Applied Intelligence (2020) 50:328–340
https://doi.org/10.1007/s10489-019-01541-1

A local community detection algorithm based on internal force between nodes

Kun Guo[1,2,3] · Ling He[1,2,3] · Yuzhong Chen[1,2,3] · Wenzhong Guo[1,2,3] · Jianning Zheng[4]

Published online: 24 July 2019

Abstract
Community structure is an important characteristic of complex networks. Uncovering communities in complex networks is currently a hot research topic in the field of network analysis. Local community detection algorithms based on seed-extension are widely used for addressing this problem because they excel in efficiency and effectiveness. Compared with global community detection methods, local methods can uncover communities without the integral structural information of complex networks. However, they still have quality and stability deficiencies in overlapping community detection. For this reason, a local community detection algorithm based on internal force between nodes is proposed. First, local degree central nodes and Jaccard coefficient are used to detect core members of communities as seeds in the network, thus guaranteeing that the selected seeds are central nodes of communities. Second, the node with maximum degree among seeds is pre-extended by the fitness function every time. Finally, the top k nodes with the best performance in pre-extension process are extended by the fitness function with internal force between nodes to obtain high-quality communities in the network. Experimental results on both real and artificial networks show that the proposed algorithm can uncover communities more accurately than all the comparison algorithms.

Keywords Complex network · Local community detection · Seed-extension algorithm · Internal force

1 Introduction

In the real world, complex systems exist in all aspects of people's lives, such as social networks, protein interaction networks and scientists' collaborative networks [1]. Complex systems are generally modeled as graphs or complex networks. A node in the network represents an individual, while edges represent connections between individuals. The existence of communities is a crucial characteristic of complex networks. The internal nodes of the community are connected closely, whereas the connections between communities are relatively sparse [2]. Uncovering communities in complex networks and mining the hidden relationships between communities are of utmost importance for complex network analysis.

The Girvan and Newman's method (GN) [3] is the first algorithm for uncovering communities in complex networks. In the last decade, numerous algorithms for uncovering communities in complex networks have been proposed [4]. These algorithms can be divided into non-overlapping community detection algorithms and overlapping community detection algorithms. Most of the early methods [5, 6] for community detection focus only on identifying non-overlapping communities in which each node belongs to a single community. However, nodes in the network usually belong to more than one community. For example, a person may be a member of his family and a personnel of his company. Palla et al. [7] proposed the first algorithm for

✉ Wenzhong Guo
guowenzhong@fzu.edu.cn

Kun Guo
gukn123@163.com

1 College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China

2 Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing, Fuzhou 350108, China

3 Key Laboratory of Spatial Data Mining and Information Sharing, Ministry of Education, Fuzhou 350108, China

4 Power Science and Technology Corporation State Grid Information and Telecommunication Group, Fuzhou 351008, China

uncovering overlapping communities in complex networks. Henceforth, numerous algorithms have been proposed for addressing the problem of uncovering overlapping communities in complex networks [8–13]. The algorithms are mainly based on local methods and global methods [14]. The global methods are not applicable to complex networks with large scale or lack of integrity because it requires the topology information of the entire network. Compared with the global methods, local methods can uncover communities without the integral structural information of the complex networks. However, the accuracy of local methods is generally affected by the initial node selection.

Seed-extension algorithms play an important role in existing overlapping community detection methods. In term of efficiency, the time complexity of seed-extension algorithms is linear to the number of nodes or edges when the network is sparse [15–17]. In term of effectiveness, algorithms based on seed-extension perform well in uncovering highly overlapping communities in many types of complex networks [18–20]. However, algorithms based on seed-extension still have weakness. They may select seeds with poor quality or cannot fully utilize the local information among nodes. In this paper, a local seed-extension algorithm based on internal force between nodes (InfoNode) is proposed. The main contributions of this paper are as follows:

(1) InfoNode uses local degree central nodes and Jaccard coefficient to detect core members in the network. The detected core members are treated as seeds, thus guaranteeing that the selected seeds are central nodes of the communities.
(2) In order to fully utilize the local information between nodes, the definition of internal force between nodes is proposed. InfoNode combines internal force between nodes with the fitness function in the community extension stage, which greatly improves the accuracy of community detection.
(3) A community pre-extension process is set up before the community extension stage. InfoNode selects the top k nodes with the best performance in the community pre-extension process and then calculate them accurately by fitness function with internal force between nodes in community extension stage, which can reduce the algorithm's running time.

The rest of the paper is organized as follows: In Section 2, we present the research related to seed-extension algorithms with various strategies in seed selection and community extension stage. Section 3 describes the proposed community detection algorithm in detail. In Section 4, a series of experimental results are given to verify the performance of our method. Finally, the conclusion is drawn in Section 5.

2 Related work

The algorithms based on seed-extension generally select a seed as the initial community and then extend it by continuously checking its neighboring nodes. The method include two essential stages: seed selection and community extension. In the stage of seed selection, its aim is to detect core members of communities based on node centrality index. In the stage of community extension, its goal is to build communities from seeds based on their influence or a greedy process with quality function. In the following, various strategies for selecting seeds and extending communities are introduced in detail, respectively.

2.1 Seed selection strategies

Increasing studies show that the formation of communities depends on core members [21, 22]. For algorithms based on seed-extension, the quality of seeds has direct affection on the algorithms' performance. In last decade, various seed selection strategies have been proposed [14]. Lancichinetti et al. [23] proposed a simple method that depends on random seed selection. The randomness brings efficiency but also makes it unable to discover high-quality communities. Lee et al. [18] proposed an algorithm that considers k-cliques in the network as initial communities so that it can uncover highly overlapping communities. However, the algorithm may ignore isolated sub-networks whose sizes are too small. In literature [24] and [25], the local degree central nodes whose degrees are greater than or equal to all their neighbors' degrees are selected as seeds. Intuitively, the core members of a community are generally local degree central nodes. The strategy can detect all the core members of communities. However, it may also select non-core members. Some algorithms [26, 27] calculate the conductance of each node in the network and select local minimum conductance nodes as seeds. This kind of method can detect high-quality seeds but with a low time efficiency.

2.2 Community extension strategies

In the stage of community extension, each seed is taken as an initial community and extended by spreading the influence of the seed throughout the network [26, 28] or

running a greedy process with a quality function [18, 23, 24]. The latter method attracts more interests of researchers. Hence, various quality functions have been proposed in recent years [29]. The quality function is optimized continuously until it gets the maximum or minimum value.

Clauset [30] proposed a local extension optimization algorithm that defines the local community modularity R, which is calculated as follows:

$$R = \frac{B_{in}}{B_{in} + B_{out}} \tag{1}$$

In (1), B is a local community. B_{in} represents the number of edges whose endpoints are all in B, while B_{out} is the number of edges those have one endpoint in B. The algorithm needs to pre-define the size of communities. It continuously adds the neighboring node that makes the largest increase of R to current community, until the current community reaches the predefined size.

Lou et al. [31] proposed the modularity M of community S, which is calculated as follows:

$$M = \frac{E_{in}}{E_{out}} \tag{2}$$

In (2), E_{in} represents the number of internal edges of community S, while E_{out} represents the number of edges between the community boundary and the external nodes. The algorithm proposes three heuristic node search methods to partially address the problem of uncovering communities in complex networks. However, it must set different thresholds for networks with different sizes.

Lancichinetti et al. [23] proposed a fitness function F_c to measure the tightness of internal nodes of the community. The fitness function is defined as follows:

$$F_c = \frac{f_{in}^c}{(f_{in}^c + f_{out}^c)^\alpha} \tag{3}$$

In (3), f_{in}^c and f_{out}^c are the total values of the internal degrees and external degrees of community c, and α is the resolution parameter used to control the size of communities detected. The quality function can effectively measure the tightness of nodes within the community, but it cannot fully utilize the local information among nodes.

3 The proposed community detection algorithm

This section is organized as follows: first, the basic notations and definitions used in the paper are presented in Section 3.1; second, Section 3.2 describes the proposed algorithm in detail; finally, the complexity analysis of our method is given in Section 3.3.

3.1 Basic notations and definitions

A network is usually modeled as $G = (V, E)$. V is the set of nodes and E is the set of edges in the network. In this paper, we only consider undirected and unweighted networks. The notations used in this paper are listed in Table 1 and basic definitions are given as follows.

Definition 1 (Jaccard coefficient) The Jaccard coefficient [32] between two nodes is defined as:

$$J(u, v) = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|} \tag{4}$$

Jaccard coefficient is defined to measure the similarity between two nodes. The larger the value, more similar the two nodes.

Definition 2 (Internal force) The internal force between two nodes is defined as:

$$F(u, v) = g \times \frac{d(u) \times d(v)}{[1 - J(u, v)]^2} \tag{5}$$

In (5), g is a parameter used to control the magnitude of internal force between two nodes, and we make its value as $1/d^2$. When internal force between nodes is calculated, a case may occur where the Jaccard coefficient between two nodes is 1, which means the two nodes are too "close".

Table 1 Formal notations used in this paper

Notation	Description
$G(V, E)$	a graph G with node set V and edge set E
V	a node set $V = \{v_1, v_2, v_3, \cdots, v_n\}$
E	an edge set $E = \{(v_i, v_j) \mid v_i \in V, v_j \in V, i \neq j\}$
n	the number of nodes in V
m	the number of edges in E
d	the average degree of nodes
q	the number of local degree central nodes
$d(v)$	the degree of node v
$N(v)$	the neighboring node set of node v
$Ns(C)$	the neighboring node set of community C

Therefore, the Jaccard coefficient formula needs to be slightly modified as follows:

$$J^{'}(u,v) = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)| + 1} \tag{6}$$

$$F^{'}(u,v) = g \times \frac{d(u) \times d(v)}{[1 - J^{'}(u,v)]^2} \tag{7}$$

Definition 3 (Fitness function with internal force) The fitness function with internal force is used to measure the tightness of a group of nodes. Its specific formula is defined as follows:

$$F_G = \frac{F_{in}^{'G}}{(F_{in}^{'G} + F_{out}^{'G})^{\alpha}} \tag{8}$$

In (8), $F_{in}^{'G}$ and $F_{out}^{'G}$ are the total value of the internal force and the external force of community G, respectively. The larger the value of fitness function of a group nodes, the more likely it is that they form a community.

Definition 4 (Node fitness to community) The fitness of node v to community c is used to determine whether the node should be added to the community. It is defined as follows:

$$F_c^v = F_{c \cup \{v\}} - F_c \tag{9}$$

In (9), if the value of F_c^v is positive, it indicates that the addition of node v to community c can make its structure more compact; on the contrary, it means that the addition of the node v to community c will make its internal structure looser.

3.2 Algorithm description

Figure 1 shows the flowchart of InfoNode that is mainly composed of two stages: seed selection and community extension. In the seed selection stage, local degree central nodes and Jaccard coefficient are used to detected core members of communities. In the community extension stage, InfoNode combines internal force between nodes with the fitness function to extend communities. In the following, the seed selection and community extension stage are described in detail, respectively.

3.2.1 Seed selection

For algorithms based on seed-extension, the quality of seeds directly affects the accuracy of communities detected. In general, the selected seeds should have a considerable "influence" in the local structure and be the core members of communities. The process of selecting seeds is given in Algorithm 1.

Algorithm 1 Seed selection.

Input: Graph $G(V, E)$, ε. //ε is the Jaccard coefficient threshold
Output: Seed set $Seeds$

1: $Seeds = \varnothing$;
2: **for** each $v \in V$ **do**
3: if v is a local degree central node, $Seeds = Seeds \cup \{v\}$;
4: **end for**
5: **for** each $v \in Seeds$ **do**
6: $J(v) = \{\sum J(u,v) | u \in N(v)\}$;
7: **end for**
8: $J_{max} = \max\{J(v) | v \in Seeds\}$;
9: **for** each $v \in Seeds$ **do**
10: $J^{'}(v) = J(v)/J_{max}$;
11: **if** $J^{'}(v) < \varepsilon$ **then**
12: $Seeds = Seeds - \{v\}$;
13: **end if**
14: **end for**

First, all local degree central nodes in the network are selected as candidate seeds (line 2-3). Obviously, these nodes have a considerable "influence" on their neighboring nodes. Second, in order to measure the "influence" of candidate seeds on their neighboring nodes, InfoNode calculates the sum of Jaccard coefficient of each candidate seed with its neighboring nodes (line 5-6). The larger the value, the greater "influence" the candidate seed has on its surrounding nodes. Finally, InfoNode normalizes the sum of Jaccard coefficient of each candidate seed and sets a threshold ε to select seeds (line 8-14). When the candidate seed is normalized to a value greater than ε, it is defined as a core member and used as the seed in community extension stage.

Figure 2 shows an example of selecting seeds in a toy network *ENZYMES_g50*[1]. The parameter ε in seed selection stage is defaulted by 0.5. In Fig. 3, we use the well-studied *Karate* network [33] to verify the performance of our seed selection method. The network has two communities, one of which is led by a class instructor (node 0), and the other is led by a club administrator (node 33). Figure 3 shows that our seed selection method can accurately find the seeds in the network.

3.2.2 Community extension

After obtaining seeds in the network, InfoNode extends them to detect communities. The process of community extension is given in Algorithm 2.

[1] http://networkrepository.com/index.php

Algorithm 2 Community extension.

Input: Graph $G(V, E, F)$, $Seeds$, k. // F represents internal force between nodes.
Output: community set C.

1: $C = \varnothing$;
2: $V_{unextended} = V$;
3: **while** $V_{unextended} \neq \varnothing$ **do**
4: current community $Cs = \varnothing$;
5: **if** $Seeds = \varnothing$ **then**
6: $v_{seed} = \{\max(d(v))|v \in Seeds\}$
7: $Seeds = Seeds - \{v_{seed}\}$;
8: **else**
9: select a node from $V_{unextended}$ randomly as the seed v_{seed};
10: **end if**
11: $Cs = Cs \cup \{v_{seed}\}$;
12: **while** $Ns(Cs) \neq \varnothing$ **do**
13: **for** each v in $Ns(Cs)$ **do**
14: F_{CS}^{v} is calculated by (3) and (9);
15: **end for**
16: the top k nodes with the best performance in pre-extension process are put into set Kn;
17: **for** each v in Kn **do**
18: F_{Cs}^{v} is calculated by (8) and (9);
19: **end for**
20: $fitness_{max} = \max(F_{Cs}^{v}|v \in Kn)$;
21: **if** $fitness_{max} > 0$ **then**
22: $Cs = Cs \cup \{v_{max}\}$; // v_{max} represents the corresponding node
23: **else**
24: break;
25: **end if**
26: update $Ns(Cs)$;
27: **end while**
28: $V_{unextended} = V_{unextended} - Cs$;
29: $C = C \cup Cs$;
30: **end while**

According to Algorithm 2, the specific community extension steps can be summarized as follows:

(1) The seeds obtained in the seed selection stage are sorted via degree in non-ascending order. When the seed set is non-empty, we select the seed with maximum degree to be extended; otherwise, we randomly select a node that have not been extended as the seed (line 5-10). If all nodes in the network are extended, the community extension stage ends.
(2) The value of fitness function of each neighboring node to the current community is calculated and the top k nodes those can maximize the value of fitness function of current community are selected (line 13-16).
(3) The top k nodes with the best performance in step (2) are accurately calculated by the fitness function with internal force between nodes. The node that can maximize the fitness value of the current community is selected(line 17-20).
(4) If the selected node increases the value of fitness function of current community, it will be added into the current community. We update neighboring nodes of the current community and the process will then return to step (2); otherwise, the current community is a final divided community and the process will return to step (1) (line 21-26).

Figure 4 shows the example of extending communities in the network. Considering the small size of the network, we take the value of k as 3. The community extension stage repeats this process until all nodes in the network are extended. In Fig. 5, we extend the seeds obtained in the seed selection stage in *Karate* network. The result shows that our algorithm accurately divides the network into two communities, and each node is divided into the right community.

3.3 Algorithm complexity analysis

In seed selection stage, the time complexity of determining whether one node is a local degree central node is $O(nd)$ and the time complexity of selecting seeds is $O(q^2d)$. Before community extension stage, we should calculate the internal force between nodes and the time complexity of this step is $O(nd)$. For algorithms based on seed-extension with a quality function, the time complexity is almost linear to the number of nodes or edges when communities are small, but the worst-case complexity is $O(n^2)$ [23]. In summary, the time complexity of InfoNode is $O(nd)$ in a general complex network. However, when the size of communities is close to n, the time complexity of InfoNode is $O(n^2)$.

For the space complexity of InfoNode, all local degree central nodes in the network need to be stored in seed selection stage and the required space is $O(q + n)$. In community extension stage, the Jaccard coefficient between the nodes must be stored and the required storage space is $O(nd + m)$. Therefore, the total space complexity of the InfoNode algorithm is $O(q + n + nd + m)$ which can be reduced to $O(m)$.

4 Experimental results and analysis

This section is organized as follows: first, the description of real and artificial dataset is given in Section 4.1; second, we introduce the experimental settings and evaluation metrics in Section 4.2; finally, experimental results and analysis are given in Section 4.3.

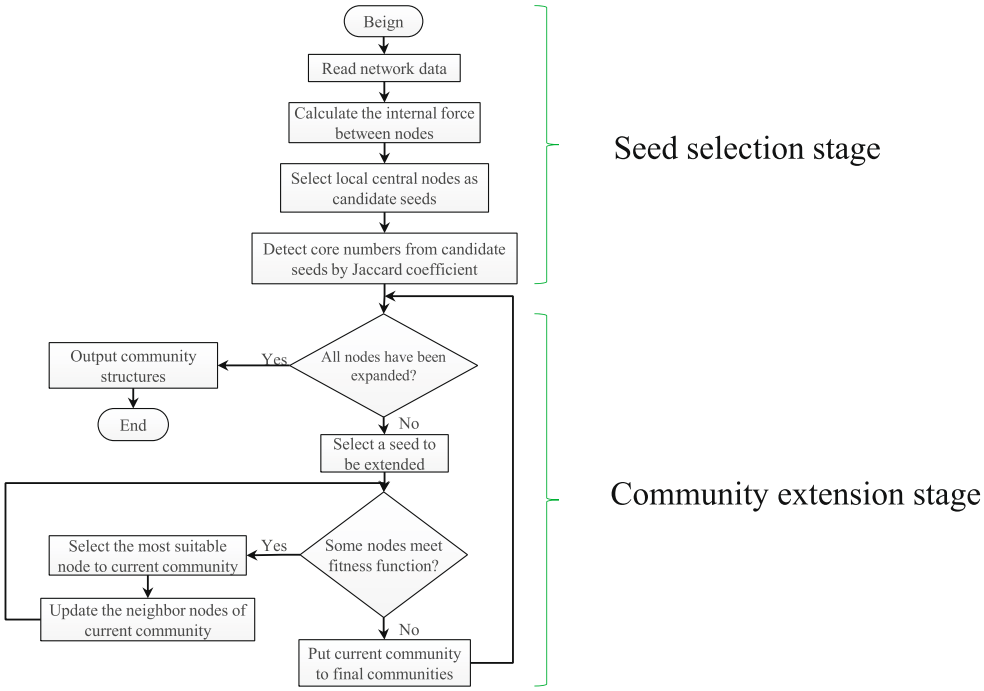


Fig. 1 The flowchart of InfoNode

4.1 Dataset description

(1) Real datasets. The real datasets used in the experiments are all well-known and widely used in community detection field. The real networks are extracted from various domains with different scales and degree distributions, which can not only verify the performance of community detection algorithms, but also challenge them in the terms of robustness and scalability. The specific information of real datasets is shown in Table 2.

(2) Artificial datasets. We use the LFR-benchmark [40] to generate artificial datasets. The network generated by this program can well control the distribution of nodes' degree and communities' size. Four groups of artificial dataset are generated by the program. The basic parameters are set as follows:

(1) $D1$: N=5000, μ=0.1-0.7, on=0.1, om=3;
(2) $D2$: N=5000, μ=0.3, on=0.1,0.3, om=2-8;
(3) $D3$: N=5000, μ=0.3, on=0.1-0.6, om=3;

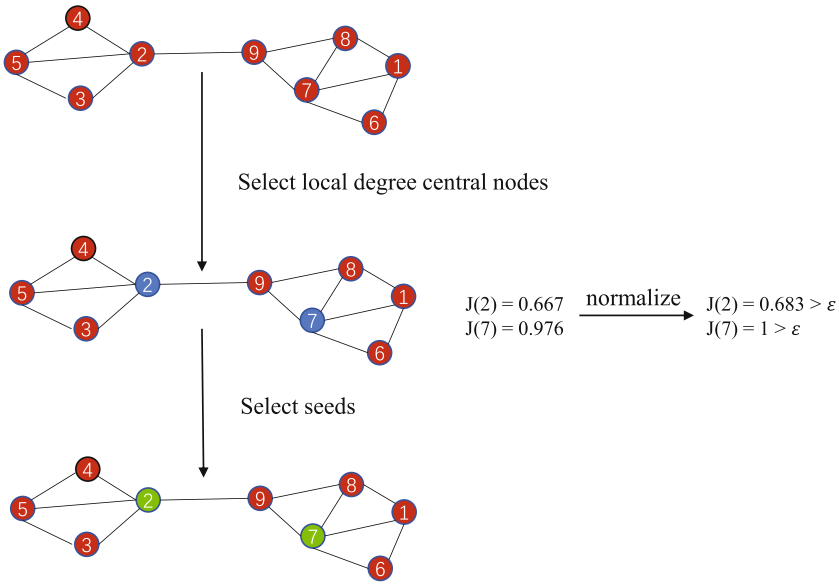


Fig. 2 Example showing the process of seed selection

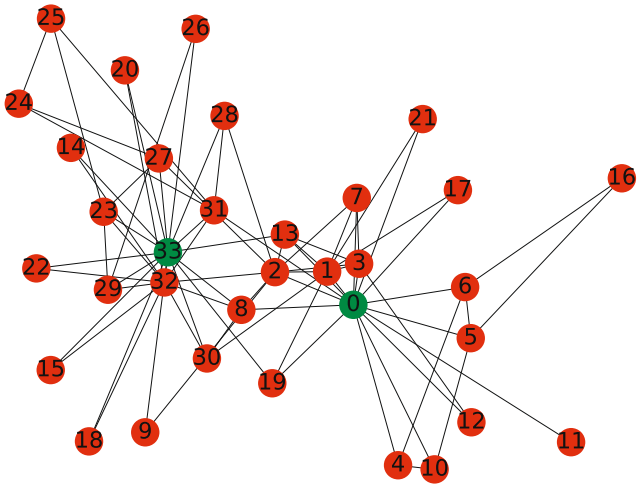


Fig. 3 The selected seeds in Karate network

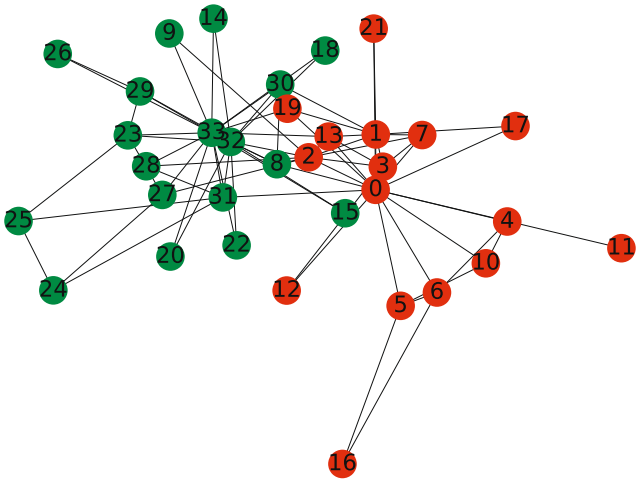


Fig. 5 Communities in the Karate network

(4) $D4$: N=2000-10000, μ=0.3, on=0.3, om=3.

The rest of the parameters are set by default: $kmax$=50, $minc$=20, $maxc$=100. The specific description of the parameters in artificial datasets is shown in Table 3. In these experiments, we takes on the ratio of the number of overlapping nodes to the total number of nodes in the network.

4.2 Experimental settings and evaluation metrics

4.2.1 Experimental settings

In the experiments, InfoNode is compared with three state-of-the-art approaches and three algorithms based on seed-extension.

The three state-of-the-art approaches are Attractor [41], Ego-Splitting [42] and MUTILCOM [43]. Attractor regards the network as a dynamic system and proposes three intuitive interaction modes to dynamically discover communities by simulating the changing distance between nodes in the network. Ego-splitting is a framework for detecting communities by leveraging local structures to de-couple overlapping communities. MUTILCOM a method for detecting multiple local communities those may overlaps. The algorithm expands seeds those are selected by embedding local networks around the initial seed set into low dimensional space.

In addition, to show the advantages of seed selection and community extension strategies used in our method, we compared InfoNode with three algorithms based on seed-extension: LFM [23], DEMON [44], and LMD [24].

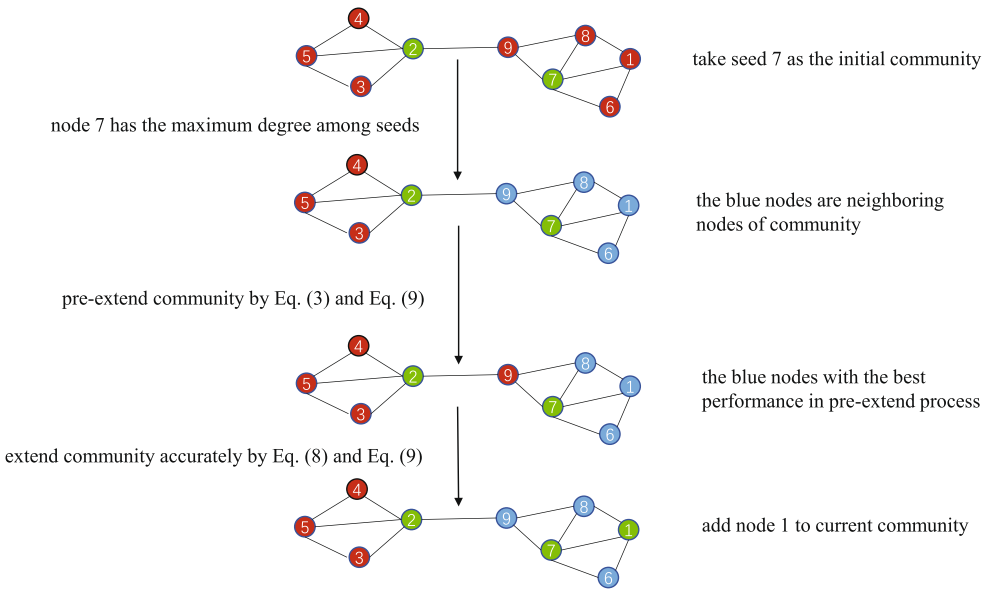


Fig. 4 Example showing the process of community extension

Table 2 Description of real datasets

Dataset	Number of nodes	Number of edges	Average degree	Communities
karate[a] [33]	34	78	4.59	2
dolphins[a] [34]	62	159	5.13	2
polbooks[a] [35]	105	441	8.4	3
football[a] [3]	115	616	10.66	13
SFI[b]	118	200	3.3898	unknown
jazz[a] [36]	198	2742	27.7	unknown
08blocks[b]	300	584	3.8933	unknown
662_bus[a]	662	906	4.737	unknown
polblogs[a] [37]	1490	16715	22.44	unknown
tech-routers[b]	2114	6632	6.27733	unknown
power[a] [38]	4941	6594	2.67	unknown
CA-GrQc[a] [39]	14845	121251	16.12	unknown

[a] http://www-personal.umich.edu/mejn/netdata/
[b] http://networkrepository.com/index.php

LFM randomly selects seeds in the network and extends communities by a greedy process with fitness function. Demon considers all nodes in the network as seeds and merges communities based on ego-networks and a community consolidation strategy. LMD selects the local degree central nodes as seeds and extends communities around the seeds.

The experimental contents include three parts: algorithms' parameter experiments, algorithms' accuracy experiments on real and artificial datasets and the scalability experiments of InfoNode.

4.2.2 Evaluation metrics

We chose two widely used evaluation metrics to verify our method: the extension of modularity (EQ) [45] and the Normalized Mutual Information (NMI) [46].

The specific calculation formula of the extended modularity is defined as follows:

$$EQ = \frac{1}{2m}\sum_{i}\sum_{v\in c_i, w\in c_i}\frac{1}{O_V O_W}(A_{vw} - \frac{k_v k_w}{2m})\delta(c_v, c_w) \quad (10)$$

Table 3 Parameter description of artificial datasets

Parameter	Description
N	number of network nodes
$kmax$	maximum degree of nodes
$minc$	minimum number of community nodes
$maxk$	maximum number of community nodes
on	number of overlapping nodes
om	number of communities the nodes belong to
μ	community mix parameter

In (10), O_v is the number of communities to which the node v belongs, A is the adjacency matrix and k_v is the degree of node v. When node v and node w is connected, the value of δ(v,w) is 1; otherwise, it is 0. The closer to 1 the value of EQ, the better quality of com7munity detection.

NMI uses information entropy to measure the difference between communities detected and the ground-truth communities. The larger the value of NMI, the better quality of community detection. The specific calculation formula is defined as follows:

$$NMI = \frac{-2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} C_{ij}\log\frac{C_{ij}N}{C_i C_j}}{\sum_{i=1}^{C_A} C_i \log\frac{C_i}{N} + \sum_{j=1}^{C_B} C_{ij}\log\frac{C_j}{N}} \quad (11)$$

In (11), $C_A(C_B)$ is the number of communities detected and ground-truth, respectively. $C_i(C_j)$ is the sum of elements of community C in row i (column j) and N represents the number of nodes in the network.

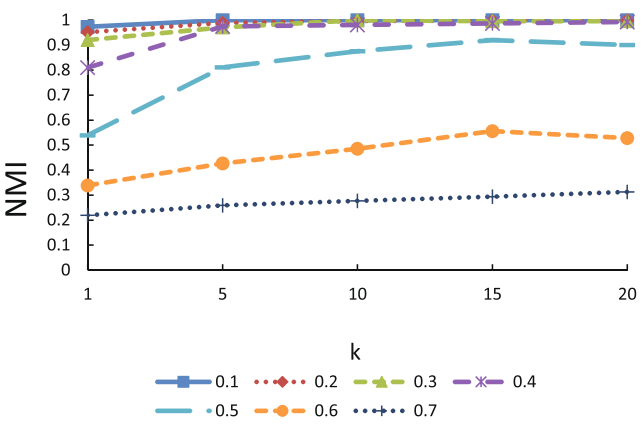


Fig. 6 Experiments on parameter k

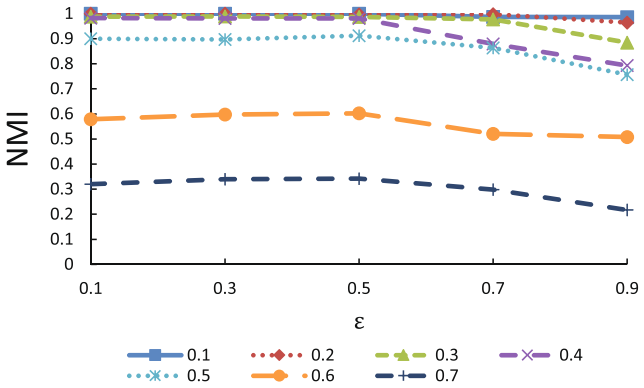


Fig. 7 Experiments on parameter ε

4.3 Experimental results

4.3.1 Experimental results on algorithms' parameters

There are three parameters used in InfoNode: α, k, and ε. Parameter α is used to control the scale of communities detected. When the value of α is large, the size of communities detected is small; vice versa. It is the same parameter α in LFM. Therefore, it is set to 0.8-1.2 as suggested in [23].

(1) *Experiments on parameter k*

Parameter k is used to select the top k nodes those have the best performance in the pre-expansion process. Obviously, the larger the value of k, the better quality of community detection, but the longer running time of InfoNode. Therefore, we should choose a appropriate value of k.

Figure 6 shows the experimental results on parameter k on the D1 dataset. As seen in the figure, as the value of k increases, the NMI increases. When the community mixing parameter μ is less than 0.5 and the value of k is 5, the NMI can reach the maximum value because the community structure in the network is clear and easy to be identified at this time. When the community mixing parameter μ reaches 0.5 or higher, the value of NMI does not increase until k reaches approximately 15 because the community structure in th network is fuzzy at this time and more nodes should be calculated by fitness function with internal force between nodes. Therefore, parameter k is suitable when its value is 5 in the network with a low community mix. It is more appropriate to take the value of k as 15 in the network with a high community mix.

(2) *Experiments on parameter ε*

The parameter ε is used to select seeds in the network. Obviously, when ε is taken as 0, all local degree central nodes in the network are selected as seeds. When ε is taken as 1, seeds are randomly selected from the nodes those are not extended in the network.

Figure 7 shows the experimental results on parameter ε on the D1 dataset. When the community mixing parameter μ is less than 0.4, the value of NMI decreases until ε is approximately 0.7 because the community structure in the network is relatively clear at this time, and the quality of seeds selected has little effect on the accuracy of communities detected. When the community mixing parameter μ reaches 0.4 or higher, the downward trend of the value of NMI occurs until ε is approximately 0.5. The quality of seeds selected has a considerable impact on the accuracy of the communities detected because the community structure in the network is relatively vague at this time. Obviously, when the value of ε is small, there are many seeds selected and the subsequent community extension stage will perform some unnecessary operations.

Table 4 EQ results on real datasets

Network	Attractor	Ego-Splitting	MUTILCOM	DEMON	LMD	LFM	InfoNode
karate	**0.3715**	0.2295	0.0486	0.1559	0.3564	0.3321	**0.3715**
dolphins	0.2744	0.2038	0.2668	0.2805	0.4553	0.4534	**0.4814**
polbooks	**0.5121**	0.4186	0.4304	0.4177	0.4635	0.4239	0.4815
football	**0.6005**	0.1393	0.4279	0.3732	0.5586	0.5104	0.5704
SFI	0.5056	0.3995	0.4704	0.3805	0.5933	0.5422	**0.6631**
jazz	**0.3839**	0.3148	0.1044	0.0508	0.2713	0.2125	0.2782
08blocks	0.7942	**0.8802**	0.4087	**0.8802**	**0.8802**	0.4152	**0.8802**
662_bus	0.7622	0.8061	0.7391	0.6019	0.7561	0.6749	**0.8268**
polblog	0.8543	0.7707	0.6983	0.6401	0.8114	0.7982	**0.8954**
tech-routers	0.3637	0.1953	0.2301	0.1508	0.3809	0.2873	**0.4278**
power	0.5323	0.1904	0.5012	0.4845	0.5089	0.4858	**0.6356**
CA-GrQc	0.6134	0.5245	0.5873	0.4296	0.6276	0.5816	**0.6716**

Bold data signify that the algorithm performs well in various real datasets which not only verifies the performance of our method, but also challenges it in the terms of robustness and scalability

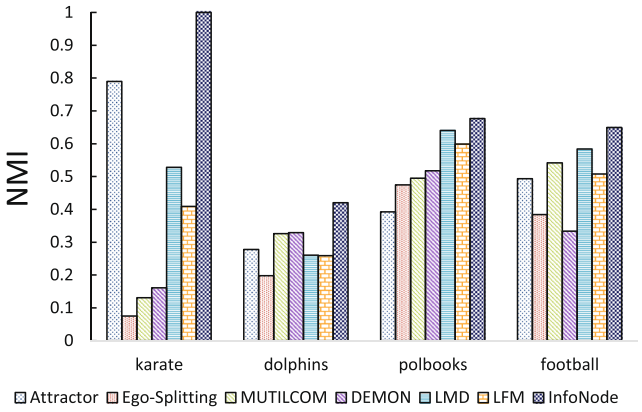


Fig. 8 NMI results on real datasets

Therefore, the suitable value of parameter ε in InfoNode is about 0.5.

4.3.2 Experimental results on real datasets

Table 4 shows the experimental results of EQ on real datasets. The accuracy of InfoNode is only slightly lower than that of Attractor on *polbooks*, *football* and *jazz*, because the average degree of nodes in the networks is quite high. In addition, links in the networks are relatively compact and more suitable for the algorithms based on distance dynamics among nodes such as Attractor. In the other nine real datasets, InfoNode has the highest EQ result owing to our new seed selection method and the fitness function with internal force in community extension stage.

On the other hand, Fig. 8 shows the NMI results on real networks whose ground-truth communities are known. It can be seen from the figure that our algorithm can detect communities more accurately than all the other algorithms on the four networks. Compared with other three algorithms based on seed-extension, the strategy in seed selection of InfoNode can obtain high-quality seeds in the network and InfoNode combines internal force with the fitness function for fully utilizing the local information between nodes.

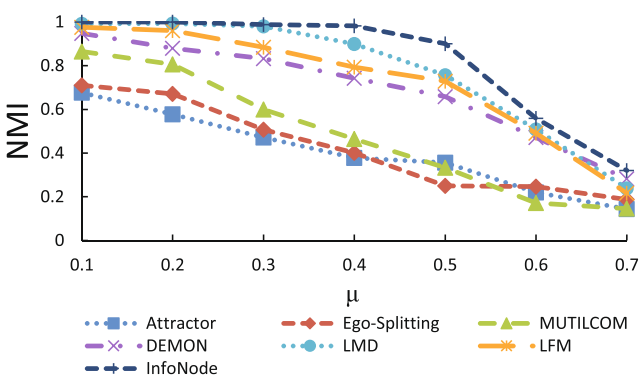


Fig. 9 NMI results on artificial datasets

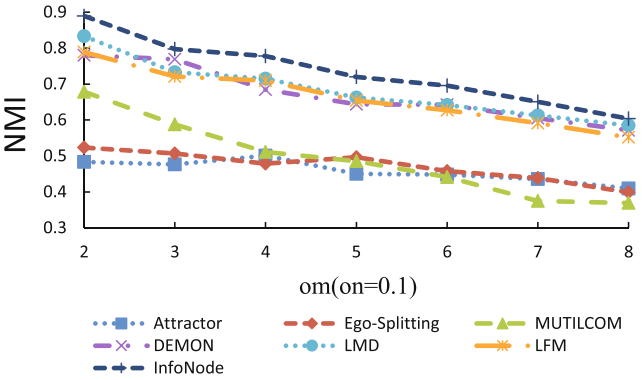


Fig. 10 NMI results on parameter *om*(on=0.1)

4.3.3 Experimental results on artificial datasets

Figure 9 shows the experimental results of different algorithms' accuracy on group D1 artificial simulation networks. The experimental results reveal that the accuracy of InfoNode is higher than that of other algorithms in all values of μ. Attractor algorithm is based on changing distance between nodes in the network. When communities in the network are not obvious, it is difficult for Attractor to simulate the changing distance between nodes. Ego-Splitting algorithm is based on the ego-nets in the network and it performs well when the value of μ is small. However, it is harder for Ego-Splitting to find ego-nets in the network as the value of μ increases. MUTILCOM algorithm selects seeds by embedding local networks around the initial seed set into low dimensional space. The quality of seeds selected by the method decreases as the value of μ increases. LFM randomly selects nodes as seeds and does not fully utilize local information between nodes. DEMON needs to fuse local communities to form optimal global communities. It tends to form multiple independent communities with the increase of the value of μ. The accuracy of community detection is then reduced. LMD performs well when the value of μ is small. The number of local degree central

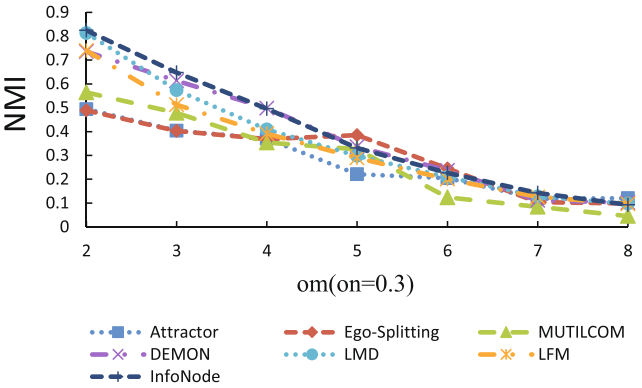


Fig. 11 NMI results on parameter *om*(on=0.3)

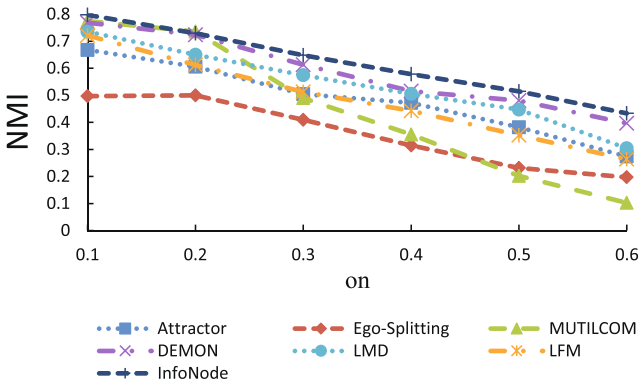


Fig. 12 NMI results on parameter *on*

nodes increases when the increase of the value of μ. LMD selects all local degree central nodes as seeds so that the quality of the seeds is reduced. In addition, LMD cannot fully utilize the local information between nodes. In conclusion, the strategies used in InfoNode can improve the accuracy of community detection.

Figures 10 and 11 show the experimental results of each algorithm on group D2 artificial networks. The experimental results indicate that as the value of *om* increases, the accuracy of each algorithm will decrease. This outcome is attributed to the artificial simulation network becoming complicated and more difficult for each algorithm to uncover communities. Overall, the accuracy of InfoNode is better than that of other algorithms because of the strategies in seed selection and community extension stage in our method.

Figure 12 shows the experimental results of each algorithm on group D3 artificial networks. The experimental results reveal that as the value of *on* increases, that is, as the number of overlapping nodes in the network increases, the accuracy of each algorithm will decrease, but InfoNode remains superior to the comparison algorithms. The accuracy of DEMON is only second to that of InfoNode, because DEMON forms local communities based on label propagation which is suitable for networks with many overlapping nodes.

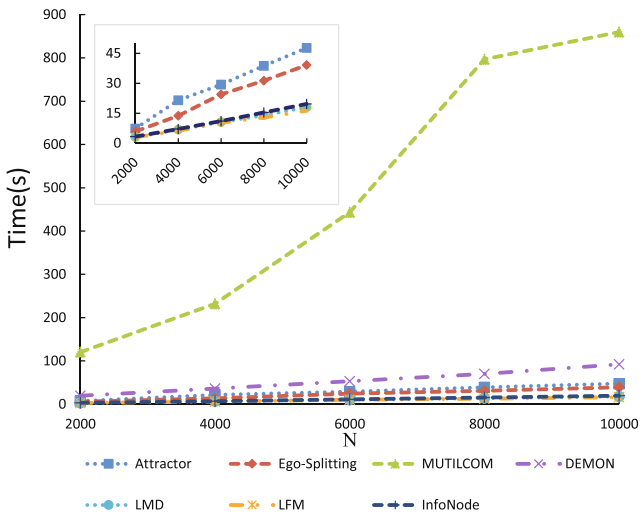


Fig. 13 Algorithms' running time(s) experimental results

4.3.4 Scalability experimental results

Figure 13 shows the running time of InfoNode and other comparison algorithms on group D4 artificial networks. The time efficiency of InfoNode is slightly lower than that of LFM and LMD. LFM randomly selects seeds and does not take advantages of internal force between nodes, so it is more efficient than other algorithms based on seed-extension. LMD is slightly more efficient than InfoNode because it does not utilize the local information between nodes. DEMON needs to extend all nodes in the network and optimize communities detected, so it has a low time efficiency. The time complexity of Attractor is linear to the number of edges in the network, so it is slightly slower that other three algorithms based on seed-extension. Ego-Splitting algorithm can easily detect all the ego-nets in the networks whose structure is not too complicated. However, its time efficiency is still inferior to LFM, LMD and InfoNode. The time complexity of MUTILCOM is high because it selects the initial seeds by embedding the local networks around the seeds into low dimensional space.

5 Conclusions

This study proposes a local community detection algorithm based on internal force between nodes, which can accurately and effectively uncover communities in the network. First, we select seeds through the local degree central nodes and Jaccard coefficient in the network. Second, the node with the maximum degree among seeds adopts the fitness function strategy for pre-extension every time. Finally, the top k nodes with the best performance in the pre-expansion process are extended by the fitness function with internal force between nodes. As experimental results show on the real and artificial datasets, our method can accurately and effectively uncover communities in complex networks.

Future work will compare InfoNode with other state-of-the-art methods. The parallelization based on MapReduce model will be considered to improve the time efficiency of InfoNode. At the same time, dynamic strategies will be introduced to adapt to uncover communities in dynamic networks, thus increasing the practicality of our method.

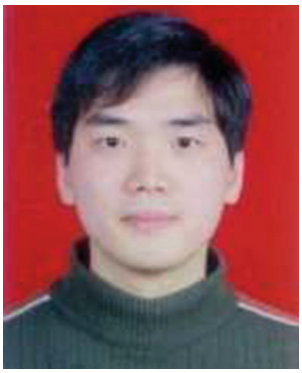
Acknowledgements This work is partly supported by the National Natural Science Foundation of China under Grant No. 61300104, No. 61300103 and No. 61672159, the Fujian Province High School

Science Fund for Distinguished Young Scholars under Grant No. JA12016, the Fujian Natural Science Funds for Distinguished Young Scholar under Grant No. 2015J06014, the Fujian Industry-Academy Cooperation Project under Grant No. 2018H6010 and No. 2017H6008, and Haixi Government Big Data Application Cooperative Innovation Center.

References

1. Newman ME (2001) The structure of scientific collaboration networks. Proc Natl Acad Sci 98(2):404–409
2. Newman ME (2004) Detecting community structure in networks. Eur Phys J B 38(2):321–330
3. Girvan M, Newman ME (2002) Community structure in social and biological networks. Proc Natl Acad Sci 99(12):7821–7826
4. Fortunato S, Hric D (2016) Community detection in networks: a user guide. Phys Rep 659(2016):1–44
5. Rosvall M, Bergstrom CT (2008) Maps of random walks on complex networks reveal community structure. Proc Natl Acad Sci 105(4):1118–1123
6. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech: Theory Exper 2008(10):P10008
7. Palla G, Derényi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. Nature 435(7043):814
8. Sattari M, Zamanifar K (2018) A spreading activation-based label propagation algorithm for overlapping community detection in dynamic social networks. Data Knowl Eng 113:155–170
9. Le BD, Shen H, Nguyen H, Falkner N (2018) Improved network community detection using meta-heuristic based label propagation. Appl Intell 49(4):1451–1466
10. Biswas A, Biswas B (2017) Analyzing evolutionary optimization and community detection algorithms using regression line dominance. Inf Sci 396:185–201
11. Fan H, Zhong Y, Zeng G (2018) Overlapping community detection based on discrete biogeography optimization. Appl Intell 48(5):1314–1326
12. Liu W, Yue K, Wu H, Fu X, Huang W (2017) Markov-network based latent link analysis for community detection in social behavioral interactions. Appl Intell 48(5915):1–16
13. Wen X, Chen WN, Lin Y, Gu T, Zhang J (2016) A maximal clique based multiobjective evolutionary algorithm for overlapping community detection. IEEE Trans Evol Comput PP(99):1–1
14. Fortunato S (2009) Community detection in graphs. Phys Rep 486(3):75–174
15. Whang JJ, Gleich DF, Dhillon IS (2016) Overlapping community detection using neighborhood-inflated seed expansion. IEEE Trans Knowl Data Eng 28(5):1272–1284
16. Bai L, Cheng X, Liang J, Guo Y (2017) Fast graph clustering with a new description model for community detection. Inf Sci 388:37–47
17. Li HJ, Bu Z, Li A, Liu Z, Shi Y (2016) Fast and accurate mining the community structure: integrating center locating and membership optimization. IEEE Trans Knowl Data Eng 28(9):2349–2362
18. Lee C, Reid F, McDaid A, Hurley N (2010) Detecting highly overlapping community structure by greedy clique expansion. arXiv:1002.1827
19. Liakos P, Ntoulas A, Delis A (2016) Scalable link community detection: A local dispersion-aware approach. In 2016 IEEE International Conference on Big Data (Big Data). IEEE, pp 716–725
20. Kanawati R (2015) Empirical evaluation of applying ensemble methods to ego-centred community identification in complex networks. Neurocomputing 150:417–427
21. Bai X, Yang P, Shi X (2017) An overlapping community detection algorithm based on density peaks. Neurocomputing 226:7–15
22. Zhi-Xiao W, Ze-chao L, Xiao-fang D, Jin-hui T (2016) Overlapping community detection based on node location analysis. Knowl-Based Syst 105:225–235
23. Lancichinetti A, Fortunato S, Kertesz J (2009) Detecting the overlapping and hierarchical community structure in complex networks. J Phys 11(3):033015
24. Chen Q, Wu TT, Fang M (2013) Detecting local community structures in complex networks based on local degree central nodes. Physica A: Stat Mech Appl 392(3):529–537
25. Tabarzad MA, Hamzeh A (2017) A heuristic local community detection method (HLCD). Appl Intell 46(1):62–78
26. Hu Y, Yang B, Wong HS (2016) A weighted local view method based on observation over ground truth for community detection. Inf Sci 355:37–57
27. Fan X, Chen Z, Cai F, Wu J, Liu S, Liao Z, Liao Z (2018) Local core members aided community structure detection. Mob Netw Appl 2017(2):1–9
28. Yao Y, Wu W, Lei M, Zhang X (2016) Community detection based on variable vertex influence. In 2016 IEEE First International Conference on Data Science in Cyberspace (DSC). IEEE, pp 418–423
29. Zhang J, Ding X, Yang J (2019) Revealing the role of node similarity and community merging in community detection. Knowl-Based Syst 165:407–419
30. Clauset A (2005) Finding local community structure in networks. Phys Rev E 72(2):026132
31. Luo F, Wang JZ, Promislow E (2008) Exploring local community structures in large networks. Web Intell Agent Syst: Int J 6(4):387–400
32. Jaccard P (1901) ÉTude comparative de la distribution florale dans une portion des Alpes et des Jura. Bull Soc Vaudoise Sci Nat 37:547–579
33. Zachary WW (1977) An information flow model for conflict and fission in small groups. J Anthropol Res 33(4):452–473
34. Lusseau D, Schneider K, Boisseau OJ, Haase P, Slooten E, Dawson SM (2003) The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations. Behav Ecol Sociobiol 54(4):396–405
35. Newman ME, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E 69(2):026113
36. Gleiser PM, Danon L (2003) Community structure in jazz. Adv Compl Syst 6(04):565–573
37. Adamic LA, Glance N (2005) The political blogosphere and the 2004 US election: divided they blog. Inproceedings of the 3rd international workshop on Link discovery. ACM, pp 36–43
38. Watts DJ (1998) Collective dynamics of 'small-world'networks. Nature 393(6684):440–442
39. Leskovec J, Kleinberg J, Faloutsos C (2007) Graph evolution: Densification and shrinking diameters. ACM Trans Knowl Discov Data (TKDD) 1(1):2
40. Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. Phys Rev E 78(4):046110
41. Shao J, Han Z, Yang Q, Zhou T (2015) Community detection based on distance dynamics. Inproceedings of the 21th ACM

SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp 1075–1084
42. Epasto A, Lattanzi S, Paes Leme R (2017) Ego-Splitting Framework: from Non-Overlapping to Overlapping Clusters. Inproceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp 145–154
43. Hollocou A, Bonald T, Lelarge M (2018) Multiple local community detection. ACM SIGMETRICS Perform Eval Rev 45(3):76–83
44. Coscia M, Rossetti G, Giannotti F, Pedreschi D (2012) Demon: a local-first discovery method for overlapping communities. Inproceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 615–623
45. Shen H, Cheng X, Cai K, Hu MB (2009) Detect overlapping and hierarchical community structure in networks. Physica A: Stat Mech Appl 388(8):1706–1712
46. Danon L, Diaz-Guilera A, Duch J, Arenas A (2005) Comparing community structure identification. J Stat Mech: Theory Exper 2005(09):P09008



Kun Guo is currently an Associate Professor with the College of Mathematics and Computer Science, Fuzhou University. He is also a member of the China Computer Federation (CCF) and the Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing. His research interests include data mining, grey system theory, and distributed parallel computation.

Ling He received the B.S. degree from the Collage of Mathematics and Computer Science, Fuzhou University, Fuzhou, China, in 2018. He is currently pursuing the M.S. degree in the Collage of Mathematics and Computer Science, Fuzhou University, Fuzhou, China. His research interest includes community detection.

Yuzhong Chen received the B.S. degree in computer engineering and the Ph.D. degree in information and communication engineering from the University of Science and Technology of China, Hefei, China, in 2000 and 2005, respectively. He is currently a Full Professor with the College of Mathematics and Computer Science, Fuzhou University. He is also a member of the CCF Young Computer Scientists & Engineers Forum and the Deputy Director of the Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing. His research interests include network information security, data mining, and social network analysis.

Wenzhong Guo received the B.S. and M.S. degrees in computer science and the Ph.D. degree in communication and information system from Fuzhou University, Fuzhou, China, in 2000, 2003, and 2010, respectively, where he is currently a Full Professor with the College of Mathematics and Computer Science. He currently leads the Fujian Provincial Key Laboratory of Network Computing and Intelligent Information Processing. His research interests include intelligent information processing, sensor networks, network computing, and network performance evaluation. He is also a member of ACM and a Senior Member of the China Computer Federation (CCF).

Jianning Zheng received the B.S. degree from the Collage of Mathematics and Computer Science, Fuzhou University, Fuzhou, China, in 2003. He is currently the Vice-Chief Engineer with the Power Science and Technology Corporation State Grid Information and Telecommunication Group, Fuzhou. His research interest includes comprehensive energy services.