Future Generation Computer Systems 105 (2020) 533–546

Contents lists available at ScienceDirect

Future Generation Computer Systems

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

LGIEM: Global and local node influence based community detection

Tinghuai Ma [a,*], Qin Liu [a], Jie Cao [b], Yuan Tian [c], Abdullah Al-Dhelaan [d], Mznah Al-Rodhaan [d]

[a] School of Computer & Software, Nanjing University of Information Science & Technology, Jiangsu, Nanjing 210-044, China
[b] Reading Academy, Nanjing University of Information Science & Technology, Nanjing 210-044, China
[c] Nanjing Institute of Technology, JiangSu, China
[d] Computer Science Department, College of Computer and Information Science, King Saud University, Riyadh 11362, Saudi Arabia

ARTICLE INFO

Article history:
Received 15 April 2019
Received in revised form 8 November 2019
Accepted 13 December 2019
Available online 18 December 2019

Keywords:
Influential nodes
Expansion strategy
SIR model
Community detection

ABSTRACT

Community detection is one of the hot topics in the complex networks. It aims to find subgraphs that are internally dense but externally sparsely connected. In this paper, a new method is proposed to identify the most influential nodes which are considered as cores of communities and achieve the initial communities. Then, by an expansion strategy, unassigned nodes are added to initial communities to expand communities. Finally, merging overlapping communities to get the final community structure. To evaluate the performance of the proposed node influence method (LGI), the susceptible–infected–removed (SIR) diffusion model are used. Testing with the synthetic networks and real-world networks, LGI can identify best nodes with high influence and is better than other centrality methods. Finally, experiments show that our proposed community detection algorithm based on influential nodes (LGIEM) is able to detect communities efficiently, and achieves better performance compared to other recent methods.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Due to the rapid development of social networks, various applications in different fields such as society, education, biology graphically represent these data. In social networks, a network can be seen as a graph, where nodes represent people, edges between them signify friendship or interaction between them. Community detection is aimed at identifying groups of nodes closely related to each other and relationship among communities is sparse. Researchers paid more attention to better identify communities, which can accurately reveal hidden information and network structures.

In the past few years, many methods for detecting communities, such as modularity-based algorithms, spectral algorithms, density-based algorithms, have been proposed. Since most of the real networks are complex with many nodes and edges, many algorithms have been executed inefficiently on large networks. Many researchers change the direction to the study of local community detection, especially seed expansion algorithms. Seed expansion can be used to extract small and relevant communities in large networks, which focuses on the identification of communities within the graph based on seed nodes.

The most important seed expansion is to identify seeds in the network. In social networks, some nodes are more important than others because of their ability of spreading information in the network. The problem above is transformed to find influential nodes. In this paper, a new metric is proposed to identify influential nodes according to global and local attributes [1] and expand these nodes to detect communities in complex networks. Our main contributions are listed as follows.

(1) A global and local node influence based community detection method (LGIEM) is proposed, which expands seeds (influential nodes) to detect communities.

(2) A method (LGI) is proposed to find influential nodes, which is based on global attributes and local attributes. We tune the parameter to identify more influential nodes. It is verified that selected influential nodes are better according to SIR model and Rank Biased Overlap (RBO).

(3) An expansion strategy based on node-cluster similarity and distance in LGIEM is proposed to find local community and all nodes in networks are in full coverage.

(4) According to our experiments, the performance of LGIEM is more accurate and effective than other similar methods on benchmark and real-world networks.

The rest of this paper is organized as follows. Section 2 reviews some related work. The main idea of our algorithm is described in Section 3. Experiments setup is described in Section 4. The experimental results on both synthetic and real networks are

* Corresponding author.
E-mail address: thma@nuist.edu.cn (T. Ma).

https://doi.org/10.1016/j.future.2019.12.022
0167-739X/© 2019 Elsevier B.V. All rights reserved.

presented in Section 5 and Section 6 is the conclusion of this paper.

2. Related works

Community structure is a common feature of networks [2] and becomes especially important in complex networks. There are many criteria for community detection, such as interests, hobbies, even work, other reasons, etc. [3].

Community detection is an unsupervised learning technique of grouping nodes into communities in consideration of the network structure. The internal connections in each community are close, external connections between communities are sparse. Various community detection methods have been proposed. Generally, these methods can be mainly divided into eight categories, feature distance, internal density, bridge detection, diffusion, closeness, structure, link clustering and no definition [4,5]. Feature distance converts community detection to clustering problem, which utilizes a distance measure to consider the edge connection, such as K-means. Internal density can find non-overlapping communities based on modularity. Most methods get robust and effective performance for community detection [4–8]. Bridge detection focuses on finding bridges which connect dense part of a network, so it can detect community structure by removing these bridges. In 2002, GN algorithm [2] was proposed by Girvan and Newman. It adopts the idea of edge betweenness to divide and process small networks. A diffusion is a process of propagation, which spreads information, similarity, interests to form communities. In the spectral clustering method [9], a similarity graph is constructed from the network, and then the community is determined based on the spectrum analysis of the similarity graph. Label propagation [10] is one of the typical algorithms and many algorithms are proposed based on its idea, such as Copra [11], SLPA [12]. Closeness is a process of detecting communities by a random walk. Walktrap and Informap are its typical algorithms [4]. A structure can be considered as the graph mining, whose aim is to find the maximal structures to satisfy constraints and structural rules. CNM algorithm [13] achieves a hierarchy of the community which contains the given node by searching the graph. Clique percolation method (CPM) [14] looks for the maximal-cliques in the network, and then these maximal-cliques are used to find the connected subgraphs of k-cliques. The community structure can be adjusted by changing the k value. However, for an unknown network, it is difficult to search all maximal cliques and it can be seen as an NP hard problem. To solve the problem above, many greedy algorithms are proposed, which consider local expansion and optimization. Its whole process can be described as selecting influential nodes as seed set, forming initial communities and greedy adding unassigned nodes into communities relying on a local benefit function. This greedy expansion stops once the value of the benefit function is negative. These greedy methods mainly consist of two parts: influential nodes selection and expansion strategy (benefit function). Link clustering mainly considers the edge connection, which detects communities by dividing edges of a network. The last one is no definition methods. These methods mainly are based on assumption and definite some pre-processing or post processing. Taking all types of community detection into consideration, we focus on the study of the structure definition.

Expansion is closely related to the core nodes, which is also called influential nodes. Therefore, algorithms of identifying influential nodes in complex networks spring up [15–18]. The most common use of methods is degree centrality (DC) [19,20], closeness centrality (CC) [21,22], betweenness centrality (BC) [23], PageRank [24] and Katz centrality [25]. According to the concept of centrality, if its centrality value is higher than that of other nodes in the network, a node is considered as an influential node.

Taking computational complexity into consideration, many centrality methods are not applicable in large-scale networks. In 2011, Kitsak et al. proposed K-shell method [26]. The algorithm considers that nodes of a network are hierarchical. The higher the kshell layer of a node, the more likely it is to be the core of the network. The idea of this method is k-shell decomposition and its biggest drawback is that multiple nodes are in the same layer and they have different degrees. So it cannot find core nodes which are a few. The larger the k-shell value of a node is, the greater its influence in the network is. In 2016, the betweenness centrality was improved by two-random-sampling. It estimates the betweenness of all nodes and pays more attention to the top-k nodes. It also can make constraint restrictions through VC dimensions [27].

In addition, as we know, the nodes inside the community are closely connected, and the connections between communities are sparse, which construct a modular network. Many methods ignore this property. The characteristic of communities rely on community structures and heterogeneous distributions of weak ties among nodes bridging communities. The main the idea of identifying influential nodes is epidemics spreading. The method relies on k-shell decomposition and connectivity of neighbors, so the weak ties and strong ties of network are processed separately which makes it more stable [28]. Later, Zakariya et al. proposed a new centrality method based on a two-dimensional vector in the modular network, which considers a local component and a global component. The local component mainly considers the influence on its own community. The global component is based on the unions of the connected parts and considers the influence on the other communities of the network [29].

You et al. [30] think all nodes revolve around their core nodes, so locally connections of core nodes or local density is very closely (i.e. higher ρ_i). In addition, the whole network consists of multiple groups formed by core nodes. The core nodes cannot be adjacent and are dispersedly distributed in the network. So they have a long distance from each other (i.e. higher δ_i). Based on the reasons above, it proposed a measure $\gamma_i = \rho_i\delta_i$ to identify core nodes. The higher the value of γ_i is, the more likely node i being the core node is. It automatically select the core nodes through a reasonable measurement method instead of using the cluster centers as the core nodes manually.

Bozorgi et al. [31] proposed an influence maximization method based on community and linear threshold model. In addition, researchers are constantly trying new directions and ideas, so more and more mathematical [32,33] and bio-inspired methods [34] are widely applied to identify influential nodes. In brief, how to prove the rationalization and effectiveness of the identifying influential nodes is still an open issue [35].

Many studies have shown the co-relations among various centrality measures and discussed these issues: which centrality measure can obtain the optimal top-k influential nodes in a given network; which centrality measure is the best fit for different type of networks etc. However, there has been no perfect solutions until now, we can only make the selected influential nodes as accurate as possible for different situations. With respect to the local benefit function, many algorithms usually adopt modularity [36,37], conductance [38] or connectivity [39] to judge whether the result of community detection is well or not. However, some of these methods ignore the topological properties of the network, so a topological approach are proposed which considers two types of comparative methods (traditional evaluation and topological evaluation) [40]. MCDM strategy measure three attributes (topological properties, quality metrics and clustering metrics) simultaneously and rank these comparative methods to judge the best community detection method [41]. The similarity

of these methods can be measured on basis of the size density distributions of the final communities. There are differences among different community detection methods [42].

Based on the description of related work, many algorithms have the following problems. (1) The algorithm cannot consider global information and local information at the same time. (2) For large networks, the effectiveness of the algorithm is reduced. To solve the problems above, many algorithms are proposed, especially seed expansion methods. What is more, a fast influence maximization framework is very formal to identify the seed set. Two-phase Competitive Influence Maximization (TCIM) is a general algorithmic framework, which can describe the ability of information propagation in the network [43]. Although the algorithm guarantees both quality and efficiency, it only considers the Competitive Influence Maximization with Partial information (CIMP) problem. TCIM does not consider the effect of global information of nodes on community detection. Such algorithms ignore the importance of global information and do not solve the problem (1) above mentioned. Algorithms based on the framework only consider partial information because of the complexity of large network and computational complexity. In addition, algorithms based on local and global information solve the above problems and can achieve more effective results. It is an open issue so it gives us more opportunities.

In this paper, a novel community detection algorithm based on influential nodes (LGIEM) is proposed, which is the abbreviation of LGI expanding and merging. It mainly consists of three phases: influential node selection, community expansion and merging overlapping communities. In the stage of influential node selection, we propose a new metric to identify influential nodes based on global and local structure (LGI). The top-k influential nodes are considered as seeds and the initial communities consist of seeds and their one-hop neighbors. In the expansion stage, a new strategy considering the similarity and distance between unsigned nodes and existing communities is proposed. In the stage of merging overlapping communities, if the number of overlapping nodes between two communities is over a threshold, the two communities are merged. We test the method on benchmark and real networks. The results show that LGIEM makes nodes be in full coverage and outperforms other recent methods in terms of modularity, normalized mutual information (NMI), accuracy (ACC) and TOPSIS.

3. Proposed method

The algorithm LGIEM can divide into three stages. Firstly, the influence of nodes in the network are calculated and select the top-k nodes as seeds. The seeds and its relevant neighbors constitute several initial communities. Secondly, a new expansion strategy is proposed in view of similarity and distance between unassigned nodes and existing communities, which can be used to expand the initial communities. Finally, communities are merged and final community structure is achieved.

3.1. The influential nodes selection process

3.1.1. Influence of nodes

A node with high influence means that the node is highly capable of spreading information in the network and has a great impact on its neighbors. Conversely, if a node has low influence, it is likely to be affected by its neighbor node with high influence [16,44]. In this paper, a new measure based on local and global information (LGI) is proposed to find influential nodes in the undirected networks.

The k-shell decomposition algorithm is used to represent the global information of nodes in the network. Considering the limitations of the k-shell algorithm itself, we add the k-shell entropy to measure global information. We propose the concept of belonging, which can measure local information. The belonging is calculated by the sum of similarities between influential nodes and their neighbors. The similarity is in view of weights of nodes. Combine global information with local information in a certain proportion to achieve influence of nodes in the whole network. The metric function for calculating the influence of nodes is as shown in the formula (1).

$$influence(i) = a(E_i) + b(B_i) \tag{1}$$

(1) global information (k-shell entropy)

K-shell is usually used to represent the layer of a node in the network. There are many nodes at the same layer so we are unable to distinguish the influence of each node at the same layer in the network. Therefore, we use k-shell entropy to measure the hierarchical information of a node. Based on the concept of network connection entropy, the initial k-shell hierarchy is reprocessed to determine the impact of nodes on each layer of nodes in the network to measure the global information throughout the network. Since the nucleus has a relatively low information entropy, the larger the value of kshell entropy is, the greater the global influence of a node is. The maximum entropy indicates that a node has the ability to connect all network layers in a complex network, and the minimum entropy indicates that all nodes are in the same network layer. The entropy of node i will reflect whether its neighbor nodes have more significant diversity, and the kshell entropy of the node is as shown in formula (2).

$$E_i = -\sum_{j=1}^{ks_{max}} p_i(x_j) \times log_2 p_i(x_j) \tag{2}$$

$$p_i(x_j) = \frac{|x_j|}{\sum_{j=1}^{ks_{max}} x_j} \tag{3}$$

where $x_j = \{1, 2, \ldots, ks_{max}\}$ denotes the kshell layer, to which neighbors of node j belongs; $p_i(x_j)$ is the probability of the neighbor of node i in the j-shell layer; $|x_j|$ represents the number of nodes in the j-shell layer.

(2) local information (belonging)

After obtaining the global information of nodes, the node's own properties are considered. The node's own properties have more performance in the relationship with neighboring nodes. Degree centrality is widely used in this aspect. On this basis, local information of a node only considers its neighbors, which directly connect to the node. The node's own attributes are judged by the similarity with neighboring nodes. The higher the sum of the similarity between the node and the neighboring nodes is, the higher the belonging of the node to the neighboring node is, expressed by B_i.

$$B_i = \sum_{j \in N(i)} s(i, j) \tag{4}$$

$$s(i, j) = \frac{2 \cdot w(i, j) + \sum_{t \in N(i) \cap N(j)} w(i, t) \cdot w(j, t)}{\sqrt{(1 + \sum_{t \in N(i)} w(i, t)^2)(1 + \sum_{t \in N(j)} w(j, t)^2)}} \tag{5}$$

$$w(m, n) = \frac{|N(m) \cap N(n)|}{|N(m) \cup N(n)|} \tag{6}$$

where $w(m, n)$ represents weight of the edge connecting node m and node n. $s(i, j)$ denotes the similarity between node i and node j, which is related to weights of nodes. $N(i)$ represents the number of neighbors of node i.

In a nutshell, the process of calculating influence of nodes is shown in Algorithm 1. Firstly, the kshell entropy of each node is calculated, indicating the global information of nodes. Then, local information based on belonging is achieved by similarity between

Algorithm 1 Pseudocode of LGI

1. initialize V = all nodes in G
2. For $i = 1 : n$
3. compute kshell by kshell decomposition algorithm
4. EndFor
5. compute k-shell entropy of nodes according to formula (2)
6. calculate N_i – the number of neighbors of node i
7. calculate $w(i, j)$ – the weight of the edge connecting node i and node j
8. calculate s(i,j) according to weight of nodes
9. achieve belonging of node i based on the sum of similarity between node i and its neighbors
10. calculate $influence(i)$ combined with kshell entropy and belonging B_i
11. normalized $influence(i)$

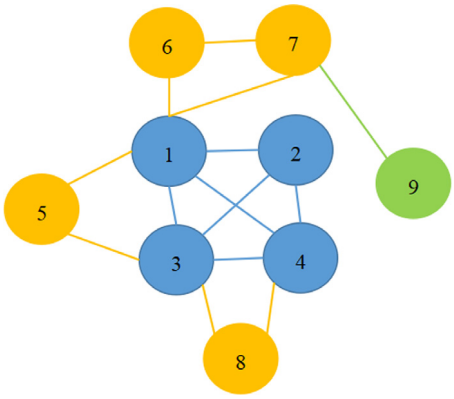

Fig. 1. A simple network. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

influential nodes and their neighbors. Therefore, we can evaluate node influence on the network according to the formula (1).

(3) A simple example

The simple example shown in Fig. 1 allows walking through the various metrics involved in the final influence score computation. Here, the parameters a and b are known, 0.6 and 0.4 respectively in Eq. (1) (discussed in Section 5.1). In this network, kshell of each node can be achieved easily. Different colors represent different shell layers separately. Node 9 belongs to 1-shell, nodes 5, 6, 7 belong to 2-shell and nodes 1, 2, 3, 4 belong to 3-shell. From the aspect of kshell, nodes 1, 2, 3, 4 are the most influential nodes which have the same influence. Based on the concept of kshell entropy, only node 1 is the most influential node in the network. Table 1 shows the local information (Belonging), node 1 shows its superiority among the whole network. According to formula (2) and (4), global influence and local influence of each node can be calculated. The ranking of influence in descending order meets the list of 1, 3, 4, 2, 8, 5, 6, 7, 9.

To sum up, it can be noticed from the simple network that when both local and global information are considered, the influential nodes are the core of networks due to their ability to spread more information to other nodes. Therefore, they can be regarded as cores of communities in the network.

3.1.2. Initial communities

$$C_1 \cup C_2 \cup \cdots \cup C_k \cup C_u = G \tag{7}$$

$$C_i = \{node_i, neighbors\ of\ node_i\}, i \in 1, 2, \ldots, k \tag{8}$$

Table 1
Local influence of each node in the simple network.

Node	Local
1	1
2	0.709
3	0.535
4	0.761
5	0.309
6	0.289
7	0.255
8	0.433
9	0

According to Section 3.1.1, the influence of nodes is achieved. The ranking list of node influence is achieved and nodes in the ranking list are sorted in descending order. The top-k nodes are selected as seeds which are considered as cores of communities. Due to the complexity of the network structure, how to determine the number of seeds is still an open issue, which is an NP hard problem [20]. The networks used in this paper have real community structure so we know the real number of communities of these networks. In order to show the effectiveness of LGIEM, we set k as the real number of communities in the network. Therefore, these seeds and their one-hop neighbors form the initial communities. In addition, the initial communities consist of these seeds and their one-hop neighbors together. The initial communities can be represented as $\{C_1, C_2, \ldots, C_k\}$, and the whole network can be represented as formula (7), where C_u is the set of unassigned nodes, $node_i$ represents the top-i influential nodes and neighbors of $node_i$ represent the neighbors of $node_i$.

3.2. The expansion process

In Section 3.1.2, initial communities are achieved. The candidate set for each community is $NS(j)$, which consists of unassigned nodes of a network. For the unassigned nodes in the candidate set, these nodes are sort in descending order. The node with higher influence has more impact on its local nodes so it is assigned into a community preferentially. We take into account both the similarity ($NC(C_i, j)$) and the distance ($DC(C_i, j)$) between a unassigned node j and a existing community C_i. The priority of a candidate node j to community C_i is defined as follows:

$$p(j, C_i) = NC(C_i, j) + D(C_i, j) = \sum_{i \in C} |N(i) \cap N(j)| + \sum_{i \in C} \frac{1}{distance(i, j)} \tag{9}$$

C_i represents the community to which node i belongs. $p(j, C_i)$ measures the closeness between a unassigned node j and a community C_i. Firstly, belonging to the same community, there are many adjacent nodes (the first item). Secondly, belonging to the same community, the sum of the distance from this node to the nodes of the community is short (the second item). Therefore, The assumption is reasonable that the measure $p(j, C_i)$ can be used to assign the candidate nodes. If a node j has the same priority for multiple communities, node j is simultaneously assigned to those communities. The nodes belonging to multiple communities is called overlapping nodes.

The structure similarity can also be used to measure the distance of nodes. The shorter the distance between nodes is, the greater the structure similarity [48] is. Therefore, we use the structure similarity to replace the second term of formula (12). Structure similarity is represented as follows:

$$r(i, j) = \frac{|N(i) \cap N(j)|}{\sqrt{N(i) \times N(j)}} \tag{10}$$

Table 2
Comparison of computational complexity of different centrality measures.

Method	Computational complexity	Description of notations
Degree centrality (DC)	$O(n)$	n: number of nodes
Closeness centrality (CC)	$O(n^3)$	
Betweenness centrality (BC)	$O(n^2 logn + nm)$	m: number of edges
K-shell (KS)	$O(m)$	
PageRank	$O(mi)$	i: number of iteration until algorithm convergence
Katz centrality (BC)	$O(n^2)$	
LGI	$O(n^2 + m)$	

Table 3
Computational complexity of different community detection methods.

Method	Computational complexity	Description of notations
Louvain	$O(n + m)$	
LPA	O(n)	
LINK [45]	$O(n * k_{max}^2)$	k_{max}: maximum node degree in the network
OCDLCE [46]	$O(n^2)$	
OCDSSE [47]	$O(cn^2)$	c: the size of maximum community
SECD [36]	$O(n^2 + h * (2blogm/\beta) + hc * logn)$	b, β: parameters
LGIEM	$O(n^2 + tm)$	t: $t = k + 1$, k is the number of top-k influential nodes

and

$$N(i) = \{j \in V | (i, j) \in E\} \cup \{i\} \tag{11}$$

Therefore, the formula (9) can also written as formula (12). In the following experiments, we use the formula (12) to expand communities.

$$p(j, C_i) = \sum_{i \in C} |N(i) \cap N(j)| + \frac{1}{\sum_{i \in C} r(i, j)} \tag{12}$$

3.3. Merging communities

In the influential nodes selection process, the top-k influential nodes and their neighbors consist of initial communities so a node can be assigned into different communities if it is connected to several seeds. In the expansion process, if a node has the same priority of belonging to several communities, the node will be assigned into these communities. So, there are many overlapping nodes in these communities. If the overlap between communities is too high, it will increase the complexity of the algorithm. Therefore, it is necessary to control the overlap of communities to optimize community structure. A parameter of overlapping rate $\delta = |C_i \cap C_j| / min\{C_i, C_j\}$ is used to control overlap of communities. The larger δ is, the more nodes are overlapping among communities. Therefore, if the overlap between two communities exceeds 0.5, the two communities will be merged. The final overlapping communities are represented as $C = \{C_1, C_2, \ldots, C_t\}$. The whole algorithm was restated in Algorithm 2.

3.4. Computational complexity

LGIEM consists of three parts of influential nodes selection, seed expansion and merging communities, so its overall complexity depends on these three parts. We assume a network G with n nodes and m edges. We will analyze these three parts separately for this network G to get the complexity of LGIEM.

The first part of LGIEM is calculating node influence (LGI). The complexity of LGI contains two aspects. Local information of nodes is calculated by the sum of similarities between influential nodes and their neighbors, which is based on the weight of nodes. The aspect of local information will take $O(n^2)$. And kshell entropy represents the global information of each node, its complexity is $O(m)$. So the complexity of LGI is $O(n^2 + m)$ (Table 2). Compared with other centrality methods, the computation complexity of LGI is not too large. The main role of seed expansion process is to assign unassigned nodes into the initial communities according to some rules. It mainly considers the similarity and distance of unassigned nodes and existing communities. The whole process will take $O(km)$, where k is the number of influential nodes, each of which takes $O(m)$ on its expansion. The third process of LGIEM is to merge overlapping communities. It will take $O(n^2)$. Thus, the complexity of LGIEM is $O(n^2 + (k + 1)m)$ (Table 3).

Algorithm 2 Pseudocode of LGIEM)

Step1: The influential nodes selection process
1. calculate influence of nodes (Algorithm 1)
2. find initial communities which consist of seeds and their nearest neighbors
3.**Step2:** The expansion process
4. $NS(j) \leftarrow$ the unassinged nodes of the network
5. sort these nodes in a descending order
6. for j in the sequential candidate set
7. for $C = C_1, C_2, \ldots, C_k$
8. $p(j, C_i) = \sum_{i \in C} |N(i) \cap N(j)| + \frac{1}{\sum_{i \in C} distance(i,j)}$
9. end for
10. if $p(j, C_m) = max_{i \in \{1,2,\ldots,k\}} p(j, C_i)$
11. if $p(j, C_n) = p(j, C_m))$ $(m, n \in \{1, 2, \ldots, k\})$
12. $j \rightarrow C_m \cup j \rightarrow C_n$
13. else
14. $j \rightarrow C_m$
15. end if
16. end if
17. end for
18.**Step3:** Merging overlapping communities
19. calculate $\delta = |C_i \cap C_j| / min(C_i, C_j), i, j = 1, 2, \ldots k (i \neq j)$
20. if δ is larger than threshold value (0.5) then
21. $C_i \leftarrow \{C_i \cup C_j\}$
22.**Output:** $C = C_1, C_2, \ldots, C_t$

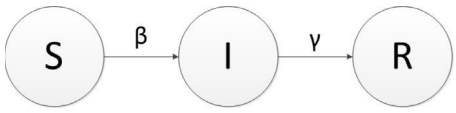

Fig. 2. Susceptible−Infected−Removed (SIR) epidemic model.

Table 4
The information of six groups of LFR networks.

Benchmark	N	k	maxk	minc	maxc	mu
1	500	10	50	10	50	0.1
2	500	10	50	10	50	0.3
3	500	10	50	10	50	0.5
4	1000	15	50	20	100	0.1
5	1000	15	50	20	100	0.3
6	1000	15	50	20	100	0.5

Table 5
The information of real-world networks.

Networks	n	m	Number of real communities
Karate Club	34	78	2
Dolphin	62	159	2
Polbook	105	441	3
Football	115	613	3
Email-Eu-Core	1005	25571	42
PGP	10680	24316	153

4. Experiments setup

4.1. Datasets

4.1.1. Benchmark networks

Synthetic networks are often used to test the accuracy of the algorithm. Through the setting of tunable parameters, the corresponding real community structure can be obtained.

The Lancichinetti–Fortunato–Radicchi (LFR) benchmark network [49] is constructed by several tunable parameters, which has similar properties as a real network. As a typical synthetic network, LFR benchmark networks are the most commonly used for community detection. Different types of networks can be generated by setting different values of related parameters.

In this paper, we adopts six groups of LFR networks. The six groups of networks can be considered as two groups according to the number of nodes N. The first three networks share the common parameters of nodes number $N = 500$, average degree $k = 10$, max degree $maxk = 50$, minimum number of communities $minc = 10$ and maximum number of communities $maxc = 50$. The parameter of mu is set to 0.1, 0.3 and 0.5, which represents the complexity of networks. The higher mu is, the more complex the network is. The last three networks have the similar properties, which are $N = 1000$, $k = 15$, $maxk = 50$, $minc = 20$ and $maxc = 100$. mu is unchanged and the network structure is expanded. The details are shown in Table 4.

4.1.2. Real networks

Like synthetic networks, real networks are also used to detect the performance of different community detection algorithms (see Table 5).

- Karate Club [50] is a network, which consists of players, a coach and a manager. Focused on the coach and the manager, the network can splits into two communities.
- Dolphin [51] is a network, whose construction is based on the observation of 62 bottlenose dolphins from 1994 to 2001. Each node of the network is a dolphin, edges represents relation between dolphins.
- Books about US politics (Polbook) [52] is a network of books about US politics. The books are published around the time of the 2004 presidential election and are sold on Amazon website. Nodes of the network represent books and an edge means that the books represented by the two nodes connecting the edge are purchased by the same buyer.
- Football [53] is a network of American College football games. Each node denotes a team and an edge between two nodes means that there is at least one game between the two teams.
- Email-Eu-core [54] is an incoming and outgoing email network among members of the research institution. Nodes of the network can spread anonymized information. A member sends at least one email to other member, there is an edge between them. The network is an internal network and has nothing to do with the outside world.
- PGP [55] is a social network. Nodes represent people, who shares confidential information using the Pretty Good Privacy (PGP) encryption algorithm.

4.2. Rank Biased Overlap (RBO)

The top-k influential nodes are considered as seeds, so we should pay more attention to the nodes with high ranks. Rank biased overlap (RBO) [56] is a method to examine the accuracy of the ranking list of top-k influential nodes. The principle of RBO is that nodes with different ranks have different weights, giving higher weights to the nodes with higher ranks. The greater the influence of the node is, the greater the weight assigned by RBO is. The value of RBO of two ranking lists A and B can be calculated as formula (13):

$$RBO(A, B, p) = (1 - p) \sum_{d=1}^{n} p^{d-1} A(A, B, d) \tag{13}$$

where parameter p ($0 \leq p \leq 1$) decides on the extent of the decline in weight. The smaller the value of p is, the higher the weights of nodes with high ranks are. If $p = 0$, it only considers the nodes with high ranks [21]. The value of RBO is in the range of 0–1, where a greater value means the two ranking lists have more similarities. If RBO is close to 0, it means the two ranking lists are completely different. On the contrary, it means that the two ranking lists are exactly the same. In formula (13), $A(A, B, d)$ is the value of overlap between two ranking lists A and B up to rank d, which calculated by formula (14), and n is the number of elements on the ranking list.

$$A(A, B, d) = \frac{|A_{1:d} \cap B_{1:d}|}{|A_{1:d} \cup B_{1:d}|} \tag{14}$$

where $A_{1:d}$ represents the elements present in rank 1 to d of list A, and $B_{1:d}$ indicates the same elements on list B.

4.3. Susceptible–infected–removed model

Various models have been presented to examine the accuracy of influential nodes. We utilize Susceptible–Infected–Removed (SIR) model [57] to simulate the spreading ability of nodes in this paper.

SIR model is an epidemic spreading models and widely used in community detection to simulate information spreading among nodes in complex networks. As shown in Fig. 2, in SIR model, each node has three discrete states: (i) susceptible S(t), represents the individuals which are not yet infected with the disease; (ii) infected I(t) denotes the ones which are infected with the disease and can spread the disease to the susceptible nodes; (iii) recovered R(t), stands for the individuals which are infected to

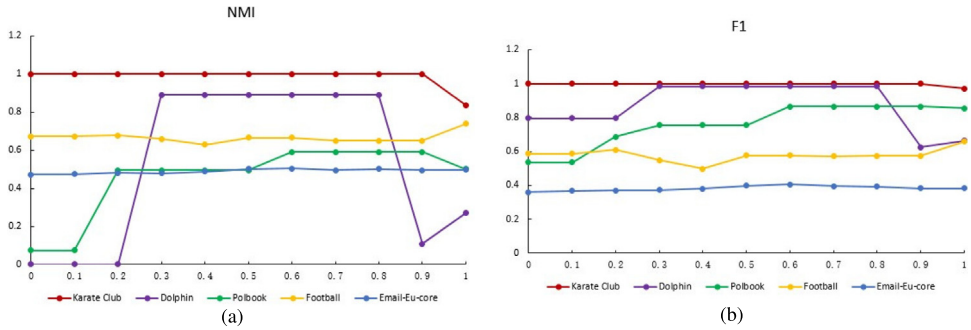

Fig. 3. The example of the karate club. (a) Represented the initial graph, (b) Represented the initial graph with two core nodes.

immunize or die. These nodes are neither infected again, nor can they transmit the infection to others.

Initially, only one node is infected and other nodes are susceptible. The infected node spread the disease and infect their susceptible neighbors with probability β (set as 0.04 in this paper) and then enter the recovered state (R) with probability γ (set as 1), where they become immunized and cannot be infected again. This spreading process is repeated until there is no infected node in the network. The spreading ability of node i represented as s_i^β is defined as the number of nodes that are finally infected by node i in the end. The higher the value of s_i^β is, the better of the spreading ability of the node i is. Finally, a ranking list can be obtained based on the spreading ability of nodes in the network.

4.4. Evaluation criteria

In this paper, F1-score [57], modularity (Q) [58], Normalized Mutual Information (NMI) [30] and ACC [59] are used to measure the quality of community detection.

- F1-score is a measure of a method's accuracy. It considers both the precision and the recall to compute the score. It considers true or false based on community ownership of each node. The F1 score is the harmonic average of the precision and recall, where an F1 score reaches its best value at 1 (perfect precision and recall) and worst at 0.

$$F1 = 2\frac{precision \cdot recall}{precision + recall} \tag{15}$$

- Modularity is widely used to measure the quality of communities, the overlapping modularity is written as follows in formula (16):

$$Q = \frac{1}{2m}\sum_{c=1}^{K}\sum_{v\in C_c}\frac{1}{O_i O_j}(A_{ij} - \frac{k_i k_j}{2m}) \tag{16}$$

where m is the number of edges in the whole graph, k_i, k_j are respectively the degree of node i and j, A_{ij} is the adjacency matrix of the graph and O_i and O_j respectively denote the number of communities which node i and j belong to the same cluster and 0 otherwise. The higher the value Q is, the more accurate the result of community detection is.

- Normalized mutual information (NMI) is used to measure the similarity between the true community structure and the community structure obtained by community detection algorithms. The higher the value of NMI is, the more accurate the community detection algorithm is. NMI is formulated as:

$$NMI(A,B) = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} N_{ij}log(\frac{N_{ij}N}{N_{i.}N_{.j}})}{\sum_{i=1}^{C_A} N_{i.}log(\frac{N_{i.}}{N}) + \sum_{j=1}^{C_B} N_{.j}log(\frac{N_{.j}}{N})} \tag{17}$$

where C_A is the number of real communities, C_B denotes the number of found communities. The matrix N represents the confusion matrix, where N_{ij} is simply the number of nodes in the real community i that appear in the detected community j. $N_{i.}$ and $N_{.j}$ are the sum over row i and column j of the confusion matrix respectively. N is the number of nodes. When NMI is equal to 1, the community structure detected by the algorithm is the same as the real community structure. Conversely, if NMI = 0, the detected community structure is entirely independent of the real and the entire network consists of one community. It indirectly proves the effectiveness of the algorithm.

- ACC is defined as:

$$ACC(C,C') = \frac{\sum_{i=1}^{n} k(C_i, PM(C_i'))}{n} \tag{18}$$

where for a given node i, C_i and C_i' represent ground-truth cluster label and the computed cluster label of i respectively. $k(C_i, PM(C_i'))$ is a Kronecker function and has two parameters C_i, $PM(C_i')$. If $C_i = PM(C_i')$, $k(C_i, PM(C_i')) = 1$, otherwise, it is equal to 0. $PM(C_i')$ is a permutation mapping function that maps C_i' to C_i. n is the total number of nodes. The larger the value of ACC is, the better community structure detected by the algorithm is.

5. Results

5.1. Parameters tuning

In Section 3.1, the influence of nodes of the network has been defined. Among them, a represents global impact factor, b represents local impact factor and $a + b = 1$. Therefore, the formula (1) can also written as follows:

$$influence(i) = aE_i + (1-a)(B_i) \tag{19}$$

Parameters a will be tuned according to the real structure of the network. It is a threshold that can control the influence of nodes detected by LGI. In the following, the performance of LGIEM impacted by different a is evaluated. Fig. 3 plots NMI and F1-score of LGIEM respectively to present the performance of LGIEM with different settings for parameter a on the five real networks.

In fact, we can obtain better results by varying parameters of LGI on different networks. In our experiments, the parameter a is set as a constant, which can simplify the experimental process. As shown in Fig. 3, when the value of a is in the range of 0.3–0.8, NMI and F1-score obtain the best performance. Therefore, we can further define the range of values of a, that is, a belongs to the range of [0.3, 0.8]. So, b belongs to the range of [0.2, 0.7]. Taking Fig. 3(a) and (b) into consideration, In Karate Club and Dolphin, LGI performs better when a belongs to the range of [0.3, 0.8]. For Polbook, LGI performs better when a is 0.6. For Football, it

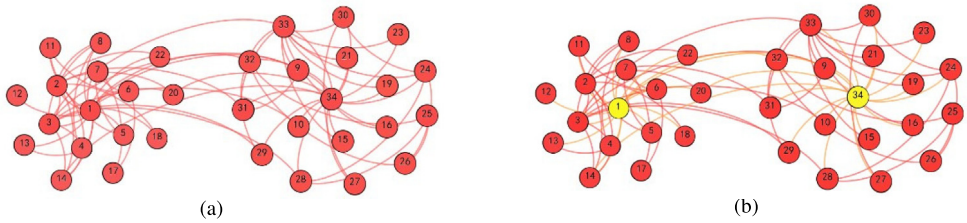

Fig. 4. The example of the karate club. (a) Represented the initial graph, (b) Represented the initial graph with two core nodes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Rank of nodes of the karate club.

Rank	DC	Kshell	PageRank	CC	BC	Katz	Isomap	LGI
1	34	1	34	1	1	1	34	34
2	1	33	1	3	34	34	33	1
3	33	34	33	34	33	3	1	33
4	3	6	3	32	3	33	3	2
5	2	7	2	9	32	32	32	3
6	4	30	32	14	9	2	9	7
7	32	2	4	33	2	9	2	6
8	9	3	24	20	14	14	14	4
9	14	27	9	2	20	20	28	30
10	24	28	14	4	6	4	31	32

performs better when a is 0.3. In Email-Eu-core, LGI performs better combining NMI and F1-score when a is 0.5. When LGI works best for different datasets, it corresponds to different value of a. For simplifying the experiments, the parameter a is set as a known constant. In Fig. 3(a), when a is set as 0.6, NMI of all networks are better. In Fig. 3(b), we can get the similar conclusion as that of the F1-score results. In addition, in the network of karate club, the coach and manager of the club are represented by node 1 and node 34, respectively. Both of them jointly manage the club, who are the core characters of this club. As shown in our experiment, when a is set as 0.6, the top-2 influential nodes of karate club are nodes 1 and 34, which is accordant with the practical situation. In summary, a is set to 0.6, b is 0.4 in this paper.

5.2. Effectiveness of LGI

5.2.1. Real world networks

As shown in Table 6, we rank nodes of the karate club and identify the influential nodes. DC, k-shell, PageRank, closeness, betweenness, katz centrality, Isomap and modular centrality methods are compared with LGI to show the effectiveness of LGI. In order to highlight the importance of influential nodes, we select top-m ($m > k$) to measure the accuracy of ranks of nodes. As mentioned in Section 5.1, the top-3 influential nodes are nodes 34, 1 and 33. The node 34 is more important than node 1 in the karate club and both of them are more important than node 33. From Table 6, DC and PageRank methods are also effective and can identify the influential nodes.

The top influential nodes achieved by different centrality measures are used to illustrate the effectiveness of LGI. The top influential nodes with different frequency of occurrence ($\geq$50%) on all centrality measures are added into a set. The higher the frequency of occurrence in the top rank of a node on all centrality measures is, the node is more likely to be considered as influential nodes. In order to achieve the proper set, we select top-m ($m > k$) to form the set. Table 7 obtained the results. The top-k influential nodes detected by LGI have high frequency of occurrence on all centrality measure. For example, in karate club, nodes 34 and 1 are considered as influential nodes by LGI and their frequency is 100%; in polbook, the top-3 influential nodes are nodes 13, 4, 67, whose frequency is above 75%; in dolphin, the top-2 influential nodes are in the final set.

Each centrality measures selects the top-m ($m > k$) influential nodes. The nodes are selected to constitute a candidate set of influential nodes, which appear more than 50% in all measures. Table 8 shows the percentage of the top-k influential nodes found in the candidate set by various measures. What is more, we illustrate that LGI can find influential nodes more effectively. As shown in Table 8, the top-k influential nodes detected by LGI are all in the candidate set in the first three networks. In the football network, the suitability of katz and closeness centrality is higher than that of LGI. In summary, LGI performs better in most cases.

Fig. 4 shows the process of LGI in detail taking by the karate club as an example. In Fig. 4(a), we can see the network structure of the karate club network, which has 34 nodes and 78 edges. In Fig. 4(b), it shows the whole network structure and two cores with yellow color. The influence of each nodes is computed by LGI. Then, the top-k influential nodes are selected on the basis of influence of nodes, which are considered as seeds (cores) in the graph. As we know, the real number of communities of karate club is 2, so k is 2 and seeds are nodes 1 and 34. They are very remarkable because they have maximal spreading ability. The Pseudo-code of the proposed method is elaborated in Algorithm 1.

5.2.2. Benchmark networks

In Section 5.2.1, we clarify that LGI performs better in detecting influential nodes in real networks. In this subsection, we mainly focus on verifying the effectiveness of LGI in benchmark networks. In the experiments, SIR model is used to simulate the spreading ability of nodes in the whole network, so a ranking list of all nodes is obtained. The ranking list serves as a standard to measure the correctness of ranking measures of influential nodes. Because we are more concerned with the top-k influential nodes with high ranks on the list, the rank biased overlap (RBO) is used to examine the accuracy of ranking lists, which are detected by different centrality measures.

As we can be seen from Table 9, LGI has higher correctness and accuracy in comparison with other methods in high ranks for most benchmark networks. In the first and forth networks, LGI also performs well and second only to Modular centrality, where the mixture parameter is set as 0.1. Modular centrality performs better when the network structure is clear. However, when the network is complex, its superiority is reduced. It shows the inferiority of other methods and the superiority of LGI at the same time. In conclusion, LGI performs better in identifying influential nodes in benchmark networks and real networks.

5.3. Results on different datasets

In this section, we verify the performance of LGIEM compared with eight community detection algorithms, namely, Louvain, LPA, LINK [45], OCDLCE [46], OCDSSE [47], SECD [36], AOCCM [60], and CAMAS [61]. Among the eight compared algorithms,

Table 7
Set of top influential nodes with different frequency of occurrence on all centrality measures.

Networks	[50%, 75%)	[75%, 100%)	100%	[50%, 100%]
Karate Club (top5)	[32]	[3, 33]	[1, 34]	[32, 3, 33, 1, 34]
Dolphin (top10)	[2, 15, 18, 21, 30, 37, 41, 46, 58]	[34, 52]	ø	[2, 15, 18, 21, 30, 37, 41, 46, 58, 34, 52]
Polbook (top10)	[59, 74, 85]	[4, 9, 13, 31, 67]	[73]	[59, 74, 85, 4, 9, 13, 31, 67, 73]
Football(top16)	[3, 4, 6, 7, 16, 17, 35, 81, 83, 93, 105, 107]	[1]	ø	[3, 4, 6, 7, 16, 17, 35, 81, 83, 93, 105, 107,1]

Table 8
The percent of top-k influential nodes in the set of top influential nodes.

	DC	Kshell	PageRank	CC	BC	Katz	Isomap	Modular	LGI
Karate Club (top2)	100%	50%	100%	100%	50%	100%	50%	100%	**100%**
Dolphin (top2)	50%	50%	100%	100%	100%	50%	50%	100%	**100%**
Polbook (top3)	100%	33.3%	100%	33.3%	33.3%	100%	100%	33.3%	**100%**
Football (top12)	58.3%	33.3%	50%	58.3%	**75%**	**75%**	50%	66.7%	66.7%

Table 9
RBO of many methods with SIR.

Benchmark	DC	Kshell	PageRank	CC	BC	Katz	Isomap	Modular	LGI
1	0.462	0.561	0.392	0.379	0.598	0.567	0.567	**0.612**	0.575
2	0.427	0.339	0.306	0.267	0.509	0.495	0.512	0.534	**0.598**
3	0.541	0	0.553	0.476	0.561	0.533	0.532	0.589	**0.597**
4	0.290	0.073	0.274	317	0.410	0.394	0.396	**0.431**	0.419
5	0.358	0	0.295	0.395	0.468	0.472	0.459	0.481	**0.483**
6	0.337	0	0.287	0.267	0.547	0.540	0.516	0.563	**0.568**

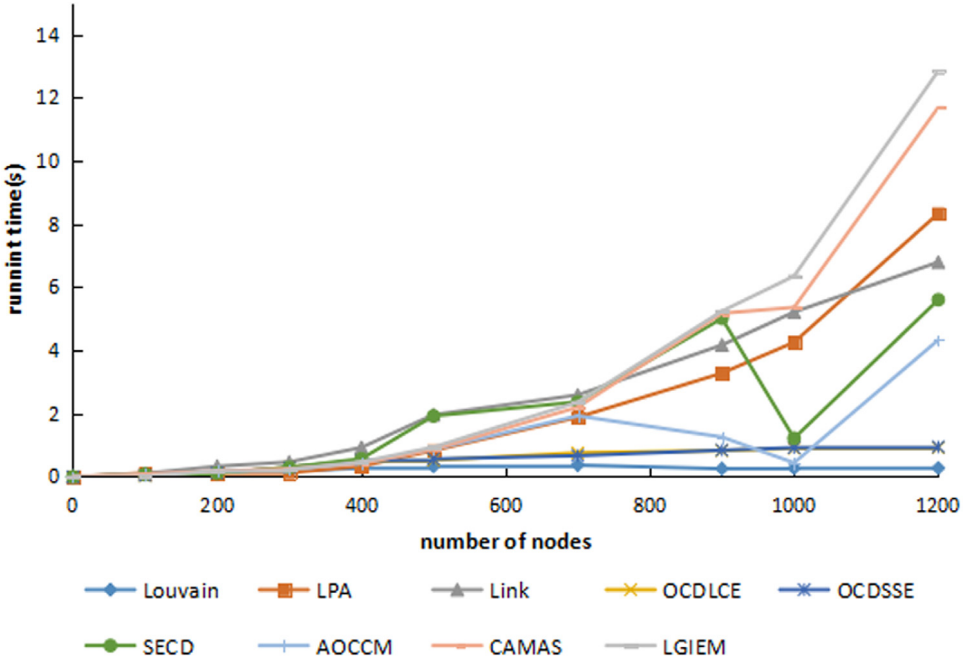

Fig. 5. Running time on benchmark network.

Louvain is based on the modularity optimization. LPA is fast but unstable, which is on the basis of information spreading. Different from other community detection algorithms, Link processes the edges rather than nodes to detect community structure. OCDLCE also operates on edges. Two nodes on either edge constitute the initial node set. Modularity function is used to judge whether nodes should be assigned to the local communities. Different from OCDLCE, both of OCDSSE and SECD are overlapping community detection algorithms using seed set expansion, which are based on nodes rather than edges. According to the idea of an autonomy-oriented computing based method, both crisp and soft communities can be detected in an incremental manner. This algorithm is called AOCCM and is practical. Based on an effective cluster-aware multi-agent system, CAMAS can act on attribute graphs and detect overlapping communities, which overcomes the drawback of AOCCM. What is more, the two algorithms measure influential nodes differently.

5.3.1. Results on benchmark networks

Tables 10 and 11 show the results of NMI and ACC on benchmark networks, both of them show the similar result. In the first four networks, Louvain and LPA perform better. When the network size increase from 500 to 1000, the performance of our method is improved and gradually highlight the advantages of LGIEM. The first three networks have 500 nodes, which means their network is small and their network structure is relatively simple. The mixture parameter of the fourth network is 0.1, so it

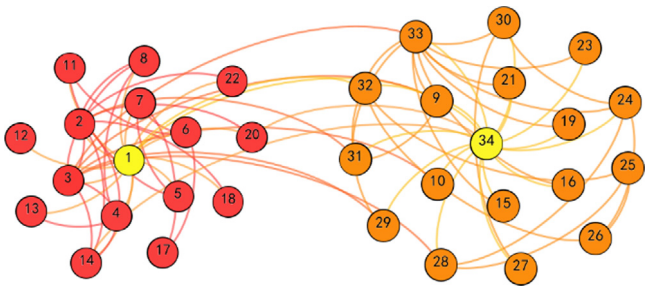

Fig. 6. Community structure of karate club (top-2 influential nodes). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

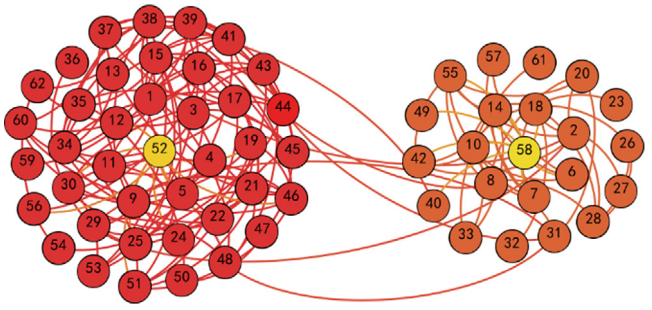

Fig. 7. Community structure of dolphin (top-2 influential nodes). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

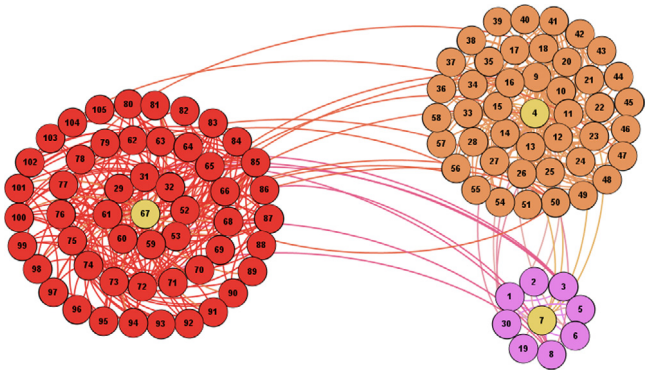

Fig. 8. Community structure of polbook (top-3 influential nodes). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 10
The NMI comparison of results on benchmark networks.

Benchmark	1	2	3	4	5	6
Louvain	0.957	0.908	**0.858**	0.802	0.682	0.625
LPA	**0.98**	**0.925**	0.83	**0.925**	0.683	0.587
Link	0.492	0.352	0.194	0.392	0.375	0.413
OCDLCE	0.789	0.667	0.697	0.665	0.675	0.712
OCDSSE	0.705	0.483	0.525	0.725	0.504	0.534
SECD	0.852	0.583	0.357	0.679	0.418	0.218
AOCCM	0.617	0.454	0.359	0.462	0.452	0.597
CAMAS	0.635	0.512	0.495	0.501	0.587	0.602
LGIEM	0.803	0.607	0.468	0.736	**0.689**	**0.725**

is easy to find its community structure. Combined these reasons above, simple network structure or obvious community structure may be the main reason why Louvain and LPA perform better in the first four networks. In addition, LGIEM performs stable in the last three networks, which have different mixture parameters. According to the analysis of Tables 10 and 11, LGIEM shows a better stability on networks with different mixture parameter and is suitable for larger networks.

Table 11
The ACC comparison of results on benchmark networks.

Benchmark	1	2	3	4	5	6
Louvain	0.976	**0.937**	**0.866**	**0.796**	0.691	0.655
LPA	**0.98**	0.912	0.841	0.792	0.683	0.657
Link	0.501	0.325	0.213	0.387	0.385	0.425
OCDLCE	0.854	0.672	0.683	0.695	0.691	0.751
OCDSSE	0.757	0.537	0.597	0.753	0.519	0.584
SECD	0.897	0.607	0.397	0.769	0.458	0.256
AOCCM	0.632	0.495	0.386	0.479	0.532	0.579
CAMAS	0.659	0.526	0.402	0.512	0.595	0.601
LGIEM	0.845	0.686	0.584	**0.796**	**0.715**	**0.783**

Table 12
The modularity comparison of results on real networks.

Networks	Karate Club	Dolphin	Polbook	Football	Email-Eu-core	PGP
Louvain	0.417	**0.516**	0.525	**0.605**	**0.289**	**0.856**
LPA	0.399	0.488	0.491	**0.605**	0.137	0.845
Link	0.027	0.149	0.203	0.083	0.127	0.388
OCDLCE	**0.482**	0.368	0.532	0.565	0.225	0.805
OCDSSE	0.312	0.377	0.516	0.41	0.07	0.735
SECD	0.399	0.401	0.659	0.595	0.137	0.773
AOCCM	0.38	0.437	0.562	0.372	0.179	0.67
CAMAS	0.375	0.396	0.528	0.401	0.172	0.655
LGIEM	0.375	0.491	**0.587**	0.572	0.292	0.791

Table 13
The NMI comparison of results on real networks.

Networks	Karate Club	Dolphin	Polbook	Football	Email-Eu-core	PGP
Louvain	0.636	0.599	0.522	0.634	0.46	0.732
LPA	0.337	0.692	0.482	0.549	0.239	0.745
Link	0.444	0.392	0.455	0.716	0.279	0.612
OCDLCE	0.612	0.487	0.479	0.725	0.327	0.701
OCDSSE	0.503	0.064	0.245	0.632	0.316	0.731
SECD	0.731	0.489	0.488	0.641	0.395	0.758
AOCCM	0.679	0.872	0.416	0.795	0.263	0.528
CAMAS	**1**	0.809	0.482	**0.809**	0.412	0.713
LGIEM	**1**	**0.890**	**0.56**	0.665	**0.540**	**0.812**

The Fig. 5 shows the running time of LGIEM on different datasets. The datasets are generated from the benchmark networks. Parameters of *mu* (mixture parameter), *maxk*, *minc* and *maxc* are fixed and set as 0.1, 15, 10 and 50 separately. The number of nodes is set within 1200. The Fig. 5 reflects time complexity indirectly. When the number of nodes is 1000, the running time of Link and AOCCM is reduced, mainly because of the network structure. From Fig. 5, the running time of LGIEM increases when the network increases. After nodes of network is more than 800, LGIEM is slower than other algorithms, which reflects the high time complexity of LGIEM. In addition, when nodes are more than 10 000, the algorithm executes very slowly and the time cannot be shown in the Fig. 5. Therefore, LGIEM should be parallelized so it can process large networks.

5.3.2. Results on real networks

The Figs. 6–9 show the communities detected by LGIEM in different networks. The nodes colored yellow are cores or top-k influential nodes of each community. The label of an influential node is set as the label of community. Nodes colored the same are belonging to the same community. In the Fig. 9, node 43 circled with red is different from other nodes because node 43 is an overlapping node which belongs to six communities (community 27, community 30, community 19, community 44, community 58, community 6).

Table 14
The ACC comparison of results on real networks.

Networks	Karate Club	Dolphin	Polbook	Football	Email-Eu-core	PGP
Louvain	0.971	**1**	0.637	0.58	0.459	0.772
LPA	0.325	0.783	0.765	0.607	0.237	0.75
Link	0.384	0.581	0.676	0.565	0.28	0.659
OCDLCE	0.882	0.785	0.796	0.572	0.369	0.724
OCDSSE	0.588	0.692	0.685	0.628	0.327	0.753
SECD	0.941	0.839	0.848	**0.678**	0.386	0.782
AOCCM	0.602	0.946	**0.886**	0.803	0.265	0.547
CAMAS	**1**	0.927	0.857	0.832	0.427	0.769
LGIEM	**1**	0.984	0.867	0.583	**0.476**	**0.886**

Table 15
Clustering metrics ranking for community detection algorithms applied on PGP.

Networks	NMI	ACC	TOPSIS
Louvain	4	3	3
LPA	3	6	4
Link	8	8	8
OCDLCE	7	7	7
OCDSSE	5	5	6
SECD	2	2	2
AOCCM	9	9	9
CAMAS	6	4	5
LGIEM	1	1	1

In order to reflect effectiveness and efficiency of LGIEM in detecting communities, it is compared with other similar methods in real networks.

In Table 12, we can see that Louvain performs better and LGIEM is second only to it. However, we cannot judge which algorithm is better because the result of modularity is too volatile in different dataset. In addition, modularity has resolution limit problem and can be affected by network structure, thus NMI and ACC can show the effectiveness of methods more accurately. As shown in Table 13, the community structure detected by our method is more similar with the real community structure of these networks. Table 14 shows the similar result of ACC comparison of networks, and the performance of LGIEM is better than other algorithms. AOCCM is better than LGIEM in polbook network and CAMAS is better in football network. However, AOCCM and CAMAS have the same drawback that all nodes of the network cannot be in full coverage. Therefore, LGIEM is more effective and accurate than them.

As shown in the three tables, Link is the worst in the real networks from the three aspects of modularity, NMI and ACC. Link divides a large community into several smaller communities, ignoring the connections among nodes within a large community. So link cannot detect reasonable communities. In the Email-Eu-core network, modularity is 0.292 which is smaller than 0.3. In other words, the network cannot form the community because its network structure is too sparse. So the value of NMI and ACC of the Email-Eu-core network is relatively small. Although LGIEM performs not better from the aspect of modularity, it performs better in most cases from NMI and ACC.

Due to the reason that NMI are not sensitive to the overlap in the community structure and overlapping can decrease the modularity, we add another metric based on the combination of quality metrics, clustering metrics and topological properties [41] to prove the superiority of LGIEM. We select clustering coefficient as topological property, modularity as the quality metric, while NMI and ACC as clustering metrics. In addition, TOPSIS is used to denote the final ranking. In this paper, we take the PGP dataset as an example to illustrate the results.

Table 15 shows the ranking of clustering metrics for PGP and the merged one using TOPSIS, which represents the final ranking. Our LGIEM method is the leading method for the merging strategy of TOPSIS. AOCCM ranks the 9th method, and is the worst method. Table 16 shows the quality and clustering metrics ranking for PGP. Although LGIEM in modularity ranks 4, it is the best algorithm according to the final rank. Louvain is second only to LGIEM and Link is the worst method. Table 17 shows the rank of all metrics and the final rank. In order to show the more accurate results, we considers as many metrics as possible and we can omit the measurement of individual strategies, such as Table 16. As shown in Table 17, LGIEM is still the leading algorithm and the Link is the worst algorithm.

Table 16
Quality and clustering metrics ranking for community detection algorithms applied on PGP.

Metrics	Quality metric	Clustering metrics		TOPSIS
	Modularity	NMI	ACC	
Louvain	1	4	3	2
LPA	2	3	6	3
Link	9	8	8	9
OCDLCE	3	7	7	5
OCDSSE	6	5	5	6
SECD	5	2	2	4
AOCCM	7	9	9	8
CAMAS	8	6	4	7
LGIEM	4	1	1	1

Table 17
Topological property, quality and clustering metrics ranking for community detection algorithms applied on PGP.

Metrics	Topological property	Quality metric	Clustering metrics		TOPSIS
	Clustering coefficient	Modularity	NMI	ACC	
Louvain	2	1	4	3	2
LPA	3	2	3	6	3
Link	9	9	8	8	9
OCDLCE	5	3	7	7	6
OCDSSE	6	6	5	5	5
SECD	4	5	2	2	4
AOCCM	8	7	9	9	8
CAMAS	7	8	6	4	7
LGIEM	1	4	1	1	1

Table 18
The final rank on real networks.

Networks	Karate Club	Dolphin	Polbook	Football	Email-Eu-core	PGP
Louvain	4	5	6	3	2	2
LPA	7	4	7	5	8	3
Link	8	9	9	9	7	9
OCDLCE	3	7	4	2	3	6
OCDSSE	6	8	8	8	9	5
SECD	2	6	2	1	5	4
AOCCM	5	2	5	7	6	8
CAMAS	1	3	3	6	4	7
LGIEM	1	1	1	4	1	1

Based on the principle, we show the final rank of all datasets based on TOPSIS, which can be seen in Table 18.

As shown in Table 18, we can see that LGIEM is the leading algorithm on almost datasets except for Football. In Football, LGIEM is inferior to SECD, OCDLCE and Louvain. It is likely because of network structure. Link can be considered as the worst algorithm, because its rank is not more than 7 and almost 9. Therefore, LGIEM can be considered as the best algorithm in the aspect of community detection.

To sum up, taking the analysis of Tables 10–14 and 18 into consideration, our method performs better, which can detect communities more effective.

5.3.3. NMI comparison on our method and other similar methods

In this section, we compute NMI between LGIEM and other algorithms and judge whether they cover similar partitions. As

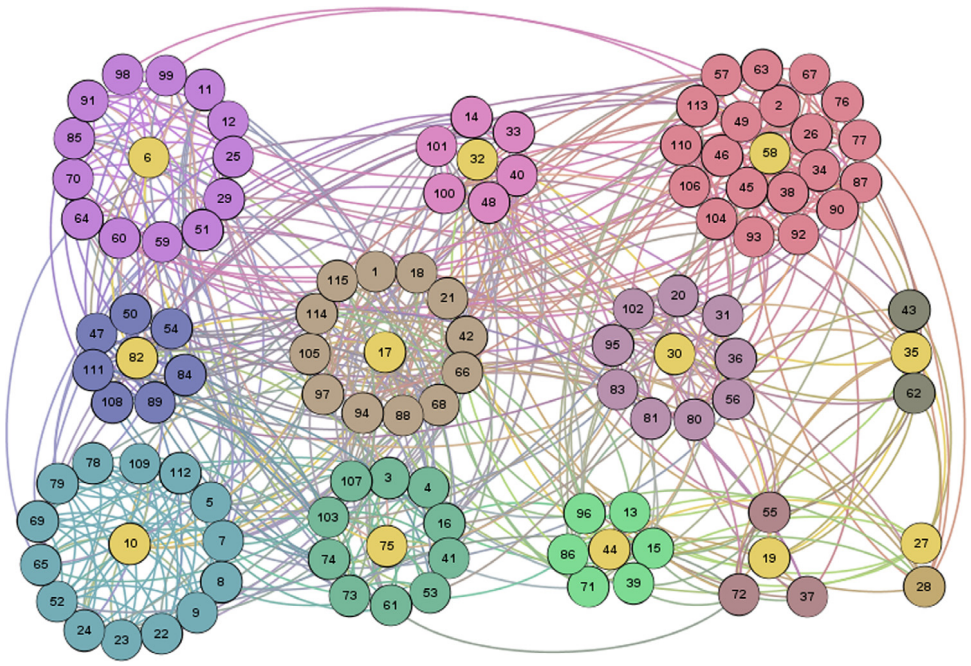

Fig. 9. Community structure of football (top-12 influential nodes). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 19
NMI Comparison on our method and other similar methods.

Networks	Karate Club	Dolphin	Polbook	Football	Email-Eu-core	PGP
Louvain	0.636	0.67	0.922	0.537	0.46	0.89
LPA	0.337	0.768	0.861	0.496	0.447	0.56
Link	0.444	0.337	0.812	0.669	0.367	0.72
OCDLCE	0.612	0.547	0.779	0.816	0.587	0.665
OCDSSE	0.503	0.023	0.094	0.89	0.595	0.703
SECD	0.731	0.487	0.861	0.827	0.639	0.758
AOCCM	0.679	0.957	0.743	0.833	0.286	0.582
CAMAS	1	0.91	0.861	0.822	0.602	0.625

seen in Table 19, we can see that the value NMI of LGIEM and other compared methods is above 0.5 mostly. In Dolphin and Polbook datasets, the value of NMI is less than 0.1 and the community structure of LGIEM is different from OCDSSE. In addition, the result is not stable in several datasets, which may be caused by instability of LPA. The value of NMI in Email-Eu-core is small, mainly because of sparse network structure. To sum up, LGIEM covers similar partition with other compared algorithms.

6. Conclusion

Identifying local communities in complex communities has attracted many research efforts in recent years. In this paper, we propose a new centrality measure LGI based on local and global information, and a community detection algorithm LGIEM based on seed expansion. It is vital to find top-k influential nodes as seeds due to their spreading ability of information in the network. Hence, there is an interesting to use LGI for finding influential nodes with high effectiveness. The proposed centrality LGI combines local information with global information of nodes. If a node has high influence, the node can be regarded as a seed or core node. In our method, these influential nodes and their neighbor constitute initial communities. In order to detect communities, an expansion strategy is proposed to assign candidate nodes into communities. The final communities are got by optimizing communities. Benchmark networks and real networks are used to test our algorithm against other similar algorithms. The experiments show that LGI is more accurate than the other centrality methods and can identify influential nodes efficiently. The results of community detection are better than other similar algorithms. Furthermore, our method LGIEM can detect communities of networks for full coverage. In terms of future research, we aim to improve the proposed method. In addition, we should focus on paralleling the method and reducing the time complexity for more complex networks.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by National Science Foundation of China (No. U1736105, No. 61572259, No. 41942017) and also supported by the National Social Science Foundation of China (No. 16ZDA054). The authors extend their appreciation to the Deanship of Scientific Research at King Saud University, Saudi Arabia for funding this work through research group No. RGP-264.

References

[1] T. Ma, S. Yu, J. Cao, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, A comparative study of subgraph matching isomorphic methods in social networks, IEEE Access 6 (2018) 66621–66631.
[2] Girvan, M.E.J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. USA 99 (12) (2002) 7821–7826.
[3] M.E.J. Newman, The structure of scientific collaboration networks, Proc. Natl. Acad. Sci. USA 98 (2) (2001) 404–409.
[4] M. Coscia, F. Giannotti, D. Pedreschi, A classification for community discovery methods in complex networks, Stat. Anal. Data Min. ASA Data Sci. J. 4 (5) (2011) 512–546.
[5] S. Fortunato, D. Hric, Community detection in networks: A user guide, Phys. Rep. 659 (2016) 1–44.

[6] T. Ma, J. Jia, Y. Xue, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, Protection of location privacy for moving kNN queries in social networks, Appl. Soft Comput. 66 (2018) 525–532.

[7] J. Xie, S. Kelley, B.K. Szymanski, Overlapping community detection in networks:the state-of-the-art and comparative study, ACM Comput. Surv. 45 (4) (2013) 1–35.

[8] T. Ma, W. Shao, Y. Hao, J. Cao, Graph classification based on graph set reconstruction and graph kernel feature reduction, Neurocomputing 296 (2018) 33–45.

[9] G. Qin, L. Gao, Spectral clustering for detecting protein complexes in protein & protein interaction (PPI) networks, Math. Comput. Modelling 52 (11/12) (2010) 2066–2074.

[10] R. Usha Nandini, A. Rka, K. Soundar, Near linear time algorithm to detect community structures in large-scale networks, Phys. Rev. E 76 (2) (2007) 036106.

[11] S. Gregory, Finding overlapping communities using disjoint community detection algorithms, 2009.

[12] J. Xie, B.K. Szymanski, X. Liu, Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process, in: IEEE International Conference on Data Mining Workshops, 2011.

[13] A. Clauset, Finding local community structure in networks, Phys. Rev. E 72 (2) (2005) 026132.

[14] I. F., T. V., G. Palla, I. Dere Nyi, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (2005) 814–818.

[15] S. Gao, J. Ma, Z. Chen, G. Wang, C. Xing, Ranking the spreading ability of nodes in complex networks based on local structure, Physica A 403 (6) (2014) 130–147.

[16] F.J.Y. Zhao, S. Li, Identification of influential nodes in social networks with community structure based on label propagation, Neurocomputing 210 (2016) 34–44.

[17] F. Xing, T. Ma, M. Tang, D. Guan, Friend circle identification in ego network based on hybrid method, IJAHUC 30 (4) (2019) 224–234.

[18] X. Zhang, J. Zhu, Q. Wang, H. Zhao, Identifying influential nodes in complex networks with community structure, Knowl.-Based Syst. 42 (2) (2013) 74–84.

[19] H. Rong, T. Ma, J. Cao, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, Deep rolling: A novel emotion prediction model for a multi-participant communication context, Inform. Sci. 488 (2019) 158–180.

[20] D. Kempe, J. Kleinberg, va Tardos, Maximizing the spread of influence through a social network, in: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2003, pp. 137–146.

[21] G. Sabidussi, The centrality index of a graph, Psychometrika 31 (4) (1966) 581–603.

[22] T. Ma, Y. Zhao, H. Zhou, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, Natural disaster topic extraction in sina microblogging based on graph analysis, Expert Syst. Appl. 115 (2019) 346–355.

[23] M. Barthlemy, Betweenness centrality in large complex networks, Eur. Phys. J. B 38 (2) (2004) 163–168.

[24] S. Brin, L. Page, Reprint of: The anatomy of a large-scale hypertextual web search engine, Comput. Netw. 56 (18) (2012) 3825–3833.

[25] J. Zhan, S. Gurung, S.P.K. Parsa, Identification of top-k nodes in large networks using katz centrality, J. Big Data 4 (1) (2017) 16.

[26] M. Kitsak, L.K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H.E. Stanley, H.A. Makse, Identification of influential spreaders in complex networks, Nat. Phys. 6 (11) (2011) 888–893.

[27] M. Riondato, E.M. Kornaropoulos, Fast approximation of betweenness centrality through sampling, Data Min. Knowl. Discov. 30 (2) (2014) 413–422.

[28] S.L. Luo, K. Gong, L. Kang, Identifying influential spreaders of epidemics on community networks, ArXiv preprint arXiv:1601.07700.

[29] Z. Ghalmane, M.E. Hassouni, C. Cherifi, H. Cherifi, Centrality in modular networks, EPJ Data Scienc. 8 (1) (2019) 15.

[30] T. You, B.C. Shia, Z.Y. Zhang, Community detection in complex networks using density-based clustering algorithm, Comput. Sci (2015).

[31] A. Bozorgi, H. Haghighi, M.S. Zahedi, M. Rezvani, Incim: A community-based algorithm for influence maximization problem under the linear threshold model, Inf. Process. Manage. 52 (6) (2016) 1188–1199.

[32] K. Zhou, A. Martin, Q. Pan, Z.G. Liu, Median evidential c-means algorithm and its application to community detection, Knowl.-Based Syst. 74 (1) (2015) 69–88.

[33] Y. Lv, T. Ma, M. Tang, J. Cao, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, An efficient and scalable density-based clustering algorithm for datasets with complex structures, Neurocomputing 171 (2016) 9–22.

[34] Y. Deng, Y. Liu, D. Zhou, An improved genetic algorithm with initial population strategy for symmetric TSP, Math. Probl. Eng. 2015 (3) (2015) 1–6, 2015, (2015-10-5).

[35] Z. Wang, C. Du, J. Fan, Y. Xing, Ranking influential nodes in social networks based on node position and neighborhood, Neurocomputing (2017).

[36] J.X. Yang, X.D. Zhang, K.A. Dawson, J.O. Indekeu, H.E. Stanley, C. Tsallis, Finding overlapping communities using seed set, Physica A 467 (2017) 96–106.

[37] H. Rong, T. Ma, M. Tang, J. Cao, A novel subgraph k^+ -isomorphism method in social network based on graph similarity detection, Soft Comput. 22 (8) (2018) 2583–2601.

[38] M. Hamann, E. Rhrs, D. Wagner, Local community detection based on small cliques, Algorithms 10 (3) (2017) 90.

[39] P. Deshpande, B. Ravindran, Mceil: An improved scoring function for overlapping community detection using seed expansion methods, in: IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, 2017, pp. 652–659.

[40] G.K. Orman, V. Labatut, H. Cherifi, Comparative evaluation of community detection algorithms: a topological approach, J. Stat. Mech. Theory Exp. 2012 (08) (2012) P08001.

[41] J. Malek, C. Hocine, C. Chantal, H. Atef, Community detection algorithm evaluation with ground-truth data, Physica A 492 (2017) 651–706.

[42] C.B.P.L. Dao, Vinh-Loc, Estimating the similarity of community detection methods based on cluster size distribution, in: Complex Networks and Their Applications, Vol. VII, Springer International Publishing, Cham, 2019, pp. 183–194.

[43] Y. Lin, J.C.S. Lui, Analyzing competitive influence maximization problems with partial information: An approximation algorithmic framework, Perform. Eval. 91 (2015) 187–204.

[44] T. Ma, H. Rong, C. Ying, Y. Tian, A. Al Dhelaan, M. Al Rodhaan, Detect structural? Connected communities based on bschef in c-dblp, Concurr. Comput. Pract. Exp. 28 (2) (2016) 311–330.

[45] Y.Y. Ahn, J.P. Bagrow, S. Lehmann, Link communities reveal multiscale complexity in networks, Nature 466 (7307) (2010) 761.

[46] Y. Xing, M. Fanrong, Z. Yong, Z. Ranran, Overlapping community detection by local community expansion, J. Inf. Sci. Eng. 31 (4) (2015) 1213–1232.

[47] B.K.B. Arjun Bhattacharya, Dipanjan Karmakar, Community detection using seed set expansion, 2016, https://www.slideshare.net/secret/IEqaSr72d0GO9A.

[48] M. Gong, J. Liu, L. Ma, Q. Cai, L. Jiao, Novel heuristic density-based method for community detection in networks, Physica A 403 (6) (2014) 71–84.

[49] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (4 Pt 2) (2008) 046110.

[50] W.W. Zachary, An information flow model for conflict and fission in small groups, J. Anthropol. Res. 33 (4) (1977) 452–473.

[51] D. Lusseau, M.E.J. Newman, Identifying the role that animals play in their social networks, Proc. R. Soc. B 271 (Suppl. 6) (2004) S477.

[52] V. Krebs, A network of co-purchased books about US politics, 2008, http://www.orgnet.com/.

[53] X. Tang, Network of American football games between division IA colleges during regular season fall 2000.

[54] J. Leskovec, A. Krevl, SNAP datasets: Stanford large network dataset collection, 2014, http://snap.stanford.edu/data.

[55] M. Bogu, R. Pastor-Satorras, A. Díaz-Guilera, A. Arenas, Models of social networks based on social distance attachment, Phys. Rev. E 70 (2) (2004) 056122.

[56] W. Webber, A. Moffat, J. Zobel, A similarity measure for indefinite rankings, ACM Trans. Inf. Syst. 28 (4) (2010) 1–38.

[57] N. Gupta, A. Singh, H. Cherifi, Centrality measures for networks with community structure, Physica A 452 (2016) 46–59.

[58] X. Liu, W. Wang, D. He, P. Jiao, D. Jin, C.V. Cannistraci, Semi-supervised community detection based on non-negative matrix factorization with node popularity, Inform. Sci. 381 (2017) 304–321.

[59] T. Ma, Y. Zhang, J. Cao, J. Shen, M. Tang, Y. Tian, A. Al-Dhelaan, M. Al-Rodhaan, KDVEM : a k-degree anonymity with vertex and edge modification algorithm, Computing 97 (12) (2015) 1165–1184.

[60] Z. Bu, Z. Wu, J. Cao, Y. Jiang, Local community mining on distributed and dynamic networks from a multiagent perspective, IEEE Trans. Cybern. 46 (4) (2016) 986–999.

[61] Z. Bu, G. Gao, H.J. Li, J. Cao, Camas: A cluster-aware multiagent system for attributed graph clustering, Inf. Fusion 37 (2017) 10–21.

Tinghuai Ma received his Bachelor (HUST, China, 1997), Master (HUST, China, 2000), PhD (Chinese Academy of Science, 2003) and was Post-doctoral associate (AJOU University, 2004) and a visiting Professor in Kyung Hee University, Korea (KHU, 2009). Now, he is a professor in Computer Sciences at Nanjing University of Information Science & Technology, China. His research interests are data mining, social network, privacy preserving, data sharing etc.

Qin Liu received her Bachelor degree in Computer Science & Technology from Nanjing University of Information Science & Technology, China in 2016. Currently, she is a candidate of Ph.D. in Nanjing University of Information Science & Technology. Her research interest is community detection.

Jie cao received his Ph.D. in Southeast University, Nanjing, China in 2005. He was an associate professor from 1999 to 2006. From 2006 to 2009, he was a Post-Doctoral Fellow at Academy of Mathematics and Systems Science, Chinese Academy of Science. From 2009, he is a professor in School of management and economic, Nanjing university of information science and technology. His research interests are system engineering, management science and technology.

Yuan Tian has received her master and Ph.D degree from KyungHee University and she is currently working as Associate Professor at College of Computer and Information Sciences, Nanjing Institute Saud University, Kingdom of Saudi Arabia. She is member of technical committees of several international conferences. In addition, she is an active reviewer of many international journals. Her research interests are broadly divided into privacy and security, which are related to cloud computing, bioinformatics, multimedia, cryptograph, smart environment, and big data.

Abdullah Al-Dhelaan, has received BS in Statistics (Hon) from King Saud University, on 1982, and the MS and Ph.D. in Computer Science from Oregon State University on 1986 and 1989 respectively. He is currently the Vice Dean for Academic Affairs, Deanship of Graduate Studies and a Professor of Computer Science, King Saud University, Riyadh, Saudi Arabia. He has guest edited several special issues for the Telecommunication Journal (Springer), and the International Journal for Computers and their applications (ISCA). Moreover, he is currently on the editorial boards of several journals and the organizing committees for several reputable international conferences. His current research interest includes: Mobile Ad Hoc Networks, Sensor Networks, Cognitive Networks, Network Security, Image Processing, and High Performance Computing.

Mznah Al-Rodhaan has received her BS in Computer Applications (Hon) and MS in Computer Science both from King Saud University on 1999 and 2003 respectively. In 2009, she received her Ph.D. in Computer Science from University of Glasgow in Scotland, UK. She is currently working as the Vice Chair of the Computer Science Department in College of Computer & Information Sciences, King Saud University, Riyadh, Saudi Arabia. Moreover, she has served in the editorial boards for some journals such as the Ad Hoc journal (Elsevier) and has participated in several international conferences. Her current research interest includes Mobile Ad Hoc Networks, Wireless Sensor Networks, Multimedia Sensor networks, Cognitive Networks, and Network Security.