



A two-stage algorithm for network reconstruction

Jianshe Wu^{a,*}, Hedi Yang^a, Yanhua Ren^a, X. Rong Li^b

^a The Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, 710071 Xi'an, China

^b The Department of Electronic Engineering, University of New Orleans, LA 70148, USA

ARTICLE INFO

Article history:

Received 21 June 2017

Received in revised form 24 April 2018

Accepted 5 June 2018

Available online 19 June 2018

Keywords:

Hybrid algorithm

Game theory

Knowledge discovery

Network reconstruction

ABSTRACT

The topology of a network is crucial to its function and behavior. In many cases, various data are obtained from the network, for example, information spreading data, gene expression microarray data, game data, but the topology of the network is unknown. Reconstructing the topology of the network from the observed data is meaningful in many applications. In this paper an evolutionary algorithm is proposed for network reconstruction from observed game data. The proposed two-stage evolutionary algorithm decomposes the network reconstruction problem as sequentially reconstructing the edges of the nodes. The edges of a node are described by the corresponding column vector of the network adjacency matrix. In the first stage, possible vectors are obtained from the proposed genetic algorithm. In the second stage, the true vector is obtained by the proposed heuristic local search. Both analyses and experiments show that the proposed evolutionary algorithm is more accurate and applicable in more general cases than the algorithm based on compressive sensing theory.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Knowing the topology of a network is fundamental for understanding its function and behavior [1]. Usually, various phenomena on a network are observed and data are obtained, but the topology is unknown. Network reconstruction aims at inferring the topology from observed data, for instance, inferring the topology of cellular gene, protein, and metabolite networks from gene expression data [2,3], inferring the topology of gene-regulatory networks from knock-out data [4,5], and inferring the topology of networks from dynamics [6,7], noise [8,9], or game data [10].

The data are usually records of measurements or observations about the actions or events being occurred in the network in many time points. Network reconstruction usually involves solving determined or underdetermined system of first-order equations (see Section 2) [11–13]. With different data, the parameters of the equations may be different, but their mechanisms are similar. Among various data on networks, a representative is the game data [10]. The phenomena of game on networks are common in many fields, ranging from biology and behavioral science to economics [14]. If a node represents an agent (or a player), an edge (or a link) rep-

resents the connections between two nodes interacting with each other by game rules (e.g., the prisoner's dilemma game), the payoffs and the strategies of the nodes are the game data. If an algorithm for network reconstruction is developed with game data, it can be extended or modified to be applicable in various data in different research fields, e.g., system biology and social science. Furthermore, game data can be easily generated according to game theory, and thus they can act as benchmark data for network reconstruction.

Assuming that the network is sparse, many heuristic methods for network reconstruction using different data have been proposed in the fields of system biology and complex networks in recent years [10–12,15]. They include, for example, singular value decomposition (SVD) based methods [11,16] and time-varying dynamic Bayesian network (TV-DBN) method for gene expression data [12], and compressive sensing (CS) based methods for game data [10,15].

The CS theory has been developed in the fields of computer science and signal processing in recent years [17–19], which provides an efficient way in sensing and reconstructing sparse signals in a compressed form. Assuming the edges of networks are sparse, the CS method for reconstructing networks from game data was proposed in [10]. Indeed, many real world networks are sparse, but more networks are not sparse—they are even dense [20]. A heuristic stochastic searching technique, e.g., an evolutionary algorithm (EA), may be more appropriate for reconstructing the topology of a network (see the analysis in Section 3). Population-based EAs mimic the evolutionary phenomena in nature, which are stochastic

* Corresponding author.

E-mail addresses: jshwu@mail.xidian.edu.cn (J. Wu), 591916249@qq.com (H. Yang), renyanhua13@163.com (Y. Ren), xli@uno.edu (X.R. Li).

global optimization methods to complicated problems, especially for those without a heuristic method or satisfactory results. EAs have been used in a wide area of practical problems, e.g., multi-objective optimization [21–24], combinatorial optimization [25], numerical optimization [26], flowshop scheduling problem [27], graph coloring [28], resource allocation [29], and other applications [30–32]. For more comprehensive discussions about various applications of EAs, refer to [33].

Based on the fuzzy cognitive map, EAs have been designed for reconstructing gene regulatory networks [34,35]. This paper attempts to tackle the general network reconstruction problems with an EA. Without assuming that the network is sparse, a two-stage algorithm (TSA) for network reconstruction from observed game data is proposed. The framework of the TSA is a genetic algorithm (GA, first stage) followed by a heuristic local search (LS, second stage). Possible vectors which satisfy one or more rounds of game data are obtained by the GA in the first stage and then the degree of each node is estimated. The proposed LS reconstructs the edges of nodes sequentially in an increasing order of the degree in the second stage.

Hybrid evolutionary algorithms (HEAs) emphasize hybridization of EAs with other methods using problem-specific knowledge, e.g., machine learning techniques and heuristics [36–39]. The HEAs are also named memetic algorithms in some literatures [36,37]. HEAs usually have an iterative framework “EA+LS”: the same process of “EA+LS” is repeated many times [40–42]. The framework of the proposed TSA is not the same as that of the HEA. The GA of our TSA is separated from the LS, though the GA (LS) itself needs to be repeated many times.

Usually in an EA, an individual of the population is a candidate of the solution for the problem. For the network reconstruction problem, a candidate solution is an adjacency matrix which describes the topology of the network. Following this direction, the search space is $N \times N$ dimensions, where N is the number of nodes in the network. Instead of reconstructing the topology of the network as a whole, the proposed TSA reconstructs the edges of nodes one by one, which reduces the search space to N dimensions. Specifically, possible vectors of each node are obtained one by one in the GA stage; in the LS stage, the edges of the node with the smallest degree will be reconstructed first, and then the node with the smallest un-reconstructed degree among the rest will be handled. This decomposition dramatically reduces the complexity of the problem (see Sections 5.2 and 5.4). The edges of a node are described by the corresponding column vector of the adjacency matrix, so the proposed crossover and LS operations (of the TSA) act on the columns of the matrix instead of the entire matrix directly.

The problem-specific knowledge may differ with different types of data. The proposed TSA deals with the prisoner’s dilemma game (PDG) data on the network. Based on the game equation of the PDG (see Section 2), the LS and crossover operators of the proposed TSA are defined (see Sections 4 and 5). From the crossover operation, one or more possible vectors which satisfy one or more game equations can be obtained, then from the LS in the corresponding solution sub-space, the true vectors which describe the edges of the nodes are obtained.

This paper has three major contributions: (a) a method for reconstructing general (non-sparse) networks; (b) analyzing the problem-specific knowledge; (c) a two-stage framework for the problem of network reconstruction (GA followed by LS).

This paper is organized as follows. Section 2 formulates the problem. Section 3 analyzes the problem-specific knowledge (solution space of the problem). Section 4 describes the proposed LS. The details of the proposed TSA are described in Section 5, which includes details of the GA. Simulation results are shown in Section 6. Section 7 concludes the paper.

2. Problem formulation

In the PDG, each of the two players has two options: Cooperate (C) and Defect (D) [14,43]. If both players cooperate, both obtain R points of payoff; if both players defect, both receive P points; if one cooperates and the other defects, the cooperator receives the lowest payoff S and the defector obtains the highest payoff T . The payoffs usually satisfy $T > R > P > S$, which guarantees that the Nash equilibrium of the game is mutual defection, and $2R > T + S$ makes mutual cooperation the globally best outcome [43,44]. The payoff matrix of the PDG is shown as follows [14,43].

	C	D
C	(R, R)	(S, T)
D	(T, S)	(P, P)

For the convenience of representation, the following matrix P_M is used, which is also called payoff matrix in cases without confusion:

$$P_M = \begin{pmatrix} R & S \\ T & P \end{pmatrix}, \tag{1}$$

where subscript “M” stands for matrix. Let $S_i(C) = (1, 0)^T$ and $S_i(D) = (0, 1)^T$ represent that node i is a cooperator and defector respectively. If nodes i and j are connected, indicating that interaction exists between them, then the payoff node i obtained by gaming with node j is

$$f_{ij} = S_i^T P_M S_j. \tag{2}$$

$F = [f_{ij}] = [f_1^T, \dots, f_i^T, \dots, f_N^T]^T$ is the network payoff matrix, where $f_i = (f_{i1}, \dots, f_{iN})$ is the i th row vector of matrix F . Obviously, row vector f_i describes the payoffs that node i may obtain by gaming with the remaining nodes. We assume $f_{ii} = 0$, indicating that node i does not interact with itself. The total payoff node i obtained by gaming with its neighbors is

$$g_i = \sum_{j \in \Gamma_i} f_{ij}, \tag{3}$$

where Γ_i is the set of nodes which are connected with node i . Denote the 0–1 adjacency matrix of the network as $A = [a_{ij}] = [a_1^T, \dots, a_i^T, \dots, a_N^T]$, where a_i^T is the i th column vector of A . In A , $a_{ij} = 1$ indicates that nodes i and j are connected; otherwise, $a_{ij} = 0$. Obviously, the topology of the network is described by A , then from (3),

$$g_i = f_i \cdot a_i^T, \quad i = 1, \dots, N. \tag{4}$$

The payoff of node i (g_i in Eq. (4)) is available from game data and f_i can be calculated from (2). If $a_i^T, i = 1, \dots, N$, can be obtained from (4), the adjacency matrix A is obtained and the network is reconstructed.

Usually, given one round game data, $g_i \in \mathfrak{R}$ and $f_i \in \mathfrak{R}^N, a_i^T$ has many solutions. Matrix A cannot be uniquely determined. If K rounds game data are available, the following K equations can be obtained:

$$\begin{aligned} g_i(1) &= f_i \cdot (1)a_i^T, \\ g_i(2) &= f_i \cdot (2)a_i^T, \\ &\dots \dots \\ g_i(K) &= f_i \cdot (K)a_i^T, \\ i &= 1, \dots, N, \end{aligned} \tag{5}$$

Download English Version:

<https://daneshyari.com/en/article/6903428>

Download Persian Version:

<https://daneshyari.com/article/6903428>

[Daneshyari.com](https://daneshyari.com)