Knowledge-Based Systems 193 (2020) 105406

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Analyses and applications of optimization methods for complex network reconstruction☆

Xiaomin Wu [a,b,*], Jianshe Wu [a,b], Jixin Zou [c], Qian Zhang [b]

[a] The Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, School of Artificial Intelligence, Xidian University, Xi'an 710071, China
[b] The 20th Research Institute of China Electronics Technology Group Corporation - Xidian University Joint Laboratory of Artificial Intelligence, Xi'an 710068, China
[c] Institute of Forensic science of China, Beijing 100000, China

ARTICLE INFO

Article history:
Received 20 February 2019
Received in revised form 12 December 2019
Accepted 18 December 2019
Available online 24 December 2019

Keywords:
Complex network
Discrete-time dynamics
Regression
Network reconstruction

ABSTRACT

Inferring the topology of a network from observable dynamics is a key topic in the research of complex network. With the observation error considered, the topology inferring is formulated as a connectivity reconstruction problem that can be solved through optimization estimation. It is found that the different optimization methods should be selected to deal with the different degrees of noise, different scales of observable time series and such other situations when it comes to the problem of connectivity reconstruction, which has not been analyzed and discussed before yet. In this paper, four regression methods, namely least squares, ridge, lasso and elastic net, are used to solve the problem of network reconstruction in different situations. In particular, a further analysis is made of the effects of each regression method on the network reconstruction problem in detail. Through simulation of a variety of artificial and real networks, as it has turned out, the four regression methods are effective in respect to network reconstruction when certain conditions are respectively satisfied. Based on the experimental results, it is possible to reach some interesting conclusions that can guide our readers to know the internal mechanisms for network reconstruction and choose the appropriate regression method in accordance with the actual situation and existing knowledge.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Dynamic phenomena such as bifurcation, chaos and unstable oscillations exist in the propagation model of real-world networks. The interaction topology of complex network strongly affects its collective dynamics and thus the function of entire system [1]. In many fields of science and society, one may be encountered with the situation in which the system of interest is composed of networked elements, called nodes, but the pattern of node-to-node interaction or the network topology is far from available. To infer the network connectivity from dynamical observable series is a crucial research on complex systems, which is known as a problem of data mining in practice [2–4]. Its potential applications include: Gene regulation network reconstruction, derived from gene expression data, aims to find the relations between genes, and such relations can help to cure and control genetic diseases [5–7]. Epidemic network reconstruction can help to reveal the process of infectious disease propagation and prevent the spread of other epidemics [8]. Protein–protein network reconstruction has an important significance in understanding the cellular functional mechanism and finding the evolutionary mechanism of organisms [9] and so on.

Methods for inferring topologies based on dynamical observation have been proposed in the recent literatures [2,10–27]. The common observations stem from many kinds of models, such as oscillatory dynamics of chaotic systems, dynamics of systems based on the modeling of evolutionary game, binary data of spreading networks and so on. Almost all of the data-based complex networks or systems can be converted into a signal reconstruction problem which in turn can be solved through adoption of such methods as reverse engineering [28], heuristic search [2,23], compressive sensing [29], regression [30,31] and so on.

Most of the above-mentioned algorithms focus on continuous-time dynamical models to describe the node dynamics [12,14,22]. However, the discrete-time dynamical models are easier to deal

☆ No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to https://doi.org/10.1016/j.knosys.2019.105406.

* Correspondence to: Xidian University, P.O. Box 224, Taibai South Road 2#, Xi'an, Shaanxi 710071, China.
E-mail addresses: wuxiaomin_0110@163.com (X. Wu), jshwu@mail.xidian.edu.cn (J. Wu), 215115170@qq.com (J. Zou), zhangqian6040@163.com (Q. Zhang).

https://doi.org/10.1016/j.knosys.2019.105406
0950-7051/© 2019 Elsevier B.V. All rights reserved.

with and they are as common as the continuous-time models in the real world. The focus of our concern, therefore, is the discrete-time-dynamics-based method for network reconstruction.

In the existing relevant literatures, some algorithms took account of the observation error [19,25,32], while others did not [10,12]. In this paper, the observation error is taken into account, and meanwhile its influence on network reconstruction is analyzed in detail, with the latter known as a blank in the existing literatures.

What is more, there has been no analysis and discussion of how to choose the most appropriate optimization method in dealing with different problems of network reconstruction. However, this move is extremely necessary in that it can help to understand the internal mechanisms for network reconstruction.

In summary, our contributions are listed as follows:

(1) Adoption of the power series expansion method is intended for the dynamics of each node to get a general representation for all nodes, and there is no need for our algorithm to know the specific expressions of chaotic systems in that it can appropriately expand the series to a relatively higher power to hold all terms.

(2) Detailed analysis is made of the effects of the degrees of observation error as well as the scale of observation series on the problem of network reconstruction.

(3) Both analysis and discussion are conducted in the light of the features and group selection effects of several regression methods on the network reconstruction whose measurements contain the observation error.

(4) Some interesting conclusions are reached in the paper. And some advice is given to our readers so that they can make choices of regression methods based on their own needs and existing knowledge when it comes to the problem of network reconstruction.

The rest of this paper is organized as follows: Section 2 is a list of some works in relation to network reconstruction published in recent years. Section 3 is concerned with the problem of formulation. Section 4 is a description of several regression methods as solutions to the problem of network reconstruction. Section 5 deals with our experiments as well as further discussion about the effects of several regression methods on network reconstruction. And the last section serves as the conclusion of this paper.

2. Related works

The past fifteen years have witnessed rapid development of contemporary complex graph theory with broad applications in interdisciplinary science and engineering. The combination of graph, information, and nonlinear dynamical systems theories with tools from statistical physics, optimization, engineering control, applied mathematics, and scientific computing enables the development of a number of paradigms to address the problem of nonlinear and complex systems reconstruction. A detailed review of such methods can be found in Ref. [16]. It reviewed some data-based methods used to address the problem of nonlinear and complex system reconstruction with its focus on compressive sensing method. Now the recent advances are listed in respect to this forefront and rapidly evolving field, with a focus on data-based network reconstruction algorithms. According to the types of observation data, there are three common categories as follows:

Firstly, the complex network is reconstructed with coupled oscillatory dynamics [10,12,19]. The basic idea is that the mathematical functions determining the dynamical couplings in a physical network can be expressed by power-series expansions. The task is then to estimate all the nonzero coefficients, which can be accomplished by some signal reconstruction methods. By presupposing a knowledge of the functional form of the dynamical units and of the coupling functions between them, Shandilya et al. came up with a solution to the inverse problem of finding the network topology through observing a time series of state variables [10]. That is, this method needs to know the specific expressions of chaotic systems in advance. In [12], their ideas were to expand the vector field or map of the underlying system into a suitable function series and then to use the compressive sensing technique [29] to estimate the various terms in the expansion. From their point of view, the functions can be expressed in power series expansions, and the interaction network can be reconstructed through learning the relation between the measured time series and their expanded power series via compressive sensing [29]. Based on adaptive control and the Barbalet's lemma, a response network with a general form was constructed to recover the unknown topology of the considered network by Zhang et al. [14].

Secondly, the complex network is reconstructed with evolutionary game dynamics [2,13,15,23]. Evolutionary game is a common type of interaction in a variety of complex networked, natural and social systems. Given such a system, uncovering the interacting structure of the underlying network is the key to understanding its collective dynamics. Game-data-based methods can always be solved by a signal reconstruction method to uncover the network topology. Wang et al. proposed a method of addressing the problem of how to uncover network topology using evolutionary-game data based on compressive sensing [13]. Ref. [15] decomposed the task of reconstructing the whole network into recovering local structures centered at each node. Wu et al. proposed a two-stage evolutionary algorithm to reconstruct the network node by node. In the first stage, possible vectors corresponding to each node were obtained by a genetic method and in the second stage, the true vector of each node was further confirmed by a heuristic local search [23]. Ref. [2] took the network reconstruction as a non-convex problem and solved this problem by a memetic algorithm.

Thirdly, the complex spreading network is reconstructed from binary data [11,20,25,26]. Among the various types of collective dynamics on complex networks, propagation or spreading dynamics is of paramount importance as it is directly relevant to issues of tremendous interests such as epidemic and disease outbreak in the society and virus spreading on computer networks. Despite stochastic spreading among the nodes, the available time series of a spreading network are always polarized (binary). Li et al. offered a solution to the network reconstruction problem by developing a data-based linearization approach for binary-state dynamics [11]. They decomposed the task of reconstructing the whole network into local structures center at each node and regarded the local structure reconstruction as a sparse signal reconstruction problem that can be addressed by employing a convex optimization method named Lasso [30]. By exploiting the expectation–maximization (EM) algorithm, Ref. [25] developed a method to ascertain the neighbors of any node in the network based on binary data, thereby recovering the full topology of the network.

In addition, there are some other dynamical data-based methods. For example, Alderisio et al. proposed an approach including a systematic algorithm to perform the *ex post* analysis of the reconstructed network and select appropriate cut-off thresholds to remove false positive [19]. A method of integrating QR decomposition and compressed sensing was proposed to solve the reconstruction problem of complex networks with the assistance of the input noise [22]. Ref. [27] took the statistical properties, such as clustering coefficient, into consideration and it investigated

the relationship between clustering property and accuracy of network reconstruction based on small-world networks. However, it aims at small-world network, which makes it very limited.

As seen from the papers mentioned above, the data-based network reconstruction, no matter what kind of observation data is, can always be converted into a solution to a signal reconstruction problem. The common methods such as reverse engineering [28], heuristic search [2,23], compressive sensing [29], regression [30,31] and so on can be used to reconstruct the unknown signal. However, little attention has been paid to how to choose the appropriate reconstruction methods when dealing with complex network in different situations, which cannot give readers good advice on how to select optimization methods. In addition, the internal mechanisms of network reconstruction can be well learned by analyzing the effect of several regression methods on time series, which was neglected by others.

3. Problem formulation

A networked system can be viewed as a large dynamical system that generates oscillatory time series at various nodes when each node is described by oscillatory dynamics. Timme [1] has proved that the network topology precisely controls the dynamics of the nodes. The dynamics of nodes in turn reflect the topological structure of networks. The dynamics of many networks can be described by

$$\boldsymbol{x}_i\left(k+1\right)=\boldsymbol{f}_i^i\left(\boldsymbol{x}_i\left(k\right)\right)+\varepsilon\sum_{j=1}^{N}\boldsymbol{C}_{ij}\boldsymbol{x}_j\left(k\right),\left(i=1,2,\ldots,N\right), \tag{1}$$

where $\boldsymbol{x}_i\left(k\right)=[x_i^1\left(k\right),x_i^2\left(k\right),\ldots,x_i^m\left(k\right)]\in\mathbb{R}^m$ represents the set of externally accessible dynamical component of node i at time $k\in\mathbb{Z}$. The function $\boldsymbol{f}_i^i\left(\cdot\right):\mathbb{R}^m\to\mathbb{R}^m$ is the intrinsic dynamics of node i. N is the number of accessible nodes and $\varepsilon\ (0<\varepsilon<1)$ is the coupling strength. And $\boldsymbol{C}_{ij}$ is the coupling matrix between the dynamical components at nodes i and j which can be denoted by

$$\boldsymbol{C}_{ij}=\begin{pmatrix} c_{ij}^{1,1} & c_{ij}^{1,2} & \cdots & c_{ij}^{1,m}\\ c_{ij}^{2,1} & c_{ij}^{2,2} & \cdots & c_{ij}^{2,m}\\ \vdots & \vdots & \ddots & \vdots\\ c_{ij}^{m,1} & c_{ij}^{m,2} & \cdots & c_{ij}^{m,m}\end{pmatrix}. \tag{2}$$

The term $c_{ij}^{n,l},(n=1,2,\ldots,m;\ l=1,2,\ldots,m)$ in $\boldsymbol{C}_{ij}$ denotes the coupling from the lth component of node j to the nth component of node i. For any two nodes, there are m^2 possible coupling terms. Nodes i and j are coupled if there is at least one nonzero element in the matrix $\boldsymbol{C}_{ij}$, i.e. there is an edge between them in the network. On the contrary, if all the elements of $\boldsymbol{C}_{ij}$ are zeros, there is no edge between nodes i and j. We assume that there is no coupling from node to itself in this paper, that is, $\boldsymbol{C}_{ii}=\boldsymbol{0}$.

3.1. Specific form

For the sake of easy understanding, a specific 1-dimensional (1-D) discrete-time chaotic system is used as an example to explain, and then a general form is given for a variety of discrete-time dynamical models.

The nodes are 1-dimensional (1-D) oscillators ($x\in\mathbb{R}$), then the $m\times m$ coupling matrix $\boldsymbol{C}_{ij}$ degenerates to a scalar.

In (1), it is assumed that $\boldsymbol{x}_i\ (i=1,2,\ldots,N)$ is available observation and $\boldsymbol{C}_{ij}$ is the variable to be solved ultimately. However, it can be found that the function $\boldsymbol{f}_i^i$ is unknown as well, which is common in practice. Thus, it is our goal to get the values of $\boldsymbol{C}_{ij}$ without knowing the exact expression of $\boldsymbol{f}_i^i$.

Concretely, we consider the networks of discrete-time chaotic Logistic system [33]:

$$f\left(x(k)\right)=\mu x(k)(1-x(k)), \tag{3}$$

where $x(k)$ represents the externally accessible dynamical variable at time k. The values of interest for the parameter μ are those in the interval (0, 4]. Different values of μ represent different chaotic behaviors. The details of this chaotic map can be seen in [33].

From (1) and (3), a network of chaotic Logistic oscillators is described by:

$$\begin{aligned} x_i\left(k+1\right)&=\mu x_i(k)(1-x_i(k))+\varepsilon\sum_{j=1}^{N}C_{ij}x_j\left(k\right),\\ &\left(i=1,2,\ldots,N\right).\end{aligned} \tag{4}$$

Since $C_{ii}=0$, (4) can be rewritten as (5),

$$x_i\left(k+1\right)=\mu x_i(k)(1-x_i(k))+\varepsilon\sum_{j=1,j\neq i}^{N}C_{ij}x_j\left(k\right). \tag{5}$$

Our goal is to estimate the value of C_{ij}, which is the coefficient of each $x_j\left(k\right),(j\neq i)$. We define

$$\begin{aligned} f_i^i(x_i(k))&=\mu x_i(k)(1-x_i(k))\\ &=[\beta_i^i]_0+[\beta_i^i]_1x_i(k)+[\beta_i^i]_2(x_i(k))^2\\ &=\sum_{l_1=0}^{2}[\beta_i^i]_{l_1}(x_i(k))^{l_1},\end{aligned} \tag{6}$$

which only contains the dynamics of node i itself. To keep in line with (6), define

$$\begin{aligned} f_j^i(x_j(k))&=\varepsilon C_{ij}x_j\left(k\right)\\ &=[\beta_j^i]_0+[\beta_j^i]_1x_j(k)+[\beta_j^i]_2(x_j(k))^2\\ &=\sum_{l_1=0}^{2}[\beta_j^i]_{l_1}(x_j(k))^{l_1},\\ &\left(j=1,\ldots,N,j\neq i\right),\end{aligned} \tag{7}$$

which only contains the dynamics coupling from node j to i. So (5) can be rewritten as

$$x_i\left(k+1\right)=\sum_{j=1}^{N}f_j^i(x_j(k))=\sum_{j=1}^{N}\sum_{l_1=0}^{2}[\beta_j^i]_{l_1}(x_j(k))^{l_1}. \tag{8}$$

We transfer (8) into matrix formation as follows:

$$x_i(k+1)=\boldsymbol{\varphi}(k)\cdot\boldsymbol{\beta}_i, \tag{9}$$

where $\boldsymbol{\varphi}(\cdot)=[1,x_1,(x_1)^2,\ldots,x_N,(x_N)^2]\in\mathbb{R}^{2N+1}$, $\boldsymbol{\beta}_i=[[\beta^i]_0,[\beta_1^i]_1,[\beta_1^i]_2,\ldots,[\beta_N^i]_1,[\beta_N^i]_2]^{\mathrm{T}}\in\mathbb{R}^{2N+1}$, and $[\beta^i]_0=\sum_{j=1}^{N}[\beta_j^i]_0$ for constant term. From (9), it can be found that the power series expansion method can expand the dynamics of each node to a series of terms with the same base (the elements in $\boldsymbol{\varphi}(\cdot)$). And the neighbors of node i are totally contained in $\boldsymbol{\beta}_i$. The problem of network reconstruction can be transferred to solve the linear model of (9).

In fact, a relative higher order of power can guarantee that all terms of chaotic system are included at the cost of a higher computational burden. Here once the highest power of expansion terms is chosen as more than 2, the dynamics of each node can be represented accurately without knowing the specific expression of chaotic system.

But in practice of signal processing, the observation error during the experiment is unavoidable. It is assumed that the observation error is $\eta_i\sim N(0,\sigma^2)$, then the measurements can

be represented as $y_i = x_i + \eta_i$. Thereafter, for q measurements of node i, we have

$$\begin{pmatrix} y_i(2) \\ \vdots \\ y_i(q+1) \end{pmatrix} = \begin{pmatrix} \widetilde{\boldsymbol{\varphi}}(1) \\ \vdots \\ \widetilde{\boldsymbol{\varphi}}(q) \end{pmatrix} \cdot \widetilde{\boldsymbol{\beta}}_i, \tag{10}$$

where $\widetilde{\boldsymbol{\varphi}}(\cdot) = [1, y_1, (y_1)^2, \ldots, y_N, (y_N)^2] \in \mathbb{R}^{2N+1}$. The initial value of each node $y_i(1)$ is assumed to be in the range (0, 1) without loss generalization. Let $\boldsymbol{y}_i = [y_i(2), \ldots, y_i(q+1)]^{\mathrm{T}} \in \mathbb{R}^q$, and $\widetilde{\boldsymbol{\Phi}} = [\widetilde{\boldsymbol{\varphi}}^{\mathrm{T}}(1), \ldots, \widetilde{\boldsymbol{\varphi}}^{\mathrm{T}}(q)]^{\mathrm{T}} \in \mathbb{R}^{q\times(2N+1)}$, (10) can be rewritten as the formation of

$$\boldsymbol{y}_i = \widetilde{\boldsymbol{\Phi}} \cdot \widetilde{\boldsymbol{\beta}}_i. \tag{11}$$

Once we compute $\widetilde{\boldsymbol{\beta}}_i$, the adjacency relations of node i with other nodes can be revealed by observing whether the corresponding terms of other nodes are zeros or not (compare the terms with a threshold δ, and set zero if the term is smaller than δ). For example, if the terms corresponding to node j have nonzero elements, there exists an edge between i and j. Otherwise, there exists no edge between i and j. Like this, the adjacency relations of other nodes can be revealed as well. Thus the network construction can be completed.

3.2. General form

It is evident that many chaotic systems, such as discrete-time Rössler system [34], Hénon map [35], Tinkerbell map [36], are polynomials of various kinds of power terms. In this section, these polynomial discrete systems will be taken into account so as to build a more general model. For each node i, it is defined that the nth component of $\boldsymbol{f}_i^i(\boldsymbol{x}_i)$ in (1) as $\left[\boldsymbol{f}_i^i(\boldsymbol{x}_i)\right]_n$ and then expand it by a power series of order up to L:

$$\left[\boldsymbol{f}_i^i(\boldsymbol{x}_i)\right]_n = \sum_{l_1=0}^{L} \cdots \sum_{l_m=0}^{L} \left[\left(\beta_i^i\right)_n\right]_{l_1,\ldots,l_m} \left(x_i^1\right)^{l_1} \cdots \left(x_i^m\right)^{l_m}, \tag{12}$$

where $x_i^n\,(n = 1, \ldots, m)$ is the nth component of the dynamical variable at node i. And $\left[\left(\beta_i^i\right)_n\right]_{l_1,\ldots,l_m}$, the coefficient of each product term, is to be determined from measurements.

To contain as many coupling terms as possible, it is defined that the nth component of each term $\boldsymbol{C}_{ij}\boldsymbol{x}_j(k)\,(j \neq i)$ in (1) as a power series with the similar form of $[\boldsymbol{f}_i^i(\boldsymbol{x}_i)]_n$ in (12) as follows:

$$\begin{aligned} \left[\boldsymbol{f}_j^i\left(\boldsymbol{x}_j\right)\right]_n &= \left[\boldsymbol{C}_{ij}\boldsymbol{x}_j\right]_n \\ &= \sum_{l_1=0}^{L} \cdots \sum_{l_m=0}^{L} \left[\left(\beta_j^i\right)_n\right]_{l_1,\ldots,l_m} \left(x_j^1\right)^{l_1} \cdots \left(x_j^m\right)^{l_m}, \\ &\quad (j = 1, \ldots, N, j \neq i)\,. \end{aligned} \tag{13}$$

Then get

$$x_i^n(k+1) = \sum_{j=1}^{N} [\boldsymbol{f}_j^i(\boldsymbol{x}_j(k))]_n. \tag{14}$$

For each dynamical component x_i^n, the observation error $\eta_i^n \sim N(0, \sigma^2)$ is taken into consideration, that is $y_i^n = x_i^n + \eta_i^n$. Similar to (8)–(10), the model can be deduced from (14) as

$$\boldsymbol{y}_i^n = \widetilde{\boldsymbol{\Phi}} \cdot \widetilde{\boldsymbol{\beta}}_i^n \tag{15}$$

where $\boldsymbol{y}_i^n = [y_i^n(2), \ldots, y_i^n(q+1)]^{\mathrm{T}} \in \mathbb{R}^q$, $\widetilde{\boldsymbol{\Phi}} = [\widetilde{\boldsymbol{\varphi}}^{\mathrm{T}}(1), \ldots, \widetilde{\boldsymbol{\varphi}}^{\mathrm{T}}(q)]^{\mathrm{T}} \in \mathbb{R}^{q\times p}$ with $\widetilde{\boldsymbol{\varphi}}(k) = \{(y_i^1(k))^{l_1} \cdots (y_i^m(k))^{l_m}\} \in \mathbb{R}^p$, $(i = 1, \ldots, N; l_n = 0, \ldots, L; n = 1, \ldots, m)$. p is the total number of power series which are served as features. After the coefficients $\widetilde{\boldsymbol{\beta}}_i^n \in \mathbb{R}^p$ of all components for node i are determined, the adjacency relations of node i with other nodes can be revealed by observing whether the corresponding terms of other nodes are zeros or not (compare the terms with a threshold δ, and set zero if the term is smaller than δ). Repetition of this process holds true for all other nodes.

4. Regression methods to infer the connectivity coefficients

Some classical regression methods, such as least squares, ridge regression [37], lasso [30], elastic net [38] and so on [31], can work out of $\widetilde{\boldsymbol{\beta}}_i$ from $\boldsymbol{y}_i = \widetilde{\boldsymbol{\Phi}} \cdot \widetilde{\boldsymbol{\beta}}_i$ with appropriate parameters respectively. A brief discussion will be made about the least squares, ridge regression, lasso and elastic net in the following. Actually, they differ from each other with different regularization terms.

The least squares model is as follows:

$$\boldsymbol{\beta}_i^{ls} = \arg\min_{\boldsymbol{\beta}_i}\{\frac{1}{2q}\left\|\boldsymbol{y}_i - \widetilde{\boldsymbol{\Phi}} \cdot \boldsymbol{\beta}_i\right\|_2^2\}, \tag{16}$$

where q is the number of the measurement data. $\|\cdot\|_2$ is L_2-norm. The closed solution to least squares is

$$\boldsymbol{\beta}_i^{ls} = (\widetilde{\boldsymbol{\Phi}}^{\mathrm{T}}\widetilde{\boldsymbol{\Phi}})^{-1}\widetilde{\boldsymbol{\Phi}}^{\mathrm{T}}\boldsymbol{y}_i. \tag{17}$$

Least squares is easy to solve for its closed solution. However, when there are many correlated nodes, which is common in some real networks such as gene network, it will lead to many correlated features in matrix $\widetilde{\boldsymbol{\Phi}}$, and their coefficients can become poorly determined. In addition, the least squares method needs to meet the requirement that the number of measurements cannot be less than the number of features. It means that enough measurements are needed to reconstruct the network by the least squares.

By imposing a size constraint on the coefficients as in (16), named ridge regression [37] in (18), the above-mentioned problem of least squares is alleviated.

$$\boldsymbol{\beta}_i^{ridge} = \arg\min_{\boldsymbol{\beta}_i}\{\frac{1}{2q}\left\|\boldsymbol{y}_i - \widetilde{\boldsymbol{\Phi}} \cdot \boldsymbol{\beta}_i\right\|_2^2 + \lambda\left\|\boldsymbol{\beta}_i\right\|_2^2\}, \tag{18}$$

where λ is a nonnegative regularization parameter which can be determined by cross-validation [31]. Its closed solution is as follows:

$$\boldsymbol{\beta}_i^{ridge} = (\widetilde{\boldsymbol{\Phi}}^{\mathrm{T}}\widetilde{\boldsymbol{\Phi}} + \lambda\boldsymbol{I})^{-1}\widetilde{\boldsymbol{\Phi}}^{\mathrm{T}}\boldsymbol{y}_i, \tag{19}$$

where $\boldsymbol{I} \in \mathbb{R}^{p\times p}$ is an identity matrix to prevent $\widetilde{\boldsymbol{\Phi}}^{\mathrm{T}}\widetilde{\boldsymbol{\Phi}}$ noninvertible. Ridge regression is also easy to solve for its closed solution. However, it shrinks the coefficients to smaller values rather than zero, which is not very suitable for most of the existing sparse network reconstruction problem.

Similar to ridge regression, lasso includes a penalty term that constraints the size of the estimated coefficients as follows [30]:

$$\boldsymbol{\beta}_i^{lasso} = \arg\min_{\boldsymbol{\beta}_i}\{\frac{1}{2q}\left\|\boldsymbol{y}_i - \widetilde{\boldsymbol{\Phi}} \cdot \boldsymbol{\beta}_i\right\|_2^2 + \lambda\left\|\boldsymbol{\beta}_i\right\|_1\}, \tag{20}$$

where $\|\cdot\|_1$ is L_1-norm. In the experiments, we use L_1-norm of the coefficient t ($\left\|\boldsymbol{\beta}_i\right\|_1 \leq t$) as tuning parameter instead of penalty term λ in (20). As the increase of t, lasso sets more coefficients to zeros. This feature selection function allows the number of measurements to be much smaller than the number of features. Thus lasso can be used to solve sparse network reconstruction problem.

The elastic net penalty is a compromise of ridge and lasso [38]. It has the form as follows:

$$\boldsymbol{\beta}_i^{en} = \arg\min_{\boldsymbol{\beta}_i}\{\frac{1}{2q}\left\|\boldsymbol{y}_i - \widetilde{\boldsymbol{\Phi}} \cdot \boldsymbol{\beta}_i\right\|_2^2 + \lambda_1\left\|\boldsymbol{\beta}_i\right\|_1 + \lambda_2\left\|\boldsymbol{\beta}_i\right\|_2^2\}, \tag{21}$$

where both λ_1 and λ_2 are tuning parameters which are chosen by cross-validation [31]. Similar to the Lasso, (t, λ_2) serves to parameterize the elastic net in the later experiments. The second regularization term λ_2 encourages highly correlated features to be averaged while the first regularization term t encourages a sparse solution in the coefficients of these averaged features. This means that elastic net has both the feature selection function and the group selection function.

Since both lasso and elastic net have no closed solutions, their problems are more complicated to solve than those of least squares and ridge. Here, the lasso and elastic net methods can be solved by LARS [39] and LARS-EN [38].

In order to further enhance the readability of our paper, our network reconstruction method can be summarized in Algorithm 1 as follows.

Algorithm 1 Network reconstruction process

Input:

The order of power series L and a threshold δ;
q-dimensional measurements of each node $\boldsymbol{y}_i^n, (n = 1, \ldots, m; i = 1, \ldots, N)$ in Eq. (15).

Output:

The topology of network.

1: Construct $\widetilde{\boldsymbol{\Phi}} = [\widetilde{\boldsymbol{\varphi}}^{\mathrm{T}}(1), \ldots, \widetilde{\boldsymbol{\varphi}}^{\mathrm{T}}(q)]^{\mathrm{T}} \in \mathbb{R}^{q\times p}$, where $\widetilde{\boldsymbol{\varphi}}(k) = \{(y_i^1(k))^{l_1} \cdots (y_i^m(k))^{l_m}\} \in \mathbb{R}^p, (i = 1, \ldots, N; l_n = 0, \ldots, L; n = 1, \ldots, m)$;
2: For each component of node i, solve the $\boldsymbol{y}_i^n = \widetilde{\boldsymbol{\Phi}} \cdot \widetilde{\boldsymbol{\beta}}_i^n$ by any regression method to obtain the estimation of $\widetilde{\boldsymbol{\beta}}_i^n$;
3: Compare the corresponding coefficients of each other node $j(j = 1, \ldots, N; j \neq i)$ in the estimation of $\widetilde{\boldsymbol{\beta}}_i^n$ with δ, and get the connectivity information of node i with other nodes;
4: Repeat 2∼3 until the whole network is reconstructed.

5. Analysis and discussion of the internal mechanisms of network reconstruction

To verify our proposed method and the application of four optimization methods for complex network reconstruction, some experiments are conducted in this section. Then for each experiment, analysis is made of the laws and mechanisms behind it in detail. And some interesting rules will be found herein.

5.1. Test networks

Some test networks involved in our experiments include LFR Benchmark [40], random network (ER) [41], small-world (SW) network [42], scale-free network (SF) [43], some real networks like Karate club network [44], Dolphins social network [45], Books about US politics (Polbook) [46], American college football (Football) [47], co-authorships network of scientists (Netscience) [48].

5.2. Parameter settings

In our paper, we take 1-D discrete-time Logistic map [33] and 2-Dimensional (2-D) discrete-time Hénon map [35] as examples to describe the dynamics of complex networks. It is assumed that the dynamics of certain types of networks conform these two different models. In fact, our algorithm is also applicable to dynamical networks that conform to other chaotic models such as discrete-time Lorenz [49], Rössler system [34] and so on.

Logistic map can be seen in Eq. (4). The parameter μ can be set to the range of (0, 4], we set $\mu = 3.7$ in order to exhibit chaotic behavior and avoid divergence. In this paper, the initial measurement for each node is randomly set to the range of (0,1).

From the expanded form of (4), it can be found that the powers of all terms are no more than 2. For simplicity, we assume the component as x (the number of components $m = 1$) and choose the power series for it such that $l_s = \{1, 2\}$, that means x, x^2. Thus for each node i, the number of the coefficients to be estimated is $p = N \cdot \max(l_s) + 1 = 2N + 1$, where N is the number of nodes, $\max(\cdot)$ is the maximum operation and "1" represents the constant term (x^0). The coupling between node dynamics is assumed to occur between the variables in the Logistic system, leading to the following coupling matrix: $C_{ij} = 1$ if nodes i and j are connected and $C_{ij} = 0$, otherwise. We set the coupling strength $\varepsilon = 0.005$ for Logistic map.

We take 2-D Hénon map [35] for an example and assume that the components are x and y. Here lists the 2-D Hénon map:

$$\begin{cases} x(k+1) = 1 - 1.4x^2(k) + y(k) \\ y(k+1) = 0.3x(k). \end{cases} \tag{22}$$

To ensure the divergence of measurements data, the initial values of Hénon map in this paper are set to continuous uniform distribution of [0, 0.631] and [0, 0.189], respectively.

From (13), we should choose the power series for the components as $0 \le l_1 \le 2$ and $0 \le l_2 \le 2$, that means 1, x, y, x^2, y^2, xy, xy^2, x^2y, x^2y^2. However, it can be found that the powers of all terms are no more than 2 in (22). Thus for simplicity, we choose the power series for the components such that $0 \le l_1 + l_2 \le 2$, that means 1, x, y, x^2, y^2, xy. Thus the total number of the coefficients to be estimated is $p = N\sum_{l_s=1}^{2}(l_s + 1) + 1 = 5N + 1$, where "1" represents the constant term (x^0y^0) and $l_s = l_1 + l_2 = \{1, 2\}$. The coupling between node dynamics is assumed to occur between y and y variables in the Hénon system, leading to the following coupling matrix: $C_{ij}^{2,2} = 1$ if nodes i and j are connected and $C_{ij}^{2,2} = 0$, otherwise. We set the coupling strength $\varepsilon = 0.005$ for Hénon map.

In fact, we can set the upper limit of l_s to a slightly larger number (e.g. 4 or 5) to ensure all terms to be contained in the expanded power series for both chaotic systems. At this time, there is no need to know the specific expressions of the chaotic systems, but at the expense of some operational efficiency. Since the core of this paper is not the choice of chaotic models, we will not discuss the models and their parameters in detail.

Set the threshold δ slightly smaller than ε such as $\varepsilon - 0.01 \times \varepsilon$ for all our following experiments.

5.3. Metric indices

In statistics, the mean squared error (MSE) of an estimator measures the average of the squares of the errors. That is, the average squared difference between the estimated value $(\hat{\boldsymbol{\beta}})$ and the actual value $(\boldsymbol{\beta})$. The MSE is always non-negative, and values closer to zero are better. It can be used to measure the difference of structures between the truth network and the reconstructed network. And it is the sum of variance and the squared bias as follows:

$$MSE(\hat{\boldsymbol{\beta}}) = E[(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^2] = Var(\hat{\boldsymbol{\beta}}) + Bias(\hat{\boldsymbol{\beta}}, \boldsymbol{\beta})^2, \tag{23}$$

where $E[\cdot]$ is the mean operation, $Var(\hat{\boldsymbol{\beta}}) = E[(\hat{\boldsymbol{\beta}} - E(\hat{\boldsymbol{\beta}}))^2]$ is the variance of the estimated value and $Bias(\hat{\boldsymbol{\beta}}, \boldsymbol{\beta})^2 = (E(\hat{\boldsymbol{\beta}}) - \boldsymbol{\beta})^2$ is the squared bias between the estimated value and the actual value.

In addition, four metrics are used to assess the performance of the network reconstruction algorithm. They are True Positive Rate (TPR, it is Recall as well), True Negative Rate (TNR), Accuracy Rate (ACC), and Positive Predicted Value (PPV, it is Precision as

well). Let N_{TP}, N_{FP}, N_{TN}, N_{FN} and $N_{Total} = N(N-1)/2$ be the total number of true positive, false positive, true negative, false negative and possible links among all the nodes in the network (undirected network applies), respectively. These metrics are defined as follows:

$$\begin{cases} \text{TPR} = \dfrac{N_{TP}}{N_{TP}+N_{FN}} \\ \text{TNR} = \dfrac{N_{TN}}{N_{TN}+N_{FP}} \\ \text{ACC} = \dfrac{N_{TP}+N_{TN}}{N_{Total}} \\ \text{PPV} = \dfrac{N_{TP}}{N_{TP}+N_{FP}}. \end{cases} \tag{24}$$

The larger the ACC, PPV, TPR and TNR values are, the better the reconstruction performance is. The network reconstruction is accomplished just when the values of TPR and TNR both reach 1 or the value of ACC reaches 1.

5.4. Effects of the measurement error on regression methods

Measurement error can affect the reconstruction of networks. To test the effects of measurement error on four regression methods, experiments are conducted on a variety of test networks.

Here we choose the Dolphins networks on 1-D Logistic map and the LFR Benchmark network on 2-D Hénon map to show the results, respectively. Set the parameters of ridge, lasso and elastic net as 10^{-6}, 50 and $(50, 10^{-6})$ for Dolphins network, respectively and as 10^{-6}, 100 and $(100, 10^{-6})$ for LFR Benchmark network, respectively. The results are shown in Figs. 1 and 2. Each experimental result is an average of 100 repetitions in the following parts.

Fig. 1(a) shows the bars of the squared bias and variance of the reconstructed Dolphins network on 1-D Logistic map. And Fig. 1(b) shows the bars of the squared bias and variance of the reconstructed LFR Benchmark network on 2-D Hénon map. The summation of the squared bias and variance is MSE which represents the estimated error of $\boldsymbol{\beta}_i$. The more accurate the estimation of $\boldsymbol{\beta}_i$ is, the higher the success rate of network reconstruction is.

It can be seen from the figures that the values of squared bias, variance and MSE of the four regression methods grow with the increase of σ. By comparing two figures in Fig. 1(a) or Fig. 1(b), we can also find that both the values of squared bias and variance grow rapidly with the increase of σ in the case of the least squares and ridge methods, while growing slowly in the case of the lasso and elastic net methods.

With the measurement error considered and the matrix $\widetilde{\boldsymbol{\Phi}}$ composed of measurement data, all four regression methods become biased estimators in estimating the $\widetilde{\boldsymbol{\beta}}_i$ of $\boldsymbol{y}_i = \widetilde{\boldsymbol{\Phi}} \cdot \widetilde{\boldsymbol{\beta}}_i$. Each bar of squared bias in Fig. 1 demonstrates this.

Fig. 2(a) shows the curves of the TPR, TNR, ACC and PPV of the Dolphins network on 1-D Logistic map. And Fig. 2(b) shows the curves of the same metrics of the LFR Benchmark network on 2-D Hénon map.

In the same range of noise, as shown in Fig. 2, lasso and elastic net can accomplish the reconstruction of the test networks while it becomes more and more difficult for the least squares and ridge to finish reconstruct of the network as the noise increases.

In summary, four regression methods are biased estimators when the measurement error is taken into consideration. The increase of measurement error results in the increases of bias and variance first, and then in the changes of MSE, TPR, TNR, ACC and PPV. And a conclusion can be drawn that the network structure, as estimated by least squares and ridge regression, will have more and more deviation from the real network structure along with the increase of measurement error, whereas lasso and elastic net are relatively insensitive to a certain range of measurement error.

5.5. Feature selection function on network reconstruction

Since lasso and elastic net have L_1-norm penalties, they both have the function of feature selection. We define #Data $= q/p$ as the ratio of the measurements to features, which can represent the ability of feature selection. Here q is the number of measurements and p is a fixed value for a particular network (for example, $p = 2N + 1 = 125$ for Dolphins network on 1-D Logistic map). Lasso and elastic net can still estimate $\widetilde{\boldsymbol{\beta}}_i$ corresponding to each node accurately when $q \ll p$, and further reconstruct the network structure completely. But least squares and ridge methods do not have the feature selection function, enough measurements are needed to completely reconstruct the networks for these two methods. Thus, #Data can represent the proportion of measurements required for completely network reconstruction and it can also represent the performance of feature selection directly. The less measurements are needed, the more robust the algorithm is.

In order to test the effect of #Data on the network reconstruction by the four regression methods, we do some experiments of Dolphins network on 1-D Logistic map. Here fix the observation error as 5×10^{-4}, set the parameters of ridge, lasso, and elastic net as 10^{-6}, 50, $(50, 10^{-6})$, respectively. ACC is used to evaluate our results.

The effect of the available measurements on the network reconstruction by the four regression methods is shown in Fig. 3. Its vertical axis represents ACC, and the horizontal axis is #Data.

Fig. 3(a) indicates the relationships between #Data and ACC when the network is reconstructed with the least squares and ridge methods. Fig. 3(b) indicates the relationships by the lasso and elastic net methods. As can be seen from Fig. 3, the more number of edges is correctly reconstructed as the #Data (available measurements) increases. By comparing (a) and (b) of Fig. 3, it is also found that the least squares and ridge need more measurements to completely reconstruct the network than the lasso and elastic net do.

Tables 1 and 2 show the values of #Data and the corresponding running time by Matlab R2014b when different networks are completely reconstructed by four regression methods on 1-D Logistic map and 2-D Hénon map respectively. The least squares and ridge are solved by computing Eqs. (16) and (18) directly (set the parameter of ridge as 10^{-6}). The lasso and elastic net are solved by LARS [39] and LARS-EN [38] respectively (set the parameter λ_2 of elastic net as 10^{-6}, set t according to the scale of networks). Polbook, Football and Netscience are directed networks by nature. But we handle them into undirected networks herein. The configurations of our computer are as follows: Intel (R) Core (TM) i3-4170 CPU @3.7 GHz, 12G RAM and 64-bit operating system.

From Tables 1 and 2, it can be found that the required running time for each regression method increases along with the gradual increase of N. When the required #Data is relatively large (e.g. #Data $>$ 0.1) for a specific test network by lasso and elastic net, least squares and ridge require less running time than lasso and elastic net. However, according to some results in Table 2, we find that when the required #Data is small enough (i.e. $q \ll p$, e.g. #Data $<$ 0.1) for a specific test network by lasso and elastic net, the running time of least squares and ridge increase dramatically, even several times more than the lasso and elastic net. This is due to the fact that the least squares and ridge methods require $O(p^3)$ operations, but the lasso and elastic net methods require $O(pt^2 + t^3)$ operations for their early stopping after t steps. When q is slightly smaller than p (i.e. #Data is large), the value of t is close to or even greater than p, then $O(pt^2 + t^3)$ is close to or even greater than $O(p^3)$. But when $q \ll p$, LARS and LARS-EN can be set to stop after t $(t < p)$ steps (early stopping). At this time $O(pt^2 + t^3) = O(pt^2) < O(p^3)$.

Tables 1 and 2 verify that lasso and elastic net have the functions of feature selection, while the least squares and ridge have

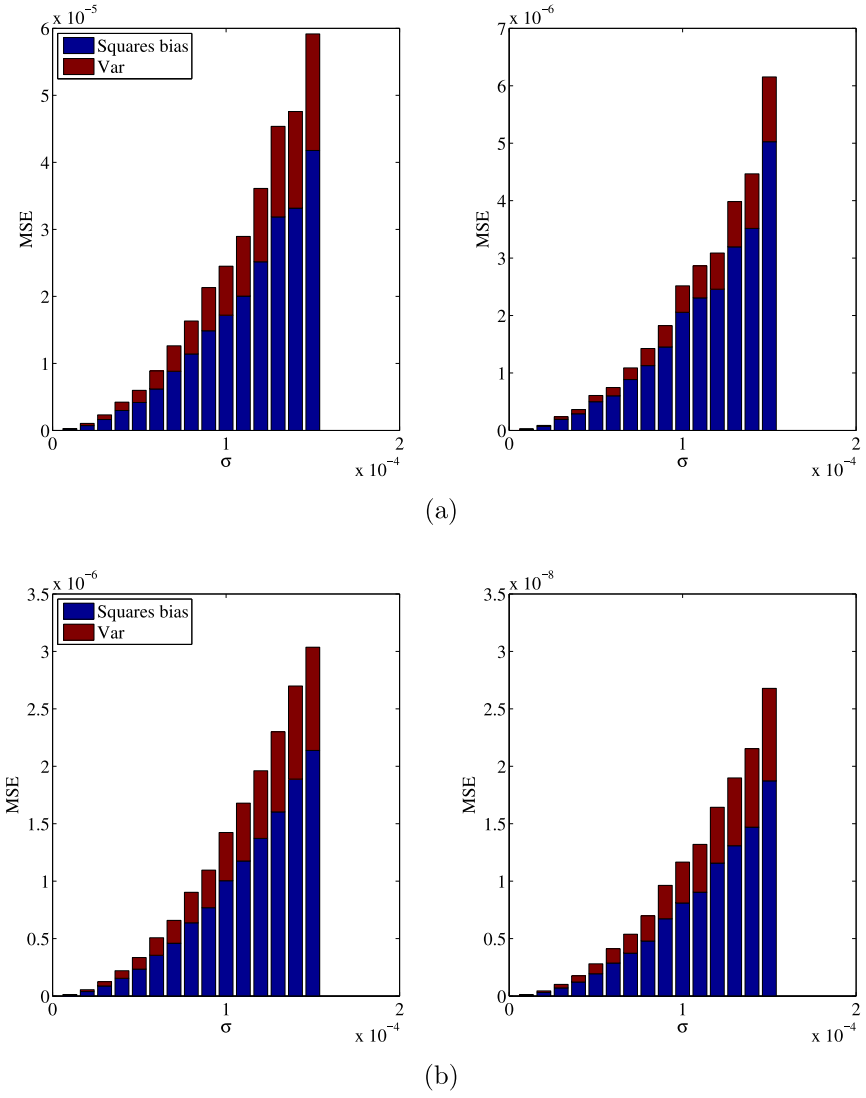

Fig. 1. Changes of Squared bias and Variance with increase measurement errors in the case of four regression methods. (a) Dolphins network ($N = 62$) on 1-D Logistic map. (b) LFR Benchmark network ($N = 100$) on 2-D Hénon map. For (a) and (b), the left ones of them mean the bars of least squares or ridge, the right ones mean the bars of lasso or elastic net. That is, the regression results of least square and ridge are same with each other under the specific parameter settings and enough measurements, and the results of lasso and elastic net are consistent under the specific parameter settings and enough measurements.

Table 1
Required #Data and running time of each test network when it is reconstructed completely with four regression methods. (1-D Logistic system, $\sigma = 10^{-4}$ for least squares and ridge, $\sigma = 10^{-3}$ for lasso and elastic net. For each regression method, the left column is #Data and the right column is running time (Unit: seconds)). N is the scale of network, and $\langle k \rangle$ means the average degree. Least squares and ridge need more observations but less time cost, while lasso and elastic net need less observations but more time cost.

Network	N	$\langle k \rangle$	Least squares		Ridge		Lasso		Elastic net	
			#Data	Time	#Data	Time	#Data	Time	#Data	Time
Karate [44]	34	4.6	1.03	0.03	1.23	0.02	0.9	0.49	0.85	0.34
SF [43]	50	3.76	1.01	0.06	1.19	0.1	0.75	1.09	0.72	1.19
Dolphins [45]	62	5.1	1.05	0.15	1.25	0.06	0.65	2.13	0.65	2.35
ER [41]	100	14.86	1.04	0.5	1.26	0.25	0.62	6.6	0.62	6.83
Polbook [46]	105	8.4	1.04	0.56	1.29	0.27	0.61	7.58	0.61	7.53
Football [47]	115	10.66	1.05	0.84	1.28	0.39	0.51	9.7	0.51	8.82
LFR Benchmark [40]	100	10.1	1.04	0.55	1.25	0.28	0.69	6.65	0.69	7.18
	500	14.8	1.03	37.88	1.26	36.93	0.46	94.76	0.46	102.69
SW [42]	300	4	1.05	8.01	1.3	5.96	0.36	29.83	0.36	25.11

no such function. In addition, as the size of network increases, #Data corresponding to lasso and elastic net methods gradually decrease. This is because p is proportional to N (we assume $p = C_1N + 1$, here C_1 is a constant and $C_1 > 1$), and the increase of N will lead to the increase of p. Meanwhile, there is $q < p$ or even $q \ll p$, and the increase of N will also lead to the increase of q by assuming $q = C_0N$ (C_0 is a constant and $C_0 < C_1$). However, the increase of q with the growth of N is not faster than p with the growth of N, then #Data will become smaller and smaller with the growth of N. The requirement of an exceptionally small

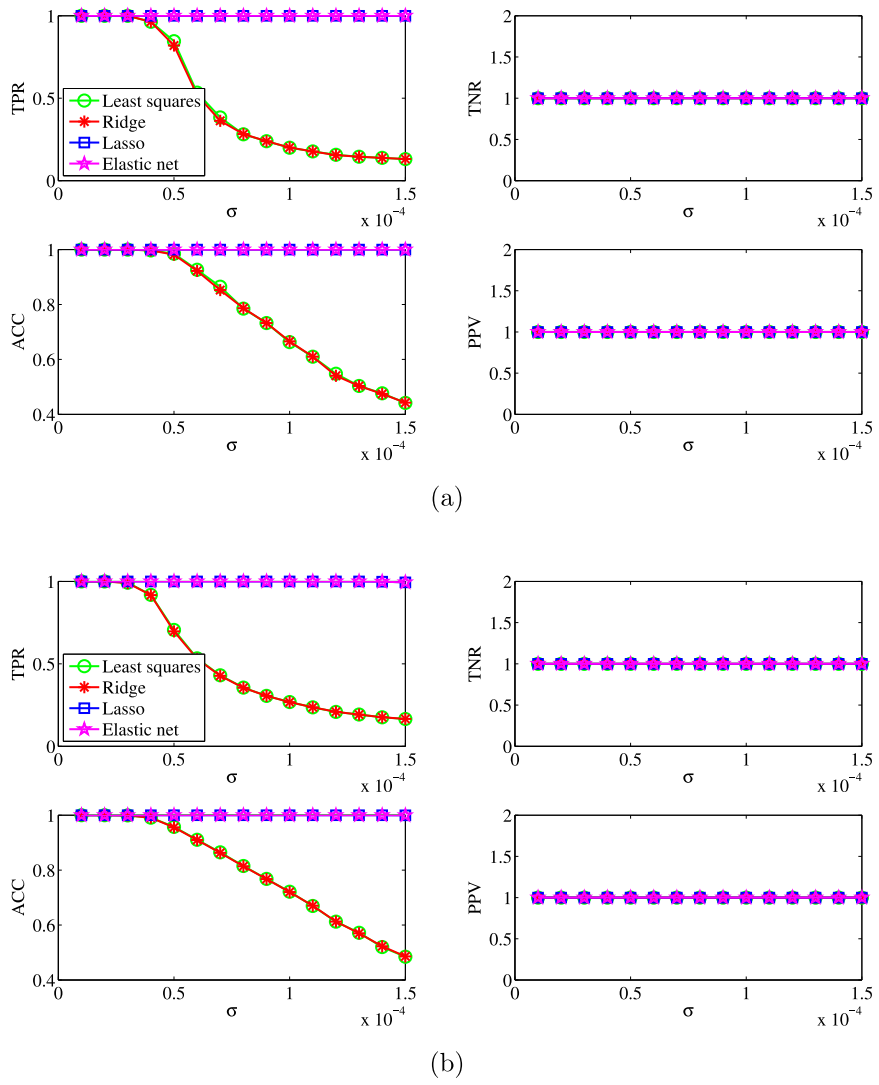

Fig. 2. Curves of four metrics named TPR, TNR, ACC and PPV with increasing measurement error on four regression methods. (a) Dolphins network ($N = 62$) on 1-D Logistic map. (b) LFR Benchmark network ($N = 100$) on 2-D Hénon map. The TPR, TNR, ACC and PPV of the lasso and elastic net methods are always 1 as the σ increases within limits, indicating that a certain range of observation errors have no effect on the lasso and elastic net to completely reconstruct the networks. But when σ reaches to a certain extent (e.g. $\sigma = 0.3 \sim 0.4 \times 10^{-4}$), least squares and ridge are impossible to reconstruct the network completely.

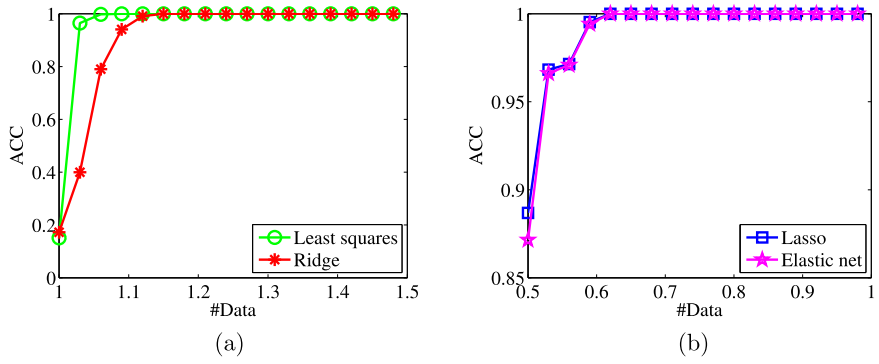

Fig. 3. ACC values of the reconstructed Dolphins network versus different #Data (available measurements) on 1-D Logistic map. (a) Least squares and ridge. (b) Lasso and elastic net. More edges are correctly reconstructed as the #Data increases for all regression methods. Specifically, the least squares and ridge need #Data of 1.09 and 1.15 at least to reconstruct the network completely respectively, whereas the lasso and elastic net, which have the feature selection function, only need #Data of 0.62 to reconstruct the network completely..

amount of data is particularly important for situations where only rare information is available.

By comparing Table 1 with Table 2, we can also find that #Data corresponding to the same network on 2-D Hénon map is

Table 2
Required #Data and running time of each test network when it is reconstructed completely with four regression methods. (2-D Hénon, $\sigma = 10^{-4}$ for least squares and ridge, $\sigma = 10^{-3}$ for lasso and elastic net. For each regression method, the left column is #Data and the right column is running time (Unit: seconds)). Lasso and elastic net need less measurements than least squares and ridge for each test network. Least squares and ridge need less running time than lasso and elastic net do when the scale of test network is small, while the opposite is true when the scale of test network is large than 500.

Network	N	$\langle k \rangle$	Least squares		Ridge		Lasso		Elastic net	
			#Data	Time	#Data	Time	#Data	Time	#Data	Time
Karate [44]	34	4.6	1.02	0.09	1.08	0.05	0.34	0.27	0.34	0.28
SF [43]	50	3.76	1.04	0.27	1.07	0.16	0.28	0.65	0.28	0.61
Dolphins [45]	62	5.1	1.04	0.37	1.1	0.28	0.23	1.19	0.23	1.27
ER [41]	100	14.86	1.05	1.78	1.1	1.27	0.21	4.97	0.21	5.41
Polbook [46]	105	8.4	1.04	2.51	1.1	1.58	0.2	5.64	0.2	7.19
Football [47]	115	10.66	1.05	3.0	1.1	2.19	0.18	6.53	0.18	8.02
LFR Benchmark [40]	100	10.1	1.05	1.69	1.1	1.32	0.21	5.18	0.21	5.45
	500	14.8	1.04	331.22	1.1	360.22	0.07	65.43	0.07	68.9
	800	25.3	1.04	2038.24	1.09	2002.7	0.06	222.55	0.06	223.8
SW [42]	500	12	1.04	343.36	1.09	318.24	0.07	64.4	0.07	65.45
Netscience [48]	1589	3.45	1.04	21677.03	1.1	21933.64	0.05	816.49	0.05	841.72

smaller than on 1-D Logistic map. This is because 2-D Hénon map can generate 2-component measurements with more information than 1-component measurements which are generated by 1-D Logistic map.

Therefore, when the size of networks is relatively large and the observations are not enough, we can consider the lasso and elastic net for network reconstruction. When the size of networks is relatively small and there are enough observations, we can consider the least squares and ridge to reconstruct the networks for their lower computational burdens. And note that when the size of network is large and the available measurements are relatively small, the lasso and elastic net have lower computational burdens than the least squares and ridge methods for their early stopping functions.

5.6. The relationship of measurement data and error

During the experiments, it can be found that for any network, the network reconstructing rate can be improved by increasing the measurements when the observation error increases. Fig. 4(a) and 4(b) show the changes of #Data with the increase of observation error under the premise of a 95% or higher reconstructing success rate (TPR $\geq$ 0.95 and TNR $\geq$ 0.95) on 1-D Logistic map and 2-D Hénon map, respectively. Test networks are Karate [44] and SF network [43], respectively. For Karate network, we set the corresponding parameters of the ridge, lasso and elastic net as 10^{-6}, 20 and (20, 10^{-6}). For SF network, we set the corresponding parameters of the ridge, lasso and elastic net as 10^{-6}, 50 and (50, 10^{-6}).

The left ones of Fig. 4(a) and 4(b) show the curves of the least squares and ridge. The right ones of Fig. 4(a) and 4(b) show the curves of the lasso and elastic net.

It can be seen from Fig. 4 that, regardless of the regression method, chaotic map and network type, when the measurement error increases, the reconstruction rate can be guaranteed by appropriately increasing the number of measurement data. This is because more measurement data can bring more effective information and appropriately reduce the problem of data inaccuracy due to the increase of measurement error.

5.7. Group selection function on network reconstruction

The ability of selecting "grouped" variables, a property is not shared by the lasso, but is owned by elastic net. Now we will analyze the effect of group selection on the problem of network reconstruction.

Qualitatively speaking, a regression method exhibits the grouping effect if the regression coefficients of a group of highly correlated variables tend to be equal. Actually in this paper, the power series expansion method will introduce some relevant columns in $\widetilde{\boldsymbol{\Phi}}$, which will lead to the similar coefficients in $\widetilde{\boldsymbol{\beta}}_i$ corresponding to these relevant columns. Specifically, for each node i, the coefficients corresponding to node j in $\widetilde{\boldsymbol{\beta}}_i$ will be similar or even the same. However, all coefficients corresponding to node j in $\widetilde{\boldsymbol{\beta}}_i$ serve as a basis for judging whether there is a coupling from node j to i. Thus the use of elastic net to solve the coefficient vector $\widetilde{\boldsymbol{\beta}}_i$ does not affect the accuracy of the network reconstruction.

Even if the coefficients corresponding to some nodes have strong correlations, the correctness of network reconstruction will not be affected as well. There is no necessary relationship between the strong correlation and edge of node pair. This is why the results of lasso and elastic net in the previous experiments are very similar with each other.

The group selection is important in prediction and classification problems while the network reconstruction is concerned with the regression coefficients themselves.

Therefore, according to our experimental results and theoretical research, the group selection is not particularly helpful to network reconstruction at least in the present situation.

5.8. Comparisons with some other methods

At present, there are many network reconstruction algorithms based on time series. Researchers took a variety of models to simulate the time series which are assumed as observation data, and then deduced the network structure according to the relationships between these time series. We compare our proposed algorithm with other state-of-the-art methods [12,15,22,25] in Table 3 without considering the observation error. Although different dynamic models are used to obtain the observation time series, these algorithms convert the network reconstruction problem into the sparse signal reconstruction problem.

In order to be consistent with the contrast algorithms, we choose the area under the receiver operating characteristic curve (AUROC) and the area under the precision–recall curve (AUPR) to evaluate the performance of network reconstruction. The AUROC is created by plotting the TPR against FPR (False Positive Rate, FPR = 1−TNR) and AUPR is created by plotting the PPV against TPR at various threshold settings. Both AUROC and AUPR equal 1, which indicates that links and null connections can be completely distinguished from each other with a certain threshold.

From Table 3, it can be seen that the performance of EM-SIS and EM-Game methods is slightly worse than other methods which can reconstruct the whole test networks with appropriate parameters. From these results, it can be found that the relationship between node dynamics and the structure of complex

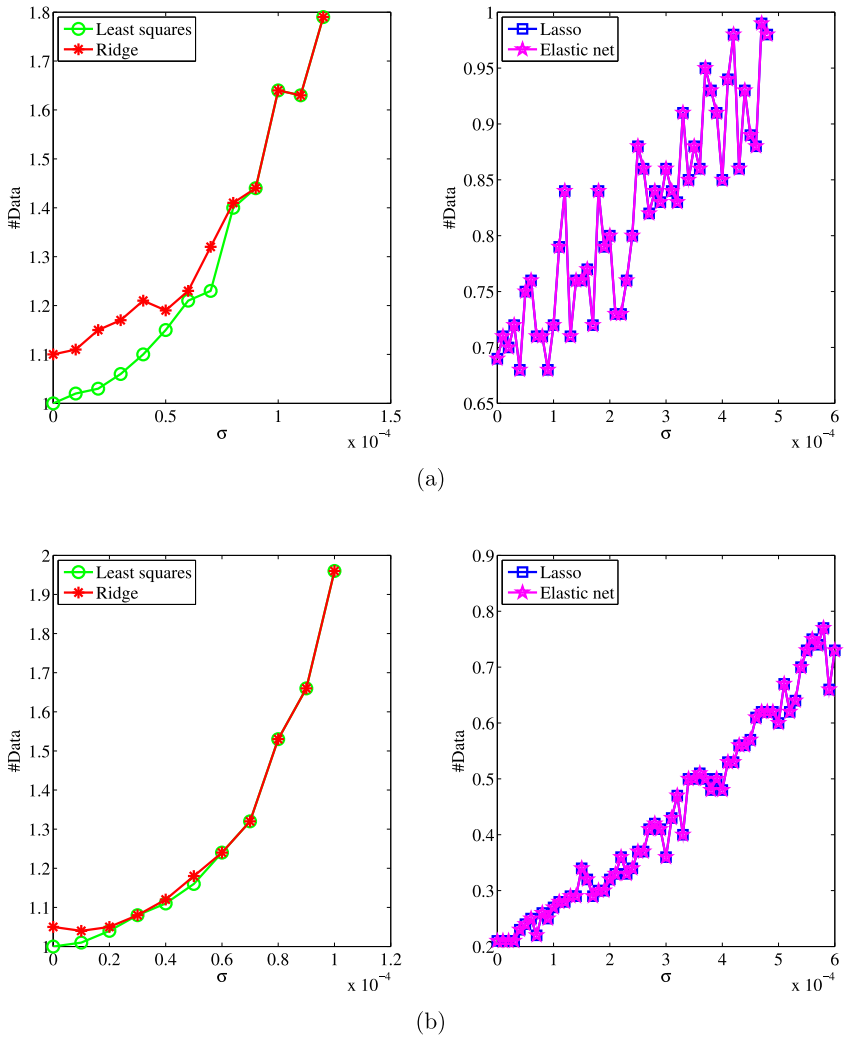

Fig. 4. The relationship curves of #Data and error about different methods. (a) Karate network ($N = 34$) on 1-D Logistic map. (b) SF network ($N = 50$) on 2-D Hénon map. For (a) and (b), the left ones of them mean the curves of least squares and ridge, the right ones mean the curves of lasso and elastic net. For all regression methods, it is possible to ensure the performance of network reconstruction by appropriately increasing the number of measurement data when the error gets bigger and bigger.

Table 3
AUROC/AUPR comparisons of our method with some state-of-the-art algorithms. EM-SIS represents the algorithm in [25] which takes the SIS dynamics model, and EM-Game takes the Game model. Sparse represents the algorithm in [15] which takes the ultimatum games (UG) dynamics model. QR means the algorithm in [22] which takes a linear network system model. CS stands for the algorithm in [12] which takes the 2-D Hénon map. Ours takes the 2-D Hénon map as well.

Network	EM-SIS [25] AUROC/AUPR	EM-Game [25] AUROC/AUPR	Sparse [15] AUROC/AUPR	QR [22] AUROC/AUPR	CS [12] AUROC/AUPR	Ours AUROC/AUPR
Karate	0.983/0.982	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0
Dolphins	0.998/0.993	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0
Football	0.996/0.973	0.999/0.998	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0
Polbook	0.94/0.864	0.994/0.991	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0
ER(500)	1.0/0.997	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0
SW(500)	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0
BA(500)	0.983/0.949	0.998/0.997	1.0/1.0	1.0/1.0	1.0/1.0	1.0/1.0

network is closely related indeed. By analyzing the time series of node dynamics, it is possible to reconstruct the complex networks completely. The regression methods, as mentioned above in our paper, can be easily extended to deal with the time series from other dynamic models.

Except for the reconstruction success rate, we are also concerned about the required measurements and noise immunity of the algorithms. When the test networks are reconstructed completely, the less requirement of the observation data is and the stronger the anti-noise ability is, the more robust and practical the algorithm is. With the same observation data from 2-D Hénon map and on the premise of reconstructing the network completely, we compare the required measurements (#Data) and anti-noise ability of our method (lasso is selected) with CS-based method [12] in Table 4.

From Table 4, we can see the comparisons of CS and our method without the noise ($\sigma = 0$) and with the noise ($\sigma = 10^{-4}$). Without the noise, CS-based method needs less measurements than ours. But it is impossible to reconstruct the networks completely when some noise is introduced. Therefore, our algorithm is a trade-off between limited measurement data and noise.

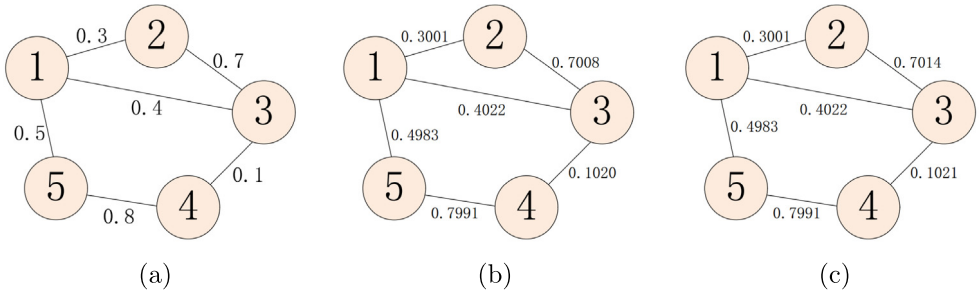

Fig. 5. The reconstruction of an undirected weighted network on 1-D Logistic map. (a) Original test network (b) Result of the least squares or lasso (c) Result of the ridge or elastic net. The value near each edge means the weight between the node pair at the ends of each edge. The results of (b) and (c) prove that we can reconstruct the undirected weighted networks with the weights which very close to the truth.

Table 4
Required #Data and anti-noise ability comparisons of our method with CS-based method [12] when the test networks are reconstructed completely. $\sigma = 0$ means no measurement error.

Network	CS ($\sigma = 0$)	CS ($\sigma = 10^{-4}$)	Ours ($\sigma = 0$)	Ours ($\sigma = 10^{-4}$)
Karate	0.22	–	0.22	0.28
Dolphins	0.13	–	0.2	0.23
Football	0.1	–	0.17	0.2

– means that the network cannot be reconstructed completely.

Table 5
Reconstruction results of the directed networks. Since four regression methods can completely reconstruct the directed network under the appropriate parameter settings, we only list the reconstruction results once which come from any one of four regression method.

Network	N	$\langle k \rangle$	t	TPR	TNR	ACC	PPV
Children's friendship	22	8.05	30	1	1	1	1
Hens	32	15.5	50	1	1	1	1
Amazon books	105	4.2	50	1	1	1	1
ER	10	3.7	15	1	1	1	1

5.9. Extension to the weighted network

The previous experiments in this section are based on the undirected and unweighted networks. However, there are not any prior conditions under which to constrain our proposed method only to be applied to the undirected and unweighted networks. Therefore, the proposed method can be easily extended to weighted networks by replacing the coupled matrix $\boldsymbol{C}_{ij}$ with a weight matrix. A very simple undirected weighted network ($N = 5$) on 1-D Logistic map is taken as an example, which is shown in Fig. 5. In this experiment, we set $\sigma = 10^{-6}$, and set the parameters of ridge, lasso and elastic net as 10^{-9}, $t = 10$, $(t, \lambda_2) = (10, 10^{-9})$. The threshold δ is used to determine whether there is a link between two nodes and the estimated coefficients are taken as the weights directly.

Fig. 5(a) is the topological graph of an undirected weighted network, Fig. 5(b) is the reconstruction result of the least squares or lasso, and Fig. 5(c) is the reconstruction result of the ridge or elastic net.

From Fig. 5, it is found that the proposed method in this paper can reconstruct all edges of the weighted network (TNR, TPR, ACC and PPV are all 1). And the reconstructed weights are very close to the true weights. Therefore, we can draw a conclusion that the proposed method is also suitable for the reconstruction of weighted networks.

5.10. Extension to the directed network

Our proposed method is not only suitable for weighted networks, but also suitable for directed networks. And it can also show the similar rules obtained in the previous parts of this section when reconstructing the directed networks by using four regression methods.

Here we test several directed networks such as Children's friendship [50], Hens [51], Amazon books [52], ER network to prove that the proposed method in this paper can completely reconstruct directed networks. We set #Data = 1.5, $\sigma = 0.3 \times 10^{-4}$, the parameters of ridge and elastic net as $\lambda = \lambda_2 = 10^{-6}$. The parameter t (stop step) of lasso or elastic net is set according to the size of test network which is listed in Table 5.

Table 5 shows the reconstruction results of directed networks by any one of four regression methods above-mentioned, but there is no separate description of them for the consistent results. As can be seen from Table 5, the four regression methods can completely reconstruct the directed network under the appropriate parameter settings.

6. Conclusions

In this paper, we adopt four typical regression methods to solve the problem of network reconstruction which takes observation error and limited observation data into consideration. We thoroughly analyze and discuss how to choose the most appropriate optimization method in the case of different network reconstruction problems, which can help to understand the internal mechanisms of network reconstruction, and conclusions are drawn as follows: Under the premise of sufficient observations, when the variance of observation error is relatively large, we can select lasso and elastic net methods which are less sensitive to observation error to reconstruct the network structure completely. Under the premise of a certain observation error, when the observations are insufficient, we can choose lasso and elastic net methods which have the functions of feature selection to reconstruct the structure of network. When the size of network is relatively small and there are enough observations, we can choose the least squares and ridge methods which have lower computational burden. We should note that when the size of network is large and the required measurements to completely reconstruct the network are relatively small enough, the computational burdens of lasso and elastic net methods are even far lower than the computational burdens of least squares and ridge for their early stopping.

The proposed method can not only reconstruct the undirected networks, but also reconstruct the directed and weighted networks. Therefore, it is concluded that the reconstruction of different types of networks has similar laws with the same regression method.

Our conclusions may help to select appropriate optimization methods based on actual situations. And our network reconstruction algorithm may help to understand the relationship between

the structure of actual complex system and its dynamics, which can be used for knowledge discovery and data mining.

CRediT authorship contribution statement

Xiaomin Wu: Conceptualization, Methodology, Software, Data curation, Writing - original draft, Writing - review & editing. **Jianshe Wu:** Investigation, Supervision, Writing - review & editing. **Jixin Zou:** Funding acquisition, Writing - review & editing. **Qian Zhang:** Resources, Funding acquisition.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. 61672405, 61572383), the Natural Science Basic Research Program of Shaanxi, China (No. B018360019) and the Funds of the Artificial Intelligence Joint Laboratory of the 20th Research Institute of CETC and the Xidian University, China.

References

[1] M. Timme, Does dynamics reflect topology in directed networks, Europhys. Lett. 76 (3) (2006) 367–373.
[2] K. Wu, J. Liu, D. Chen, Network reconstruction based on time series via memetic algorithm, Knowl.-Based Syst. 164 (15) (2019) 404–425.
[3] M. Nabi-Abdolyousefi, M. Mesbahi, Network identification via node knock-out, in: Decision and Control, 2010, pp. 2239–2244.
[4] P.K. Pandey, B. Adhikari, A parametric model approach for structural reconstruction of scale-free networks, IEEE Trans. Knowl. Data Eng. 29 (10) (2017) 2072–2085.
[5] T.S. Gardner, D. Di Bernardo, D. Lorenz, J.J. Collins, Inferring genetic networks and identifying compound mode of action via expression profiling, Science 301 (5629) (2003) 102–105.
[6] F. Geier, J. Timmer, C. Fleck, Reconstructing gene-regulatory networks from time series, knock-out data, and prior knowledge, BMC Syst. Biol. 1 (1) (2007) 1–16.
[7] M. Salleh, F. Hani, S. Zainudin, M. Raih, M. Firdaus, Reconstruction of large-scale gene regulatory networks using regression-based models in: 2018 IEEE Conference on Big Data and Analytics, 2019, pp. 129–134.
[8] S. Clémençon, H. De Arazoza, F. Rossi, V.C. Tran, A statistical network analysis of the HIV/AIDS epidemics in Cuba, Soc. Netw. Anal. Min. 5 (1) (2015) 1–14.
[9] R. Schweiger, M. Linial, N. Linial, Generative probabilistic models for protein-protein interaction networks–the biclique perspective, Bioinformatics 27 (13) (2011) i142–i148.
[10] S. Shandilya, M. Timme, Inferring network topology from complex dynamics, New J. Phys. 13 (1) (2011).
[11] J. Li, Z. Shen, W.X. Wang, C. Grebogi, Y.C. Lai, A universal data based method for reconstructing complex networks with binary-state dynamics, Physics (2016).
[12] W.X. Wang, R. Yang, Y.C. Lai, V. Kovanis, M.A.F. Harrison, Time-series-based prediction of complex oscillator networks via compressive sensing, Europhys. Lett. 94 (4) (2011).
[13] W.X. Wang, Y.C. Lai, C. Grebogi, J. Ye, Network reconstruction based on evolutionary-game data via compressive sensing, Phys. Rev. X 1 (2) (2011).
[14] S. Zhang, X. Wu, J.A. Lu, H. Feng, J. L, Recovering structures of complex dynamical networks based on generalized outer synchronization, IEEE Trans. Circuits Syst. I. Regul. Pap. 61 (11) (2014) 3216–3224.
[15] X. Han, Z. Shen, W.-X. Wang, Z. Di, Robust reconstruction of complex networks from sparse data, Phys. Rev. Lett. 114 (2) (2015) 028701.
[16] W.X. Wang, Y.C. Lai, C. Grebogi, Data based identification and prediction of nonlinear and complex dynamical systems, Phys. Rep. 644 (2016) 1–76.
[17] X. Wang, Q.Y. Wang, J.H. L, Topology reconstruction of complex networks with time-varying parameters nodes, in: Chinese Control Conference 2017, 2019.
[18] C. Ma, H.F. Zhang, Y.C. Lai, Reconstructing complex networks without time series, Phys. Rev. E 96 (2) (2017) 022320.
[19] F. Alderisio, G. Fiore, M. Di Bernardo, Reconstructing the structure of directed and weighted networks of nonlinear oscillators, Phys. Rev. E 95 (4) (2017) 042302.
[20] J.W. Li, Z.S. Shen, W.X. Wang, C. Grebogi, Y.C. Lai, Universal data-based method for reconstructing complex networks with binary-state dynamics, Phys. Rev. E 95 (2017) 032303.
[21] H.J. van Waarde, P. Tesi, M.K. Camlibel, Topology reconstruction of dynamical networks via constrained Lyapunov equations, 2017, URL arXiv:1706.09709v1.
[22] L. Li, D. Xu, H. Peng, J. Kurths, Y. Yang, Reconstruction of complex network based on the noise via QR decomposition and compressed sensing, Sci. Rep. 7 (1) (2017) 15036.
[23] J.S. Wu, H.D. Yang, Y.H. Ren, X.R. Li, A two-stage algorithm for network reconstruction, Appl. Soft Comput. 70 (2018) 751–763.
[24] P.K. Pandey, V. Badarla, Reconstruction of network topology using status-time-series data, Physica A 490 (15) (2018) 573–583.
[25] C. Ma, H.S. Chen, Y.C. Lai, H.F. Zhang, Statistical inference approach to structural reconstruction of complex networks from binary time series, Phys. Rev. E 97 (2–1) (2018) 022301.
[26] H.F. Zhang, F. Xu, Z.K. Bao, C. Ma, Reconstruction of networks with binary-state dynamics via generalized statistical inference, IEEE Trans. Circuits Syst. I. Regul. Pap. 66 (4) (2019) 1608–1619.
[27] W.F. Deng, K.K. Huang, C.H. Yang, Effect of clustering property on complex network reconstruction via compressed sensing, Physica A 528 (15) (2019) 121357.
[28] J. Bongard, H. Lipson, Automated reverse engineering of nonlinear dynamical systems, Proc. Natl. Acad. Sci. 104 (2007) 9943–9948.
[29] D.L. Donoho, Compressed sensing, IEEE Trans. Inform. Theory 52 (4) (2006) 1289–1306.
[30] R. Tibshirani, Regression shrinkage and selection via the lasso, J. R. Stat. Soc. Ser. B Stat. Methodol. (1996) 267–288.
[31] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, second ed., Springer, New York, 2009.
[32] S. Shahrampour, V.M. Preciado, Topology identification of directed dynamical networks via power spectral analysis, IEEE Trans. Automat. Control 60 (8) (2015) 2260–2265.
[33] R.M. May, et al., Simple mathematical models with very complicated dynamics, Nature 261 (5560) (1976) 459–467.
[34] M. Itoh, T. Yang, L.O. Chua, Conditions for impulsive synchronization of chaotic and hyperchaotic systems, Int. J. Bifurcation Chaos 11 (02) (2001) 551–560.
[35] M. Hénon, A two-dimensional mapping with a strange attractor, Comm. Math. Phys. 50 (1) (1976) 69–77.
[36] H.E. Nusse, J.A. Yorke, Dynamics: Numerical Explorations: Accompanying Computer Program Dynamics, Springer, New York, 2012.
[37] A.E. Hoerl, R.W. Kennard, Ridge regression: Biased estimation for nonorthogonal problems, Technometrics 12 (1) (1970) 55–67.
[38] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc. Ser. B Stat. Methodol. 67 (2) (2005) 301–320.
[39] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, Least angle regression, Ann. Statist. 32 (2) (2004) 407–499.
[40] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (4) (2008).
[41] P. Erdös, A. Rényi, On the evolution of random graph, Publ. Math. Inst. Hung. Acad. Sci. 5 (1960) 17–61.
[42] D.J. Watts, S.H. Strogatz, Collective dynamics of small-worldnetworks, Nature 393 (6684) (1998) 440–442.
[43] A.L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512.
[44] W.W. Zachary, An information flow model for conflict and fission in small groups, J. Anthropol. Res. (1977) 452–473.
[45] D. Lusseau, K. Schneider, O.J. Boisseau, P. Haase, E. Slooten, S.M. Dawson, The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations, Behav. Ecol. Sociobiol. 54 (4) (2003) 396–405.
[46] K. Valdis, A network of books about US politics, URL http://www-personal.umich.edu/~mejn/netdata/.
[47] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proc. Nat. Acad. Sci. 99 (12) (2002) 7821–7826.
[48] M.E. Newman, Finding community structure in networks using the eigenvectors of matrices, Phys. Rev. E 74 (3) (2006).
[49] E.N. Lorenz, Deterministic nonperiodic flow, J. Atmos. Sci. 20 (2) (2004) 130–141.
[50] C.J. Anderson, S. Wasserman, B. Crouch, A p* primer: logit models for social networks, Social Networks 21 (1) (1999) 37C66.
[51] A.M. Guhl, Social behavior of the domestic fowl, Trans. Kans. Acad. Sci. 71 (3) (1968) 379–384.
[52] V. Krebs, http://www.orgnet.com/ (unpublished).