Physica A 497 (2018) 118–125

Contents lists available at ScienceDirect

Physica A

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

Reconstruction of networks from one-step data by matching positions

Jianshe Wu *, Ni Dang, Yang Jiao

Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, International Research Center for Intelligent Perception and Computation, Joint International Research Laboratory of Intelligent Perception and Computation, Xidian University, Xi'an, Shaanxi Province, 710071, China

HIGHLIGHTS

- A method for reconstructing the topology of networks with only one step data.
- This algorithm utilizes the natural sparsity and degree distribution of real world networks.
- The problem of matching position is defined and a simple algorithm is developed.
- The matching position can be used in network reconstruction and other problems.
- The proposed method is verified in networks with ten thousands of vertices.

ARTICLE INFO

Article history:
Received 2 March 2017
Received in revised form 2 July 2017
Available online 11 January 2018

Keywords:
Network reconstruction
Topology of network
Time series data
Reconstruction algorithm
Matching positions

ABSTRACT

It is a challenge in estimating the topology of a network from short time series data. In this paper, matching positions is developed to reconstruct the topology of a network from only one-step data. We consider a general network model of coupled agents, in which the phase transformation of each node is determined by its neighbors. From the phase transformation information from one step to the next, the connections of the tail vertices are reconstructed firstly by the matching positions. Removing the already reconstructed vertices, and repeatedly reconstructing the connections of tail vertices, the topology of the entire network is reconstructed. For sparse scale-free networks with more than ten thousands nodes, we almost obtain the actual topology using only the one-step data in simulations.

1. Introduction

Usually, the phenomena (e.g., phase information) of coupled agents occurred in a network are observed, but the topology of the network is unknown. Estimating the topology of the network from those phenomena becomes a key in a multidisciplinary research field [1–3]. Network reconstruction is a reverse engineering problem and a number of methods have been proposed to address this problem [4–9]. Since the network function and dynamics are various [10–12], the data measured for networks reconstructions are various, e.g., response dynamics [10], avalanche dynamics [11], random phase-resetting [12], knock-out data [13], noise [9], etc. In network reconstruction, a collection of unprecedented amount data measured in series time points, named as time series data is needed, especially for large scale networks. However, it is often

* Corresponding author.
E-mail address: jshwu@mail.xidian.edu.cn (J. Wu).

https://doi.org/10.1016/j.physa.2018.01.011
0378-4371/

that only limited observable data are available. In Ref. [14], the difficulty of reconstruction using short time series is tackled, and it is still a challenge.

When the time series is short, a convenient method is using the natural sparsity of networks [7,15,16]. Natural sparsity of networks indicates that the number of connections of a network is much less than the number of all possible connections in real world. Natural sparsity makes it possible to approximately reconstruct the topology with a short time series data, much less than the network size. But these methods are impossible to reconstruct with only one-step data. New method is required.

In this paper, the matching positions problems (MPPs) are defined firstly, and then the matching positions (MP) algorithm is developed to reconstruct the topology of a network with only one-step data. Besides the natural sparsity, the degree distribution of real world networks is approximately power-law [17,18], that is called scale-free. This indicates most of the vertices have few, even one or two edges, but small number of vertices has large amount of connections. The MP algorithm reconstructs the connections of the vertices with the smallest degree firstly. Removing the reconstructed vertices, the remained network is still scale-free. By repeatedly reconstructing the connections of the vertices with the smallest degree, the topology of entire network is reconstructed.

2. Problem formulation

We consider a general un-weighted network model of coupled agents, in which the phase of vertex i at time step k is determined by [19,20]:

$$\theta_i(k+1) = \sum_{j=1}^{N} \frac{a_{ij}}{k_m}\theta_j(k), \quad i = 1, \ldots, N, \tag{1}$$

where N is the number of vertices and a_{ij} is the elements of the network adjacency matrix $A \in R^{N\times N}$. If there is an edge between vertices (agents) i and j, $a_{ij} = a_{ji} = 1$; otherwise $a_{ij} = 0$. k_m is normalization parameter, which may has different values in different applications. In the communication network model [15], k_m equals the number of neighbors of vertex i. In the original Vicsek model [19], k_m equals the number of neighbors of vertex i plus one (including i itself). In this paper, without loss generalization, we assume $k_m = 1$ for clarity:

$$\theta_i(k+1) = \sum_{j=1}^{N} a_{ij}\theta_j(k), \quad i = 1, \ldots, N. \tag{2}$$

In matrix form, Eq. (2) is written as Eq. (3) if the length of time series data is K.

$$\theta(k+1) = A\theta(k), \quad k = 1, \ldots, K, \tag{3}$$

where $\theta(k) = (\theta_1(k), \ldots, \theta_N(k))^T$. Network reconstruction means estimating A from time series data $\theta(k)$, $k = 1, \ldots, K$. If $K = 1$, only $\theta(1)$ and $\theta(2)$ are available, that is the one-step data.

In mathematics, when $K \geq N$, we may obtain A by direct solving matrix equation (3). When $K < N$, the equation is underdetermined, and there are a set of possible solutions for A. Since the natural sparsity, we can choose the sparsest one from those possible solutions for the adjacency of the network [7,15,16]. These methods perform well when K is not too small and are impossible when $K \to 1$. In this paper, we address the challenge problem of network reconstruction when $K = 1$.

Real world networks are formulated by two ingredients: growth and preferential attachment [17,18]. In the BA scale-free network model, from an initial network of m_0 vertices, a new vertex with m edges is added to the network at each time step [18]. The new vertex links to m vertices with probability $\prod(k_i) = k_i/\sum k_i, i = 1, \ldots, N$, where k_i is the degree of vertex i.

Inspired by the natural networks formulation, it is reasonable to reconstruct the topology of a network by sequentially reconstructing the connections of the vertices. As a reverse engineering problem [1,4], we treat the network reconstruction as a reverse process of the network formulation. The edges of the latest added vertex which has the smallest degree, are reconstructed firstly. Reconstructing the connections of a vertex from one-step data is one of the MPPs that will be defined and analyzed in the next section. Removing the constructed vertex and its edges, the remained unconstructed network is still scale-free. Repeat the above process until the edges of all vertices are obtained (see Fig. 1).

3. Matching positions with single vertex

For clarity in representation, suppose vertex i has the smallest degree k_s in an N-vertex network. The k_s edges may have $l = C_N^{k_s} = \frac{N!}{k_s!(N-k_s)!}$ possible combinations; each of them is denoted as a 0–1 row vector $d_j \in R^{1\times N}$. The dictionary matrix $D = (d_1, \ldots, d_l)^T \in R^{l\times N}$ is formulated by all those possible row vectors. An example dictionary for a 4-vertex network with $k_s = 2$ is

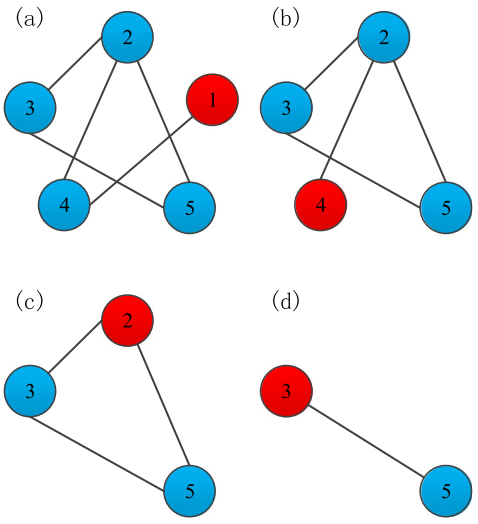

Fig. 1. Sequence of reconstructing the connections of vertices: increasing order of the degree of vertices. (a) Reconstruct vertex 1 with $k_s = 1$, remove vertex 1; (b) Reconstruct vertex 4 with $k_s = 1$, remove vertex 4; (c) Reconstruct vertex 2 with $k_s = 2$ (no vertex with $d_s = 1$), remove vertex 2; (d) Reconstruct vertex 3 with $k_s = 1$, the topology of the network is reconstructed.

$$D = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

The prediction of the phase of the vertex in the next time step is

$$\hat{\theta}(2) = D\theta(1), \tag{4}$$

where $\hat{\theta}(2) = (\hat{\theta}_1(2), \ldots, \hat{\theta}_j(2), \ldots, \hat{\theta}_l(2))^T$ and $\hat{\theta}_j(2)$ is the prediction of the phase corresponding to the jth row vector in dictionary D.

The MPPs include matching position with single vertex (MPSV) and matching positions with multiple vertices (MPMV). The MPSV means to find a row vector (e.g., d_j) in D such that the corresponding prediction element in $\hat{\theta}(2)$ matches an element (e.g., $\theta_i(2)$) in $\theta(2)$:

$$d_j = \arg\min_{\substack{1\le i\le N \\ 1\le j\le l}} \{\theta_i(2) - \hat{\theta}_j(2)\}, \tag{5}$$
$$\text{s.t. } \{|\theta_i(2) - \hat{\theta}_j(2)| < \varepsilon\},$$

where ε is the threshold for matching. If $\theta_i(2)$ matches $\hat{\theta}_j(2)$, the connections of vertex i are reconstructed and the edges are described by vector d_j (see Fig. 2). MPSV is equivalent to matching a vertex with a row vector of the dictionary.

Since the natural sparsity: $k_s \ll N$, most elements of the dictionary matrix D are zero. For simplification in computation, we define location matrix $L \in R^{l\times ks}$ to represent the locations of "1" in the dictionary matrix. For example, the location matrix L of the above mentioned dictionary D for 4-vertex network above is

$$L = \begin{pmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 2 & 3 \\ 2 & 4 \\ 3 & 4 \end{pmatrix}.$$

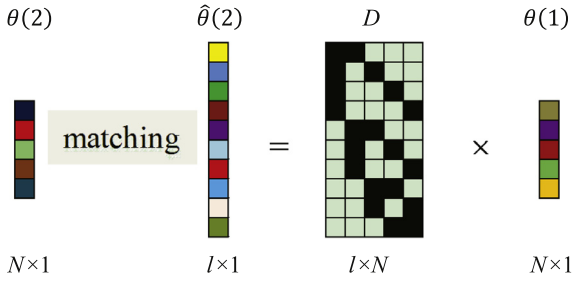

Fig. 2. A pictorial representation of the MPSV.

The edges described by a row vector in dictionary matrix can be equivalently described by the corresponding row vector of the location matrix. Phase matrix $P \in R^{l\times ks}$ is defined to represent the initial phases ($\theta(1)$) of the vertices corresponding to the location matrix. Column vector $S \in R^{l\times}1$ is the row sum of the phase matrix P, which is actually the prediction.

Take the network in Fig. 1 for illustration. For $k_s = 1$, $L = (1, 2, 3, 4, 5)^T$, $P = (\theta_1(1), \theta_2(1), \theta_3(1), \theta_4(1), \theta_5(5))^T$, and $S = (s_1, s_2, s_3, s_4, s_5)^T = (\theta_1(1), \theta_2(1), \theta_3(1), \theta_4(1), \theta_5(1))^T$. In the data $\theta_i(2)$, $i = 1, \ldots, 5$, if $\theta_1(2)$ matches s_4, then there is an edge e_{14}. If it does not match in $k_s = 1$, perform MPSV further in $k_s = 2$.

For $k_s = 2$, $l = C_5^2 = 10$:

$$L = \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \\ 1 & 5 \\ 2 & 3 \\ 2 & 4 \\ 2 & 5 \\ 3 & 4 \\ 3 & 5 \\ 4 & 5 \end{bmatrix}, \quad P = \begin{bmatrix} \theta_1(1) & \theta_2(1) \\ \theta_1(1) & \theta_3(1) \\ \theta_1(1) & \theta_4(1) \\ \theta_1(1) & \theta_5(1) \\ \theta_2(1) & \theta_3(1) \\ \theta_2(1) & \theta_4(1) \\ \theta_2(1) & \theta_5(1) \\ \theta_3(1) & \theta_4(1) \\ \theta_3(1) & \theta_5(1) \\ \theta_4(1) & \theta_5(1) \end{bmatrix}, \quad S = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \\ s_6 \\ s_7 \\ s_8 \\ s_9 \\ s_{10} \end{bmatrix} = \begin{bmatrix} \theta_1(1) + \theta_2(1) \\ \theta_1(1) + \theta_3(1) \\ \theta_1(1) + \theta_4(1) \\ \theta_1(1) + \theta_5(1) \\ \theta_2(1) + \theta_3(1) \\ \theta_2(1) + \theta_4(1) \\ \theta_2(1) + \theta_5(1) \\ \theta_3(1) + \theta_4(1) \\ \theta_3(1) + \theta_5(1) \\ \theta_4(1) + \theta_5(1) \end{bmatrix}.$$

If $\theta_i(2)$ matches s_j, then there are two edges connect from vertex i to the two corresponding vertices described by the jth row vector of L. If it does not match, perform MPSV further in $k_s = 3$. The process continues until a successful match.

4. The detailed process of the algorithm

When the edges of one vertex with the smallest degree are reconstructed by the MPSV, the vertex and its edges are removed firstly from the original network, and then reconstruct the vertex with the smallest degree in the remaining network. Repeat the process until the network is reconstructed completely.

One important problem need to address is that the one-step phase data cannot be directly applied in reconstructing the second vertex after the first vertex is removed. In fact, after one vertex is removed in the process, the phase data $\theta(2)$ is updated by

$$\theta(2) = \theta(2) - a_i \cdot \theta_i(1), \tag{6}$$

where a_i is the 0–1 column vector corresponding to the neighbors of reconstructed vertex i in last step. For example the network in Fig. 1, if vertex 1 is reconstructed in the first step and $a_1 = (0, 0, 0, 1, 0)^T$, then the phase of vertex 4 is updated by $\theta_4(2) = \theta_4(2) - \theta_1(1)$ from (6).

From the BA model [18], in each step a new vertex with m edges is added to the network, then the smallest degree is m and the average degree $\langle k\rangle \approx 2$ m, thus $m \approx \langle k\rangle/2$. The length of vector $\hat{\theta}(2)$ is $C_N^{\langle k\rangle/2}$. The time complexity for matching between $\hat{\theta}(2)$ and $\theta(2)$ is $NC_N^{\langle k\rangle/2}$. Reconstructing the network needs matching $N-1$ times. Thus, the average time complexity is expected to be less than $N^2C_N^{\langle k\rangle/2}$; though a rigorous upper bound is not yet available.

5. Matching positions with multiple vertices

In Eq. (5) for MPSV, a single vertex with the smallest degree k_s matches a single row vector. In fact, there are multiple vertices with the same degree k_s usually. We define the MPMV as follows:

$$d_j = \underset{\substack{1\le i\le N \\ 1\le j\le l}}{\arg} \{|\theta_i(2) - \hat{\theta}_j(2)| < \epsilon\}. \tag{7}$$

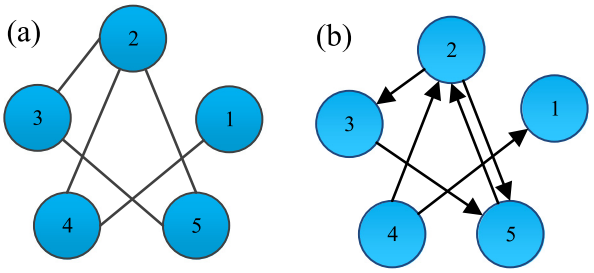

Fig. 3. Undirected and directed networks. (a) Undirected network; (b) Directed network.

The MPMV matches multiple vertices to the same number of row vectors of the dictionary simultaneously in one step. If MPSV is replaced by MPMV, it does not need $N-1$ steps to reconstruct the entire network, so the process is speeded up. The average time complexity is expected to be less than $N^{\alpha}C_N^{\langle k\rangle/2}$, where $\alpha < 2$.

After one step of MPMV, the phase data $\theta\,(2)$ should be computed by

$$\theta\,(2) = \theta\,(2) - A_c\theta\,(1)\,, \tag{8}$$

where $A_c \in R^{N\times N}$ is a 0–1 matrix that describes the already reconstructed part of the adjacency matrix A. The elements of the unconstructed part of A_c are zeros. If all the connections of the network are correctly reconstructed in the final step, $A_c = A$. Note that in Eq. (8), the updated part in each step is only A_c. The $\theta\,(2)$ in the right part of Eq. (8) is always the initial data, and is not cumulatively updated, which is different from Eq. (6) for MPSV.

6. Reconstructing directed networks

Sections 4 and 5 describe the process for undirected networks. Similar as Eq. (2), for a directed network, the phase of vertex i at time step k is determined by:

$$\theta_i\,(k+1) = \sum_{j=1}^{N} a_{ij}\theta_j\,(k)\,, \quad i = 1, \ldots, N, \tag{9}$$

where the meanings of N and $\theta_i\,(k)$ are the same as those in Eq. (2). But a_{ij} here is different from that in Eq. (2). In Eq. (9), $a_{ij} = 1$ does not indicates $a_{ji} = 1$ and vice verse.

In matrix form, Eq. (9) is written as Eq. (10) if the length of time series data is K.

$$\theta\,(k+1) = A\theta\,(k)\,, \quad k = 1, \ldots, K, \tag{10}$$

where $\theta\,(k) = (\theta_1\,(k)\,, \ldots, \theta_N\,(k))^T$. Reconstructing the topology of a directed network is actually estimating A from time series data $\theta\,(k)$. Obviously, the adjacency matrix A for a directed network is asymmetric.

For inferring the topology of a directed network, the process of removing the vertex and its edges in Sections 4 and 5 no longer applies. Thus the process of reconstructing all the vertices of a directed network is the same as the process of reconstructing the first vertex. Being similar to undirected networks, a single vertex is reconstructed in each step for MPSV and several vertices may be reconstructed in each step for MPMV.

Since the process of removing the vertex and edges is no longer applicable in reconstructing directed networks, the average time complexity is expected larger than reconstructing an undirected network with similar structure. However, when reconstructing a directed network, the average in-degree to be concerned is usually less than that in an undirected network with similar structure. As shown in Fig. 3, the average degree of the undirected network (Fig. 3(a)) is $(1+3+2+2+2)/5 = 2$, while the average in-degree of the directed network (Fig. 3(b)) is $(1+2+1+0+2)/5 = 1.2$.

7. Simulations

To test the proposed method, we generate one-step data of a given network by (2) or (3) with $K = 1$. Suppose A and A_r are the true and reconstructed adjacency matrices respectively. The error matrix is defined as

$$A_e = A_r - A. \tag{11}$$

The elements of A_e may be '0', '1', or '-1'. Both '1' and '-1' indicate an error. If $A_e = 0$, the network is reconstructed perfectly. As shown in Eqs. (12) and (13) respectively, the error rate and accuracy rate are all global evaluating indexes [21], which can

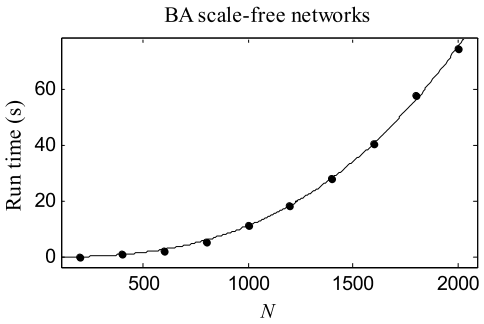

Fig. 4. Average run time of the proposed algorithm varies with the network scale ($\langle k\rangle$=4). Acc=1 for all the simulations. Each point is the average on 10 networks, and in each network the algorithm runs 10 times.

Table 1
Run time and ErrorEdge with different $\langle k\rangle$ and N.

Run time(s) /ErrorEdge <k> \ N	200	500	800	1400	2000	3000
2	0.013/ 0	0.027/ 0	0.047/ 0	0.097/ 0	0.177/ 0	0.356/ 0
4	0.618/ 0	1.557/ 0	5.296/ 0	28.092/ 0	64.765/ 0.4	211.225/ 1.1
6	13.441/ 0	360.505/ 3.4	4782.046/ 12.95			

be utilized to evaluate the performance of the proposed algorithm.

$$Err = \frac{FP + FN}{TP + FN + FP + TN}, \tag{12}$$

$$\text{Acc} = \frac{TP + TN}{TP + FN + FP + TN} = 1 - Err, \tag{13}$$

where *FP*, *FN*, *TP*, and *TN* represent the number of false positive, false negative, true positive, and true negative, respectively. All the simulations run on a personal computer with a Intel(R) Core(TM) i5-3230M 2.60 GHz processor and a 4 GB RAM. In all simulations, the proposed MP algorithm runs on the MPMV and $\varepsilon = 1\ \text{E}{-}10$. We test the performance of the proposed algorithm on scale-free networks, modular networks, random networks, and gene networks (directed).

Scale-free networks. The scale-free networks are generated by the BA model with $m_0 = 3$ and $m = 2$. The network scale N varies from 200 to 2000. In each network scale, 10 networks are generated, for each network we run the algorithm 10 times. In all the experiments, the networks are correctly reconstructed: Acc=1. Fig. 4 shows the average run time varying with the network scale.

To test the reconstruction efficiency of scale-free networks with different average degrees, we generate BA networks with N from 200 to 3000, $m_0 = 3$, and the $\langle k\rangle$ varies from 2 to 6 (see Table 1). For each average degree, 10 networks are generated, and for each network we run the algorithm 10 times. The average run times and the average number of error edges (ErrorEdge) are shown in Table 1. It is seen that the run time increases with the average degree rapidly and increase almost linearly with the network scale (N). When the average degree and network scale is small, all edges are correctively reconstructed (ErrorEdge = 0). With the increasing of the average degree $\langle k\rangle$ and the network scale N, the error edges appear and the number gets larger.

As analyzed and simulated above, the proposed MP algorithm is suitable for scale-free networks, in which the degree distribution is power-law. Here we compare the MP algorithm on sparse BA scale-free networks and ER random network [22]. The degree distribution of the vertices of ER networks is Poisson [22]. We generate 50 BA and ER networks at each network scale (N) with the same average degree ($\langle k\rangle = 2$), and the simulation results are shown in Fig. 5.

From Fig. 5, it is seen that the number of error edges is zero for all the BA networks even when $N = 10\,000$, but there are error edges for ER networks even when $N = 2000$. In addition, the run time of ER networks is much larger than that of BA networks.

Modular networks. Modular networks are generated by the LFR benchmark network model, in which the degree distribution is also power-law [23]. The LFR model has six parameters: the average degree $\langle k\rangle$, the maximum degree k_{max}, the number of vertices N, the mixing parameter μ, the exponent for vertex degree distribution γ, and the exponent for community size distribution β. We generate networks with five constant parameters: $\langle k\rangle = 3$, $k_{max} = 15$, $\mu = 0.1$, $\gamma = 2$, $\beta = 1$, and the

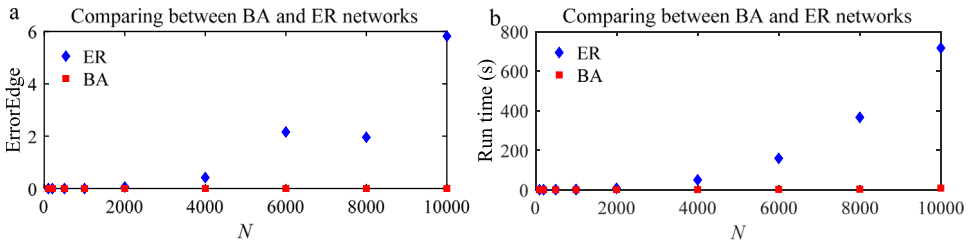

Fig. 5. Comparing between BA and ER networks ($\langle k\rangle = 2$). (a) The number of error edges (ErrorEdge); (b) Run time. Each point in the figure is the average on 50 networks.

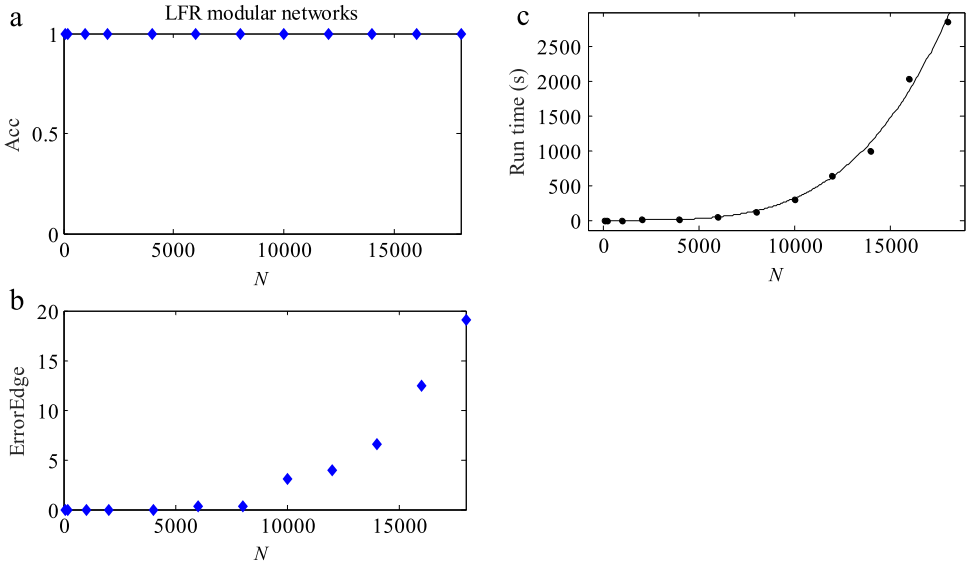

Fig. 6. Experiments on the LFR modular networks. (a) Acc; (b) ErrorEdge: the number of error edges; (c) Average run time. Each point in the figure is the average on 5 networks, and for each network the algorithm runs 5 times.

network scale N varies from $N = 100$ to $N = 18\,000$. In each network scale, 5 networks are generated. We run the algorithm 5 times for each network.

It is seen in Fig. 6(a) that the accurate rate is almost 1 even when $N = 18\,000$. So the number of error edges is given in Fig. 6(b), and it is seen that ErrorEdge = 0 when $N < 5000$ and ErrorEdge < 20 even when $N = 18\,000$.

Directed networks. Fig. 7 show the gene regulatory network of TF family AP1 in rat [24,25]. Ellipses are TFs. Boxes are genes. The hexagon is the clustered genes; the number of the genes is shown inside. Red lines indicate the protein–DNA binding is known. Only official gene symbols are used in the network. For more details of the gene regulatory network, please see in Ref. [24].

The network has 26 vertices, with the average degree $\langle k\rangle = 1.423$ and the maximum degree $k_{max} = 5$. We run the MP algorithm 10 times and the network is correctively reconstructed in all the 10 times (ErrorEdge = 0). The average run time is 1043.642 s.

8. Conclusion

Reconstructing the network structure with short time series data is a challenge, especially in the extreme case that only one-step data is available. We define the problem of network reconstruction with only one-step data as the matching positions problems (MPPs). The matching positions (MP) algorithm for network reconstruction is proposed, which is capable of reconstructing the topology of networks with only one-step data. Though the proposed MP algorithm is applicable in both undirected networks and directed networks, it is more suitable for undirected scale-free networks.

In principle, if the dynamic behavior is predictable, the proposed mechanism can be used. This paper only illustrates the MP algorithm in a simple phase transformation model. Future researches are supposed to include extending the MP mechanism in other networks with different dynamic behaviors and exploring other problems of MPPs other than network reconstruction.

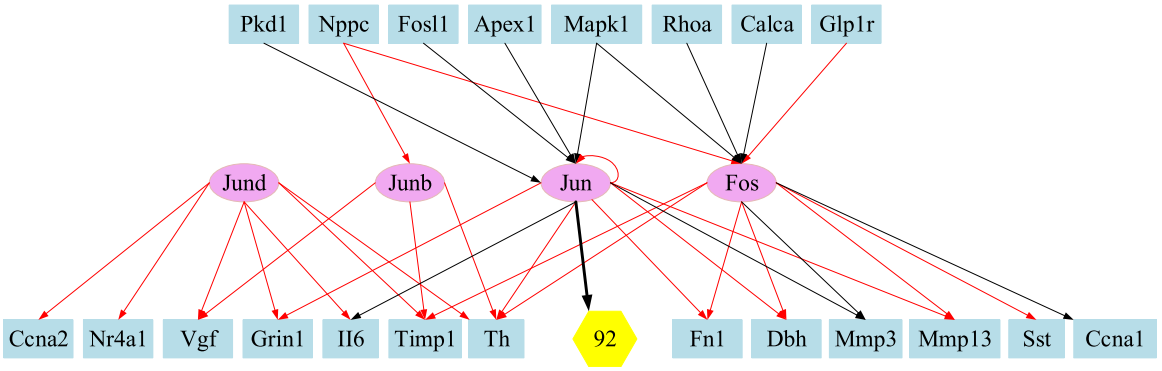

Fig. 7. The gene regulatory network of TF family AP1 in rat. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Nos. 61571342. 61672405, 61472306, 61572383).

References

[1] M.K.S. Yeung, J. Tegnér, J.J. Collins, Reverse engineering gene networks using singular value decomposition and robust regression, Proc. Natl. Acad. Sci. USA 99 (2002) 6163–6168.
[2] E.S.C. Ching, P.-Y. Lai, C.Y. Leung, Reconstructing weighted networks from dynamics, Phys. Rev. E (2015) 030801(R).
[3] S.R. Gujarathi, C.L. Farrow, C. Glosser, L. Granlund, P.M. Duxbury, Ab-initio reconstruction of complex Euclidean networks in two dimensions, Phys. Rev. E 89 (2014) 053311.
[4] R. Laubenbacher, B. Stigler, A computational algebra approach to the reverse engineering of gene regulatory networks, J. Theoret. Biol 229 (2004) 523–537.
[5] 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] B.N. Kholodenko, A. Kiyatkin, F.J. Bruggeman, E. Sontag, H.V. Westerhoff, J.B. Hoek, Untangling the wires: A strategy to trace functional interactions in signaling and gene networks, Proc. Natl. Acad. Sci. USA 99 (2002) 12841–12846.
[7] W.-X. Wang, Y.-C. Lai, C. Grebogi, J. Ye, Network reconstruction based on evolutionary-game data via compressive sensing, Phys. Rev. X 1 (2011) 021021.
[8] W.-X. Wang, Q.-F. Chen, L. Huang, Y.-C. Lai, M.A.F. Harrison, Scaling of noisy fluctuations in complex networks and applications to network prediction, Phys. Rev. E 80 (2009) 016116.
[9] J. Ren, W.-X. Wang, B. Li, Y.-C. Lai, Noise bridges dynamical correlation and topology in coupled oscillator networks, Phys. Rev. Lett. 104 (2010) 058701.
[10] M. Timme, Revealing network connectivity from response dynamics, Phys. Rev. Lett. 98 (2007) 224101.
[11] S.G. Shandilya, M. Timme, Inferring network topology from complex dynamics, New J. Phys. 13 (2011) 013004.
[12] Z. Levnajic, A. Pikovsky, Network reconstruction from random phase-resetting, Phys. Rev. Lett. 107 (2011) 034101.
[13] F. Geier, J. Timmer, C. Fleck, Reconstructing gene-regulatory networks from time series knock-out data, and prior knowledge, BMC Syst. Biol. 1 (2007) 11.
[14] S. Hempel, A. Koseska, J. Kurths, Z. Nikoloski, Inner composition alignment for inferring directed networks from short time series, Phys. Rev. Lett. 107 (2011) 054101.
[15] X. Han, Z. Shen, W.-X. Wang, Z. Di, Robust reconstruction of complex networks from sparse data, Phys. Rev. Lett. 114 (2015) 028701.
[16] M. Andrecut, S.A. Kauffman, On the sparse reconstruction of gene networks, J. Comput. Biol. 15 (1) (2008) 21–30.
[17] R. Albert, A.-L. Barabási, Statistical mechanics of complex networks, Rev. Modern Phys. 74 (2002) 47–97.
[18] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512.
[19] T. Vicsek, A. Czirok, E. Ben-Jacob, I. Cohen, O. Shochet, Novel type of phase transition in a system of self-driven particles, Phys. Rev. Lett. 75 (1995) 1226–1229.
[20] W. Ren, R.W. Beard, Consensus seeking in multiagent systems under dynamically changing interaction topologies, IEEE Trans. Automat. Control. 49 (2004) 622–629.
[21] G.E.A.P.A. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, ACM Sigkdd Explor. Newsl. 6 (2004) 20–29.
[22] P. Erdös, A. Rényi, On the evolution of random graphs, Publ. Math. Inst. Hung. Acad. Sci. 5 (1960) 17–60.
[23] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (2008) 046110.
[24] http://rulai.cshl.edu/TRED/GRN/AP1htm.
[25] https://cb.utdallas.edu/cgi-bin/TRED/tred.cgi?process=home.