Reconstructing Heterogeneous Networks via Compressive Sensing and Clustering

Yichi Zhang, Chunhua Yang, Keke Huang, *Member, IEEE*, Marko Jusup, Zhen Wang, and Xuelong Li, *Fellow, IEEE*

***Abstract*—Reconstructing complex networks from observed data is a fundamental problem in network science. Compressive sensing, widely used for recovery of sparse signals, has also been used for network reconstruction under the assumption that networks are sparse. However, heterogeneous networks are not exactly sparse. Moreover, when using compressive sensing to recover signals, the projection matrix is usually a random matrix that satisfies the restricted isometry property (RIP) condition. This condition is much harder to satisfy during network reconstruction because the projection matrix depends on time-series data of network dynamics. To overcome these shortcomings, we devised a novel approach by adapting the alternating direction method of multipliers to find a candidate adjacency matrix. Then we used clustering to identify high-degree nodes. Finally, we replaced the elements of the candidate adjacency vectors of high-degree nodes, which are likely to be incorrect, with the corresponding elements of small-degree nodes, which are likely to be correct. The proposed method thus overcomes the shortcomings of compressive sensing and is suitable for reconstructing heterogeneous networks. Experiments with both artificial scale-free and empirical networks showed that the proposed method is accurate and robust.**

***Index Terms*—Complex networks, network reconstruction, node degree, hub nodes, sparsity.**

I. Introduction

COMPLEX networks are ubiquitous in the real world and, among others, encompass the World Wide Web [1], social relationships [2], [3], and biological networks [4]. In recent years, researchers have focused on the geometric features [4], control [5], and synchronization [6] of complex networks based on fixed or time-varying network structures [7]. In particular, accurate structural information about a complex network is both necessary and sufficient for a comprehensive understanding of network behavior and dynamics [8], [9]. Owing to experimental cost and measurement technology restrictions, it is hard to obtain structural information directly. Thus, a general method that can be used to mine network structural information from limited measurement data is important.

Recently, network structure reconstruction, the inverse of the network analysis problem, has attracted a great deal of attention across multiple disciplines. Not to be confused with network recovery [10], which deals with network functionality depending on the functional status of its nodes, network reconstruction treats the network as a black box and uses observational data to infer the true network structure. Among the many reconstruction methods, data-driven methods that infer a network's connectivity from limited observable data are increasing in popularity. For example, Huang *et al.* [11] studied how incorporating latent constraints enhances the performance of network reconstruction, Ma *et al.* [12] demonstrated the synergy of two different methods in reconstructing the structure of heterogeneous networks, Liu *et al.* [13] exploited the power-law property to propose a reconstruction method for scale-free networks, and Mei *et al.* [14] devised a solution to the network-reconstruction problem for two-layer networks using the Alternating Direction Method of Multipliers (hereafter ADMM).

Data-driven network structure reconstruction is challenging for the following reasons. First, because the structural information of a network is mostly implicit, which is sometimes referred to as gray information [15], the structure cannot be directly observed in measurable data. In gene regulatory networks, for instance, relationships between genes are accessible only by means of specialized instrumentation and laborious experimentation that paint only a partial picture. Second, as the number of nodes in a complex network increases, the number of dimensions of its possible structural configuration grows exponentially. Using again gene regulatory networks as an example, if the expected number of regulatory interactions is in the order of 10^5, which implies $2^{10^5} \approx 9 \cdot 10^{30102}$ structural configurations even if the network is assumed to be unweighted. Third, because collecting data from a complex network is expensive and the data collected is extremely scarce, network structure reconstruction is, in general, an ill-posed problem, i.e., we need to solve a mathematically underdetermined system of equations. Even our best data on gene regulatory networks usually cover just a small percentage of the expected number of regulatory interactions.

Manuscript received February 12, 2020; revised April 21, 2020; accepted May 12, 2020. Date of publication June 8, 2020; date of current version November 23, 2021. This work was supported in part by the National Natural Science Foundation of China under Grants 61703439, 61860206014, and 61621062, in part by the Innovation-Driven Project of Central South University, China under Grant 2019CX020, in part by the Natural Science Foundation of Hunan Province under Grant 2019JJ50777, and in part by the 111 Project, China under Grant B17048. *(Corresponding author: Keke Huang.)*

Yichi Zhang, Chunhua Yang, and Keke Huang are with the School of Automation, Central South University, Changsha 410083, China (e-mail: zndxzyc@csu.edu.cn; ychh@csu.edu.cn; huangkeke@csu.edu.cn).

Marko Jusup is with the World Hub Research Initiative, Institute of Innovative Research Tokyo Institute of Technology, 152-8550 Tokyo, Japan (e-mail: mjusup@gmail.com).

Zhen Wang and Xuelong Li are with the Center for Optical Imagery Analysis and Learning, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: zhenwang0@gmail.com; xuelong_li@ieee.org).

Digital Object Identifier 10.1109/TETCI.2020.2997011

Compressive sensing, which was originally used in the fields of numerical computing, signal processing, and computer vision [16]–[18], is a novel approach that is increasingly being applied in the domain of network reconstruction [19]. Specifically, according to compressive sensing theory, the task of sparse signal recovery is to transform a linear system into an optimization problem that can be solved by some greedy or convex optimization method. To guarantee effective signal recovery, a random matrix is often selected as the projection matrix because such a matrix can satisfy the Restricted Isometry Property (RIP) condition with high probability [20], [21]. In the context of the network reconstruction problem, a compressive sensing-based method decomposes the process of reconstructing the entire network into many local signal recovery problems, and the network structure, namely, the adjacency matrix, is reconstructed column by column. Wang *et al.* [22] and Han *et al.* [19] respectively used the basic pursuit method and the LASSO method to solve the compressive sensing problem for complex networks reconstruction. Chen *et al.* [23] demonstrated the effectiveness of compressive sensing in reconstructing wireless sensor networks, whereas Pan *et al.* [24] applied the approach to the reconstruction of biochemical reaction networks. The results of these previous works reveal that recovered network adjacency vectors are sparse, which suits compressive sensing-based methods. However, a critical difference from the sparse signal recovery task using compressive sensing theory is that the projection matrix in the network structure reconstruction task is defined by network dynamics that often fails to satisfy the RIP condition, especially when the network is heterogeneous.

In a heterogeneous network, there are always hub nodes, which act as bottlenecks in the network structure reconstruction task. The term “heterogeneity” here refers to node-degree distributions, such as power laws, for which a possible number of per-node connections spans several orders of magnitude. Because the adjacency vectors of the hub nodes may not be sparse, a larger scale projection matrix and more observed data are needed for precise reconstruction of those vectors. Owing to experimental cost or measurement technology restrictions, it is almost impossible to collect such huge amounts of data in actual situations, which means that it is not possible to obtain accurate local structure around hub nodes. This problem is an inevitable flaw of compressive sensing theory when applied directly to network reconstruction, yet the problem has been ignored by previous studies. Fortunately, in heterogeneous networks, hub nodes are often linked to small-degree nodes, and the adjacency vectors of the small-degree nodes are often sparse. A straightforward question arises: Can we use such latent structural information to get around the described flaw in the application of compressive sensing theory?

Because many complex networks are heterogeneous and there is a great need to determine the topology of such networks, we propose a compressive sensing-based methodology for undirected and unweighted network reconstruction that aims to significantly improve performance in the reconstruction of heterogeneous networks. Specifically, we supplemented compressive sensing with a clustering-based approach (CBA) to mine the structural information from time-series data and thus improve the reconstruction accuracy. We applied this methodology to artificial scale-free networks [25]–[27] with two different scales, as well as to two real networks, known as the Karate network [28], football network [29] and the Adjnoun network [30]. The results show a large improvement in network reconstruction performance compared with the original compressive sensing theory-based method. In summary, there are three main contributions of the method proposed herein. First, we demonstrate how to turn a traditional decomposed network reconstruction process into an integrated global optimization problem that improves the accuracy of the results. Second, using the linear program (i.e., box-constraint) relaxation, we transform a Boolean linear programming formulation of the problem, which is most accurate for mining unweighted networks, into a convex optimization problem that is faster to calculate, and then design an efficient ADMM solver to perform the actual calculations. Third, to exploit the latent structural information of undirected networks and thus account for the shortcomings of compressive sensing in relation to non-sparse hub nodes, we employ a clustering method to separate hub nodes from normal nodes, followed by reusing the reconstruction results for normal nodes to improve the reconstruction results for hub nodes.

The rest of the paper is organized as follows. In Section II, we formulate the problem of network reconstruction according to evolutionary game data, and then incorporate the reconstruction problem into the compressive sensing framework. In Section III, we propose our CBA method. In Section IV, we conduct a series of numerical experiments with several artificial and real networks to demonstrate that our method can effectively improve network structure reconstruction results. Finally, Section V comprises a discussion and concluding remarks.

II. Problem Formulation

Here, we introduce the game dynamics in complex networks, and then review the compressive sensing-based method for using time-series data to solve the network reconstruction problem.

A. Game Dynamics in Networks

In a typical game, agents occupy different nodes of a network with a known structure, and they usually adopt different strategies, from a given set of strategies, to achieve a maximum payoff under a certain mode of interaction. The three fundamental elements of the evolutionary game dynamics in networked systems are (i) the network structure, (ii) the strategy set and strategy selection model, and (iii) the game model.

The Prisoner’s Dilemma Game (PDG) is commonly used to model cooperation among selfish agents in a multi-agent system [19], [22], [31]. There are many interesting recent results about the effects of the network structure of complex networks on the PDG cooperation ratio [32]–[36]. In the networked PDG, each node is occupied by one agent, and each agent participates in games with their neighbors, determined by the network structure, in a pair-wise fashion. The strategies of the PDG are defined as follows:

1) If both agents of the game choose cooperation, recorded as (C, C), each one gets the "reward for mutual cooperation," R.
2) If one agent chooses cooperation and the other chooses defection, recorded as (C, D) or (D, C), the defector gets T, the "temptation to defect," and the cooperator gets S, the "sucker's payoff".
3) If both agents choose to defect, recorded as (D, D), then each of them gets P, "punishment for mutual defection".

Accordingly, the payoff matrix becomes

$$P_{\mathrm{G}} = \begin{pmatrix} R & S \\ T & P \end{pmatrix}. \tag{1}$$

We use the well-known weak PDG as the game model [37], for which the payoff matrix is

$$P_{\mathrm{PDG}} = \begin{pmatrix} 1 & 0 \\ b & 0 \end{pmatrix}, \tag{2}$$

where $1 < b < 2$ quantifies the temptation to defect. Mathematically, cooperation (C) can be defined with strategy vector $s(C) = (1, 0)^T$. Similarly for defection, $s(D) = (0, 1)^T$. The payoff of agent i gained by playing with agent j can be calculated as $s_i^T(t) P_{\mathrm{PDG}} s_j(t)$. In each round, agent i plays the game with their immediately adjacent neighbors, and thus earns payoff

$$u_i(t) = \sum_{j \in N_i} s_i^T(t) P_{\mathrm{PDG}} s_j(t), \tag{3}$$

where N_i is the set comprising the neighbors of agent i. After each game round, the agents update their strategies with reference to their neighbors' strategies. We use the proportional imitation rule as the strategy selection model. According to this rule, agent i randomly picks one of their neighbors j. If $u_i > u_j$, agent i maintains the same strategy in the next round of the game as in the current round. Otherwise, agent i adopts j's strategy with probability [38]

$$P(s_i(t+1) \leftarrow s_j(t)) = \frac{u_j - u_i}{b \max\{k_i, k_j\}} \tag{4}$$

where k_i and k_j are the node degrees of agents i and j, respectively. If the payoffs and strategies of all of the agents are recorded, the resulting time-series data can be used for the network reconstruction task.

B. Compressive Sensing for Network Reconstruction Using Evolutionary Game Data

The key to reconstructing a network on the basis of compressive sensing is the relationship between strategies and payoffs. Suppose that we can represent the relationships of each node in a certain network by adjacency matrix A with dimensions $N \times N$, where N is the number of nodes in the network. If agents i and j can interact with each other, then they are connected in the network and element a_{ij} in matrix A is equal to 1; otherwise, $a_{ij} = 0$. In general, if we do not know the interaction relationship between all agent pairs, the relationship between the strategies and payoffs of agent i in the tth round can be expressed by

$$u_i(t) = \sum_{N_i} a_{ij} F_{ij}(t), \tag{5}$$

where $F_{ij}(t) = s_i^T(t) P_{\mathrm{PDG}} s_j(t)$ represents the potential payoff, which is decided by the strategies of the two sides during one round of the game. If agent i has a connection to agent j, the potential payoff will be i's actual payoff.

Accordingly, we can stack the equations for each time instance into a combined equation based on the time-series data of strategies $S_i = [s_i(t_1), s_i(t_2), \ldots, s_i(t_M)]^T$ and payoffs $U_i = [u_i(t_1), u_i(t_2), \ldots, u_i(t_M)]^T$ of agent i from $t = 1$ to $t = M$, i.e.,

$$\begin{bmatrix} u_i(t_1) \\ \vdots \\ u_i(t_M) \end{bmatrix} = \begin{bmatrix} F_{i1}(t_1) & \cdots & F_{iN}(t_1) \\ \vdots & \ddots & \vdots \\ F_{i1}(t_M) & \cdots & F_{iN}(t_M) \end{bmatrix} \times \begin{bmatrix} a_{i1} \\ \vdots \\ a_{iN} \end{bmatrix}. \tag{6}$$

For notational consistency, we rewrite (6) as

$$Y_i = \Phi_i X_i, \tag{7}$$

where

$$Y_i = [u_i(t_1), \ldots, u_i(t_M)]^T \tag{8}$$

$$\Phi_i = \begin{bmatrix} F_{i1}(t_1) & \cdots & F_{iN}(t_1) \\ \vdots & \ddots & \vdots \\ F_{i1}(t_M) & \cdots & F_{iN}(t_M) \end{bmatrix} \tag{9}$$

$$X_i = [a_{i1}, \ldots, a_{iN}]^T. \tag{10}$$

It is clear from (7)–(10) that Φ_i and Y_i can be acquired from the time-series data of strategies and payoffs, and the only unknown is the interaction relationship, expressed as X_i in (7). Because most of the complex network is sparse, we can make full use of this structural information and incorporate it into the compressive sensing framework according to

$$\begin{aligned} &\min \|X_i\|_1 \\ &\text{s.t. } Y_i = \Phi_i X_i. \end{aligned} \tag{11}$$

The adjacency vectors of the remaining agents in the network can be calculated similarly. After the adjacency vectors of all of the agents have been obtained, adjacency matrix $A = [X_1, X_2, \ldots, X_N]^T$ can be acquired by stacking all of the adjacency vectors together.

III. Clustering-Based Approach (CBA)

In compressive sensing-based network reconstruction, an existing link can be distinguished from a nonexistent link by setting a single threshold value for the elements of reconstructed adjacency vectors [19], [39], giving good results when the dynamics and topology of each node are of the same type. However, many real-world networks follow a power-law degree distribution, reflecting large heterogeneity in node connectivity [40]–[42]. The application of compressive sensing theory directly to the reconstruction of this kind of networks cannot achieve good results. Thus, the method needs to be improved to generate more accurate reconstruction. To this end, we first introduce new constraints to enhance the reconstruction of binary elements in the adjacency matrix, and then we design a solution for the optimization problem through the ADMM framework. Second, we analyze the shortcomings of the compressive sensing method and propose a method by which to overcome them.

In an unweighted network, $a_{ij} = 1$ when there is a link between i and j; otherwise, $a_{ij} = 0$. Recalling that $X_i = \{a_{i1}, a_{i2}, \ldots, a_{iN}\}^T$, it is necessary to add a binary constraint, $X_i \in \{0,1\}^N$, $i = 1, \ldots, n$, thus turning this problem into an instance of Boolean linear programming (BLP)

$$\min \|X_i\|_1 \quad \text{s.t.} \begin{cases} Y_i = \Phi_i X_i \\ X_i \in \{0,1\}^N. \end{cases} \tag{12}$$

Because (12) is a combinatorial optimization problem, it is in general difficult to solve. Thus, to simplify the calculation, the problem is reformulated using Lagrangian relaxation

$$\min \|X_i\|_1 \quad \text{s.t.} \begin{cases} Y_i = \Phi_i X_i \\ a_{ij}(1 - a_{ij}) = 0, j = 1, \ldots, N \end{cases} \tag{13}$$

or, alternatively, linear program (LP) relaxation [43]

$$\min \|X_i\|_1 \quad \text{s.t.} \begin{cases} Y_i = \Phi_i X_i \\ X_i \in [0,1]^N. \end{cases} \tag{14}$$

The difference between (12) and (13) is the way of solving the problem. The main method for (12) is usually exhaustion, but for (13), a dual problem formulation is used to find the lower BLP bound. The way of finding this lower bound is called the Lagrangian relaxation [44]. As for Eq. (14), it just changes the constraints $X \in \{0,1\}^N$ to the linear inequalities $X \in [0,1]^N$, thus making the constraints continuous [45]. The way of replacing original Boolean constraints with linear inequalities is usually treated as an LP relaxation of the BLP [43]. These forms of the optimization problem are far easier to solve and give a lower bound on the optimal value of the BLP. Moreover, it can be proven that the lower bound obtained via Lagrangian relaxation and that obtained via LP relaxation are the same [43]. We opted for the LP relaxation and integrated it into the ADMM framework.

In general, the adjacency matrix reconstruction problem is decomposed into tasks of inferring local structures centered around each node, which is the local method for network reconstruction, but this inference process is time-consuming [11]. To improve efficiency, we can stack all of the tasks together. First, we vectorize the adjacency matrix to a single vector, represented by $X = \mathrm{vec}(A) = [X_1^T, X_2^T, \ldots, X_N^T]^T$, where X_i is the ith row of the adjacency matrix. At the same time, payoffs Y_i can be rewritten as $Y = [Y_1^T, Y_2^T, \ldots, Y_N^T]^T$, while the potential payoff matrix can be rewritten as $\Phi = \mathrm{diag}\{\Phi_1, \ldots, \Phi_N\} \in \mathbb{R}^{MN \times N^2}$. Thus, (7) is transformed into a global format for network reconstruction

$$Y = \Phi X, \tag{15}$$

which allows the global network structure to be reconstructed in one calculation process. The optimization problem in (12) then becomes

$$\min \|X\|_1 \quad \text{s.t.} \begin{cases} Y = \Phi X \\ X \in [0,1]^{N^2}. \end{cases} \tag{16}$$

Traditional compressive sensing usually reconstructs the adjacency matrix column by column. Here, the matrix reconstruction problem is transformed into a vector reconstruction problem which needs to be run only once. Thanks to the decentralized nature of the ADMM algorithm, the reconstruction process can be deployed to a computer cluster to enhance parallel computing performance [46]. Therefore, the proposed network reconstruction algorithm is faster and more efficient.

Next, we transform (16) into an ADMM algorithm.

$$\min f(X) + g(z_1) + \|z_2\|_1 \quad \text{s.t.} \begin{cases} X - z_1 = 0 \\ X - z_2 = 0, \end{cases} \tag{17}$$

where f is the indicator function of $\{x \in \mathbb{R}^n | \Phi X = Y\}$, and g is the indicator function of $\{z_1 \in [0,1]^{N^2}\}$. The augmented Lagrangian is formed by applying the method of multipliers

$$\begin{aligned} L_\rho(X, z_1, z_2, \lambda_1, \lambda_2) &= f(X) + g(z_1) + \|z_2\|_1 + \lambda_1^T(X - z_1) \\ &\quad + \lambda_2^T(X - z_2) + \frac{\rho}{2}\|X - z_1\|_2^2 \\ &\quad + \frac{\rho}{2}\|X - z_2\|_2^2 \end{aligned} \tag{18}$$

where $\lambda \in \mathbb{R}^p$ is the multiplier and $\rho > 0$ a penalty parameter. Then ADMM consists of the following iterations

$$\begin{aligned} X^{k+1} &= \arg\min L_p(X, z_1^k, z_2^k, \lambda_1^k, \lambda_2^k) \\ z_1^{k+1} &= \arg\min L_p(X^{k+1}, z_1, z_2^k, \lambda_1^k, \lambda_2^k) \\ z_2^{k+1} &= \arg\min L_p(X^{k+1}, z_1^{k+1}, z_2, \lambda_1^k, \lambda_2^k) \\ \lambda_1^{k+1} &= \lambda_1^k + \rho(X - z_1^{k+1}) \\ \lambda_2^{k+1} &= \lambda_2^{k+1} + \rho(X - z_2^{k+1}) \end{aligned} \tag{19}$$

For simplicity, the ADMM iterations can be rewritten in a slightly different, scaled form by combining linear and quadratic terms in the augmented Lagrangian and scaling the dual variable. If we take z_1 as an example, suppose the residual $r_1 = x - z_1$, then

$$\lambda_1^T r_1 + \frac{\rho}{2}\|r_1\|_2^2 = \frac{\rho}{2}\|r_1 + u_1\|_2^2 - \frac{\rho}{2}\|u_1\|_2^2 \tag{20}$$

where $u_1 = \frac{1}{\rho}\lambda_1$ is the scaled dual variable. ADMM iteration (19) is thus expressed in the scaled form as:

$$\begin{aligned} X^{k+1} &= \arg\min_x \Bigg(f(X) \\ &\quad + \frac{\rho}{2}\left(\frac{1}{2}\left\|X - z_1^k + u_1^k\right\|_2^2 + \frac{1}{2}\left\|X - z_2^k + u_2^k\right\|_2^2\right)\Bigg) \\ z_1^{k+1} &= \arg\min \left(g(z_1) + \frac{\rho}{2}\left\|X^{k+1} - z_1 + u_1^k\right\|_2^2\right) \end{aligned}$$

$$\begin{aligned} z_2^{k+1} &= \arg\min\left(\|z_2\|_1 + \frac{\rho}{2}\left\|X^{k+1} - z_2 + u_2^k\right\|_2^2\right) \\ u_1^{k+1} &= u_1^k + X^{k+1} - z_1^{k+1} \\ u_2^{k+1} &= u_2^k + X^{k+1} - z_2^{k+1} \end{aligned} \tag{21}$$

The residuals in the kth iteration are defined as $r_1^k = X^k + z_1^k$ and $r_2^k = X^k + z_2^k$, while u_1^k and u_2^k are the running sums of the residuals

$$\begin{aligned} u_1^k &= u_1^0 + \sum_{j=1}^{k} r_1^j \\ u_2^k &= u_2^0 + \sum_{j=1}^{k} r_2^j \end{aligned} \tag{22}$$

Next, we turn the iteration into an explicit solution:

$$\begin{aligned} X^{k+1} &= \Pi_1\left(\frac{1}{2}(z_1^k - u_1^k) + \frac{1}{2}(z_2^k - u_2^k)\right) \\ z_1^{k+1} &= \Pi_2(X^{k+1} + u_1^k) \\ z_2^{k+1} &= S_{1/\rho}(X^{k+1} + u_2^k) \\ u_1^{k+1} &= u_1^k + X^{k+1} - z_1^{k+1} \\ u_2^{k+1} &= u_2^k + X^{k+1} - z_2^{k+1}, \end{aligned} \tag{23}$$

where Π_1 is the projection operator in the Euclidean norm onto $\{X \in \mathbb{R}^n | \Phi X = Y\}$, i.e.,

$$\Pi_1(a) = \arg\min_{X^*}\{\|X^* - a\|_2^2 \,|\Phi X^* = Y\}, \tag{24}$$

Here, $a = \frac{1}{2}((z_1^k - u_1^k) + \frac{1}{2}(z_2^k - u_2^k))$, Π_2 is the projection operator of set $\{z_1 \in [0,1]^{N^2}\}$, i.e.,

$$\Pi_2(c) = \max(1, \min(0, c)) \tag{25}$$

where $c = X^{k+1} + u_1^k$, and S is the soft-thresholding operator:

$$S_d(t) = \begin{cases} t - d, & t > d \\ 0, & |t| \leqslant d \\ t + d, & t < -d. \end{cases} \tag{26}$$

Remark 1: The updating process for x can be rewritten explicitly as

$$X^{k+1} = \left(I - \Phi^T(\Phi\Phi^T)^{-1}\Phi\right) a + \Phi^T(\Phi\Phi^T)^{-1} y. \tag{27}$$

By setting $a = \frac{1}{2}((z_1^k - u_1^k) + \frac{1}{2}(z_2^k - u_2^k))$, the Lagrangian function becomes $L(X, \nu) = \frac{1}{2}\|X^* - a\|_2^2 + \nu\|\Phi X^* - Y\|_1$, where ν is the penalty parameter. Furthermore, Karush-Kuhn-Tucker (KKT) conditions lead to matrix form

$$\begin{pmatrix} I & \Phi^T\Phi \\ \Phi & 0 \end{pmatrix} \begin{pmatrix} X^* \\ \nu \end{pmatrix} = \begin{pmatrix} a \\ Y \end{pmatrix}, \tag{28}$$

from which it follows that

$$X^* = \left(I - \Phi^T(\Phi\Phi^T)^{-1}\Phi\right) a + \Phi^T(\Phi\Phi^T)^{-1} Y. \tag{29}$$

Thus, ADMM iterations become

$$\begin{aligned} x^{k+1} &= \left(I - \Phi^T\left(\Phi\Phi^T\right)^{-1}\Phi\right) a + \Phi^T\left(\Phi\Phi^T\right)^{-1} y \\ z_1^{k+1} &= \max(0, \min(c, 1)) \\ z_2^{k+1} &= S_{1/\rho}(x^{k+1} + u_2^k) \\ u_1^{k+1} &= u_1^k + x^{k+1} - z_1^{k+1} \\ u_2^{k+1} &= u_2^k + x^{k+1} - z_2^{k+1} \end{aligned} \tag{30}$$

where $a = \frac{1}{2}((z_1^k - u_1^k) + \frac{1}{2}(z_2^k - u_2^k))$ and $c = X^{k+1} + u_1^k$.

In practice, observing all constituents of, or all interactions in, complex systems is unfeasible, causing random errors or systematic bias in sampling [47]. If the available data is erroneous in addition to being incomplete, we can write the ADMM framework:

$$\begin{aligned} &\min \frac{1}{2}\|\Phi x - y\| + g(z_1) + \|z_2\|_1 \\ &\text{s.t.} \begin{cases} x - z_1 = 0 \\ x - z_2 = 0 \end{cases} \end{aligned} \tag{31}$$

and the iterations turn to

$$\begin{aligned} x^{k+1} &= (\rho I + \Phi^T\Phi)^{-1} \\ &\quad \times \left(\Phi^T y + \rho\left(\frac{1}{2}\left(z_1^k - u_1^k\right) + \frac{1}{2}(z_2^k - u_2^k)\right)\right) \\ z_1^{k+1} &= \max(0, \min(a, 1)) \\ z_2^{k+1} &= S_{1/\rho}(x^{k+1} + u_2^k) \\ u_1^{k+1} &= u_1^k + x^{k+1} - z_1^{k+1} \\ u_2^{k+1} &= u_2^k + x^{k+1} - z_2^{k+1} \end{aligned} \tag{32}$$

where $a = \frac{1}{2}((z_1^k - u_1^k) + \frac{1}{2}(z_2^k - u_2^k))$, and the process of deriving the iterations is the same as in the case without noise. Additionally, term $\rho I + \Phi^T\Phi$ is always invertible because $\rho > 0$. Thus, the numerical instability caused by the ill-conditioned projection matrix is avoided.

The spark of a matrix is defined as the smallest number of linearly dependent column-vectors from this matrix, i.e.,

$$\text{spark}(A) = \min_{b \neq 0} \|b\|_0 \quad \text{s.t.} \quad Ab = 0 \tag{33}$$

where $b \in \mathbb{R}^N$ is a K-sparse vector and $A \in \mathbb{R}^{M \times N}$ is an observation matrix. The term K-sparse means that the number of nonzero elements of b is less than K. According to [17], given a sparse signal, a linear system can be changed from having an infinite number of solutions to having a unique solution in compressive sensing theory under the condition that the spark of the projection matrix is larger than twice the sparsity of the signal vector, i.e., $\text{spark}(A) > 2K$, where K represents the sparsity of the vector. However, in practical network reconstruction, the projection matrix is defined by network structure and dynamics, which do not always satisfy the RIP condition, especially with heterogeneous networks. The main reason is that scale-free and other heterogeneous networks include hub nodes whose adjacency vectors are far from being sparse. The RIP condition is thus not guaranteed, making the reconstruction results practically equivalent to random guessing; we call this an "inevitable flaw" of compressive sensing in the reconstruction of scale-free or other heterogeneous complex networks. Assuming undirected links, however, the node on either side of a link should yield the same solution. If for hub nodes we encounter situations that a_{ij} is not equal to a_{ji}, then the reconstruction process has failed and needs to be improved. More formally, the adjacency matrix of an undirected network is symmetric, and the network reconstruction process should comply with the property that $a_{ij} = a_{ji}$. This property can be utilized to replace the unreliable values in adjacency vectors of hub nodes with the more reliable values indicated by small-degree nodes. For example, if node i is a hub node and node j is a small-degree node, then from the perspective of compressive sensing a_{ij} is unreliable whereas a_{ji} is reliable—a_{ij} should be replaced with a_{ji}.

The next problem is how to identify hub nodes. After performing multiple tests, we found that an efficient unsupervised clustering algorithm, K-means clustering [48], possesses the desired traits, and can be exploited in combination with compressive sensing to develop a robust reconstruction method for heterogeneous networks. The algorithm partitions network nodes into two clusters, one for hub nodes, and the other for common, small-degree nodes. K-means clustering can be viewed as a highly effective structural classifier for hub and common nodes in heterogeneous networks. Intuitively, an agent who is popular in a certain group, can interact with more agents than other, common agents; thus, the popular agent may have more chances to play the game with others to gain a higher payoff, whereas a common agent gains a much lower payoff under the same conditions. We can conclude that in a networked evolutionary game, the payoff of a hub node in a single round is higher than the payoff of common nodes. To distinguish hub nodes from common nodes, we stack the payoffs of each agent into a vector, and then group all of the agents' payoff vectors into two clusters, one with more elements and the other with fewer elements. The cluster with fewer elements is identified as the hub node cluster because the number of hub nodes is small relative to the total number of nodes in a heterogeneous network. After finding these hub nodes, we use the reconstruction results for small-degree nodes to replace those of the hub nodes. The replacement process after clustering is implemented using the following steps:

1) Group the time-series-payoff data into two clusters and choose the smaller one as the hub node set, denoted Γ.
2) Choose one of the hub nodes from Γ and pick its adjacency vector from A. For example, node i is a hub node, and X_i is picked for revision.
3) Suppose that j indexes nodes linked to i. If j belongs to Γ, then a_{ij} is kept unchanged because j is also a hub node; otherwise, replace a_{ij} with a_{ji} because the structural information on j is more reliable.

The steps of the proposed method are shown in detail in Algorithm 1 (sidebar), and an example of the application of the proposed method is shown in Fig. 1.

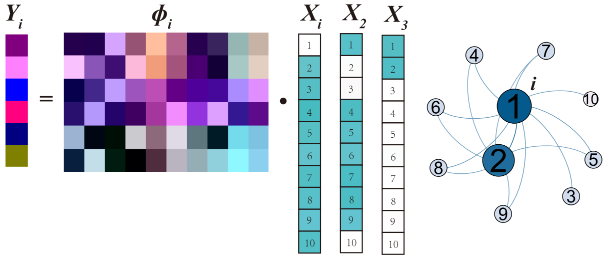

Fig. 1. An example of the application of the proposed method for network structure reconstruction. The real structure of the network is shown on the right. Quantity Y_i represent observed values, and Φ_i represent the projection matrix, where different colors indicate different real numbers. The two-color squares in X_i represent whether there is (cyan) or there is not (white) a link between node i and another network node (identified by the number in the square). In this network, node 1 is connected to all other nodes, so all squares in its adjacency vector are cyan except its own square (which is irrelevant). Nodes 2 and 3 have fewer green squares than node 1 because they have fewer neighbors. We can see that node 3 is sparse, unlike nodes 1 and 2, and it is linked to node 1, but not to node 2. Therefore, node 3's adjacency vector can be used to fix the estimated adjacency vector of node 1, following Algorithm 1, by replacing a_{13} with a_{31}. However, node 3 cannot be used for a similar purpose in conjunction with node 2, nor can node 2 be used to infer the links of node 1.

Algorithm 1: Clustering-Based Approach (CBA).

```
Input:
  Strategy vectors $S_i = [s_i(t_1), s_i(t_2), \ldots, s_i(t_M)]^T, \forall i$.
  Payoff vectors $Y_i = [u_i(t_1), u_i(t_2), \ldots, u_i(t_M)]^T, \forall i$.
Output:
  Reconstructed adjacency matrix $\hat{A}$.
Step 1 (Data Preprocessing)
  Step 1.1: Stack payoff vectors in $Y = [Y_1, Y_2, \ldots, Y_N]^T$.
  Step 1.2: Use strategy vectors to calculate potential payoffs $F_{ij}(t)$ and stack them into matrix $\Phi$ from (15).
Step 2 (Preliminary Reconstruction):
  Step 2.1: Calculate adjacency vector $X$ based on $Y$, $\Phi$, and ADMM algorithm as in (16), (17), and (30).
  Step 2.2: Reshape vector $X$ into adjacency matrix $\hat{A}_1$ of the same dimensions as input adjacency matrix $A$.
Step 3 (Fine Tuning):
  Step 3.1: Use $K$-means clustering to divide payoffs $Y$ into two clusters, then choose the smaller one as the set of large-degree nodes denoted $\Gamma$.
  Step 3.2:
  for $i$ in $\Gamma$:
    Apply fine tuning to $i$'s adjacency vector in $\hat{A}_1$ to obtain the final reconstructed matrix, $\hat{A}$.
  end for
return $\hat{A}$.
```

IV. Numerical Experiments

A. Experiments With Accurate Observations

To illustrate the effectiveness of the proposed method, we designed two scale-free (SF) networks [25] of different scales. One SF network has 50 nodes and the other has 100 nodes, where both networks were created by preferentially attaching four new nodes in each step of the generative process. The adjacency matrix is the ground truth we want to reconstruct and the standard by which to validate the proposed method. Data on every round

TABLE I
CONFUSION MATRIX FOR THE BINARY CLASSIFICATION OF PREDICTED OUTCOMES

		Predicted Label	
		Target Class	Negative Class
Actual Label	Target Class	TP	FN
	Negative Class	FP	TN

of the network game, i.e., each node's strategy and payoff, are recorded in the form of time series. The amount of data used is quantified by $R_D = L/N$, where L is the length of the recorded time-series data, and N is the number of nodes in the network. To evaluate the reconstruction performance of the proposed method, we employed two standard indices: the area under the receiver operating characteristic curve (AUROC) and the area under the precision-recall curve (AUPR). Additionally, we used a binary classification test method. By comparing the standard, i.e., the true adjacency matrix, with the predicted outcome, the result for each node can be classified as a true positive (TP), false positive (FP), true negative (TN), or false negative (FN). The confusion matrix is shown in Table I. Based on this confusion matrix, we calculated the method's accuracy as

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}. \tag{34}$$

Note that we set a threshold to regularize values in the predicted adjacency matrix. For example, if 0.1 is chosen as a threshold, then the predicted value is set to 1 if it falls within interval [0.9, 1.1], whereas it is set to 0 if it falls within interval $[-0.1, 0.1]$. We used 0.1 as the threshold.

In numerical simulation, we compare the proposed method to four other algorithms used in network reconstruction. First is a standard algorithm based on Compressive Sensing Theory (CST) that has been extensively documented in literature [19], [22]. Matching Pursuit (MP) [49] and Orthogonal Matching Pursuit (OMP) [50] are two famous dictionary learning algorithms that can be applied to the network reconstruction problem because the underlying mathematical abstraction is equivalent. The OMP method is an upgraded version of the MP method. Finally, Relevance Vector Machine (RVM) [51] is a sparse probability model similar to Support Vector Machine (SVM), and represents a novel supervised learning method that incorporates the elements of Bayesian theory. The RVM method is, among others, applicable to network reconstruction problems [52].

Fig. 2 depicts the reconstruction results as a function of R_D in two artificial SF networks of different scale. The performance of all methods increases with the amount of data, except that of Matching Pursuit (MP). The performance of MP is close to 0.5 across the board, meaning that this method is just like random guessing and the results are unreliable. The method proposed herein (CBA) performs the best with the same amount of data, whether measured by the AUPR or the AUROC value. The mean values obtained using CBA are always higher than those obtained by other classical algorithms, even when the amount of data is relatively small.

We also tested the performance in two real networks with different sizes. The first network is the famous social network of friendships among 34 members of a karate club at a U.S. university in the 1970s [28]. The Karate network has 34 nodes and 78 edges. The second network is an adjacency network of common adjectives and nouns in the novel David Copperfield by Charles Dickens [30], which contains English adjectives and nouns commonly occurring in the novel. This Adjnoun network has 112 nodes and 425 edges. The node degree distributions of the two networks are shown in Fig. 3.

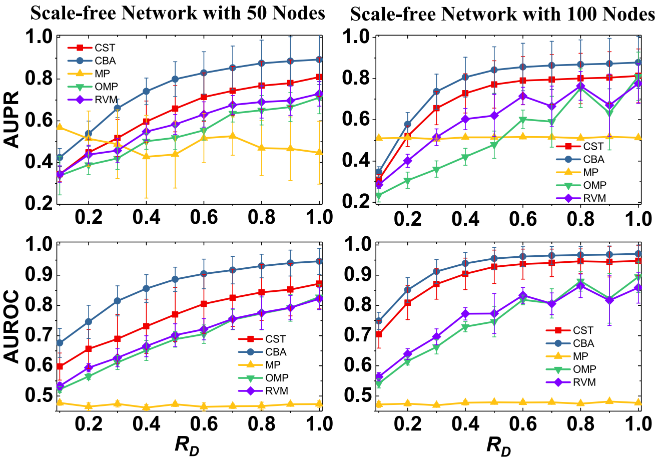

Fig. 2. AUPR and AUROC values, as functions of R_D, of the reconstruction results for SF networks of two different scales, based on time-series data obtained from over 50 independent experiments. One network contains 50, and the other 100 nodes. The red squares and lines depict the results obtained by the traditional compressive sensing theory-based method, the blue circles and lines depict those obtained by the proposed method, the yellow triangles and lines depict those obtained by MP, the green triangles and lines depict those obtained by OMP, and violet diamonds and lines depict those obtained by RVM. The symbols represent mean values, and the error bars denote standard deviations.

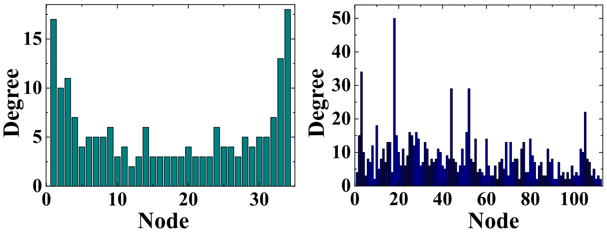

Fig. 3. Degree distributions of Karate (left) and Adjnoun (right) networks. In both networks there are nodes with exceptionally high degrees. These are the hub nodes for which standard CST reconstruction fails.

In Fig. 3, we can see that both networks have a heterogeneous degree distribution. It is obvious that agents 1 and 34 in the Karate network, and nodes 1, 18, 44, and 52 in the Adjnoun network, are hubs. Close to 50% of the elements in the adjacency vectors of hub nodes are nonzero elements, which means that these adjacency vectors are not sparse at all. Thus, in practice, it is very difficult to reconstruct the hub nodes accurately, so these nodes become bottlenecks in the CST performance. Fortunately, our CBA method can find such hub nodes precisely and improve the reconstruction of their adjacency vectors.

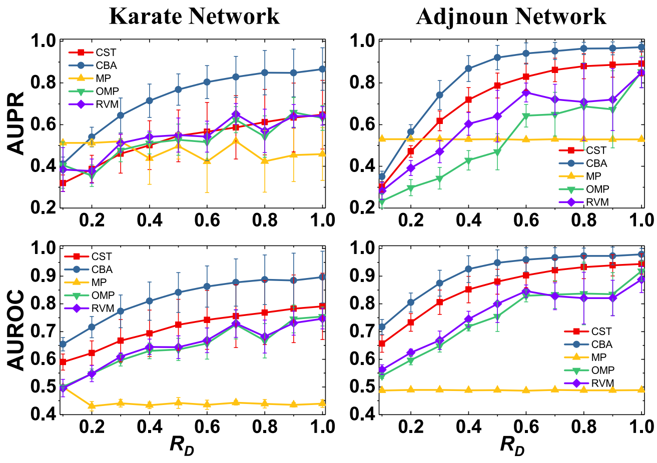

Fig. 4. Comparing the performance of the proposed method to that of four different algorithms for network reconstruction II. Shown are the AUPR and AUROC curves as functions of R_D obtained for the Karate network (left) and the Adjnoun network (right) over the course of 50 independent numerical experiments. Red squares, blue circles, yellow triangles, green inverted triangles, and violet diamonds respectively denote the standard compressive sensing algorithm (CST), matching pursuit (MP), orthogonal matching pursuit (OMP), and Relevance Vector Machine (RVM). The symbols indicate mean values, and the error bars denote standard deviations.

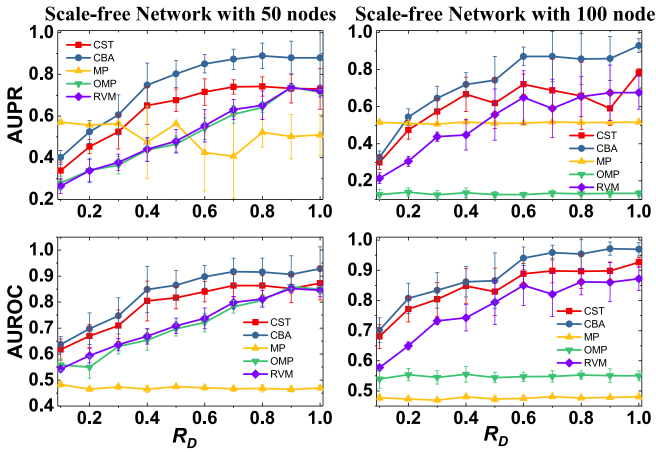

Fig. 5. AUPR and AUROC values, as functions of R_D, of the reconstruction results for SF networks with two different scales, based on noisy time-series data obtained from over 50 independent experiments. One network contains 50, and the other 100 nodes. The red squares and lines depict the results obtained by the traditional compressive sensing theory-based method, the blue circles and lines depict those obtained by the proposed method, the yellow triangles and lines depict those obtained by MP, the green triangles and lines depict those obtained by OMP, and violet diamonds and lines depict those obtained by RVM. The symbols represent mean values, and the error bars denote standard deviations. Noise amplitude $\sigma = 0.12$.

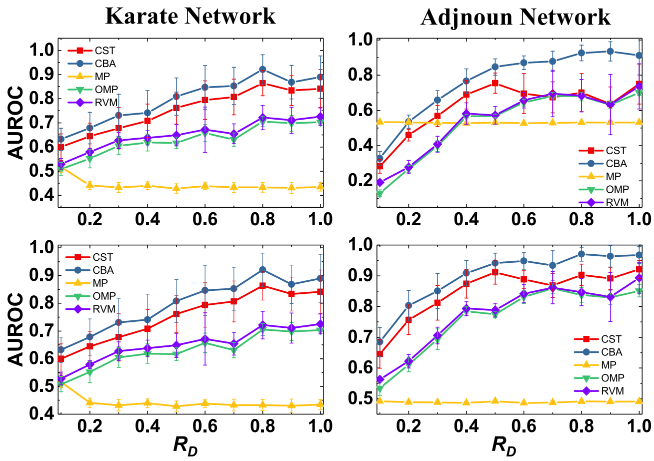

Fig. 6. Comparing the performance of the proposed method to that of four different algorithms for network reconstruction IV. Shown are the AUPR and AUROC curves as functions of R_D obtained for the Karate network (left) and the Adjnoun network (right) over the course of 50 independent numerical experiments. Game data is contaminated with noise of amplitude $\sigma = 0.12$. Red squares, blue circles, yellow triangles, green inverted triangles, and violet diamonds respectively denote the standard compressive sensing algorithm (CST), matching pursuit (MP), orthogonal matching pursuit (OMP), and Relevance Vector Machine (RVM). The symbols indicate mean values, and the error bars denote standard deviations.

Fig. 4 shows the reconstruction results for two real-world networks as a function of R_D. Irrespective of whether performance is evaluated by the AUPR or the AUROC values, CBA performs better than other control algorithms when R_D is held fixed. In the Karate network, both AUPR and AUROC curves increase more than other curves as R_D increases. In the Adjnoun network, although the performance difference between CBA and CST is not obvious when R_D is either very large or very small, for intermediate R_D values, especially from 0.3 to 0.7, the AUPR results for CBA are greatly improved compared with those for other methods. The AUROC results do not differ as much, but CBA performance is always better than CST performance. In summary, these results show that the proposed method is efficient in improving the reconstruction performance in heterogeneous networks based on limited time-series data without noise.

B. Experiments With Noise-Contaminated Observations

Although an ideal situation in which the observed data are accurate is often assumed, in reality, the observed data contain noise or include gaps for a variety of reasons, such as measurement error or low-resolution observers. To evaluate the robustness of our proposed method, we tested the effect of noise on reconstruction performance. The observed noise is assumed to originate from a uniform distribution with a range of $[0, \sigma]$, where σ is the noise amplitude. When two agents in a network interact with each other, their observed payoffs contain noise.

Fig. 5 shows AUPR and AUROC reconstruction curves for artificial networks, while Fig. 6 depicts the results for real networks. The observed data sets for both types of network are contaminated by noise ($\sigma = 0.12$). The results show that CBA is more resistant to noise interference; regardless of the size of the network or the amount of data used for network reconstruction, the performance remains stable.

Lastly, we summarize the effect of noise on CBA results in Fig. 7. Even with noise amplitude $\sigma = 1.2$, which is equal to the maximum value in the payoff matrix, the accuracy of reconstruction for the two larger networks, calculated by (34), is higher than 0.70, and the reconstruction accuracy for the two smaller networks is higher than 0.55. Thus, CBA can efficiently resist noise corrosion, even when the noise level is very high. Furthermore, it is interesting to note that as the network size

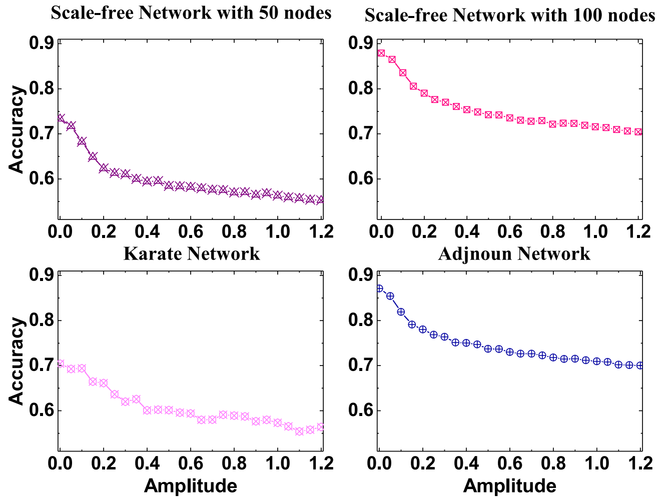

Fig. 7. Changes in the accuracy of the reconstruction results for the four networks with increasing noise amplitude, based on 100 independent experiments. The symbols represent mean values. The upper panels show the results for artificial scale-free networks generated by the Barabási-Albert model with 50 and 100 nodes, and the lower panels show the results for the real Karate and Adjnoun networks. The data ratio R_D is 0.4.

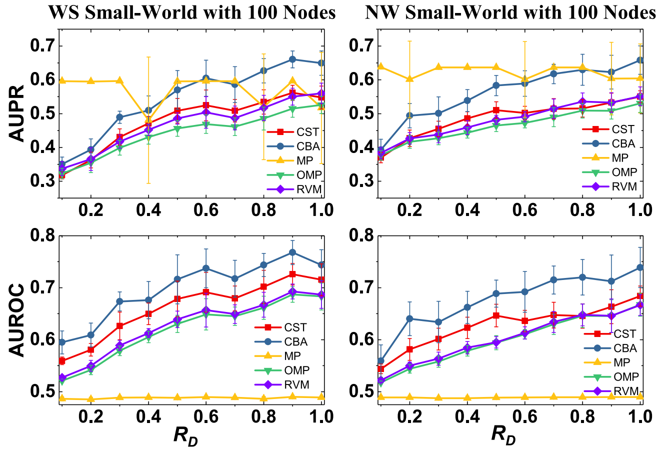

Fig. 8. Comparing the performance of the proposed method to that of four different algorithms for network reconstruction V. Shown are the AUPR and AUROC curves as functions of R_D obtained for the Watts-Strogatz (left) and the Newman-Watts (right) small-world networks over the course of 50 independent numerical experiments. Both networks have 100 nodes and the average degree of 20. Red squares, blue circles, yellow triangles, green inverted triangles, and violet diamonds respectively denote the standard compressive sensing algorithm (CST), matching pursuit (MP), orthogonal matching pursuit (OMP), and Relevance Vector Machine (RVM). The symbols indicate mean values, and the error bars denote standard deviations.

increases the results become more accurate. The reason may lie in the fact that, as the network size increases, the networks become sparser, which improves the efficiency of the proposed method because this method is ultimately based on compressive sensing.

C. Experiments on Homogeneous Networks

While our focus has been on heterogeneous networks, we still wanted to test the performance of the proposed method in homogeneous networks. Figs. 8 and 9 respectively show the results of numerical experiments with accurate and noise-contaminated observations. With accurate observations, CBA maintains relatively good performance compared to the other algorithms. This is because we stack the adjacency matrix into one vector and perform reconstruction globally instead of locally (when columns of the adjacency matrix are reconstructed one by one). With noise-contaminated observations, however, the performance is less impressive, although not any worse than that of the other methods. The fine-tuning step by means of clustering has no effect on the results because none of the nodes are truly hubs, meaning that adjacency matrix elements a_{ij} and a_{ji} are reconstructed with the same reliability.

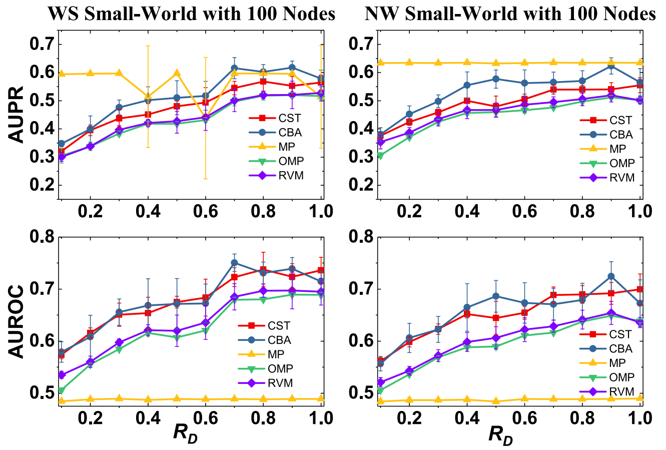

Fig. 9. Comparing the performance of the proposed method to that of four different algorithms for network reconstruction VI. Shown are the AUPR and AUROC curves as functions of R_D obtained for the Watts-Strogatz (left) and the Newman-Watts (right) small-world networks over the course of 50 independent numerical experiments. Both networks have 100 nodes and the average degree of 20. Game data is contaminated with noise of amplitude $\sigma = 0.12$. Red squares, blue circles, yellow triangles, green inverted triangles, and violet diamonds respectively denote the standard compressive sensing algorithm (CST), matching pursuit (MP), orthogonal matching pursuit (OMP), and Relevance Vector Machine (RVM). The symbols indicate mean values, and the error bars denote standard deviations.

V. Conclusion

It is difficult to reconstruct the underlying structure of complex networks from observed data. Despite all the effort devoted to this endeavor, few studies have considered the latent structural information hidden in time-series data. Herein, we focused on the reconstruction of heterogeneous network structures globally based on game dynamics data. By analyzing the features of heterogeneous networks as well as the shortcomings of compressive sensing methodology, we developed a clustering-based approach that could resolve the bottlenecks of network structure reconstruction caused by hub nodes. Furthermore, we incorporated the binary feature of undirected networks into the proposed method, and used the ADMM algorithm to solve the optimization problem, further enhancing the accuracy. By comparing the reconstruction results with multiple other methods, we showed that the proposed method can efficiently improve reconstruction performance and resist noise contamination. It is worth mentioning that, although we designed our approach with heterogeneous networks in mind, the reconstruction of homogeneous networks is also possible.

We tested the algorithm on two artificial networks and two real networks based on evolutionary game data, but the proposed method can be extended to even more complex networks and to other types of network dynamics. For example, it might be worthwhile to extend the method to reconstruction from the oscillator dynamics of synchronous networks. At present, we have tested only undirected and unweighted networks, but the proposed method may inspire future extensions to directed and weighted networks with different types of constraints.

There is an important limitation to be addressed in future work. Our method integrates the traditional decomposed network reconstruction process into one optimization problem to use the global nature of the network and improve the accuracy of the results. However, as the number of nodes increases the memory space required in the computer becomes larger and larger. When the number of network nodes rises to a certain level, a normal computer cannot satisfy the reconstruction task. How to make a normal computer complete the task of large network reconstruction, or in other words, how to disassemble the network vector into multiple parts for distributed solution while ensuring the global nature is a key future research direction. In the meantime, our work sheds lights on network reconstruction problem from the perspective of heterogeneous network structure, and provides a useful tool for network analysis and many other potential applications.

REFERENCES

[1] A. J. Chakravarti, G. Baumgartner, and M. Lauria, “The organic grid: Self-organizing computation on a peer-to-peer network,” *IEEE Trans. Syst., Man, Cybern., Part A: Syst. Humans*, vol. 35, no. 3, pp. 373–384, May 2005.

[2] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, “Complex networks: Structure and dynamics,” *Phys. Reports*, vol. 424, no. 4-5, pp. 175–308, 2006.

[3] H.-J. Li, Z. Bu, A. Li, Z. Liu, and Y. Shi, “Fast and accurate mining the community structure: Integrating center locating and membership optimization,” *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2349–2362, Sep. 2016.

[4] T. Nakano, “Biologically inspired network systems: A review and future prospects,” *IEEE Trans. Syst., Man, Cybern., Part C Appl. Rev.*, vol. 41, no. 5, pp. 630–643, Sep. 2011.

[5] Y.-Y. Liu, J.-J. Slotine, and A.-L. Barabási, “Controllability of complex networks,” *Nature*, vol. 473, no. 7346, pp. 167–173, 2011.

[6] R. Lu, W. Yu, J. Lü, and A. Xue, “Synchronization on complex networks of networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 11, pp. 2110–2118, Nov. 2014.

[7] A. Ahmed and E. P. Xing, “Recovering time-varying networks of dependencies in social and biological studies,” *Proc. Nat. Acad. Sci.*, vol. 106, no. 29, pp. 11 878–11 883, 2009.

[8] Z. Wang, A. Szolnoki, and M. Perc, “Interdependent network reciprocity in evolutionary games,” *Scientific Reports*, vol. 3, pp. 1–7, 2013.

[9] M. Lu, Z. Zhang, Z. Qu, and Y. Kang, “LPANNI: Overlapping community detection using label propagation in large-scale complex networks,” *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 9, pp. 1736–1749, Sep. 2019.

[10] Y. Shang, “Localized recovery of complex networks against failure,” *Scientific Reports*, vol. 6, 2016, Art. no. 30521.

[11] K. Huang, Z. Wang, and M. Jusup, “Incorporating latent constraints to enhance inference of network structure,” *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 466–475, Mar. 2020.

[12] L. Ma, X. Han, Z. Shen, W.-X. Wang, and Z. Di, “Efficient reconstruction of heterogeneous networks from time series via compressed sensing,” *PLOS ONE*, vol. 10, no. 11, 2015, Art. no. e0142837.

[13] J. K. Liu *et al.*, “Inference of neuronal functional circuitry with spike-triggered non-negative matrix factorization,” *Nature Commun.*, vol. 8, no. 1, p. 149, 2017.

[14] G. Mei, X. Wu, Y. Wang, M. Hu, J.-A. Lu, and G. Chen, “Compressive-sensing-based structure identification for multilayer networks,” *IEEE Trans. Cybern.*, vol. 48, no. 2, pp. 754–764, Feb. 2018.

[15] Y. Shang, “Robustness of scale-free networks under attack with tunable grey information,” *Europhysics Lett.*, vol. 95, no. 2, 2011, Art. no. 28005.

[16] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[17] D. L. Donoho, “Compressed sensing,” *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[18] E. J. Candès and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[19] X. Han, Z. Shen, W.-X. Wang, and Z. Di, “Robust reconstruction of complex networks from sparse data,” *Phys. Rev. Lett.*, vol. 114, no. 2, 2015, Art. no. 028701.

[20] E. Candes and J. Romberg, “Sparsity and incoherence in compressive sampling,” *Inverse Problems*, vol. 23, no. 3, pp. 969–985, 2007.

[21] E. J. Candes and T. Tao, “Decoding by linear programming,” *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[22] W.-X. Wang, Y.-C. Lai, C. Grebogi, and J. Ye, “Network reconstruction based on evolutionary-game data via compressive sensing,” *Phys. Rev. X*, vol. 1, no. 2, 2011, Art. no. 021021.

[23] W. Chen and I. Wassell, “Energy efficient signal acquisition via compressive sensing in wireless sensor networks,” in *Proc. 6th Int. Symp. Wireless Pervasive Comput.*, 2011, vol. 2325, pp. 1–6.

[24] W. Pan, Y. Yuan, J. Gonçalves, and G.-B. Stan, “Reconstruction of arbitrary biochemical reaction networks: A compressive sensing approach,” in *Proc. IEEE 51st Conf. Decis. Control*, 2012, pp. 2334–2339.

[25] A.-L. Barabási and R. Albert, “Emergence of scaling in random networks,” *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[26] F. C. Santos and J. M. Pacheco, “Scale-free networks provide a unifying framework for the emergence of cooperation,” *Phys. Rev. Lett.*, vol. 95, no. 9, 2005, Art. no. 098104.

[27] P. K. Pandey and B. Adhikari, “A parametric model approach for structural reconstruction of scale-free networks,” *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 10, pp. 2072–2085, Oct. 2017.

[28] W. W. Zachary, “An information flow model for conflict and fission in small groups,” *J. Anthropological Res.*, vol. 33, no. 4, pp. 452–473, 1977.

[29] M. Girvan and M. E. J. Newman, “Community structure in social and biological networks,” *Proc. Nat. Acad. Sci. United States America*, vol. 99, no. 12, pp. 7821–7826, 2002.

[30] M. E. Newman, “Finding community structure in networks using the eigenvectors of matrices,” *Phys. Rev. E*, vol. 74, no. 3, 2006, Art. no. 036104.

[31] M. Perc and A. Szolnoki, “Social diversity and promotion of cooperation in the spatial prisoner! s dilemma game,” *Phys. Rev. E*, vol. 77, no. 1, 2008, Art. no. 011904.

[32] M. Perc, J. J. Jordan, D. G. Rand, Z. Wang, S. Boccaletti, and A. Szolnoki, “Statistical physics of human cooperation,” *Phys. Reports*, vol. 687, pp. 1–51, 2017.

[33] K. Huang, Y. Cheng, X. Zheng, and Y. Yang, “Cooperative behavior evolution of small groups on interconnected networks,” *Chaos, Solitons Fractals*, vol. 80, pp. 90–95, 2015.

[34] D. A. Gianetto and B. Heydari, “Catalysts of cooperation in system of systems: The role of diversity and network structure,” *IEEE Syst. J.*, vol. 9, no. 1, pp. 303–311, Mar. 2015.

[35] M. Perc, “Evolution of cooperation on scale-free networks subject to error and attack,” *New J. Phys.*, vol. 11, no. 3, 2009, Art. no. 033027.

[36] J. Tanimoto, “Dilemma solving by the coevolution of networks and strategy in a 2 × 2 game,” *Phys. Rev. E*, vol. 76, no. 2, 2007, Art. no. 021126.

[37] M. A. Nowak and R. M. May, “Evolutionary games and spatial chaos,” *Nature*, vol. 359, no. 6398, pp. 826–829, 1992.

[38] K. Huang, X. Zheng, Z. Li, and Y. Yang, “Understanding cooperative behavior based on the coevolution of game strategy and link weight,” *Scientific Reports*, vol. 5, 2015, Art. no. 14783.

[39] Z. Shen, W.-X. Wang, Y. Fan, Z. Di, and Y.-C. Lai, “Reconstructing propagation networks with natural diversity and identifying hidden sources,” *Nature Commun.*, vol. 5, p. 4323, 2014.

[40] R. Albert, H. Jeong, and A.-L. Barabási, “Internet: Diameter of the world-wide web,” *Nature*, vol. 401, no. 6749, pp. 130–131, 1999.

[41] A.-L. Barabási and E. Bonabeau, “Scale-free networks,” *Scientific Amer.*, vol. 288, no. 5, pp. 60–69, 2003.

[42] A.-L. Barabási, "Scale-free networks: A decade and beyond," *Science*, vol. 325, no. 5939, pp. 412–413, 2009.

[43] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[44] V. M. Manquinho and J. Marques-Silva, "Effective lower bounding techniques for pseudo-Boolean optimization [EDA applications]," in *Proc. IEEE Des., Autom. Test Europe*, 2005, pp. 660–665.

[45] D. Malioutov and M. Malyutov, "Boolean compressed sensing: LP relaxation for group testing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2012, pp. 3305–3308.

[46] S. Boyd *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[47] Y. Shang, "Subgraph robustness of complex networks under attacks," *IEEE Trans. Syst., Man, Cybern.: Syst.*, vol. 49, no. 4, pp. 821–832, Apr. 2019.

[48] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *J. Roy. Statistical Soc. Series C Appl. Statist.*, vol. 28, no. 1, pp. 100–108, 1979.

[49] S. G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[50] T. T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4680–4688, Jul. 2011.

[51] M. E. Tipping, "Sparse bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.

[52] K. Huang, W. Deng, Y. Zhang, and H. Zhu, "Sparse bayesian learning for network structure reconstruction based on evolutionary game data," *Physica A: Statistical Mech. Appl.*, vol. 541, 2020, Art. no. 123605.

Yichi Zhang received the B.A. degree in automatic control from Central South University, Changsha, China, in 2016. He is currently working toward the master's degree. His research interests include complex network and game theory.

Chunhua Yang received the M.Eng. degree in automatic control engineering and the Ph.D. degree in control science and engineering from Central South University, Changsha, China, in 1988 and 2002, respectively. From 1999 to 2001, She was with the Department of Electrical Engineering, Katholieke Universiteit Leuven, Leuven, Belgium She is currently a Full Professor with the Central South University. Her research interests include modeling and optimal control of complex industrial processes, intelligent control systems, and complex network.

Keke Huang (Member, IEEE) received the B.A. degree in automatic control from Northeastern University, Shenyang, China, in 2012, and the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China, in 2017. He is currently an Associate Professor with Central South University, Changsha, China. His research interests include complex networks, evolutionary game theory, and big data.

Marko Jusup received the B.S. degree in physics and mathematics from the University of Rijeka, Rijeka, Croatia, in November 2003, the MBA degree in finance and banking from the Zagreb School of Economics and Management, Zagreb, Croatia, in July 2008, and the Ph.D. degree in environmental risk management from Yokohama National University, Yokohama, Japan, in March 2012. He. He is currently an Assistant Professor with the Institute for Innovative Research, Tokyo Institute of Technology, Tokyo, Japan. Previously, he held a similar position for two years with the Center of Mathematics for Social Creativity, Hokkaido University. He was also a Japan Society for the Promotion of Science (JSPS) Postdoctoral Fellow with the Faculty of Science, Kyushu University. His research career started in Febuary 2004 with Rudjer Boskovic Institute (RBI), the largest Croatian national research institution. Over time, he contributed to multiple international scientific projects, including large Pan-European collaborations within the Framework Programmes for Research and Technological Development. He actively supports the creation of Free Content (e.g., Open Access, Open-source Software, etc.) and other projects improving human liberties.

Zhen Wang received the Ph.D. degree from Hong Kong Baptist University, Hong Kong, in 2014. From 2014 to 2016, he was a JSPS Senior Researcher with the Interdisciplinary Graduate School of Engineering Sciences, Kyushu University, Fukuoka, Japan. Since 2017, he has been a Full Professor with Northwestern Polytechnical University, Xi'an, China. His current research interests include complex networks, complex system, big data, evolutionary game theory, behavioral economics, and brain science. Thus far, he has authored or cauthored more than 100 scientific papers, his total citation number is around 7,300 and H-index is 42. He was the recipient of the 1000 National Talent Plan Program of China.

Xuelong Li (Fellow, IEEE) is currently a Full Professor with the Center for Optical Imagery Analysis and Learning, Northwestern Polytechnical University, Xi'an, China.