 1(a) shows the solutions of $\hat{X}$ as a function of $\hat{X}_0$ under an orthogonal design matrix for the signal lasso method, where $\lambda_1 = 0.6$ and $\lambda_2 = 0.4$, and the 45° line of $\hat{X} = \hat{X}_0$ is for reference. The signal lasso method not only shrinks the small values of the parameter to zero but also shrinks large values to 1, and therefore outperforms the lasso and CS methods in network reconstruction for signal parameters [14]. However, this method still has some unsatisfactory aspects in shrinking signal parameters. First, although larger values such that $1 + \delta_2 \leqslant \hat{X}_{k0} \leqslant 1 + \delta_1$ can be shrunk to 1 and values in the interval $-\delta_1 \leqslant \hat{X}_{k0} \leqslant \delta_2$ to 0, values in the interval $(\delta_2, 1 + \delta_2)$ only shift by a constant δ_2, making some parameters unidentifiable.

Compared to the pattern shown in Fig. 1(a), the pattern shown in Figs. 1(b) and 1(c), obtained from our method, is more favorable, as the middle part between 0 and 1 can be shrunk toward two directions. Second, the signal lasso involves two tuning parameters, making the computation costly, even if cross-validation is available. To overcome these weaknesses, we propose an efficient modification by giving a weight in penalized terms of signal lasso. We find that the estimation of parameters in model in (5) can be completely shrunk to 0 or 1, and we only need to select one tuning parameter, which makes the computation fast and greatly improves its accuracy.

III. ADAPTIVE SIGNAL LASSO

A. Method

To deal with the above-mentioned problems, we propose the following penalized least-squares function:

$$L(X|\lambda_1, \lambda_2) = \tfrac{1}{2}||Y - \Phi X||_2^2 + PF(X, \lambda_1, \lambda_2), \tag{8}$$

with the penalty function $PF(X, \lambda_1, \lambda_2)$ given by

$$PF(X, \lambda_1, \lambda_2) = \lambda_1 \sum_{j=1}^{p} \omega_{1j}|X_j| + \lambda_2 \sum_{j=1}^{p} \omega_{2j}|X_j - 1|,$$

where weight coefficients ω_{1j} and ω_{2j} are functions of $\hat{X}_{j0}$, an initial estimator of X_j, which can be an ordinary least-squares estimator for $p < n$, or a ridge estimator for $p > n$. A new estimator of X, defined by

$$\hat{X} = \arg\min_X L(X|\lambda_1, \lambda_2), \tag{9}$$

is called an adaptive signal lasso. For the choice of weights, we expect that the first term of the penalty will have a lower weight, and the second term will have a large weight when $\hat{X}_{k0}$ is close to 1 (similar to when $\hat{X}_{k0}$ is close to 0). Motivated by the adaptive lasso [28], we can choose that

$$\omega_{1k} = |\hat{X}_{k0}|^{-\nu}, \quad \omega_{2k} = |\hat{X}_{k0}|^{\gamma},$$

with $\nu, \gamma > 0$ for $k = 1, \ldots, p$. After comparison and analysis, we find the best candidates for the weights are $\omega_{1k} = 1$ and $\omega_{2k} = |\hat{X}_{k0}|$. This is effective in the analysis and will be used throughout this paper if not specified otherwise (see the Appendix for discussions).

We now give the geometry of adaptive signal lasso in the case of an orthogonal design matrix Φ with $p < n$. After some calculations (see Appendix 2), the solution is given by

$$\hat{X}_k = \begin{cases} \{(1 - \lambda_2)\hat{X}_{k0} + \lambda_1\}_-, & \hat{X}_{k0} \leqslant 0, \\ \{(1 + \lambda_2)\hat{X}_{k0} - \lambda_1\}_+, & 0 < \hat{X}_{k0} \leqslant \alpha_2, \\ \max\{1, (1 - \lambda_2)\hat{X}_{k0} - \lambda_1\}, & \hat{X}_{k0} > \alpha_2, \end{cases} \tag{10}$$

where $\alpha_2 = (1 + \lambda_1)/(1 + \lambda_2)$. In this case, the adaptive signal lasso with $0 < \lambda_1 < \lambda_2 < 1$ will enjoy satisfactory technical properties. Figure 1(b) shows the solutions of $\hat{X}$ as a function of $\hat{X}_0$ for the adaptive signal lasso in the special case of (10), where the main difference from the signal lasso occurs in the interval (0,1). It is of interest to see that the values of $\hat{X}_0$ in (α_1, α) will be shrunk toward 0, while the values of $\hat{X}_0 \in (\alpha, \alpha_2)$ will be shrunk toward 1, where $\alpha = \lambda_1/\lambda_2$ and $\alpha_1 = \lambda_1/(1+\lambda_2)$. The line $X = (1+\lambda_2)X_0 - \lambda_1$ has a slope of $1+\lambda_2$, which indicates a kind of shrinkage strength. When λ_2 increases to 12 and we keep $\alpha = 0.5$ (which means $\lambda_1 = 6$), the pattern given in Fig. 1(c) shows that almost all parameter estimation in the middle part can be shrunk almost to 0 or 1.

Thus we reparametrize λ_1 and λ_2 by $\lambda = \lambda_2$ and $\alpha = \lambda_1/\lambda_2$ and rewrite the penalty function by

$$PF(X,\lambda,\alpha) = \lambda\left\{\alpha\sum_{j=1}^{p}|X_j| + \sum_{j=1}^{p}|\hat{X}_{j0}||X_j - 1|\right\}. \tag{11}$$

It can be easily proved from Eq. (10) that when α is fixed and let $\lambda \to +\infty$, then we have

$$\hat{X}_k \to \begin{cases} 1, & \hat{X}_{k0} > \alpha, \\ 0, & \hat{X}_{k0} < \alpha, \end{cases} \tag{12}$$

since $\alpha_2 \to \alpha$ and $\alpha_1 \to \alpha$ in these scenarios. The parameter can be assigned randomly as 0 or 1 if $\hat{X}_{k0} = \alpha$. This result indicates that if we set λ large enough, the estimators from the adaptive signal lasso can be completely shrunk to either 0 or 1, thus removing the unidentified set that will be presented in the signal lasso method. Another advantage is that we only need to select the tuning parameter α, which dramatically reduces the computation time compared with the signal lasso. In addition, the range for selecting α can be set in a small interval such as (0.2,0.8), since a smaller or larger α is inappropriate in practice. Figure 1(d) shows the functional form of the penalty to show how the penalties behave for different values of $\hat{X}_0$. For example, when $\hat{X}_{10} = 0.1, \hat{X}_{20} = 0.9$, the adaptive penalty function tends to shrink toward (0,1), while it shrinks toward (1,0) when $\hat{X}_{10} = 0.9, \hat{X}_{20} = 0.1$.

Figure 2 shows constraint regions $PF(X) = c$ for different shrink estimation methods. Figure 2(a) shows the graphs for adaptive lasso methods. Figure 2(b) shows a signal lasso and Figs. 1(c) and 1(d) show an adaptive signal lasso with different values of $(\hat{X}_{10}, \hat{X}_{20})$. For example, for $\hat{X}_0 = (0.9, 0.1)$, the contours for the adaptive signal lasso will be centered at $X = (1, 0)$, gradually converging to it when c becomes small. It is of interest to see that the shape of constraint regions $PF(X) = c$ in the adaptive signal lasso varies by target point.

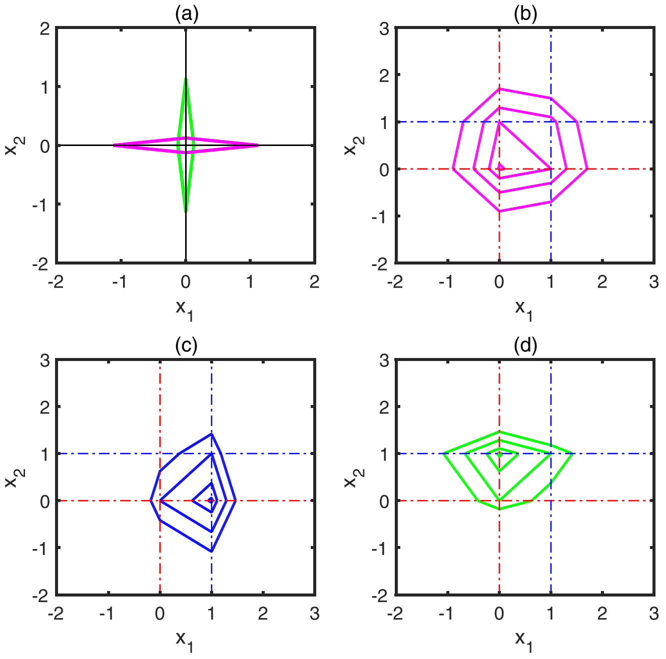

FIG. 2. Constraint regions of $PF(x) = c$ for some constant c under four estimation methods in the two-dimensional case ($p = 2$). (a) Adaptive lasso estimation with penalty function $PF(x) = \sum_{j=1}^{2}|\hat{X}_{j0}|^{-1}|x_j|$, $c = 1$, where the green line is for the case of $\hat{X}_0 = (0.1, 0.9)$ and the magenta line for $\hat{X}_0 = (0.9, 0.1)$. (b) Signal lasso with penalty function $PF(x) = \lambda_1\sum_{j=1}^{2}|x_j| + \lambda_2\sum_{j=1}^{2}|x_j - 1|$, $\lambda_1 = 0.6, \lambda_2 = 0.4, c = 1.7, 1.3, 1, 0.82$. (c) Adaptive signal lasso with penalty function $PF(x) = \lambda_1\sum_{j=1}^{2}|x_j| + \lambda_2\sum_{j=1}^{2}|X_{j0}||x_j - 1|$, $\lambda_1 = 0.6, \lambda_2 = 1.2, \hat{X}_0 = (0.9, 0.1), c = 1, 0.8, 0.6, 0.5$. (d) Adaptive signal lasso with penalty function $PF(x) = \lambda_1\sum_{j=1}^{2}|x_j| + \lambda_2\sum_{j=1}^{2}|X_{j0}||x_j - 1|$, $\lambda_1 = 0.4, \lambda_2 = 0.6, \hat{X}_0 = (0.1, 0.9), c = 1, 0.8, 0.6, 0.5$.

B. Algorithm and computation

It is noted that the penalty function in Eq. (8) is convex. Hence, the optimization problem in (9) does not suffer from multiple local minima, and its global minimizer can be efficiently solved. We provide an algorithm using the coordinate descent method [29], an iterative algorithm that updates the estimator by choosing a single coordinate to update and then performing a univariate minimization over it. Since ω_{1k} and ω_{2k} are known, we denote $\lambda_{1k}^* = \omega_{1k}\lambda_1$, $\lambda_{2k}^* = \omega_{2k}\lambda_2$. Define a threshold function by

$$S_{\theta_1,\theta_2}(z) = \begin{cases} (z+\theta_1)_-, & z \leqslant 0, \\ (z-\theta_2)_+, & 0 < z \leqslant 1+\theta_2, \\ \max\{1, z-\theta_1\}, & z > 1+\theta_2. \end{cases} \tag{13}$$

Then the update can proceed as

$$\hat{X}_k^{t+1} \leftarrow S_{\delta_{1k}^*,\delta_{2k}^*}\left(\hat{X}_k^t + \frac{\langle \hat{r}^t, \phi_k\rangle}{\langle \phi_k, \phi_k\rangle}\right), \tag{14}$$

where $\delta_{1k}^* = (\lambda_{1k}^* + \lambda_{2k}^*)/\langle \phi_k, \phi_k\rangle$, $\delta_{2k}^* = (\lambda_{1k}^* - \lambda_{2k}^*)/\langle \phi_k, \phi_k\rangle$, $\langle z_1, z_2\rangle$ denotes the inner product of vectors z_1 and z_2, ϕ_k is the kth column of Φ, $\hat{X}_k^t$ is the estimator of X_k in the tth step, and $\hat{r}^t = Y - \Phi\hat{X}^t$. The algorithm applies this update repeatedly in a cyclical manner, updating the coordinates of $\hat{X}$ along the way. Once an initial estimator of X is given, for example, by lasso or ridge estimation, updating can continue until convergence. These results are proved in the Appendix.

If Eq. (9) is formulated by parameters λ and α and $\omega_{1k} = 1$, $\omega_{2k} = |\hat{X}_{0k}|$, then

$$\delta_{1k}^* = \frac{\lambda(\alpha + |\hat{X}_{k0}|)}{\langle \phi_k, \phi_k\rangle}, \; \delta_{2k}^* = \frac{\lambda(\alpha - |\hat{X}_{k0}|)}{\langle \phi_k, \phi_k\rangle}.$$

Let $\lambda \to +\infty$. It is clear that Eq. (14) will shrink the negative solution to 0 and a solution larger than 1 to 1, because

TABLE I. Measures for accuracy of network reconstruction.

	Predicted class		
Actual class	Signal	Nonsignal	Unclassified
Signal class	True positive (TP)	False negative (FN)	Unclassified positive (UCP)
Nonsignal class	False positive (FP)	True negative (TN)	Unclassified negative (UCN)

$\delta^*_{1k} \to +\infty$. For $\hat{X}_{k0} \in (0, 1)$, when $\hat{X}_{k0} > \alpha$ and $\lambda \to +\infty$, then $\delta^*_{2k} \to -\infty$ and only the last condition in Eq. (13) holds. Therefore the solution gives a result of 1, since $\delta^*_{1k} \to +\infty$ in this case. When $\hat{X}_{k0} < \alpha$ and $\lambda \to +\infty$, then $\delta^*_{2k} \to +\infty$, and only the first two conditions in Eq. (13) are possible. However, they both equal 0 since $\delta^*_{1k} \to +\infty$. This indicates that the conclusion given in Eq. (12) still holds in the general case (meaning without the limitations of $p < n$ and a design matrix with orthogonal columns), which will be helpful for selecting the tuning parameters in the adaptive signal lasso.

C. Tuning the parameter

From the previously mentioned properties based on λ and α in Eq. (12), we can specify a large value for λ and only tune α using cross-validation (CV), which only involves one parameter and will greatly reduce computation. Furthermore, since α represents the proportion of data compressed to 0 in the interval (0, 1), it should be less than 1 and greater than 0. For the choice of α, we also need to conduct cross-validation, which minimizes the mean square prediction error (MSPE). In more detail, we first divide the full data set into some number of groups $K > 1$. The typical choices of K might be 5 or 10. We fix one group as the test set, and the remaining $K - 1$ group are designed as training set. We then apply the adaptive signal lasso method based on training set to obtain a fitted model for a range of α and use each fitted model to predict the responses in the test set, which leads to a mean square prediction error for each α. The procedure can be repeated K times and the average MSPE recorded, and the value of α with the minimum MSPE is selected as tje best choice of tuning parameters in our method. More details can be found in Ref. [14]. We use $\lambda = 1000$ as a large value for calculation in this paper and find that it is good enough in our calculations and can remove the unclassified portion in network reconstruction after conducting CV for α.

D. The metrics of reconstruction accuracy

To measure the accuracy of the estimation method, we have to define some metrics in the signal identification problem. As shown in Table I, we adopt common notation as in binary classification, where a true positive (TP) is the number of correctly identified true signals, a true negative (TN) is the number of correctly identified nonsignals, a false positive (FP) is the number of nonsignals incorrectly identified as signals, and a false negative (FN) is the number of signals incorrectly identified as nonsignals. However, in our analysis some lasso-type methods have points that cannot be classified (e.g., the parameter X is classified as signal if $\hat{X} \in 1 \pm 0.1$ and nonsignal if $\hat{X} \in 0 \pm 0.1$, and the remainder are unclassified [5,6,14]); hence we have the additional classes of unclassified positive (UCP) and unclassified negative (UCN), as shown in Table I. In traditional classification problems, the predicted class is completely classified into two classes, and therefore most common indexes for measuring accuracies are the true positive rate (TPR, sensitivity or recall), true negative rate (TNR, or specificity), and precision (positive prediction value, PPV), as well as the areas under the receiver operating characteristic curve (AUROC) and under the precision recall curve (AUPR) [5,12,30], where TPR and TNR are defined by

$$TPR = \frac{TP}{TP + FN}, \quad TNR = \frac{TN}{TN + FP}. \tag{15}$$

However, these measures have a problem when size of signals and nonsignals are unbalanced [31]. An alternative solution employs the Matthews correlation coefficient (MCC),

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, \tag{16}$$

which correctly takes into account the size of the confusion matrix elements [31,32]. MCC is widely used in machine learning to measure the quality of binary classifiers and is an overall measure of accuracy at the detection of signal and nonsignal classes. It is generally regarded as a balanced measure, which can be used even if classes have very different sizes [32,33]. Its values range from –1 to 1, and a large value indicates good performance.

In most cases of network construction, some links are unclassified; hence, the success rates for the detection of existing links (SREL) and nonexisting links (SRNL) are defined to study the performance of network reconstruction [5,6,14], where

$$SREL = \frac{TP}{TP + FN + UCP}, \quad SRNL = \frac{TN}{TN + FP + UCN}, \tag{17}$$

which considers the effects of nonclassifiability in Table I and is more reasonable for measuring the reconstruction accuracy of a network structure.

To address the effect of nonclassifiability on accuracy measures of reconstruction, we define an adjusted MCC, MCCa, by replacing FN with FN+UCP (i.e., the number of signals that are not correctly predicted), and FP with FP+UCN (the number of incorrectly predicted nonsignals) in MCC. It is clear when an unclassified set disappears, MCCa reduces to MCC. It is easy to see that MCCa plays a similar role to MCC when nonclassifiability occurs. We find that MCCa performs

TABLE II. Simulation results in linear regression model based on seven methods: lasso, adaptive lasso (A-lasso), SCAD, MCP, elastic net, signal lasso (S-lasso), and adaptive signal lasso (AS-lasso). Each of the results is averaged over 200 independent realizations, where n is the sample size, p is the number of explanatory variables, and p_1 is the number of signals (number of $\beta = 1$). The first panel is for the case of $p < n$, and the signal is sparse with $p_1 = 6$. The second panel is for $p > n$ and sparse signals. The third panel considers the nonsparse signal case, where $p_1 = 20$. The noise is introduced by $\sigma = 0.4, 1$, respectively, in each panel.

Method	MSE/SREL/SRNL/TPR/TNR/UCR/MCC/MCCa	
(n, p, p_1, σ)	(100, 30, 6, 0.4)	(100, 30, 6, 1)
Lasso	0.0014/0.920/0.987/0.999/1.000/0.027/1.000/0.919	0.0083/0.492/0.850/0.979/0.999/0.221/0.980/0.347
A-lasso	0.0015/0.925/0.976/0.999/1.000/0.343/1.000/0.900	0.0095/0.465/0.843/0.980/1.000/0.232/0.985/0.316
SCAD	0.0008/0.937/0.987/0.999/1.000/0.023/1.000/0.937	0.0043/0.522/0.985/0.999/1.000/0.108/1.000/0.632
MCP	0.0008/0.935/0.992/1.000/1.000/0.019/1.000/0.943	0.0046/0.525/0.980/1.000/1.000/0.111/1.000/0.633
Elastic net	0.0012/0.835/0.999/1.000/1.000/0.333/1.000/0.892	0.0066/0.385/0.979/0.970/1.000/0.139/0.970/0.503
S-lasso	0.0007/0.972/0.992/0.999/1.000/0.012/1.000/0.966	0.0046/0.671/0.916/0.999/1.000/0.133/1.000/0.609
AS-lasso	0.0001/1.000/0.997/0.999/1.000/0.002/1.000/0.993	0.0006/0.980/0.990/0.999/1.000/0.012/1.000/0.965
(n, p, p_1, σ)	(50, 150, 6, 0.4)	(50, 150, 6, 1)
Lasso	0.0016/0.640/0.969/0.989/1.000/0.043/0.990/0.529	0.0085/0.333/0.859/0.969/1.000/0.162/0.970/0.107
A-lasso	0.0026/0.585/0.937/0.979/1.000/0.076/0.980/0.375	0.0130/0.288/0.830/0.859/1.000/0.191/0.860/0.061
SCAD	0.0002/0.822/1.000/0.999/1.000/0.007/1.000/0.896	0.0111/0.271/0.987/0.567/1.000/0.037/0.609/0.305
MCP	0.0002/0.820/0.999/1.000/1.000/0.007/1.000/0.892	0.0130/0.246/0.991/0.541/1.000/0.032/0.588/0.309
Elastic net	0.0007/0.556/0.998/1.000/1.000/0.019/1.000/0.712	0.0037/0.263/0.989/0.850/1.000/0.039/0.850/0.359
S-lasso	0.0004/0.806/0.994/0.999/1.000/0.013/1.000/0.839	0.0028/0.570/0.964/0.999/1.000/0.051/1.000/0.482
AS-lasso	0.0009/0.973/0.999/0.985/0.999/0.001/0.989/0.976	0.0213/0.996/0.974/0.998/0.980/0.005/0.821/0.781
(n, p, p_1, σ)	(50, 150, 20, 0.4)	(50, 150, 20, 1)
Lasso	0.0279/0.198/0.885/0.860/0.999/0.202/0.905/0.091	0.0431/0.162/0.826/0.755/1.000/0.255/0.847/-0.01
A-Lasso	0.0104/0.429/0.925/0.964/0.999/0.140/0.975/0.372	0.0305/0.220/0.842/0.882/0.999/0.237/0.918/0.057
SCAD	0.1556/0.019/0.969/0.037/0.999/0.085/0.095/-0.01	0.1565/0.029/0.971/0.050/0.999/0.082/0.124/0.057
MCP	0.1709/0.017/0.973/0.026/0.998/0.072/0.078/-0.01	0.1771/0.022/0.974/0.032/0.999/0.069/0.091/-0.00
Elastic net	0.0142/0.294/0.968/0.934/1.000/0.119/0.961/0.362	0.0276/0.183/0.942/0.812/1.000/0.153/0.883/0.172
S-Lasso	0.0015/0.850/0.972/0.999/1.000/0.044/1.000/0.815	0.0092/0.680/0.912/0.996/0.999/0.118/0.997/0.549
AS-Lasso	0.0368/0.806/0.983/0.827/0.998/0.005/0.840/0.819	0.0400/0.794/0.982/0.812/0.998/0.004/0.827/0.809

well in network reconstruction to measure the accuracy of the method from either simulations or real examples.

IV. NUMERICAL STUDIES

We conduct simulation studies using three kinds of model: (1) a standard linear regression model with different assumptions; (2) a dynamics model of an evolutionary game [14], but with the PDG game; and (3) the Kuramoto model in synchronization dynamic, which is a special case of Eq. (2). Models (2) and (3) use three network topology structures: Erdös-Rényi (ER) random networks, Barabási-Albert (BA) scale-free networks, and small-world Watts-Strogatz (WS) networks. In each case we evaluate performance under situations of sparse and dense signals. When an adaptive signal lasso is used, we set $\lambda = 1000$ as a large value and tune α by the cross-validation method.

A. Linear regression models

We first use the following standard linear model,

$$Y = \mathbf{1}_n \mathbf{x}_0 + \Phi_1 \mathbf{X}_1 + \Phi_2 \mathbf{X}_2 + \epsilon, \tag{18}$$

for simulation study, where Y is a $n \times 1$ response variable, $\mathbf{x}_0$ is the intercept, $\mathbf{1}_n$ is an $n \times 1$ vector with all elements equal to 1, $\mathbf{X} = (\mathbf{X}_1', \mathbf{X}_2')'$, $\mathbf{X}_1 \in R^{p_1}$ represents the signal parameter, $\mathbf{X}_2 \in R^{p_2}$ denotes the nonsignal parameter, and ϵ is the error term. A smaller p_1 is called a sparse signal, and a larger p_1 (comparing with n and $p = p_1 + p_2$) is called a dense signal. In generation of the data set, we set $\mathbf{X}_1 = \mathbf{1}_{p_1}$ and $\mathbf{X}_2 = \mathbf{0}_{p_2}$. Each design matrix comes from a standard normal score with mean 0 and variance 1, but the columns in $\Phi = (\Phi_1, \Phi_2)$ are correlated in such a way that the correlation coefficient between ϕ_i and ϕ_j is given by $r^{|i-j|}$ with $r = 0.5$ (see similar uses in [27,34]). The error variable ϵ is generated from a Gaussian distribution with mean zero and variance σ^2. We also calculate the results from several well-known shrinkage estimates, including lasso, adaptive lasso, elastic net, SCAD, MCP, and signal lasso [14,27–29,34]. The above-mentioned first five methods do not shrink the parameter X in the model to 1, as they were designed to shrink irrelevant variables to zero using different penalty functions. Hence, we call these lasso-type methods, while signal lasso and adaptive signal lasso are called a signal-lasso-type method in this paper.

The results are presented in Table II, with the first panel showing the case of $p < n$ and sparse signals with $n = 100$, $p = 30$, $p_1 = 6$, and $\sigma = \{0.4, 1\}$. The second panel shows the case of $p > n$, with $n = 50$, $p = 150$, and the same parameters as the first panel. The third panel presents the case of nonsparse signals, where $p_1 = 20$ and other parameters are the same as in the second panel. The case of $p > n$ corresponds to high-dimensional variable selection, which is often encountered in network reconstruction with a small number of

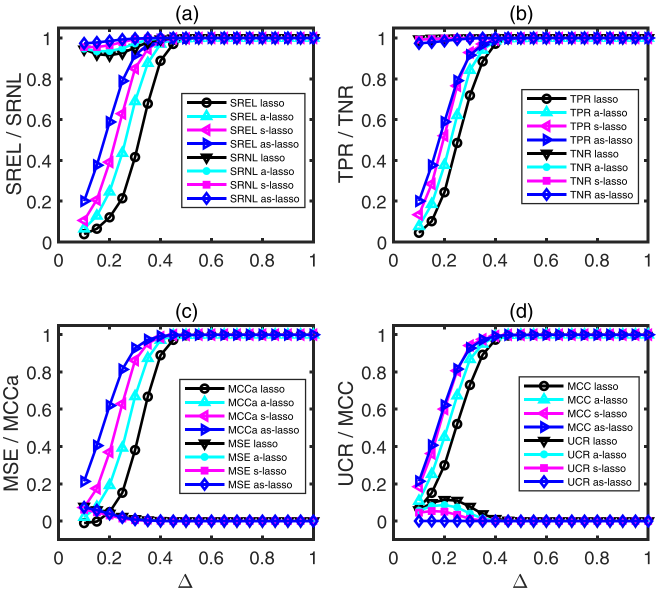

FIG. 3. Accuracy in the reconstruction vs $\Delta = L/N$ for PDG game with small-world (WS) network attained by lasso, adaptive lasso (a-lasso), signal lasso (s-lasso), and adaptive signal lasso (as-lasso) methods. (a) Measures using SREL and SRNL. (b) Measures of TPR and TNR. (c) Measures of MCCa and MSE. (d) Measures of MCC and UCR. The network size $N = 100$ with average degree 6. Each point is averaged over ten simulations.

observations. The method is tested for robustness by adding Gaussian noise with variances of 0.4 and 1. In the table we present all measures discussed in Sec. III D, but we only need to focus on MCC, MCCa, MSE, and UCR, as MCC is a synthesis measure of TPR and TNR, and MCCa is for SREL and SRNL.

In the first panel we can see that the measures of accuracy for the adaptive signal lasso method are overwhelmingly superior to other existing methods. The mean square error (MSE) and UCR values are the smallest, and the MCC and MCCa values are the largest among the seven methods. The signal lasso method is the second best, outperforming the first five nonsignal lasso methods based on all measures. It is also worth noting that the MCCa of the adaptive signal lasso method for large noise ($\sigma = 1$) remains at a high value, indicating its robustness. In the second panel, where $p \gg n$, the adaptive signal lasso method gives the largest values of MCCa and the smallest values of UCR for all cases. For small noise ($\sigma = 0.4$), the MCC of the adaptive signal lasso method is slightly lower and the MSE is slightly higher than other methods, but the difference is negligible. For large noise with $\sigma = 1$, the MCC of the signal lasso method is the highest; however, the MCCa of the adaptive signal lasso method is much higher than other methods. The MSE for all methods is small, but the adaptive signal lasso is not the smallest one; this is because adaptive signal lasso almost shrinks the parameter completely to 0 or 1, causing a larger deviation with the true value once it is falsely classified. In the last panel we can see the results for the cases of nonsparse signals with $p < n$, $p_1 = 20$, and $\sigma = \{0.4, 1\}$, respectively. As before, the adaptive signal lasso method outperforms other methods in terms of MCCa and UCR. The signal lasso method in this case has the largest MCC values and the smallest MSE. It is clear that the lasso-type methods (the first five methods) perform poorly, especially for the case of large noise.

In summary, we find that the adaptive signal lasso method outperforms other methods, followed by the signal lasso method. The first five methods (nonsignal lasso type) are not capable of efficiently identifying the correct signals, especially in cases of large noise and nonsparse signals. The signal lasso method is only competitive with the adaptive signal lasso method in cases of nonsparse signals and small noise. However, the most important advantage of the adaptive signal lasso method is that it can almost classify all parameters as 0 or 1 (with UCR close to zero), as its theory indicates. The adaptive signal lasso method is robust against noise, as it still performs well for larger σ in all cases.

B. Evolutionary-game-based dynamical model

We illustrate our method by iterative game dynamics [Eq. (2)] through Monte Carlo simulation. We take a simple structure with $R = 1, T = b = 1.15$, and $P = S = 0$ in our simulation. The total payoff is $F_i = \sum_j a_{ij} P_{ij}$ [Eq. (3)], where $a_{ij} = 1$ if the ith player and jth player interact, and 0 otherwise. In each round of the game, each player calculates its total payoff and then imitates with a certain probability the two strategies (p, q) of a randomly selected player in its direct neighborhood. That is, player x adopts the strategy of player y with probability $W = 1/\{1 + \exp[(F_x - F_y)/K]\}$ (Szabo *et al.*, 2007), where K is the uncertainty in strategy transition, with a value of 0.1 in this paper. The game iterates forward in a Monte Carlo manner, and player i (the focal player) acquires its fitness (total payoff) F_i by playing the game with all its direct neighbors, i.e., $F_i = \sum_{j=1}^{N} a_{ij} P_{ij}$. The focal player then randomly picks a neighbor j, which similarly acquires its fitness. Following the definition of fitness, a strategy update then occurs between its direct neighbors in a given network. Player i tries to imitate the strategy of player j with Fermi updating probability $W = 1/(1 + \exp[(F_i - F_j)/K])$, where $K = 0.1$ [35,36]. To make the model more realistic, we account for mutation at very small rates.

Now $F_i = \sum_{j=1, j\neq i}^{N} a_{ij} P_{ij}$ can be written as a linear regression model,

$$Y_i = \Phi_i \tilde{X}_i + e_i, \tag{19}$$

where $Y_i = (F_i(t_1), F_i(t_2), \dots, F_i(t_L))'$, $\tilde{X}_i = (a_{i1}, \dots, a_{iN})'$, and Φ_i has the form of

$$\begin{pmatrix} P_{i1}(t_1) & \cdots & P_{i,i-1}(t_1) & P_{i,i+1}(t_1) & \cdots & P_{i_N}(t_1) \\ P_{i1}(t_2) & \cdots & P_{i,i-1}(t_2) & P_{i,i+1}(t_2) & \cdots & P_{i_N}(t_2) \\ \vdots & \vdots & \vdots & \vdots & & \\ P_{i1}(t_L) & \cdots & P_{i,i-1}(t_L) & P_{i,i+1}(t_L) & \cdots & P_{i_N}(t_L) \end{pmatrix}.$$

Let $Y = (Y_1', \dots, Y_N')'$, $X = (\tilde{X}_1', \dots, \tilde{X}_N')'$, $\Phi = \mathrm{diag}(\Phi_1, \Phi_2, \dots, \Phi_N)$, then Eq. (19) can be converted into the general form of Eq. (5).

Figure 3 plots the measures discussed in Sec. III D against the data length $\Delta = L/N$ in the PDG model with a small-world WS network, using the methods of lasso, adaptive lasso, signal lasso, and adaptive signal lasso. The adaptive lasso

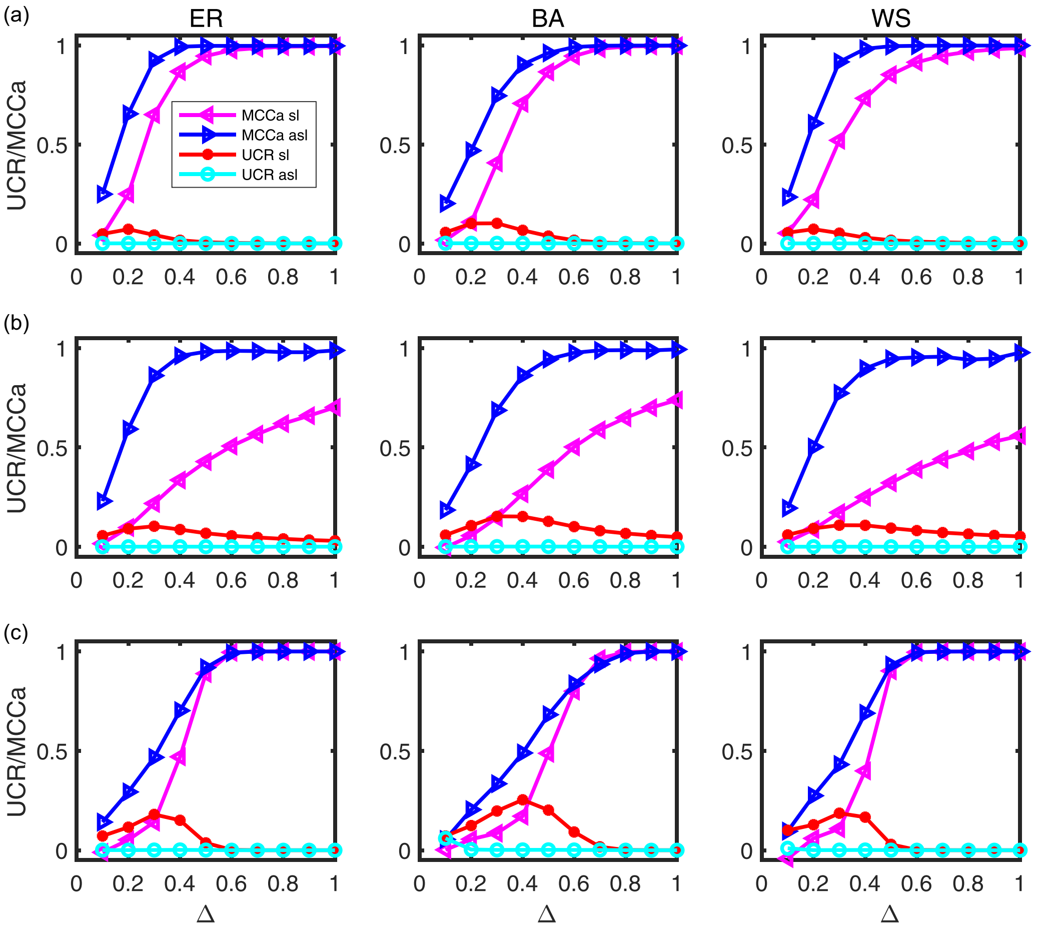

FIG. 4. Accuracy measures MCCa and UCR in the reconstruction vs $\Delta = L/N$ for the PDG model attained by the signal lasso (sl) and adaptive signal lasso (asl) methods in three kinds of networks. Row A shows results for the PDG game with noise $\sigma^2 = 0.1$ and average degree 6. Row B shows results for the PDG game with noise $\sigma^2 = 0.3$ and average degree 6. Row C shows the results for the PDG game with average degree 20 without noise. Three columns correspond to the results based on the Erdös-Rényi (ER) random network, Barabási-Albert (BA) scale-free network, and small-world (WS) network, respectively. The network size $N = 100$, and each point is averaged over ten simulations.

method represents the best performing method among the lasso-type methods, and lasso is also included as a commonly used method. As shown in the Supplemental Material [37], it is clear that the adaptive signal lasso method performed best based on most measures, followed by signal lasso, adaptive lasso, and lasso. For the measures TNR and MSE, the adaptive signal lasso method is not absolutely superior, but the differences between the four methods are very small. It is notable that the UCR of the adaptive signal lasso method is close to zero, even in the case of small $\Delta = L/N$, which is an appealing property.

In Fig. 4 we compare the reconstruction accuracy measures of MCCa and UCR in the ER, WS, and BA networks, with a focus on the signal lasso and adaptive signal lasso methods, as they performed better than other methods. In Fig. 4 rows A and B, we observe that the adaptive signal lasso method is superior to signal lasso when there is noise in the data, and that the UCR of the adaptive signal lasso method remains close to zero. Additionally, Fig. 4 row C shows that in a nonsparse network with an average degree of 20, the adaptive signal lasso has higher MCCa values than the signal lasso at small Δ, and the two methods perform similarly at large Δ. However, the UCR of the adaptive signal lasso is always better than that of the signal lasso. Overall, the results indicate the robustness and superiority of the adaptive signal lasso method in the presence of noise.

C. Kuramoto model in synchronization problem

For problems introduced in Eq. (2), we use the Kuramoto model [38–40] to illustrate the reconstruction of the network in a complex system. This model has the following governing equation:

$$\frac{d\theta_i}{dt} = \omega_i + c\sum_{j=1}^{N} a_{ij}\sin(\theta_j - \theta_i), \tag{20}$$

$i = 1, \dots, N$, where the system is composed of N oscillators with phase θ_i and coupling strength c, each of the oscillators has its own intrinsic natural frequency ω_i, a_{ij} is the adjacency matrix of a give network and needs to be estimated in network reconstruction. Using the same framework of Ref. [14], the Euler method can be employed to generate a time series with an equal time step h. Let $Y_i = (y_{i1}, \dots, y_{iL})'$, $y_{it} = [\theta_i(t + h) - \theta_i(t)]/h$, and $\Phi_i = [\phi_{ij}(t)]$ is a $L \times N$ matrix [14] with

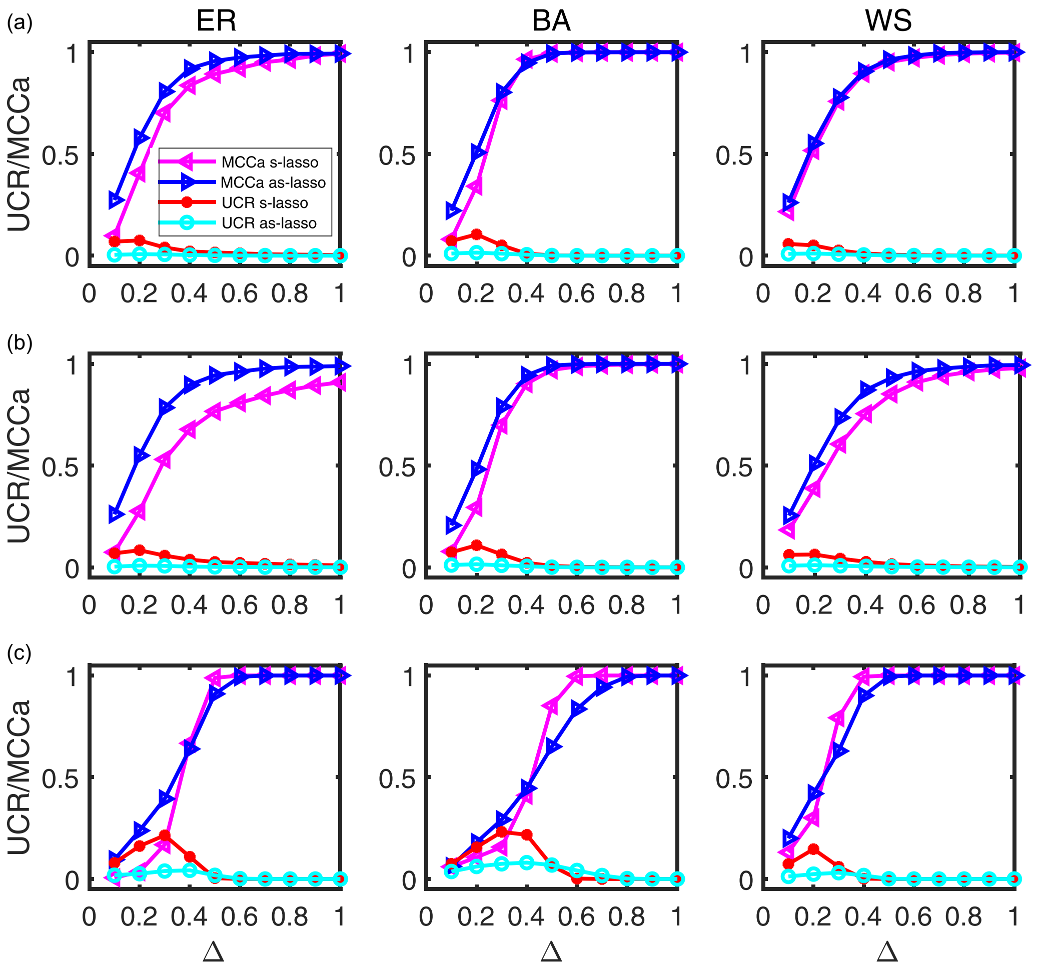

FIG. 5. Accuracy measures MCCa and UCR in the reconstruction vs $\Delta = L/N$ for the Kuramoto model attained by the signal lasso (sl) and adaptive signal lasso (asl) methods in three kinds of networks. The top row, A, contains the measures from the Kuramoto model in the network with average 6, while the middle panel, B, contains the results for the case with noise of $\sigma^2 = 0.3$, and the network with average 6. The bottom panel, C, gives the results for a nonsparse network with average degree 20 without noise. The three columns correspond to the results based on Erdös-Rényi (ER) random networks, Barabási-Albert (BA) scale-free networks, and small-world (WS) networks, respectively. The network size $N = 100$, and coupling $c = 10$ for all cases. Each point is averaged over ten simulations.

elements

$$\phi_{ij}(t) = c \times \sin(\theta_j(t) - \theta_i(t))$$

for $t = 1, \ldots, L$ and $j = 1, \ldots, N$, $\tilde{X}_i = (a_{i1}, \ldots, a_{iN})'$; then a reconstruction model can be written as

$$Y_i = \omega_i \mathbf{1}_L + \Phi_i \tilde{X}_i, \tag{21}$$

where $\mathbf{1}_L$ denotes a $L \times 1$ vector of all element 1.

In Fig. 5 we analyze the performance of the adaptive signal lasso method in the Kuramoto model in ER, WS, and BA networks with $N = 100$ and coupling strength $c = 10$. Row A shows the results without noise, and we find similar results to those previously analyzed. Row B shows the results with noise variance equal to 0.3, and it also indicates that the adaptive signal lasso method is robust against noise. Row C shows the results for a nonsparse network with an average degree of 20. It is interesting to observe that the adaptive signal lasso performed better than the signal lasso for small $\Delta = L/N$, while the signal lasso became better than the adaptive signal lasso for large $\Delta = L/N$. However, the UCR always performed well for both methods. These results imply that the signal lasso is also useful in nonsparse networks, but the adaptive signal lasso is more computationally convenient since it only requires tuning one parameter.

V. EXAMPLES

A. Human behavioral data

The real data from a human behavior experiment [22] is used to illustrate social network reconstruction. There were three trials with 135 volunteers participating in the experiments where participants played an iterative prisoner's dilemma game on different types of networks. The data was analyzed to show that the signal lasso method outperformed other methods in a previous study [14]. In this study the performance of the proposed adaptive signal lasso method was evaluated and the results are given in Fig. 6, where rows A, B, and C refer to the results of the experimental ring, homogeneous random, and heterogeneous random network, respectively. The figures in column (a) list the measures of MCC and MSE, and column (b) lists the measures of MCCa

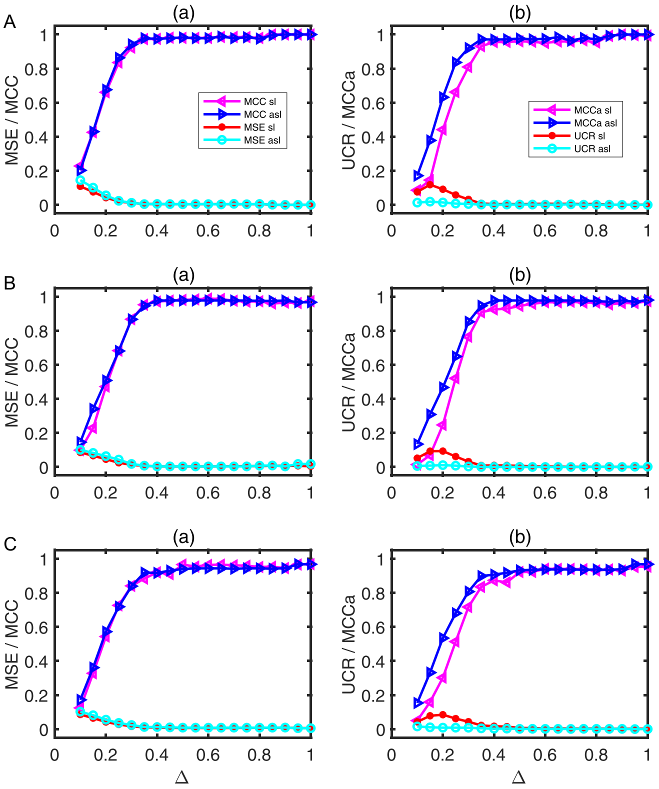

FIG. 6. Accuracy in the reconstruction vs $\Delta = L/N$ for three real trials using the signal lasso (sl) and adaptive signal lasso (asl) methods. Row A shows the results of the experimental ring network, (a) is a plot of MCC and MSE vs Δ, and (b) refers to the MCCa and UCR criterion for the signal lasso and adaptive signal lasso, respectively. There are 35 nodes and 140 links, and the degree of each node is 4. Row B shows results of the experimental homogeneous random network. There are 50 nodes and 200 links, and the degree of each node is 4. Row C shows the results of the experimental heterogeneous random network. There are 50 nodes and 200 links, and the average degree of each node is 4.

and UCR. It is clear that the adaptive signal lasso outperforms the signal lasso in terms of MCC and MCCa, especially for MCCa. The MSE of the adaptive signal lasso is slightly larger than that of the signal lasso for smaller values of Δ. The UCR of the adaptive signal lasso is close to zero for all values of Δ, indicating high reconstruction accuracy, while the UCR of the signal lasso shows that it still contains some unclassified parts for smaller Δ, reducing reconstruction accuracy.

B. World trade web

A network formed by import/export relationships between countries, the world trade web (WTW) has been extensively studied. Some empirical studies focused on the WTW as a complex network and investigated its architecture [41]. When the available information on the system is incomplete or partial, reconstruction methods of the whole network have been proposed, such as maximum entropy [12,42] and the configuration model (CM) [43–45]. Much of the network reconstruction research in WTW focuses on ensemble models, which means that a model is defined to be not a single network but a probability distribution over many possible networks [46]. We use a different perspective to illustrate the construction of a trade network using our proposed signal lasso method, which gives a model-based estimation of adjacency matrix A.

In this paper we obtained a database of trade network using data reported by countries to the United Nations Statistical Division. The data, which is available for free on their website, includes bilateral trade flows for over 5000 products and 200 countries from 1995 to 2018. We deleted any missing data and focused on data from 1995 to 2018 involving 215 countries. This resulted in a network of all trade products with 215 nodes and 36 296 links, which is an extremely dense network, as reported in [41].

We only consider an undirected binary network, defining two countries as connected if one has output to another. Let Y_{it} denote total foreign trade (in U.S. dollars) of the ith country at time t, including imports and exports with other countries. Let $w_{ij}(t)(=w_{ji}(t))$ denote the import and export values of the ith country with the jth country at time $t(t=1,\dots,24)$. Then we can formulate the following equation:

$$Y_{it} = \sum_{j=1}^{N} a_{ij} w_{ij}(t) + \epsilon_{it}, \qquad (22)$$

where a_{ij} is an element of a connectivity matrix with $a_{ii}=0$, $a_{ij}=a_{ji}$ for $i \neq j$, and $N=215$ is the number of countries. It is noted that a_{ij} is independent of t and represents a network connectivity during an observed period. For some country pairs, the export (or import) bilaterally or unilaterally at some time periods are missing, which means in some years during the observations there are no trade between two countries or it is missing because the trading quantity is small and can be ignored. Hence, the edges during 24 years might be incompletely linked, which is unbalanced.

In many economic network analyses and spatial panel data models, the adjacency matrix must be predetermined. A simple method to obtain an adjacency matrix is to define $a_{ij}=1$ if there is a connection at least one time between two nodes and $a_{ij}=0$ otherwise [47,48]. However, this may be unreasonable if two countries only have one year of very small or negligible trade. An alternative method is to use the percentage of the trade between two countries of the total trade amount to measure their connectivity, but they need a cutoff value, which might be subjective. An interesting problem is studying how to use the structural relationship, such as Eq. (22), to estimate the network connectivity during the observed periods; this problem can be dealt with using the shrinkage theory proposed in this paper, since in this case a_{ij} is either 0 or 1. It is noted that the model in Eq. (22) relates to the reconstruction of binary topology, and we assume this topology is fixed during the observed time period, which is usually favorable in statistical network models [47,48].

For comparison, we define the reference connectivity as $\tilde{a}_{ij}=1$ if at least one connection occurs during the observed years and zero otherwise. This network is not assumed to be a correct adjacency matrix, but it can be used for comparison and analysis. The results are listed in Table III, using $\tilde{a}_{ij}$ as a baseline for comparison. It is surprising that the lasso and adaptive lasso perform poorly in terms of SREL (TPR), MCC, MCCa, and MSE, even with high values of SRNL (TNR),

TABLE III. Estimation accuracy of reconstruction of trade network.

Measures	Lasso	A-lasso	S-lasso	AS-lasso
SREL	0.0055	0.0202	0.9632	0.9710
SRNL	1	1	1	1
TPR	0.0075	0.0263	0.9652	0.9722
TNR	1	1	1	1
MCC	0.0246	0.0563	0.9659	0.9722
MCCa	0.0185	0.0446	0.9636	0.9282
UCR	0.2296	0.2038	0.0103	0.0022
MSE	1.3×10^8	1.9×10^3	0.266	0.1358

perhaps because the trade network is highly dense, which was verified in our simulation of Sec. IV A. The performance of both the signal lasso and the adaptive signal lasso is outstanding. Specifically, the signal lasso yields a larger MCCa value, whereas the adaptive signal lasso demonstrates smaller values for MSE and UCR. All methods can correctly identify nonexistent links (SRNL and TPR have values of 1) because there are just a few zero edges (about 12%) in WTW. The values of SREL (TNR) of the adaptive signal lasso can exceed 96%, even with $\Delta = 24/215 = 0.11$.

Figure 7 shows some basic statistics calculated from the reference and estimated adjacency matrix. Figure 7(a) shows the evolution of the average degree and average-nearest-neighbor degree (ANND) over the time calculated from $\tilde{a}_{ij}$, which shows that these two quantities are time dependent and have increasing degrees of nodes, but there is a downward trend in the last few years. Figure 7(b) shows the absolute difference of reference adjacency matrix $\tilde{A}$ and estimated adjacency matrix $\hat{A}$ (denoted by $D(\tilde{A};\hat{A}) = |\tilde{a}_{ij} - \hat{a}_{ij}|$) against their weight coefficients:

$$\tilde{w}_{ij} = \frac{\sum_t w_{ij}(t)}{\sum_t \sum_{i<j} w_{ij}(t)}.$$

The small value of $\tilde{w}_{ij}$ means small imports/exports between two countries or few connections during 24 years. $D(\tilde{A},\hat{A}) = 1(0)$ means inconsistent (consistent) results between $\tilde{A}$ and $\hat{A}$, and $0 < D(\tilde{A};\hat{A}) < 1$ indicates the unclassified cases. It is clear that inconsistency only occurs at very small weight coefficients, and unclassified cases occur mostly at smaller weight coefficients. Further investigation finds that these two cases occur at $J = 1, 2, 3$, where J is the number of years from 1995 to 2018 for which import/export statistics are available (namely, nonzero). These results show that our shrinkage estimation can eliminate unimportant linkages between nodes.

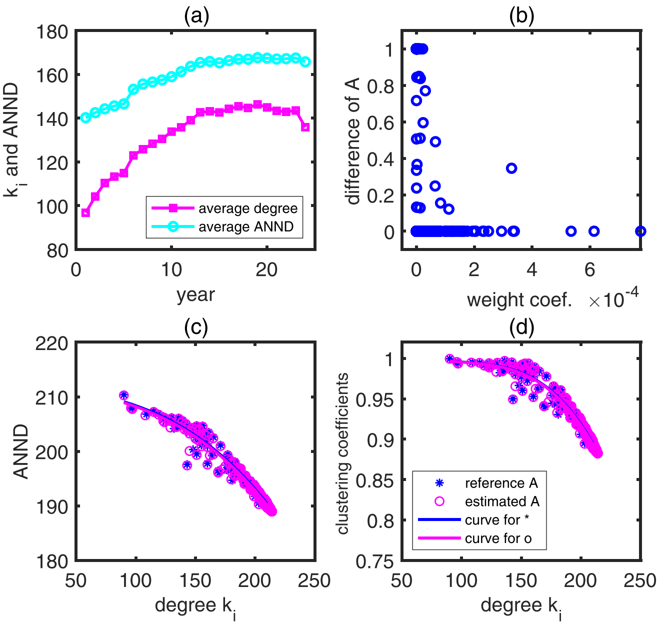

FIG. 7. The reconstruction results for undirected WTW using the adaptive signal lasso method. (a) Plots of average degree and average-nearest-neighbor degree (ANND) vs year, where 1 represents 1995 and 24 for 2018. (b) Plot of difference between the reference and estimated adjacency matrix of the network. (c) Plot of ANND vs average degree. (d) Plot of clustering coefficients of network vs average degree. In (c) and (d), curves are fitted based on a polynomial of order 3.

Figures 7(c) and 7(d) show plots of ANND and clustering coefficients, respectively, versus average degree. We find the results from $\tilde{A}$ and $\hat{A}$ are highly coincident, where decreasing trends have been found in previous studies employing different datasets in WTW [41]. The decreasing trend in Fig. 7(c) is known as disassortivity (i.e., countries trading with highly connected countries have few trade partners, and those trading with poorly connected countries have many) in WTW [41].

VI. CONCLUSIONS

We present the adaptive signal lasso, an enhanced version of the traditional signal lasso method. By introducing a weight to the penalty terms of the original signal lasso method, we have managed to enhance its overall performance. Our theoretical examinations and simulations have established that the adaptive signal lasso demonstrates superior efficacy in the detection of signals within complex networks. It offers three key advantages over the traditional signal lasso and other lasso-based methodologies: (1) it necessitates only a single tuning parameter, thereby minimizing computational expenditure; (2) it exhibits a higher degree of robustness against noise and contamination, as evidenced by simulations based on the PDG evolutionary model and the Karumoto model; and (3) it is adept at shrinking parameters to either 0 or 1 completely, surpassing the network reconstruction capabilities of other lasso-based approaches. However, in the context of nonsparse networks, the traditional signal lasso method maintains its merit and holds its own against the adaptive signal lasso. This parity is reflected in simulations involving evolutionary-game-based dynamics and the Kuramoto model, where the signal lasso outperforms the adaptive signal lasso when the data length Δ is larger (refer to row C of Figs. 4 and 5) in terms of the MCCa metric. In examining the real-world network of WTW, it is also observed that the MCCa value obtained via the signal lasso exceeds that obtained from the adaptive signal lasso.

To exemplify our method's efficacy in reconstructing nonsparse or dense signal networks, we employed a straightforward example from the WTW network. Our findings revealed that the proposed method and the signal lasso both perform well. The adaptive signal lasso exhibited smaller MSE and

UCR values, whereas the signal lasso demonstrated larger MCCa values. Importantly, this WTW example primarily serves to assess the method's effectiveness; a reliable method should indeed yield a high recognition rate. In directed networks, our signal-lasso-type methods can be equivalently applied and have demonstrated satisfactory performance, as shown in the Supplemental Material [37]. In real-world scenarios, only partial information might be available, such as in-degree or out-degree values in a binary network. These circumstances can be managed by replacing the w_{ij} in Eq. (22) with an estimated value sourced from an existing method [12].

Our study focused on detecting signal parameters (0 or 1) within binary networks. However, in weighted networks, weight coefficients exist within the interval [0,1] [6,12,45]. In these instances the proposed adaptive signal lasso could be applied by adjusting the λ parameter in Eq. (11) to a suitable value rather than allowing it to approach infinity as we did in this study. This potential modification and its implications will be the focus of our future research.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (No. 12271471 and No. 11931015) and Major Projects of the National Philosophy and Social Science Foundation of China (No. 22&ZD158 and No. 22VRC049) to L.S., JSPS Postdoctoral Fellowship Program for Foreign Researchers (No. P21374) and an accompanying Grant-in-Aid for Scientific Research to C.S., the National Philosophy and Social Science Foundation of China (No. 23BTJ046) to L.J., and the National Natural Science Foundation of China (No. 12071414) to D.Y.

APPENDIX

1. Comparison of weight choice

If we assign the weight in both penalty terms, then $\omega_{1i} = |\hat{X}_{i0}|^{-\nu}$ and $\omega_{2i} = |\hat{X}_{i0}|^{\gamma}$ are two appropriate choices in adaptive signal lasso, where $\hat{X}_{i0}$ is the ordinary least square (OLS) estimator of X. Figure 8(a) shows the curves of the penalty solution of the adaptive signal lasso as a function of $\hat{X}_0$ when $\nu = 0, \gamma = 1$, which is the formula used throughout this paper. Figures 8(b)–8(d) show the solution of the adaptive signal lasso as a function of $\hat{X}_0$ in the three cases of $\nu = 1$ and $\gamma = 1$, $\nu = 0$, and $\gamma = 1$, and $\nu = 0$ and $\gamma = 1.5, 2$, respectively. Although the adaptive signal lasso in (b)–(d) also has the function of shrinking the values between 0 and 1 in two directions (0 or 1), it is too tedious and complicated, and sometime is unstable. Case (a), with $\nu = 0$ and $\gamma = 1$, is the simplest, but it can achieve our purpose very well. In addition, case (a) can reveal an appealing property [such as the result in Eq. (12)] in selecting the tuning parameters in the adaptive signal lasso, as we show in Eq. (12).

2. The proof of Eqs. (10) and (12)

In order to study the geometry of the signal lasso, we assume that the columns of Φ are orthogonal to each other and $p < n$. The ordinary least-squares estimate in this special case then has the form of $\hat{X}_0 = \Phi' Y$. Let $\hat{Y}_0 = \Phi \hat{X}_0$, and we then have

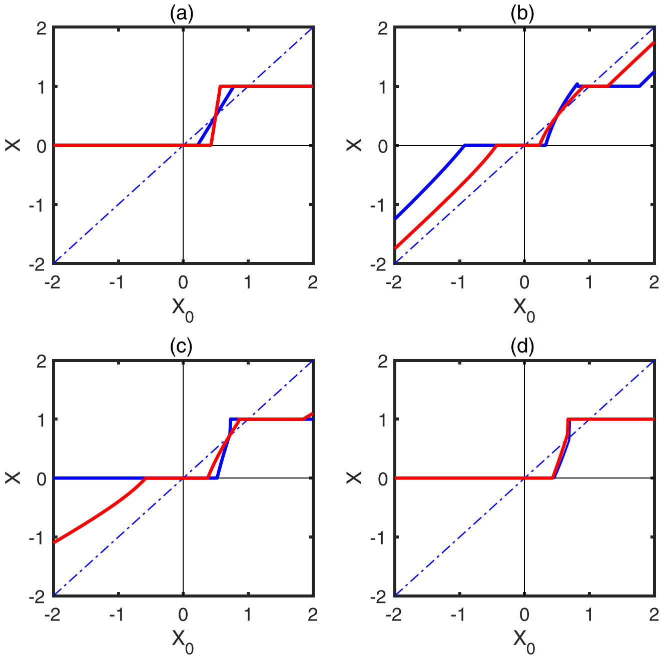

FIG. 8. Solution of X under orthogonal design in adaptive signal lasso method with different weights. (a) Solution of adaptive signal lasso as a function of OLS $\hat{X}_0$ with $\omega_{1j} = 1$ and $\omega_{2j} = |\hat{X}_{j0}|$, where the red line is for $\lambda_1 = 2$ and $\lambda_2 = 4$, and the blue line is for $\lambda_1 = 0.4$ and $\lambda_2 = 0.8$. (b) Solution of the adaptive signal lasso as a function of OLS $\hat{X}_0$ with $\omega_{1j} = |\hat{X}_{j0}|^{-1}$ and $\omega_{2j} = |\hat{X}_{j0}|$, where the red line is for $\lambda_1 = 0.2$ and $\lambda_2 = 0.8$, the blue line is for $\lambda_1 = 0.6$ and $\lambda_2 = 1.2$. (c) Solution of the adaptive signal lasso as a function of OLS $\hat{X}_0$ with $\omega_{1j} = |\hat{X}_{j0}|^{-1}$ and $\omega_{2j} = 1$, where the red line is for $\lambda_1 = 0.1$ and $\lambda_2 = 0.2$, the blue line is for $\lambda_1 = 0.3$ and $\lambda_2 = 0.6$. (d) Solution of the adaptive signal lasso as a function of OLS $\hat{X}_0$ with $\omega_{1j} = 1$ and $\omega_{2j} = |\hat{X}_{j0}|^{\gamma}$, with $\lambda_1 = 2$ and $\lambda_2 = 4$, where the blue line is for $\gamma = 1.5$ and red line for $\gamma = 2$.

$$L(X|\lambda_1, \lambda_2) = \tfrac{1}{2}\|Y - \hat{Y}_0\|_2^2 + \tfrac{1}{2}\|X - \hat{X}_0\|_2^2 + \lambda_1\|\omega_1 \circ X\|_1 + \lambda_2\|\omega_2 \circ (X - \mathbf{1}_p)\|_1, \tag{A1}$$

where ω_1 and ω_2 are $p \times 1$ known weight vectors and $\circ$ denotes the Hadamard product. Note that the first term is constant with respect to X and $\|X(X - \hat{X}_0)\|_2^2 = \|(X - \hat{X}_0)\|_2^2$. Using the fact that

$$\frac{\partial |X_i|}{\partial X_i} = \begin{cases} 1, & if\ X_i > 0, \\ -1, & if\ X_i < 0, \\ \in\ [-1, 1] & if\ X_i = 0, \end{cases} \tag{A2}$$

and differentiating $L(X, \lambda_1, \lambda_2)$ with respect to X and setting it to zero, after some calculations we have

$$\hat{X}_j = \begin{cases} (\hat{X}_{j0} + \lambda_1\omega_{1j} + \lambda_2\omega_{2j})_-, & \hat{X}_{j0} \leqslant 0, \\ (\hat{X}_{j0} - \lambda_1\omega_{1j} + \lambda_2\omega_{2j})_+, & 0 < \hat{X}_{j0} \leqslant \delta, \\ \max\{1, \hat{X}_{j0} - \lambda_1\omega_{1j} - \lambda_2\omega_{2j}\}, & \hat{X}_{j0} > \delta, \end{cases} \tag{A3}$$

in which δ is the solution of function $\hat{X}_{j0} - \lambda_1\omega_{1j} + \lambda_2\omega_{2j} = 1$. From our analysis in the main content of this paper, we choose $\omega_{1j} = |\hat{X}_{j0}|^{-\nu}$ as adaptive lasso used and

$\omega_{2j} = |\hat{X}_{j0}|^{\gamma}$. When $\nu = 0$ and $\gamma = 1$, Eq. (10) can be derived easily from Eq. (A3).

Now we reparametrize λ_1 and λ_2 by λ and α, and then we have

$$\hat{X}_j = \begin{cases} \{\hat{X}_{j0} + \lambda(\alpha - \hat{X}_{j0})\}_{-}, & \hat{X}_{j0} \leqslant 0, \\ \{\hat{X}_{j0} - \lambda(\alpha - \hat{X}_{j0})\}_{+}, & 0 < \hat{X}_{j0} \leqslant \alpha_2, \\ \max\{1, \hat{X}_{j0} - \lambda(\alpha + \hat{X}_{j0})\}, & \hat{X}_{j0} > \alpha_2, \end{cases} \quad \text{(A4)}$$

which will immediately lead to the results in Eq. (12).

3. Coordinate descent algorithm

Differentiating (8) with respect to X_k and setting it equal to zero, we have

$$-\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p}\phi_{ij}X_j\right)\phi_{ik} + \lambda_{1k}^{*}s_k^{(1)} + \lambda_{2k}^{*}s_k^{(2)} = 0, \quad \text{(A5)}$$

where $s_k^{(1)} = \partial|X_k|/\partial X_k = \text{sgn}(X_k)$, $s_k^{(2)} = \partial|X_k - 1|/\partial X_k = \text{sgn}(X_k - 1)$; sgn$(z)$ takes values of sgn(z) for $z \neq 0$, and some value lying in $[-1, 1]$ for $z = 0$. Let $r^{(k)} = (r_1^{(k)}, \dots, r_n^{(k)})'$ denote the partial residual, where $r_i^{(k)} = y_i - \sum_{j\neq k}\phi_{ij}X_j$. Then, using formula (A2) and after some calculation, we have

$$X_k = \begin{cases} [z_k + \delta_{1k}^{*}]_{-}, & z_k \leqslant 0, \\ [z_k - \delta_{2k}^{*}]_{+}, & 0 < z_k \leqslant 1 + \delta_{2k}^{*}, \\ \max\{1, [z_k - \delta_{1k}^{*}]\}, & z_k > 1 + \delta_{2k}^{*}, \end{cases} \quad \text{(A6)}$$

where $\langle z, y\rangle$ denotes the inner product of vectors z and y, $\delta_{1k}^{*} = (\lambda_{1k}^{*} + \lambda_{2k}^{*})/\langle\phi_k, \phi_k\rangle$ and $\delta_{2k}^{*} = (\lambda_{1k}^{*} - \lambda_{2k}^{*})/\langle\phi_k, \phi_k\rangle$, and $z_k = \langle r^{(k)}, \phi_k\rangle/\langle\phi_k, \phi_k\rangle$. From the definition of $S_{\theta_1,\theta_2}(z)$, it is easy to see that

$$X_k = S_{\delta_{1k}^{*},\delta_{2k}^{*}}\left(\frac{\langle r^{(k)}, \phi_k\rangle}{\langle\phi_k, \phi_k\rangle}\right). \quad \text{(A7)}$$

Note that $\langle r^{(k)}, \phi_k\rangle = \langle r, \phi_k\rangle + X_k\langle\phi_k, \phi_k\rangle$, where $r = Y - \Phi X$, and we have

$$X_k = S_{\delta_{1k}^{*},\delta_{2k}^{*}}\left(X_k + \frac{\langle r, \phi_k\rangle}{\langle\phi_k, \phi_k\rangle}\right). \quad \text{(A8)}$$

Therefore the update can be written as

$$\hat{X}_k^{t+1} \leftarrow S_{\delta_{1k}^{*},\delta_{2k}^{*}}\left(\hat{X}_k^{t} + \frac{\langle\hat{r}^{t}, \phi_k\rangle}{\langle\phi_k, \phi_k\rangle}\right), \quad \text{(A9)}$$

where $\hat{X}_k^t$ denotes the estimator of X_k in the tth step, and $\hat{r}^t = Y - \Phi\hat{X}^t$. The overall algorithm operates by applying this update repeatedly in a cyclical manner, updating the coordinates of $\hat{X}$ along the way. Once an initial estimator of X is given, for example, by lasso estimation or ridge estimation, the update can be continued until convergence.

The algorithm for computation of methods such as lasso, adaptive lasso, SCAD, MCP, and ElasticNet can be found in R-software and has been widely used in scientific research (see Supplemental Material [37] for details). The R-code using the coordinate descent method for the adaptive signal lasso method can be found on GitHub [49].

[1] S. H. Strogatz, Exploring complex networks, Nature (London) **410**, 268 (2001).

[2] A.-L. Barabási, The network takeover, Nat. Phys. **8**, 14 (2012).

[3] R. Albert and A.-L. Barabási, Statistical mechanics of complex networks, Rev. Mod. Phys. **74**, 47 (2002).

[4] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, Complex networks: Structure and dynamics, Phys. Rep. **424**, 175 (2006).

[5] X. Han, Z. Shen, W.-X. Wang, and Z. Di, Robust reconstruction of complex networks from sparse data, Phys. Rev. Lett. **114**, 028701 (2015).

[6] W.-X. Wang, Y.-C. Lai, C. Grebogi, and J. Ye, Network reconstruction based on evolutionary-game data via compressive sensing, Phys. Rev. X **1**, 021021 (2011).

[7] T. P. Peixoto, Reconstructing networks with unknown and heterogeneous errors, Phys. Rev. X **8**, 041011 (2018).

[8] T. S. Gardner, D. Di Bernardo, D. Lorenz, and J. J. Collins, Inferring genetic networks and identifying compound mode of action via expression profiling, Science **301**, 102 (2003).

[9] F. Geier, J. Timmer, and C. Fleck, Reconstructing gene-regulatory networks from time series, knock-out data, and prior knowledge, BMC Syst. Biol. **1**, 11(2007).

[10] S. Grün, M. Diesmann, and A. Aertsen, Unitary events in multiple single-neuron spiking activity. I. Detection and significance, Neural Comput. **14**, 43 (2002).

[11] K. Supekar, V. Menon, D. Rubin, M. Musen, and M. D. Greicius, Network analysis of intrinsic functional brain connectivity in Alzheimer's disease, PLoS Comput. Biol. **4**, e1000100 (2008).

[12] T. Squartini, G. Caldarelli, G. Cimini, G. Andrea, and D. Garlaschelli, Reconstruction methods for networks: The case of economic and financial systems, Phys. Rep. **757**, 1 (2018).

[13] L. Shi, C. Shen, Q. Shi, Z. Wang, J. Zhao, X. Li, and S. Boccaletti, Recovering network structures based on evolutionary game dynamics via secure dimensional reduction, IEEE Trans. Netw. Sci. Eng. **7**, 2027 (2020).

[14] L. Shi, C. Shen, L. Jin, Q. Shi, Z. Wang, and S. Boccaletti, Inferring network structures via signal lasso, Phys. Rev. Res. **3**, 043210 (2021).

[15] W.-X. Wang, Y.-C. Lai, and C. Grebogi, Data based identification and prediction of nonlinear and complex dynamical systems, Phys. Rep. **644**, 1 (2016).

[16] S. Raimondo and M. De Domenico, Measuring topological descriptors of complex networks under uncertainty, Phys. Rev. E **103**, 022311 (2021).

[17] F. C. Santos and J. M. Pacheco, Scale-free networks provide a unifying framework for the emergence of cooperation, Phys. Rev. Lett. **95**, 098104 (2005).

[18] M. A. Nowak and R. M. May, Evolutionary games and spatial chaos, Nature (London) **359**, 826 (1992).

[19] C. Hauert and M. Doebeli, Spatial structure often inhibits the evolution of cooperation in the snowdrift game, Nature (London) **428**, 643 (2004).

[20] G. Szabó and G. Fath, Evolutionary games on graphs, Phys. Rep. **446**, 97 (2007).

[21] M. Perc, J. J. Jordan, D. G. Rand, Z. Wang, S. Boccaletti, and A. Szolnoki, Statistical physics of human cooperation, Phys. Rep. **687**, 1 (2017).

[22] X. Li, M. Jusup, Z. Wang, H. Li, L. Shi, B. Podobnik, H. E. Stanley, S. Havlin, and S. Boccaletti, Punishment diminishes the benefits of network reciprocity in social dilemma experiments, Proc. Natl. Acad. Sci. USA **115**, 30 (2018).

[23] Z. Wang, M. Jusup, R.-W. Wang, L. Shi, Y. Iwasa, Y. Moreno, and J. Kurths, Onymity promotes cooperation in social dilemma experiments, Sci. Adv. **3**, e1601444 (2017).

[24] Z. Wang, M. Jusup, L. Shi, J.-H. Lee, Y. Iwasa, and S. Boccaletti, Exploiting a cognitive bias promotes cooperation in social dilemma experiments, Nat. Commun. **9**, 2954 (2018).

[25] L. Shi, I. Romić, Y. Ma, Z. Wang, B. Podobnik, H. E. Stanley, P. Holme, and M. Jusup, Freedom of choice adds value to public goods, Proc. Natl. Acad. Sci. USA **117**, 17516 (2020).

[26] G. Szabó, J. Vukov, and A. Szolnoki, Phase diagrams for an evolutionary prisoner's dilemma game on two-dimensional lattices, Phys. Rev. E **72**, 047107 (2005).

[27] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B. Stat. Methodol. **58**, 267 (1996).

[28] H. Zou, The adaptive lasso and its oracle properties, J. Am. Stat. Assoc. **101**, 1418 (2006).

[29] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity: The Lasso and Generalizations* (CRC Press, Boca Raton, FL, 2015).

[30] D. Marbach, J. C. Costello, R. Küffner, N. M. Vega, R. J. Prill, D. M. Camacho, K. R. Allison, The DREAM5 Consortium, M. Kellis, J. J. Collins, and G. Stolovitzky, Wisdom of crowds for robust gene network inference, Nat. Methods **9**, 796 (2012).

[31] D. Chicco, Ten quick tips for machine learning in computational biology, BioData Mining **10**, 35 (2017).

[32] D. Chicco and G. Jurman, The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation, BMC Genomics **21**, 6 (2020).

[33] J. Fan, Y. Feng, and Y. Wu, Network exploration via the adaptive LASSO and SCAD penalties, Ann. Appl. Stat. **3**, 521 (2009).

[34] J. Fan and R. Li, Variable selection via nonconcave penalized likelihood and its oracle properties, J. Am. Stat. Assoc. **96**, 1348 (2001).

[35] G. Szabó and C. Tőke, Evolutionary prisoner's dilemma game on a square lattice, Phys. Rev. E **58**, 69 (1998).

[36] G. Szabó and C. Hauert, Evolutionary prisoner's dilemma games with voluntary participation, Phys. Rev. E **66**, 062903 (2002).

[37] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevResearch.5.043200 for the detailed notes for the use of computational software of all methods, the generation method of different networks and more simulation results.

[38] S. Boccaletti, G. Bianconi, R. Criado, C. I. del Genio, J. Gómez-Gardenes, M. Romance, I. Sendina-Nadal, Z. Wang, and M. Zanin, The structure and dynamics of multilayer networks, Phys. Rep. **544**, 1 (2014).

[39] M. Timme, Revealing network connectivity from response dynamics, Phys. Rev. Lett. **98**, 224101 (2007).

[40] X. Wu, W. Wang, and W. X. Zheng, Inferring topologies of complex networks with hidden variables, Phys. Rev. E **86**, 046106 (2012).

[41] T. Squartini, G. Fagiolo, and D. Garlaschelli, Randomizing world trade. II. A weighted network analysis, Phys. Rev. E **84**, 046118 (2011).

[42] P. E. Mistrulli, Assessing financial contagion in the interbank market: Maximum entropy versus observed interbank lending patterns, J. Bank. Finance **35**, 1114 (2011).

[43] D. Garlaschelli and M. I. Loffredo, Fitness-dependent topological properties of the world trade web, Phys. Rev. Lett. **93**, 188701 (2004).

[44] D. Garlaschelli and M. I. Loffredo, Generalized Bose-Fermi statistics and structural correlations in weighted networks, Phys. Rev. Lett. **102**, 038701 (2009).

[45] G. Caldarelli, A. Chessa, G. Andrea, F. Pammolli, and M. Puliga, Reconstructing a credit network, Nat. Phys. **9**, 125 (2013).

[46] J. Park and M. E. J. Newman, Statistical mechanics of networks, Phys. Rev. E **70**, 066117 (2004).

[47] L. Anselin, *Spatial Econometrics: Method and Models* (Springer Science & Business Media, New York, 2013), Vol. 4.

[48] X. Zhu, W. Wang, H. Wang, and W. K. Härdle, Network quantile autoregression, J. Econometrics **212**, 345 (2019).

[49] https://github.com/shilei65/adaptive-signal-lasso-code.git.