An Improved Game-theoretic Approach to Uncover Overlapping Communities

Hong-liang Sun*, Eugene Ch'ng†

NVIDIA Joint-Lab on Mixed Reality,
International Doctoral Innovation Centre,
University of Nottingham,
Ningbo, 315100, P. R. China
**zx17898@nottingham.edu.cn*
†*eugene.chng@nottingham.edu.cn*

Xi Yong

Water Information Centre, Ministry of Water Resources,
Beijing, 100053, P. R. China
yongxi@mwr.gov.cn

Jonathan M. Garibaldi

School of Computer Science, University of Nottingham,
Nottingham, NG8 1BB, UK
jon.garibaldi@nottingham.ac.uk

Simon See

NVIDIA AI Technology Centre,
NVIDIA, Singapore 138522
Center for High Performance Computing,
Shanghai Jiao Tong University
Shanghai 200240, P. R. China
ssee@nvidia.com

Duan-bing Chen‡

Web Sciences Center, Big Data Research Center,
University of Electronic Science and Technology of China,
Chengdu, 611731, P. R. China

‡*dbchen@uestc.edu.cn*

Received 16 April 2017
Accepted 21 August 2017

How can we uncover overlapping communities from complex networks to understand the inherent structures and functions? Chen et al. firstly proposed a community game (Game) to study this problem, and the overlapping communities have been discovered when the game is convergent. It is based on the assumption that each vertex of the underlying network is a rational game player to maximize its utility. In this paper, we investigate how similar vertices affect the formation of community game. The Adamic-Adar Index (AA Index) has been employed to define the new utility function. This novel method has been

evaluated on both synthetic and real-world networks. Experimental study shows that it has significant improvement of accuracy (from 4.8 percent to 37.6 percent) compared with Game on 10 real networks. It is more efficient on Facebook networks and Amazon co-purchasing networks than on other networks. This result implicates that "friend circles of friends" of Facebook are valuable to understand overlapping community division.

Keywords: Overlapping Community Detection; Game Theory; Complex Networks

PACS Nos.: 89.75.Fb, 89.20.Ff, 89.75.Hc

1. Introduction

Networks are natural representations of real world complex systems. During the past few decades, the study of complex networks has attracted extensive researchers from physics, computer science, social sciences and other disciplines. Thinking from the perspective of network science can lead to better understand about function and dynamic processes of underlying complex systems. It contributes to gain deep insight into networks including economic networks, biological networks, scientific collaboration networks [1], software execution dependency networks [2], Twitter social networks [3,4] and etc.

A key property of network is community [5,6], where there are dense connections within communities but sparse connections between them. The precise definition of community is still not well understood [7]. In the past few decades, researchers have proposed extensive assumptions to uncover the mechanism of community formation [8]. The automatic discovery of community can reveal coarse-gained structures of networks, which are too large for humans to understand at the level of individual vertices.

Researchers have further discovered that members do not necessarily belong to disjoint communities. Instead, they have multiple chances to select which communities they can belong to. Palla et al. firstly proposed the Clique Percolation Method (CPM , implemented with the name CFinder) to study the overlapping community in 2005 [9]. It is based on the idea that internal edges of a community are likely to form cliques due to their high density. On the other hand, this method requires the parameter k to obtain k-clique (i.e. a complete graph with k vertices). In practice, CFinder is sensitive to the selection of parameter k to identify k-cliques. Chen et al. designed a community game [10] to discover overlapping communities via complex networks. Each vertex of the network is assumed as a rational player to maximize its utility. The game continues until no one wants to change its strategies with respect to choices of other players. The decisions of all players naturally lead to the division of communities via the underlying network.

Many algorithms have been proposed based on the soft clustering of vertices. However, it has been pointed out that overlapping communities in real social networks reflect different association types between people [11,12]. It is not necessarily to discover community merely from methods of vertex clustering. Link clustering methods shed light on new roads with respect to edge clustering rahter than vertex

clustering. Line graphs are proposed to solve the problem of overlapping communities [11]. Link density clustering (LDC) extends the idea of edge clustering based on density peaks [13]. But from the recent study of Fortunato et al. [14], there is no clear evidence to support that edge clustering methods are better than vertex clustering methods and vice versa. The answer to this question depends on real-world networks under investigation.

There are other assumptions such as modularity optimization [15,16], distance dynamics [17], community core expansion [18], local iterative expansion [19,20,21,22], label propagation [23,24], non-negative matrix factorization [25,26], a variant of non-negative matrix factorization using neighbour node degree [27], node similarity [28] and etc. Directed and weighted networks are also studied based on methods related to undirected and unweighted networks [29]. Due to the limitation of space, we limit the scope without more novel and effective methods. Fortunato et al. have summarized excellent and comprehensive survey of overlapping community detection [8,14] for interested researchers.

In this paper, we study the overlapping community detection from a game theoretic way[30,31]. Most social networks represent complex systems of human interactions and behaviours. Game theoretic approaches enable us to understand how players from social networks behave with respect to activities of other members. Chen et al. have firstly studied this problem and designed a community game where each vertex is considered as a rational player [10]. Maximization of the individual pay-off leads to the Nash equilibrium in the end. Communities are naturally divided from each player's final choice.

We rethink the community game and extend it from the interactive model of nodes. Previous method mainly focuses on the vertices which are connected with each other. It is discovered that disconnected yet similar nodes also affect the community formation. Contributions are listed as following:

Intuitive Interaction: A community game is formulated from the interactive ways including both connected and disconnected yet similar vertices. A novel gain function has been developed to quantify the contributions of similar nodes using AA Index [32] in the community game.

Parameter Free: From the new point of view on vertex interaction, a new model GExplorer has been proposed to play the community game. It is a parameter free method to support users in practice.

High Performance: From the study on synthetic networks, it is known that GExplorer outperforms Game and CFinder in various conditions. Experimental study on real networks demonstrates that this new method has an improvement of accuracy from 4.8 percent to 37.6 percent compared with Game. It has a time complexity $O(m^2)$, which is effective to deal with networks with thousands vertices and tens of thousands edges in a few minutes.

The remainder of this paper is presented as following: at the very beginning, a brief introduction is given to interpret related concepts and definitions in section 2.1. A comprehensive study of community game is proposed in section 2.2, 2.3,

and 2.4. Experimental studies on both synthetic and real networks are discussed in section 3. Finally main contributions are concluded in details in section 4.

2. Materials and Methods

In this section, a novel method will be presented to discover overlapping community in complex networks. The philosophy behind this method is that we design an improved community game with respect to connected nodes, disconnected yet similar nodes. It starts with some preliminary definitions in section 2.1. Then the existing game-theoretic method is discussed in section 2.2. Furthermore, a new method GExplorer is proposed to gain deep insight in section 2.3. Finally, the time complexity and efficiency will be discussed in section 2.4.

2.1. *Preliminaries*

Aiming to discuss the community game, some preliminary definitions are introduced first.

DEFINITION 1 (NEIGHBOURS OF VERTEX u) Given a graph G, the neighbour set of a vertex u contains its adjacent vertices.

$$N(u) = \{u \in V, v \in V \mid (u, v) \in E\} \tag{1}$$

DEFINITION 2 (COMMON NEIGHBOURS OF VERTEX u, v) Given a graph G, the common neighbours of vertex u and v $CN(u, v)$ is the intersection set of $N(u)$ and $N(v)$.

$$CN(u, v) = N(u) \cap N(v) \tag{2}$$

Based on previous definitions, we further introduce the vertex similarity of graph G. A pair of vertices are similar if they share common neighbours. Adamic-Adar Index (AA Index) [32] is employed because it is a local similarity measure, time efficient and accurate. In addition, other measures have also been well studied by Liu et al. for the purpose of further discussions and comparison [33].

DEFINITION 3 (AA INDEX) Given a graph G, how to measure the vertex similarity based on network topology? AA Index [32] is defined as following, where $d(i)$ is the degree of vertex i.

$$AA(u, v) = \sum_{i \epsilon CN(u,v)} \frac{1}{ln(d(i))} \tag{3}$$

DEFINITION 4 (OVERLAPPING COMMUNITY DETECTION) Given a graph $G = (V, E)$, we aim to divide the network into overlapped communities where there are dense connections within the communities and sparse connections between them. Let $C_1, C_2, ..., C_k$ denote such k communities, there are at least i and j, where $C_i \cap C_j \neq \varnothing$, $1 \leqslant i \leqslant k$ and $1 \leqslant j \leqslant k$.

Table 1: Symbol table

Symbol	Descriptions
$G(V, E)$	Graph G with vertex set V and edge set E, where $\|V\| = n$ and $\|E\| = m$.
S_u	The community set vertex u belongs to, namely the strategy space of vertex u.
S_{-u}	Strategies of other $n-1$ vertices except vertex u.
Join (u, c)	$S_u = S_u \cap \{c\}$
Leave (u, c)	$S_u = S_u - \{c\}$
Switch (u, c, c')	$S_u \leftarrow (S_u - \{c\}) \cup \{c'\}$

2.2. *Community Game*

Game theory comes from the study of human behaviour and interaction in social sciences and economics. On one hand, the human behaviour is cooperative, such as students work together on a team project within the campus. On the other hand, it can also be competitive. For example, more than two companies compete for the market share of the same products. Due to the mutual shaping of cooperation and competition of human behaviour, game theory provides insightful ways to study such interplay between cooperation and competition.

A traditional game necessarily consists of following elements [34]:

- A list of players.
- Complete descriptions about how players behave.
- How much players know about other players' behaviour?
- How players' actions lead to final outcomes?
- A specification of the players' preferences on such outcome.

Chen et al. firstly introduced the game-theoretic approach to the study of overlapping community detection[10]. Symbols used next are described in Table 1. In the community game, each vertex behaves as a rational player. It can join, leave, switch communities or take no actions depending on the utility gain. All communities a player can belong to consist of its strategy space. When a player joins the game, it knows the strategy spaces of its neighbours. The game continues until no one wants to change its strategy space, the game is convergent. Strategy spaces of all players can naturally contribute to the community division of the underlying network. A community game is shown in Algorithm 1.

Utility determines how players select different strategies. It can be calculated from gain function and loss function.

DEFINITION 5 (UTILITY) Given a graph $G(V, E)$, the utility of vertex $u \in V$ in the community game, $ut_u(S_{-u}, S_u)$, strikes a balance between gain function g_u and loss function l_u.

$$ut_u(S_{-u}, S_u) = g_u(S_{-u}, S_u) - l_u(S_{-u}, S_u) \tag{4}$$

DEFINITION 6 (GAIN FUNCTION) Gain function of vertex u $g_u(S_{-u}, S_u)$ in

Algorithm 1 Community Game

Input:
Given an undirected and unweighted Network $G(V, E)$.
Output:
k communities $C_1, C_2, ..., C_k$.
procedure GAME($G(V, E)$)
  Initialize $|V|$ players from G.
  while Not convergence from all players' utilities **do**
    Randomly select a player u from $|V|$ players.
    Select the best operations from *Join*, *Leave*, *Switch* and *No Operation*.
    ▷ The best operation is determined by the maximisation of utility gain.
    Update the strategy space S_u and S_{-u}
  end while
  Generate the communities $C_1, C_2, ..., C_k$ from $|V|$ players' strategy spaces.
  return k communities $C_1, C_2, ..., C_k$
end procedure

the community game is defined as below. It comes from direct impact $DI(u, v)$ of adjacent neighbors of u.

$$g_u(S_{-u}, S_u) = \frac{1}{2m} \sum_{u \epsilon V} \sum_{v \epsilon V, v \neq u} DI(u, v) \tag{5}$$

The gain function of player u comes from the interaction patterns of vertex u and its neighbours. From the previous work of Chen et al.[10], they considered the direct interaction of adjacent vertices.

DEFINITION 7 (DIRECT IMPACT) Direct interaction between vertex u and its adjacent neighbour v is defined as Eq. 6.

$$DI(u, v) = \delta(u, v) - \frac{d(u)d(v)}{2m} \tag{6}$$

$d(u)$ is the degree of u in graph G, m is the number of edges. If vertex u and v are in the same community $\delta(u, v) = 1$, otherwise $\delta(u, v) = 0$. It comes from the idea of personalized modularity given by Chen et al.[10]. Such personalized modularity of each vertex u sums up to the modularity score given by Newman et al.[15].

DEFINITION 8 (LOSS FUNCTION) Loss function of vertex u is $l_u(S_{-u}, S_u)$. It simply takes some cost from each community vertex u joins. Initially, each vertex u belongs to its singleton community and no cost is taken. It is directly proportional to $|S_u| - 1$ [10].

$$l_u(S_{-u}, S_u) = \frac{|S_u| - 1}{2m} \tag{7}$$

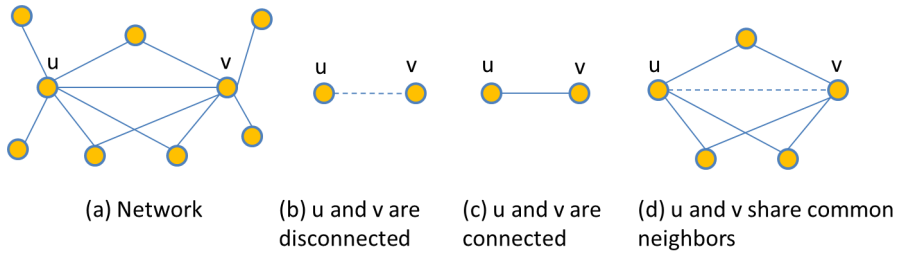

Fig. 1: Interaction patterns of vertices and their neighbours.

A community game is shown in Algorithm 1. At the very beginning, each vertex is considered as a rational player to maximize its utility. Each one belongs to a singleton community. That means every player's strategy space initials with itself. The game continues with random selections of players until no one wants to change its current strategy space. At each round of the game, selected player makes a decision considering its neighbours' strategy spaces. It can join, leave, switch or take no actions depending on the utility gain calculated from Eq. 4. When the game stops, all players' strategy spaces can be converted to the discovered communities.

If the game is repeated with the same networks, it is not necessarily to obtain the same results. This problem can be caused by two factors. One is the random selection of players, which causes the game to be played in different ways. The other one is that players make decisions from strategy spaces of their neighbours rather than from all others. Because the latter case is a NP hard problem and infeasible in practice, local decision making from neighbours is selected. We also consider which local optimum should be chosen as the final result. In real practice, the game is repeatedly played r times and the best one is selected from different measures. Evaluation of the game will be further discussed in details in section 3.

2.3. *Improved Community Game GExplorer*

Game only considers the direct interaction of players in Eq. 6. It is further discovered that indirect interaction from similar neighbours also can contribute to the utility gain. From Fig. 1 it is shown that connected nodes, disconnected yet similar nodes sharing common neighbors. Liu et al. have investigated the stability of several node similarity measures from bipartite networks [33]. Lü et al. have studied link prediction problem via local similarity information [35,36]. In this paper, AA Index [32] is employed to calculate the node similarity because its stability and accuracy. We have proposed a novel indirect interaction using AA index in Eq. 8. Then the improved gain function is given in Eq. 9.

DEFINITION 9 (INDIRECT IMPACT) Indirect interaction between vertex u and its similar vertex v are defined as Eq. 8.

$$SI(u,v) = \frac{AA(u,v)}{2m} \tag{8}$$

$AA(u,v)$ is the AA Index discussed in section 2.1 and m is the number of edges of graph G.

DEFINITION 10 (IMPROVED GAIN FUNCTION) Gain function of vertex u $g_u(S_{-u}, S_u)$ in the community game GExplorer is defined as below. It comes from direct impact $DI(u,v)$ of adjacent neighbors of u and indirect impact $SI(u,v)$ of similar neighbors of u.

$$g_u(S_{-u,u}) = \frac{1}{2m} \sum_{u \epsilon V} \sum_{v \epsilon V, v \neq u} DI(u,v) + SI(u,v) \tag{9}$$

2.4. *Time Complexity*

There are two options for players of community game to make decisions, from all other players or from their neighbours. The former one requires the strategy spaces of all players. It is a NP-hard problem but can guarantee the global optimum of community division. In real practice, the latter one can ensure the community game to be convergent in polynomial time complexity to reach local optimum. As it has been proofed by Chen et al., it takes at most $O(m^2)$ steps to reach Nash equilibrium [10]. Usually there are multiple local optimums due to random selections, thus it runs r times to select the best one in practice. In addition, interaction between vertices sharing common neighbours does not require more steps, but at each step it consumes more time. Thus the time complexity of GExplorer is still $O(m^2)$.

3. Experimental Study

In this section, we evaluate GExpolorer on both synthetic networks and real networks.

Selected comparison methods. To evaluate the performance of GExplorer, we select two overlapping community discovery methods: Palla et al. firstly proposed the Clique Percolation Method[9] (CPM and it was also implemented as CFinder [a] in 2005. Another one was the game-theoretic approach proposed by Chen et al.[10]. We select several clique size using CFinder and the best one is chosen as the final result. Due to the random selection of game player, both Game and GExplorer execute r times on each data set and the best one has been selected.

There are two ways to evaluate the performance of overlapping community detection methods. One measure comes from the synthetic networks generated by LFR benchmark[37]. We can compare ground truth with communities detected by existing algorithms. One benefit of the method is that we can vary the parameters of the

[a] www.cfinder.org

LFR to change the network topological structures. One drawback is that synthetic networks do not always reflect the real-world networks. The other one depends on real networks. 10 real networks are selected to study the performance of GExplorer.

Experiments on synthetic networks are carried out in section 3.1. Further discussions on real networks are described in section 3.2.

Evaluation measures. To study and compare different overlapping community methods, there are two ways. If the overlapping community division over underlying networks are given in advance, Normalized Mutual Information (NMI)[38] is applied to give a score ranging from 0 to 1. Otherwise, modularity[15] is employed. The original modularity is defined only for non-overlapping community discovery. But Nicosia et al. have extended the definition for overlapping community methods. We use this overlap modularity Q_{ov}[23,39] for the experimental study in this paper. A higher value means more intra-community edges than those expected by random selections.

Node similarity measures. It is for the purpose to study which node similarity better fits the utility function, we study 10 node similarity measures on 10 real networks shown in **Appendix**. It is concluded that AA Index is better in most cases.

All the experiments have been carried out on a PC with Intel i7 3.4 GHz CPU and 16 GB memory.

3.1. *Synthetic Data Sets*

This section describes experiments on artificial networks. These benchmark networks come from a network generator LFR benchmark by Lancichinetti et al.[37].

We set various parameters to determine network topological features: the number of network vertices N, the community size C, the average degree k, the mixing parameter μ, the number of overlapping vertices on and the number of memberships of overlapping vertices om. By default, $k = 20$, $maxk = 50$, and $om = 2$. N, C, μ and on vary according to different datasets in our experiment.

We also compare the performance of GExplorer with CFinder[9] and Game[10].

From Fig. 2, those sub figures correspond to four rows: small networks with small communities (SS, $N = 1000, C \in [10, 50]$), small networks with large communities (SL, $N = 1000, C \in [20, 100]$), large networks with small communities (LS, $N = 5000, C \in [10, 50]$) and large networks with large communities (LL, $N = 5000, C \in [20, 100]$). We aim to measure how NMI[38] scores change with respect to mixing parameter μ and the fraction of overlapping vertices x.

Small networks with small communities (SS). It is shown in Fig. 2 (a-c) that if mixing parameter is low ($\mu = 0.1$ and $\mu = 0.3$), our method performs nearly the same as Game, but both of them are better than CFinder. On one hand, they all have high NMI scores which are more than 0.9. On the other hand, they remain stable when fraction of overlapping nodes x increases from 0 to 0.8. If the mixing parameter μ is high ($\mu = 0.5$), our method is still better than others. But the NMI

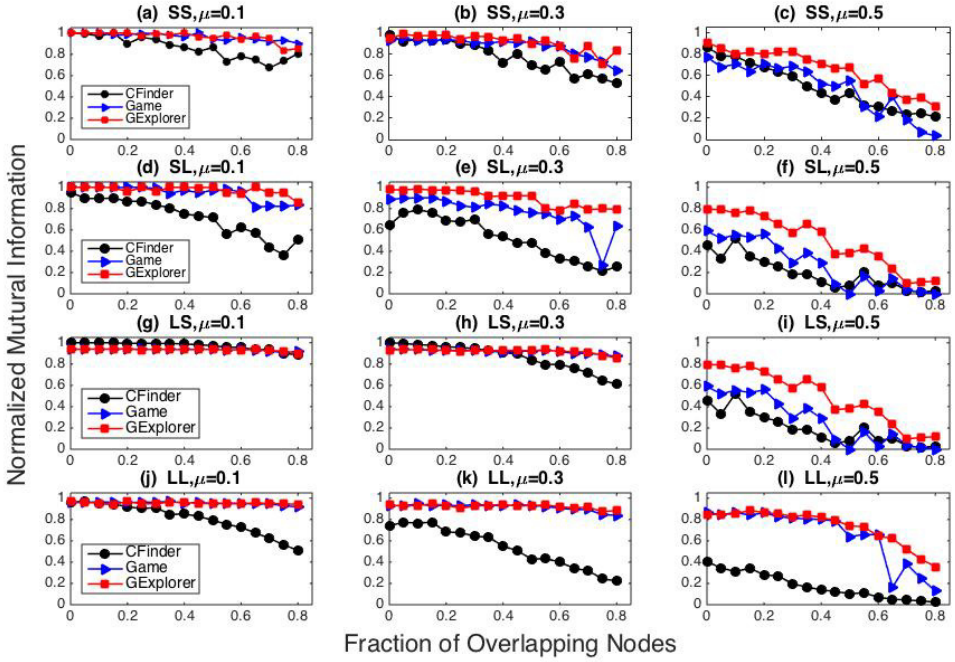

Fig. 2: The synthetic network size N is either 1000 or 5000, community size C is in the range [10, 50] or [20, 100], mixing parameter $\mu \in (0.1, 0.3, 0.5)$. (1) The first row represents small networks with small communities (SS, $N = 1000, C \in [10, 50]$). Such three figures (a), (b) and (c) have different mixing parameter $\mu = 0.1$, $\mu = 0.3$ and $\mu = 0.5$. (2) The second row contains small networks with large communities (SL, $N = 1000, C \in [20, 100]$).Such three figures (d), (e) and (f) have different mixing parameter $\mu = 0.1$, $\mu = 0.3$ and $\mu = 0.5$. (3) The third row includes large networks with small communities (LS, $N = 5000, C \in [10, 50]$). Such three figures (g), (h) and (i) have different mixing parameter $\mu = 0.1$, $\mu = 0.3$ and $\mu = 0.5$. (4) The last row contains large networks with large communities (LL, $N = 5000, C \in [20, 100]$). Such three figures (j), (k) and (l) have different mixing parameter $\mu = 0.1$, $\mu = 0.3$ and $\mu = 0.5$.

decreases when the fraction of overlapping nodes x increases.

Small networks with large communities (SL). It is shown in Fig. 2 (d-f) that when mixing parameter is low ($\mu = 0.1$), our method performs nearly the same as Game, but both of them are better than CFinder. When the mixing parameter μ is high ($\mu = 0.3, \mu = 0.5$), our method is still better than others.

Large networks with small communities (LS). It is shown in Fig. 2 (g-i) that if mixing parameter is low ($\mu = 0.1, \mu = 0.3$), our method performs slightly better than Game. But both of them outperform CFinder. If the mixing parameter

is high ($\mu = 0.5$), GExplorer is more efficient.

Large networks with large communities (LL). It is shown in Fig. 2 (j-l) that when mixing parameter is low ($\mu = 0.1, \mu = 0.3$), our method performs nearly the same as Game, but both of them are better than CFinder. When the mixing parameter is high($\mu = 0.5$), GExplorer performs nearly the same if the fraction of overlapping nodes x is less than 0.6. It outperforms others when x is larger than 0.6.

It concludes that if the mixing parameter is low ($\mu = 0.1, \mu = 0.3$), then GExplorer is slightly better than Game. GExplorer is more efficient if the mixing parameter is high ($\mu = 0.5$). Mixing parameter μ indicates the ratio of external degree to total degree. In the case that mixing parameter is high, indirect relations are more efficient because there are more chances for external members to interact with internal members.

One pitfall of our method is that it does not strictly decrease when the fraction of overlapping nodes increases. It was found that the random selection of vertices can lead to local optimum rather than global optimum. However, the fluctuation is small and can be accepted.

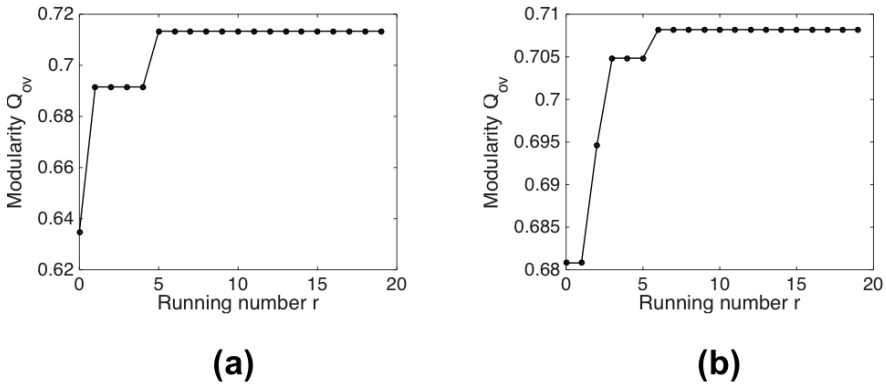

Fig. 3: The running number r of the programme influences the local maximum modularity Q_{ov} in GExplorer. (a) In ZKC, maximisation of Q_{ov} is obtained when $r = 6$. (b) In dolphin network, local optimum Q_{ov} score is reached when $r = 7$.

Table 2: Performance of CFinder, Game and GExplorer on real networks

Network	$\|V\|$	$\|E\|$	*CFinder*		*Game*		*GExplorer*			
			(Q_{ov})	(C)	(Q_{ov})	(C)	(Q_{ov})	(C)	(ΔQ_{ov}^{G})	(ΔQ_{ov}^{CF})
ZKC	34	78	0.515	3	0.594	5	**0.713**	4	0.119	0.198
$Dolphin$	62	159	0.662	4	0.617	11	**0.708**	8	0.091	0.046
PB	105	441	0.786	4	0.549	12	**0.796**	7	0.247	0.01
CF	115	613	0.641	13	0.625	12	**0.673**	11	0.048	0.032
$FN414$	150	1698	0.869	3	0.826	8	**0.887**	4	0.061	0.018
$FN686$	168	1656	0.551	3	0.400	13	**0.616**	5	0.216	0.065
$FN348$	224	3192	0.587	3	0.336	18	**0.625**	14	0.289	0.038
$FN0$	333	2519	0.707	13	0.403	36	**0.779**	23	0.376	0.072
$FN107$	1034	26749	*None*	*None*	0.508	40	**0.795**	25	0.287	*None*
AG	5242	14496	0.581	835	0.526	1323	**0.657**	1219	0.131	0.076

3.2. *Real-world Data Sets*

In this section, we will study the performance of GExplorer on real networks. All real networks are public available from the UCI network data repository[b] and SNAP[c] network data sets.

Zachary's Karate Club (ZKC). Zachary investigated the members of a karate club and studied the friendships of such members. Given a karate club network with 34 vertices and 78 edges, we employ GExplorer to uncover the overlapping communities. Due to the impact of random selection of players, GExplorer has been repeated for r times to obtain the local optimum result. It is shown in Fig. 3 (a) that the local optimum is reached when $r = 6$. As a result, 4 communities are discovered, which are shown in Table 2. GExploer (Q_{ov}=0.713) is 11.9 percent better than the second best method Game (Q_{ov}=0.594). It also outperforms CFinder (Q_{ov}=0.515) nearly 20 percent , where clique size $k \in [3, 5]$ and $k = 4$ is the best choice.

Dolphin. D. Lusseau, et al. contributed a dolphin social network of frequent associations between 62 dolphins in a community living off Doubtful Sound, New Zealand [40]. GExplorer has been repeated for r times to obtain the local optimum result shown in Fig. 3 (b). It is shown from Table 2 that GExplorer discovers 8 overlapping communities (Q_{ov}=0.708). It outperforms Game (Q_{ov}=0.617, $C = 11$) nearly 9 percent and is 4.6 percent better than CFinder (Q_{ov}=0.662, $C = 4$). CFinder has clique size $k \in [3, 5]$ and $k = 4$ is selected with highest Q_{ov} score.

Political Books (PB). A network of books about US politics published around 2004 presidential election and sold by the online book seller Amazon. Edges between nodes represent frequent co-purchasing of books by the same buyers. From Table 2, GExplorer (Q_{ov}=0.796, $C = 7$) is slightly better than CFinder (Q_{ov}=0.786, $C = 4$).

[b]https://networkdata. ics.uci.edu/index.php
[c]http://snap.stanford.edu/

But it is 24.7 percent better than Game (Q_{ov}=0.549, $C = 12$). We selected clique size $k = 3$ from $[3, 6]$ in CFinder.

College Football (CF). A network of American football games between Division IA colleges during regular season Fall 2000. From Table 2 it is known that GExplorer (Q_{ov}=0.673, $C = 11$) is 4.8 percent better than Game (Q_{ov}=0.625, $C = 12$) and outperforms CFinder (Q_{ov}=0.641, $C = 13$) 3.2 percent. We set $k = 4$ from [3,9] as the best clique size in CFinder.

Facebook Networks (FN). This dataset consists of 'friend list' from Facebook. They were collected from survey participations using Facebook. Due to the requirement of this experimental study, 5 networks are selected from 10 networks and some small networks are ignored. They are named Facebook Network 414 (FN414), Facebook Network 686 (FN686), Facebook Network 348 (FN348), Facebook Network 0 (FN0) and Facebook Network 107 (FN107) in Table 2. The first four networks consist of hundreds of nodes and thousands of edges. The last one Facebook Net. 107 is challenging because it has thousands of nodes and tens of thousand of edges. Generally speaking, GExplorer is better than CFinder and Game. It has the smallest improvement over Game on FN414, where GExplorer (Q_{ov}=0.887, $C = 4$) is 6.1 percent better than Game (Q_{ov}=0.826, $C = 8$). It also has the largest improvement on FN0, where GExplorer (Q_{ov}=0.779, $C = 23$) is 37.6 percent better than Game (Q_{ov}=0.403, $C = 36$). For the challenging network FN107, GExplorer (Q_{ov}=0.795, $C = 25$) is still 28.7 percent better than Game (Q_{ov}=0.508, $C = 40$). But CFinder does not give out the final result because time exceeds the maximum limitation. It takes more than 4 hours during our experimental study.

ArXiv GR-QC (AG). arXiv GR-QC (General Relativity and Quantum Cosmology) collaboration network is from the e-print arXiv and covers scientific collaborations between authors from papers submitted to General Relativity and Quantum Cosmology category. If an author i co-authored a paper with author j, the graph contains an undirected edge from i to j. If the paper is co-authored by k authors this generates a completely connected sub-graph on k nodes. The data covers papers in the period from January 1993 to April 2003 (124 months). From Table 2 it is known that GExplorer (Q_{ov}=0.657, $C = 1219$) is 13.1 percent better than Game (Q_{ov}=0.526, $C = 1323$) and 7.6 percent better than CFinder (Q_{ov}=0.581, $C = 835$). We selected clique size $k = 3$ from $[3, 44]$ in CFinder. Finally, the runtime of GExplorer is less than 2 minutes. Thus it demonstrates that GExplorer can be applied to large networks with thousands of nodes and tens of thousand edges in practice.

GExplorer is evaluated in authentic small networks and large networks shown in Table 2. The experimental results are listed in descending order with respect to accuracy gain compared with Game. From the number of communities discovered from GExplorer, Game and CFinder, it is known that CFinder aims to uncover large communities while Game always detects small ones. To strike a balance between large communities and small ones, GExplorer can discover reasonable medium

size communities. GExplorer is more efficient than Game because indirect interaction is considered when players join the game. Thus each player has larger view from neighbours extending to include disconnected by similar nodes. Such factors contribute to adjacent smaller communities merge into larger and reasonable ones. Due to the acceptable size of communities, GExplorer is more efficient than both CFinder and Game.

In real practice, there are various different types of networks where topology diversity is one of the key factors. It is discovered from Table 2 that GExplorer is more efficient on Facebook networks and Political Books networks (PB), where it outperforms Game more than 20 percent except one case FN414. One of the reasons for this exception is FN414 has already been well divided by Game ($Q_{ov} = 0.826$) as well as CFinder ($Q_{ov} = 0.887$). Thus it leaves less room for further improvement.

We conclude that indirect interaction has different impacts on distinct social behaviours. Due to the larger view of players on disconnected yet similar nodes, more accurate communities have been discovered by GExplorer. In practice, indirect impact has large impacts on Facebook friend circles and Amazon online co-purchasing behaviours. The result from Facebook networks have implicated that online individuals have been influenced by the "friend circles of friends" greatly with respect to community. In addition, accurate community division method can support various businesses to understand valuable online co-purchasing behaviours in Amazon.

4. Summary and Discussions

In this paper, an improved game-theoretic approach is proposed to discover overlapping communities via complex networks. Based on the previous work by Chen et al.[10], new method GExplorer extends to study how vertices sharing common neighbours influence the community division using AA Index[32]. Such a community game can effectively divide underlying networks into overlapping communities in the time complexity of $O(m^2)$.

We have studied the performance on both synthetic networks and real-world networks. The experimental study shows that GExplorer can find high-quality overlapping communities compared with Game and CFinder on different synthetic networks. It is evaluated on small networks with small communities (SS), small networks with large communities (SL), large networks with small communities (LS) and large networks with large communities (LL). It is discovered that when mixing parameter μ is low ($\mu = 0.1, \mu = 0.3$), GExplorer is slightly better than Game. Furthermore, if μ is high ($\mu = 0.5$), GExplorer is more efficient compared with Game. This result indicates that indirect interaction is more efficient to community discovery if communities are densely mixed with others.

Further study about GExplorer was carried out on 10 real networks ranging from small to large networks. As shown in **Appendix**, AA Index better fits most of the networks and is selected in GExplorer. Then we compare GExplorer with Game and CFinder on 10 real networks. It is obtained that the improvement of accuracy ranges

from 4.8 percent to 37.6 percent compared with Game. GExplorer enables players to have larger views not only on neighbours, but also on disconnected yet similar ones. This factor determines that smaller communities merge into larger and more reasonable ones. Thus we get larger and more accurate results from GExplorer. On the other hand, communities discovered by CFinder are roughly divided and less efficient. It is concluded that GExplorer strikes a balance between large size and small size, a medium and reasonable community size contributes to accurate results. In addition, GExplorer can also be applied to large networks with thousands of nodes and tens of thousands of edges in a few minutes.

It concludes that the indirect interaction influences community division together with direct impact. It might also enlighten further exploration on existing community discovery methods only considering influences from connected neighbour nodes.

We obtain good results on Facebook networks and Amazon co-purchasing networks. These results have implicated that "friend circles of friends" are valuable in online social community division. Meanwhile, they influence online co-purchasing behaviours in Amazon greatly.

In future work, we plan to focus on abstraction and visualisation of large networks based on this game-theoretic model. Community evolution via dynamic networks is another direction which we have interests. Such study can contribute to understand complex networks and communities with a few simple words.

Appendix. A. Comparing node similarity measures

In this section, we aim to study which node similarity index is more suitable to the utility function in the community game GExplorer. Here, local neighbour similarity information is studied by 10 different node similarity measures, including CN (Common Neighbours), HDI (Hub Depressed Index), HPI (Hub Prompted Index), JI (Jaccard Index), LHN (Leicht Holme Newman Index), PA (Preferential Attachment), RA (Resource Allocation Index), SAL (Salton Index), SI (Sϕrensen Index) and AA (AdamicAdar Index) [28,36].

The results shown in Table A.1 demonstrate that utility function using AA Index outperforms others on 7 networks including ZKC, Dolphin, CF, FN414, FN686, FN0 and AG. It concludes that in most cases, AA Index is more suitable to be selected.

Acknowledgments

This work is jointly supported by the International Doctoral Innovation Centre (IDIC) scholarship scheme from the University of Nottingham Ningbo, China (UNNC), the National Natural Science Foundation of China (with Grant Nos. 61433014 and 61673085), and the Fundamental Research for the Central Universities (Grant Nos. ZYGX2014Z002 and ZYGX2015J152).

We also acknowledge the support received from NVIDIA Joint-Lab on Mixed Reality from UNNC, the University of Nottingham United Kingdom (UNUK), Big Data Research Center of University of Electronic Science and Technology of China, Ningbo Education Bureau, Ningbo Science and Technology Bureau and China

Table A1: Comparing 10 different node similarity measures of utility function. Modualrity Q_{ov} scores are obtained on 10 networks by ten similarity measures including CN (Common Neighbours), HDI (Hub Depressed Index), HPI (Hub Prompted Index), JI (Jaccard Index), LHN (LeichtHolmeNewman Index), PA (Preferential Attachment), RA (Resource Allocation Index), SAL (Salton Index), SI (Sϕrensen Index) and AA (Adamic Adar Index).

Measure	*ZKC*	*Dolphin*	*PB*	*CF*	*FN*414	*FN*686	*FN*348	*FN*0	*FN*107	*AG*
CN	0.709	0.702	0.761	0.671	0.884	0.611	0.648	0.765	0.805	0.651
HDI	0.691	0.703	0.808	0.673	0.882	0.595	**0.654**	0.762	**0.839**	0.649
HPI	0.710	0.701	0.799	0.658	0.883	0.603	0.635	0.777	0.813	0.646
JI	0.677	0.691	0.738	0.663	0.884	0.599	0.651	0.773	0.819	0.641
LHN	0.710	0.704	0.806	0.670	0.883	0.610	0.625	0.760	0.772	0.639
PA	0.707	0.697	**0.816**	0.664	0.848	0.481	0.552	0.757	0.771	0.636
RA	0.702	0.694	0.792	0.664	0.880	0.610	0.653	0.770	0.801	0.631
SAL	0.691	0.703	0.796	0.670	0.880	0.590	0.644	0.761	0.830	0.627
SI	0.700	0.696	0.806	0.671	0.879	0.599	0.643	0.773	0.816	0.637
AA	**0.713**	**0.708**	0.796	**0.673**	**0.887**	**0.616**	0.625	**0.779**	0.795	**0.657**

MOST.

References

1. M. E. J. Newman, *Proceedings of the National Academy of Sciences* **98**, 404 (2001).
2. G. Y. Huang, P. Zhang, B. Zhang, T. T. Yin and J. D. Ren, *International Journal of Modern Physics C* **27** (2016).
3. E. Ch'ng, *Industrial Management & Data Systems* **115**, 612 (2015).
4. X. J. Wang, W. Leroy, C. Xu and E. Ch'ng, *Industrial Management & Data Systems* **115**, 1724 (2015).
5. M. E. J. Newman, *Oxford University Press Inc., New York* , p. 371 (2010).
6. M. E. J. Newman and M. Girvan, *Physical Review E* **69** (2004).
7. M. E. J. Newman, *Nature Physics* **8**, 25 (2012).
8. S. Fortunato, *Physics Reports* **486**, 75 (2010).
9. G. Palla, I. Derényi, I. Farkas and T. Vicsek, *Nature* **435**, 814 (2005).
10. W. Chen, Z. M. Liu, X. R. Sun and Y. J. Wang, *Data Mining and Knowledge Discovery* **21**, 224 (2010).
11. B. J. P. Ahn Yong Yeol and S. Lehmann, *Nature* **466**, 761 (2010).
12. T. Evans and R. Lambiotte, *Physical Review E* **80** (2009).
13. L. Huang, G. S. Wang, Y. Wang, W. Pang and Q. Ma, *International Journal of Modern Physics B* **30**, 165 (2016).
14. S. Fortunato and D. Hric, *Physics Reports* **659**, 1 (2016).
15. M. E. J. Newman, *Proceedings of the National Academy of Sciences* **103**, 8577 (2006).
16. V. D. Blondel, J. L. Guillaume, R. Lambiotte and E. Lefebvre, *Journal of Statistical Mechanics: Theory and Experiment* **2008** (2008).
17. J. M. Shao, Z. C. Han, Q. L. Yang and T. Zhou, *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* , 1075 (2015).
18. M. S. Shang, D. B. Chen and Z. Tao, *Chinese Physics Letters* **27** (2010).

19. D. B. Chen, M. S. Shang, Z. H. Lv and Y. Fu, *Physica A: Statistical Mechanics and its Applications* **389**, 4177 (2010).
20. D. B. Chen, Y. Fu and M. S. Shang, *Physica A: Statistical Mechanics and its Applications* **388**, 2741 (2009).
21. J. L. He and D. B. Chen, *Physica A: Statistical Mechanics and its Applications* **429**, 87 (2015).
22. J. L. He, D. B. Chen, C. J. Sun, Y. Fu and W. J. Li, *Physica A: Statistical Mechanics and its Applications* **469**, 438 (2017).
23. S. Gregory, *New Journal of Physics* **12** (2010).
24. Y. X. Zhao, S. H. Li and S. L. Wang, *Advances in Complex Systems* **17** (2014).
25. J. Yang and J. Leskovec, *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining* , 587 (2013).
26. D. R. Lai, X. J. Wu, H. T. Lu and C. Nardini, *International Journal of Modern Physics C* **22**, 1173 (2011).
27. L. Y. Tang, S. N. Li, J. H. Lin, Q. Guo and J. G. Liu, *International Journal of Modern Physics C* **27**, p. 165046 (2016).
28. Y. Pan, D.-H. Li, J.-G. Liu and J.-Z. Liang, *Physica A: Statistical Mechanics and its Applications* **389**, 2849 (2010).
29. X. Y. Wang and X. M. Qin, *International Journal of Modern Physics C* **28**, p. 1750006 (2016).
30. W. J. Liu, J. F. Liu, M. M. Cui and M. He, *International Conference on Genetic and Evolutionary Computing* , 386 (2010).
31. A. S. Li and X. Yong, *Scientific Reports* **4** (2014).
32. L. A. Adamic and E. Adar, *Social Networks* **25**, 211 (2003).
33. J. G. Liu, L. Hou, X. Pan, Q. Guo and T. Zhou, *Scientific Reports* **6** (2016).
34. R. B. Myerson, *Game theory* (Harvard University Press, 2013).
35. L. Y. Lü, C. H. Jin and T. Zhou, *Physical Review E* **80** (2009).
36. T. Zhou, L. Y. Lü and Y. C. Zhang, *European Physical Journal B* **71**, 623 (2009).
37. A. Lancichinetti, S. Fortunato and F. Radicchi, *Physical Review E* **78** (2008).
38. A. Lancichinetti, S. Fortunato and J. Kertész, *New Journal of Physics* **11** (2009).
39. V. Nicosia, G. Mangioni, V. Carchiolo and M. Malgeri, *Journal of Statistical Mechanics: Theory and Experiment* **2009** (2009).
40. D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten and S. M. Dawson, *Behavioral Ecology and Sociobiology* **54**, 396 (2003).