Weighted compactness function based label propagation algorithm for community detection

Weitong Zhang, Rui Zhang, Ronghua Shang, Licheng Jiao

PII: S0378-4371(17)31078-6
DOI: https://doi.org/10.1016/j.physa.2017.11.006
Reference: PHYSA 18774

To appear in: *Physica A*

Received date : 27 June 2017
Revised date : 6 November 2017

Please cite this article as: W. Zhang, R. Zhang, R. Shang, L. Jiao, Weighted compactness function based label propagation algorithm for community detection, *Physica A* (2017), https://doi.org/10.1016/j.physa.2017.11.006

Highlights

This paper presents a label propagation based on an weighted compactness function for large-scale networks.

We search the sets of core nodes which have great influence on the network.

We assign weights to the nodes according to the similarity of the nodes between the core nodes sets and the nodes degree.

We propose a compactness function as the objective function.

We adopt the adjustment strategy to correct the result of network partition.

Weighted compactness function based label propagation algorithm for community detection

Weitong Zhang[a], Rui Zhang[b], Ronghua Shang[a], Licheng Jiao[a]

([a] Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi’an, China, 710071. [b] College of Information Science and Technology, Agricultural University of Hebei, Baoding Hebei 071000. 13833231366@163.com)

Abstract: Community detection in complex networks, is to detect the community structure with the internal structure relatively compact and the external structure relatively sparse, according to the topological relationship among nodes in the network. In this paper, we propose a compactness function which combines the weight of nodes, and use it as the objective function to carry out the node label propagation. Firstly, according to the node degree, we find the sets of core nodes which have great influence on the network. The more the connections between the core nodes and the other nodes are, the larger the amount of the information these kernel nodes receive and transform. Then, according to the similarity of the nodes between the core nodes sets and the nodes degree, we assign weights to the nodes in the network. So the label of the nodes with great influence will be the priority in the label propagation process, which effectively improves the accuracy of the label propagation. The compactness function between nodes and communities in this paper is based on the nodes influence. It combines the connections between nodes and communities with the degree of the node belongs to its neighbor communities based on calculating the node weight. The function effectively uses the information of nodes and connections in the network. The experimental results show that the proposed algorithm can achieve good results in the artificial network and large-scale real networks compared with the 8 contrast algorithms.

Keywords: Community detection; compactness function; node weight; label propagation.

1. Introduction

The structure of real networks is often abstracted as complex network model, such as the Internet, power network, chain network and social network [1]. We can deeply understand the evolution process and the function of the systems by studying the complex network model [2]. The community structure in the network, that is, the set of nodes with relatively tight internal connections and the relatively sparse external connections [3,4]. Similar to the problem of graph clustering, community detection is to find the local structure closely connected in the network [5]. Today, the amount of information is growing rapidly, and the problem of community detection in large-scale network needs to be solved urgently [6].

In recent years, more and more community detection algorithms have been proposed and improved. They can be roughly divided into: split method, merge method, heuristic optimization method, network dynamics based method and so on according to the division way. Kernighan-Lin algorithm [7] defines a revenue function of the network partition. It uses greedy search to get the maximum value of the income function of the network. The spectral bisection method [8,9] is a method of Laplace matrix characteristic value of the network, which divides the network into two parts based on the feature vector corresponding to the second smallest eigenvalue of the Laplace matrix of the network. Markov clustering algorithm [10] is a graph clustering method based on random walk process [11]. GN algorithm [2] is a method for decomposing a network by removing the edges. The Fast Newman algorithm [12] is a greedy algorithm which directly optimizes the modularity function [1] value of by the combination of nodes. Merge-split algorithm [13] proposed by Mei et al. is a kind of strategy based on single node mobility. The performance of the algorithm depends on the perturbation probability p_r. Simulated annealing algorithm (SA) [14] is a kind of stochastic global heuristic optimization method. The optimal value of the objective function can be obtained by searching the state space. Extremal optimization (EO) algorithm [15] is a heuristic optimization algorithm too. Duch and Arenas [16] put forward decomposing the modularity function into a local variable represented by a single node as the fitness function. This method randomly divided the network into two parts with equal size, and selected the node with lower degree. Then move the chosen node from one community to another community, to search for the optimal modularity function value. Tabu search algorithm [17] is often used to optimize the value of modularity function. Genetic algorithm (GA) [18] introduced the concept of community score, and the optimal network partition is obtained by maximizing the community score. The Meme-Net algorithm [19] used the modularity density [20] as the optimization function, and the hill-climbing method was for the local search strategy. The communities in the network is detected by adjusting the parameters λ of the objective function. MOEA/D-Net [21] algorithm is a kind of multi-objective evolutionary algorithm. This algorithm divided the modularity density into two parts: the ratio association and the ratio cut as the objective functions at the same time. The multi-objective optimization problem is decomposed into a series of scalar optimization problems. The MODPSO algorithm [22] used the decomposition mechanism to solve the clustering problem, and transformed them into a series of scalar problems. The update rules and representation of the particles has been redefined in the discrete case. The diversity between these sub problems of leaded to the diversity of the population. GDPSO algorithm [23] also introduced the decomposition concept in the particle swarm optimization algorithm. At the same time, the algorithm proposed a greedy strategy to assign particles to appropriate regions. Infomap algorithm [11] is a weighted directional method for community detection. Its detection result is a map which simplified and projected the regularity of community structure and its relationship.

Bagrow [24] studied the label propagation in the network in his L-layer method for the first time. Raghavan et al. [25] proposed strategy using label propagation for network community detection. This algorithm is easy to implement, good classification effect and the time complexity is linear, especially for community detection in large-scale network. But the traditional label propagation algorithm has strong randomness and weak robustness. LPAm algorithm [26] changed the node label update strategy to optimize the local modularity function, so that each node label each iteration update is always moving in the direction of increasing the local modularity. The final optimization results of the algorithm corresponded to the optimization of modularity function. But this method is very easy to fall into local optimum. In order to improving the shortcomings of the LPAm algorithm that easily falling into the local optimum, Liu [27] added a multi step greedy fusion strategy similar to the MSG algorithm [28] on the basis of the LPAm algorithm, so that the final optimization results were improved. At the same time, it also increases the complexity of the algorithm. In 2014, Lin et al. proposed a community kernel based label propagation algorithm (CKLPA) [29], which improves the randomness of the original label propagation algorithm and reduced the complexity of the algorithm. However, the number of community kernel in the algorithm needs to be given in advance, so that the detection results of the network may be random.

In this paper, we propose a node weight based label propagation strategy for large-scale complex network community detection. First of all, according to the node degree, we find the set of core nodes with more influence in the network. Then, according to the similarity of nodes and the core nodes as well as the node degree, the weights of nodes in the network are given. Finally, we propose a weighted compactness function of nodes and communities as the objective function for label propagation strategy. The function combines the connections between nodes and communities with the degree of the node belongs to its neighbor communities. This makes full use of the nodes and edges information in the network. In this paper, the algorithm can ensure the priority of the influence of the label in the label propagation process, and improve the accuracy of network partition.

2. Related works

In this section, we will introduce several relevant definitions that are used in the paper, mainly including: label propagation algorithm, core node and node similarity function.

2.1 The label propagation algorithm

Label propagation algorithm, that is, labels propagate between nodes. When the labels no longer change, all node labels represent the network partition results. We can understand the label propagation process through the spread of virus. A and B are friends (in the network can be understood as the node A and the node B has a connection between them). If A is infected with the virus, he may infect his friend B. If C is a friend of B, C and A do not know each other (in the network can be understood as the node C and node A is not connected). Then C may be infected by B. Namely, A, B, C may be infected with the virus (in the network can be understood as they belong to the same community). The label propagation algorithm is used to update the node label to the largest number of tags in the neighbor, which has high efficiency, but low accuracy. The classic label propagation algorithm is based on the label number of nodes of the neighbor labels, and the label update formula is as follows [25]:

$$l(i) = \arg\max_k \sum_{j \in N(i)} \sigma(l(j), k) \tag{1}$$

where *i*, *j* are nodes in the network, *N*(*i*) is a set of neighbors of the node *i*, *l*(*i*) is the label of the node *i*, *l*(*j*) is the label of the node *j*, *k* is the new label of the node *i*, and when *a*=*b*, σ(*a*, *b*) =1, when *a*≠*b*, σ(*a*, *b*) =0.

In this paper, we propose a function that represents the degree of closeness between the nodes and the communities based on the node weights. We use it as the objective function to carry out label propagation on the networks.

2.2 The core nodes

The center of a complex network is often with high importance and affects the function of the entire network [30-31]. Many times the network center is not only one. Each community may have a different center. The community center is usually connected with other nodes more closely. The amount of information received and transmitted by the community center is larger. We call such a community center as the core node for short [29]. There are two main types of core nodes. One is “destructive decision critical” [32]. That is, if removing or altering a node can change the structure or reliability of the network, then the node is the core node of the network. The other is “significant equals to the key” [33]. The centrality of the node is judged by analyzing the centrality degree index of the network structure. The node centrality measure index is mainly divided into the following several kinds [34]: the centrality degree index based on the degree of closeness, the centrality index based on the edge betweenness and the centrality index based on the node degree and so on. considering the complexity of the algorithm, this paper uses the node degree as a measure of the node centrality index [35] and choose the core nodes according to the node degree. Moreover, this paper will not only find the core nodes of communities in the network, but also to detect the possible set of core nodes of communities through the core nodes neighbors. This can further collect and utilize the information of the community center to prevent the wrong partition.

2.3 The node similarity function

The node degree of the core nodes in the network are different. So, the influence of the core nodes on their neighbors may be different. Also the different core nodes may have the same neighbor nodes. Therefore, we can calculate the similarity of core nodes and their neighbor nodes to give weight values for the nodes in the network. Thus we can highlight the familiarization between nodes and core nodes (i.e. potential community center) in order to improve the detection accuracy of the community detection. There are many kinds of similarity measurement functions. Here are several precise similarity functions: the RA [36] index, the Hub promoted index [37], the Leicht-Holme-Newman index [38] and the cosine similarity [39]. In this paper, the RA index is used as the node similarity calculation function. Since it is calculated according to the amount of resources transferred between nodes.

3. Weighted compactness function based label propagation algorithm

In this paper, we propose a node weight based label propagation strategy for community detection in large-scale complex network. By detecting the similarity of the nodes and the core nodes, as well as the node degree and other information, we can use the network topology information more efficiently.

3.1 The set of the core nodes

As described in the section 2.2, the core nodes often carry and transmit more information as the center of the communities. Therefore, by calculating the similarity of nodes and core nodes, we can detect communities in network more accurately. However, the information carried by a single core node may not be complete enough. So we use the nodes with higher node degree in the neighborhood of the core nodes as the members in the set of the core nodes, which can be used to compute the similarity between the nodes and the core nodes set.

In this paper, we use the centrality measure index based on node degree to find the kernel node. The node with the highest degree in its neighborhood is the first node we need. Obviously, each core node is not connected to each other. Then, take each node as the center, and find the nodes with higher node degree in its neighbors as the members of the core node set to complete the search of the set of core nodes.

The specific algorithm for finding the core nodes set is as follows:

```
Algorithm 1: Finding the core nodes set.
Input: The nodes number in the network $n$, node connection information $Edge$;
Output: The set of core nodes in the network $K$;
Step1: Calculate the node degree of all nodes in the network according to the network node connection information $Edge$;
Step2: Taking the nodes with the highest degree in their neighborhood in the network as the core nodes. They are the
       potential community center;
Step3: Taking each node in the Step2 as the potential center of communities. Then add the nodes with higher node degree in
       their neighborhood to the set of core nodes $K$;
Step4: Output the set of core nodes in the network $K$.
```

3.2 Assign weights for nodes according to the node similarity and node degree

Most of the community detection methods are based on the node and node connection information of the network, so that the network topology information is not fully utilized. The proposed algorithm assigns weights for nodes in network according to the node similarity and node degree. This will further strengthen the influence of nodes, and highlight the nodes that have high ability carrying and disseminate information of nodes. Thereby the detection accuracy of the community is improved. In the 3.1 section, we have got a set of core nodes with more influence in the network. This section describes a method for assigning weights to nodes in the network.

The node weight W is composed of two parts. The first part of w_1 is obtained by calculating the similarity between the nodes and the core set members. If the node is a member of the core set, then w_1=2. Otherwise, calculate the similarity value between each node and each member of the core set, and select the largest value as w_1. In this paper, the node similarity value is calculated by RA exponent [36]. This index calculates the similarity between nodes for the amount of information transmitted between nodes, and makes use of the connection information between nodes effectively. RA index formula is as follows:

$$F = \sum_{a \in N(\mathrm{i}) \cap N(j)} \frac{1}{d(a)} \tag{2}$$

where F is the similarity between node i and node j, where i and j are nodes in the network, $N(i)$ is a set of neighbors of the node i, $N(j)$ is a set of neighbors of the node j, a is a common neighbor of both node i and node j, $d(a)$ is the node degree of node a.

The second part w_2 of the node weight is given through the node degree, w_2= (node degree / the maximum node degree in the network). Thus obtain the formula W=w_1+w_2, which is the final node weight.

We take a small scale network Karate network [40] for example to demonstrate the case assigning weights for nodes. The Karate network consists of 34 nodes (representing members of the club) and 78 sides (representing the social relationships between the members). The Karate network node connection is shown in Fig. 1:

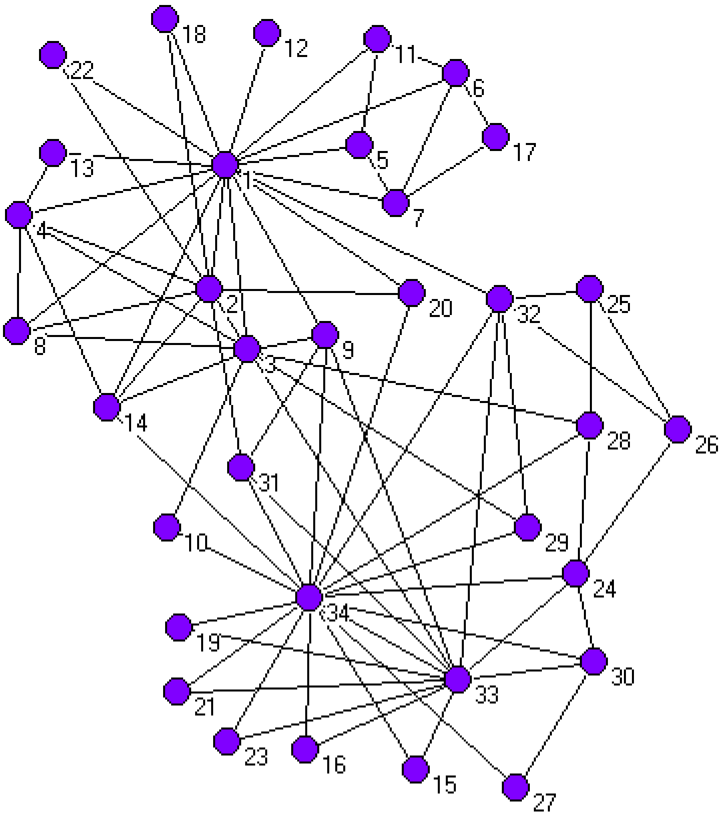


Fig. 1 Node connection condition of the Karate network

According to the method introduced in this section for the node weights in the network, we can get the weights of the nodes in the Karate network, as shown in Table 1.

Table 1 The weights of nodes in the Karate network

node	1	2	3	4	5	6	7	8	9
weight	2.94	2.57	2.15	1.51	0.75	0.81	0.81	0.61	0.62
node	10	11	12	13	14	15	16	17	18
weight	0.21	0.75	0.05	0.28	0.67	0.20	0.20	0.61	0.22
node	19	20	21	22	23	24	25	26	27
weight	0.20	0.28	0.20	0.22	0.20	0.87	0.59	0.54	0.36
node	28	29	30	31	32	33	34	-	-
weight	0.43	0.44	1.01	0.54	0.76	4.27	3	-	-

As can be seen from Table 1, the weight of node 1, 2, 3, 33, 34 is higher than other nodes. As can be seen in Fig. 1, these 5 nodes above have more connections with other nodes. That is, their node degree is higher, and the influence on other nodes is greater. In this way, we can ensure that the node with higher degree is with higher weight. That is to say, this can ensure the purpose of this paper: highlighting the nodes with higher influence in the network.

3.3 Weighted compactness function

The traditional label propagation algorithm is simple and easy to implement, and the classification effect is good. Its time complexity is also linear, especially suitable for large-scale network community detection. However, the traditional label propagation algorithm has strong randomness and weak robustness. Our algorithm inherits the low complexity of the label propagation algorithm, and improves the condition of the process of label propagation. According to the content of the section 3.2, this paper proposes a weighted compactness function based in the node weights of the nodes and their neighbor community labels. The weighted compactness function formula is as follows:

$$F_c = \frac{l(i,c) + w(\mathrm{i},c)}{d(i)} \tag{3}$$

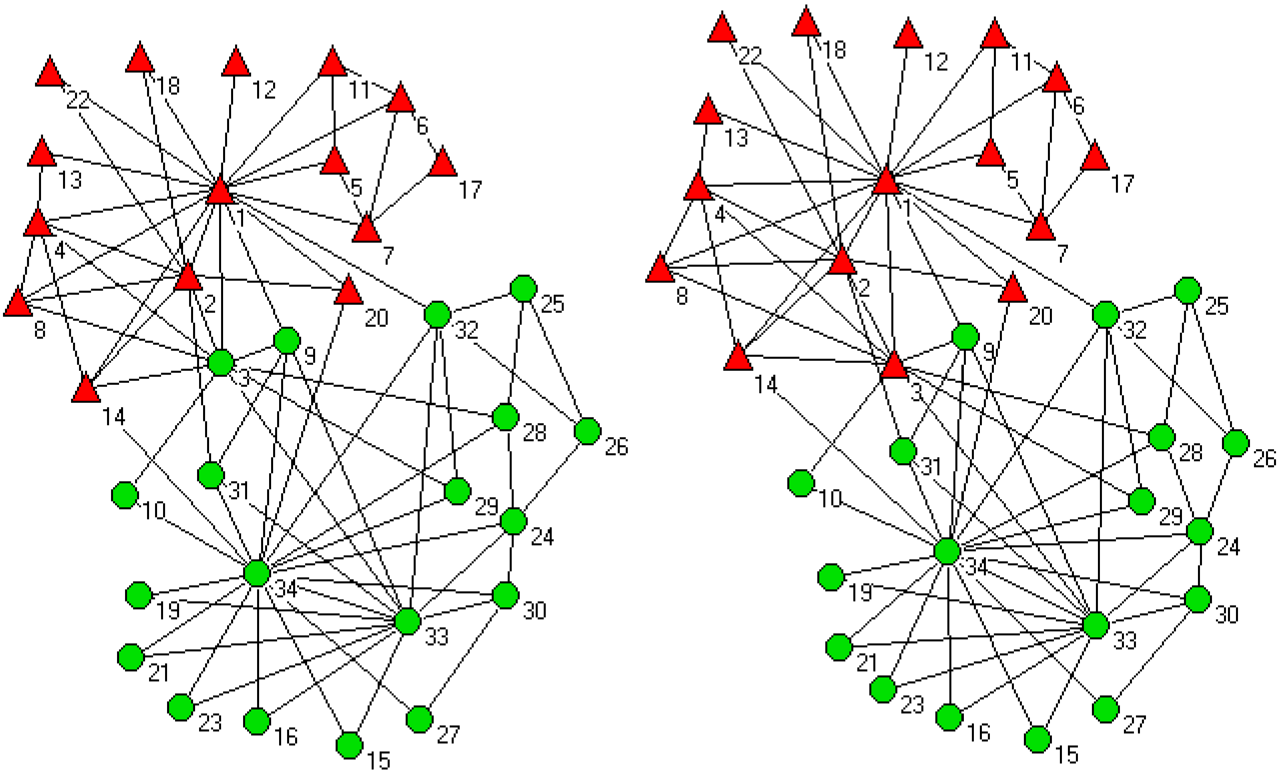
where F_c is the weighted compactness function of the node *i* and the community *c*, *i* is node in the network, *c* is the neighbor community of the node *i*, *l*(*i*, *c*) is the number of the connections between the node *i* and the community *c*, that is the number of the neighbor nodes of the node *i* in the community *c*, *w*(*i*, *c*) is the sum of the neighbor nodes weights of the node *i* in the community *c*, which is as follows:

$$w(i,c) = \sum_{j \in N(i)} W_j \tag{4}$$

where *N*(*i*) is the neighbor set of the node *i*, *j* is the neighbor node of the node *i*, W_j is the weight of the node *j*.

The following results are given for the small scale network karate network and dolphin network [41] after calculating the weighted compactness function in the section 3.3 of the algorithm. Fig. 2 is the partition result of the karate network and the standard division of karate network.

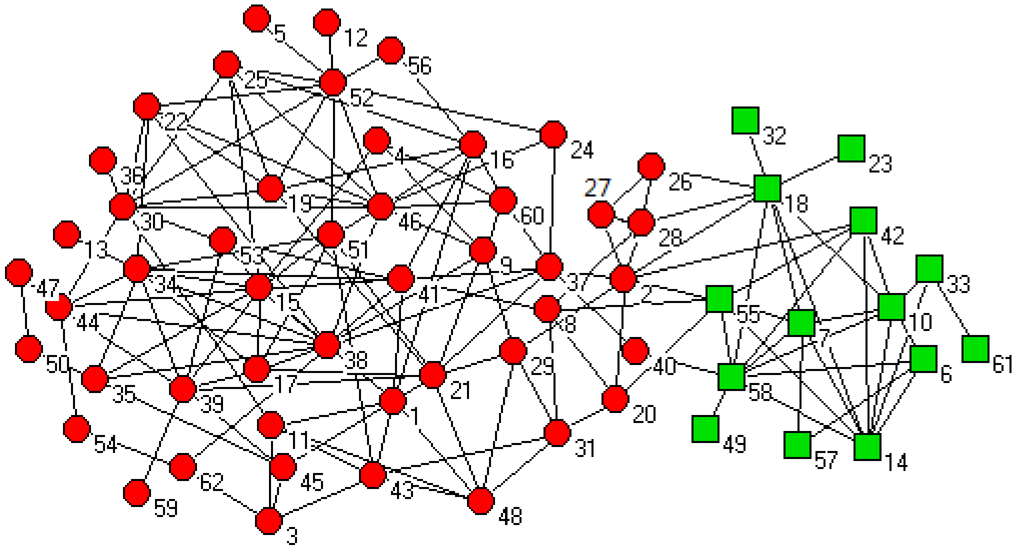
As can be seen from Fig. 2, after the algorithm described in section 3.3, we can effectively divide the karate network into two parts of communities, where only the node 3 is wrongly divided into the community with the core node 34 in it.



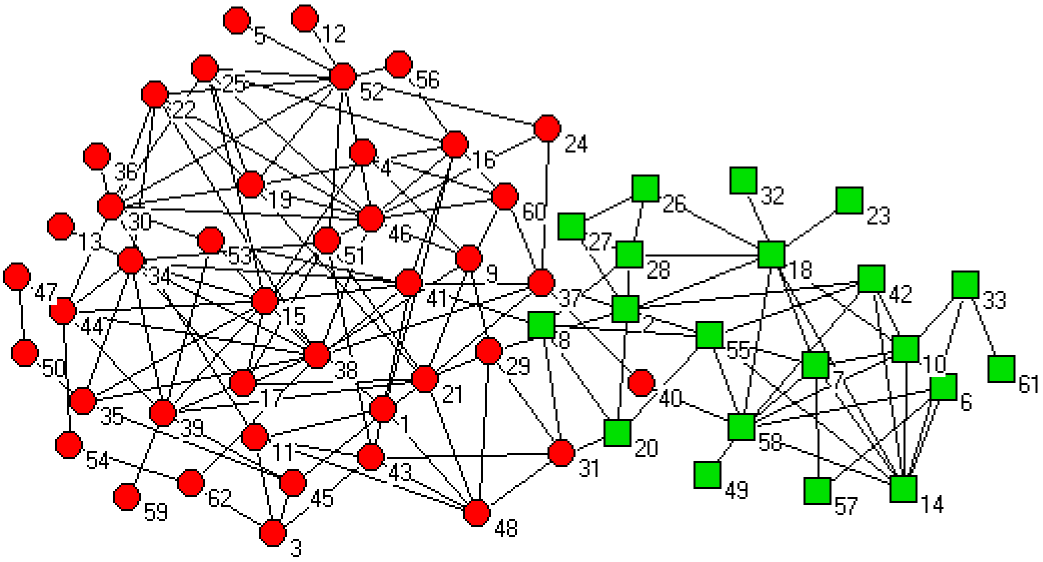
(a) (b)

Fig. 2 The karate network. (a) the partition result of the karate network after implementing Algorithm 2; (b) the standard division of karate network

Fig. 3 is the partition result of the dolphin network and the standard division of dolphin network.



(a)



(b)

Fig. 3 The dolphin network. (a) the partition result of the dolphin network after implementing Algorithm 2; (b) the standard division of dolphin network

As can be seen from Fig. 3, the Algorithm 2 described in the 3.3 section of this paper accurately divides the dolphin network into two communities. Although there are still 6 nodes are divided into the wrong community, but it is very close to the correct division.

The label propagation algorithm based on the node and community weighted compactness function is as Algorithm 2.

Algorithm 2：The label propagation algorithm based on the node and community weighted compactness function.

```
Input: The nodes number in the network $n$, node connection information $Edge$;
Output: Network partition result $f$;
Step1: Arrange the network nodes in descending order according to the nodes weight, get the weight matrix $W_n$ of n row and 2
    column. The first column of the matrix $W_n$ corresponds to the node number. The second columns of matrix $W_n$ corresponds
    to node weights;
Step2: Starting from the first node $i$ in matrix $W_n$;
Step3: Find the neighbor communities $N_c=\{c_1,c_2,\ldots c_m\}$ of the node $i$, m is the number of the communities;
Step4: Calculate the weighted compactness function $F_c(x)$ between the node and its neighbor communities, $x=1,2,3\ldots m$;
Step5: Find the neighbor community $c_g$ corresponding to the max weighted compactness function in Step3, where $g=1,2,\ldots m$,
    update the label of the node $i$ into the label of the neighbor community, $i=i+1$;
Step6: If all the labels are updated, turn to Step7. Otherwise, turn to Steps 3;
Step7: Output the Network partition result $f$.
```

3.4 Adjustment Strategy

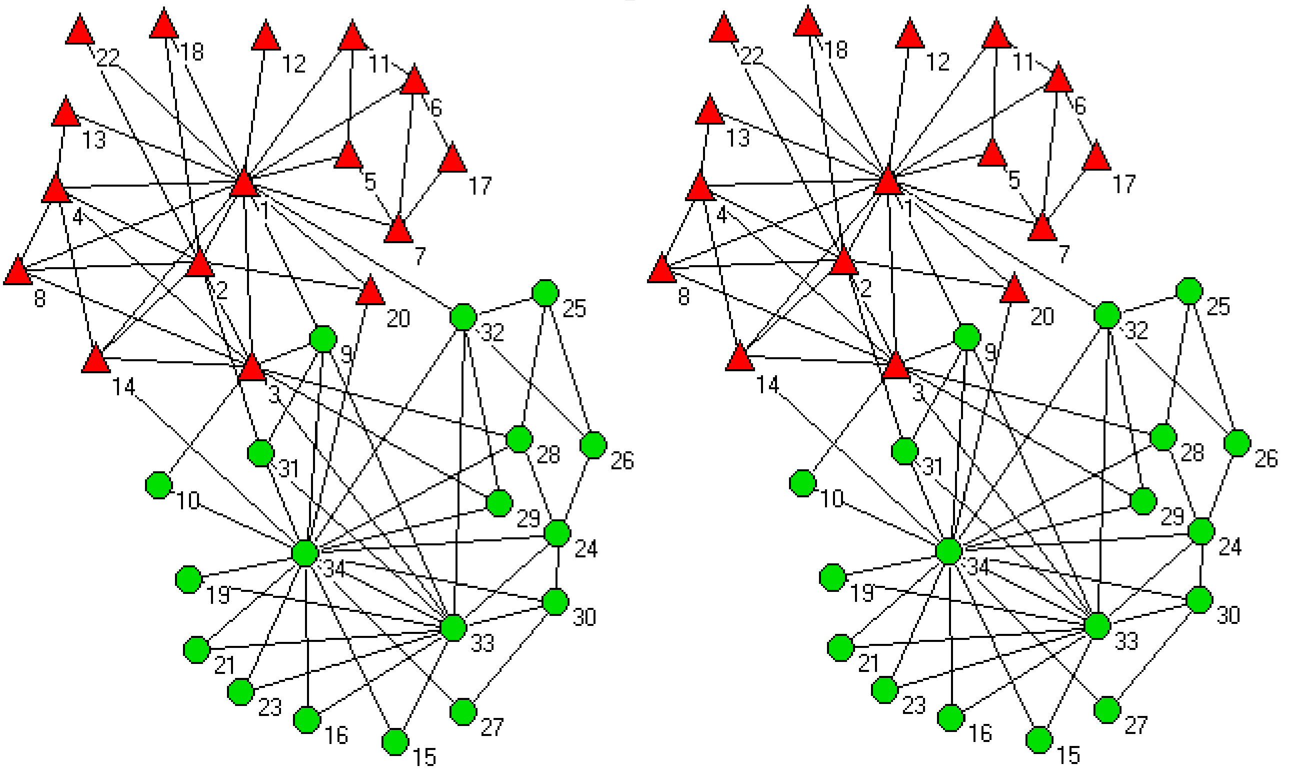
Through the improved label propagation algorithm, we can get the result of large-scale network partition more efficiently. However, in the process of label propagation, there may still be individual nodes are mistakenly classified in the wrong community. Therefore, this paper adopts an adjustment strategy based on the membership degree of nodes and communities [42].

This strategy is based on the consideration of the relationship between the node and the neighbor communities. It will adjust the node which is divided by error. The adjustment function is defined as follows:

$$R = \alpha \cdot \frac{l(i,c)}{d_i} + \beta \cdot \frac{l(i,c)}{|c|} \tag{5}$$

where $l(i,c)$ is the link number of the node i and its neighbor community c, d_i is the node degree of the node i, $|c|$ is the node number of the community c, α and β are parameters, i=1, 2, ...n, α、$\beta \in [0,1]$, $\alpha+\beta \neq 0$.

After the adjustment of the network, the detection result of the network will be more accurate. Fig. 4 is the result of the final community partition of the karate network.



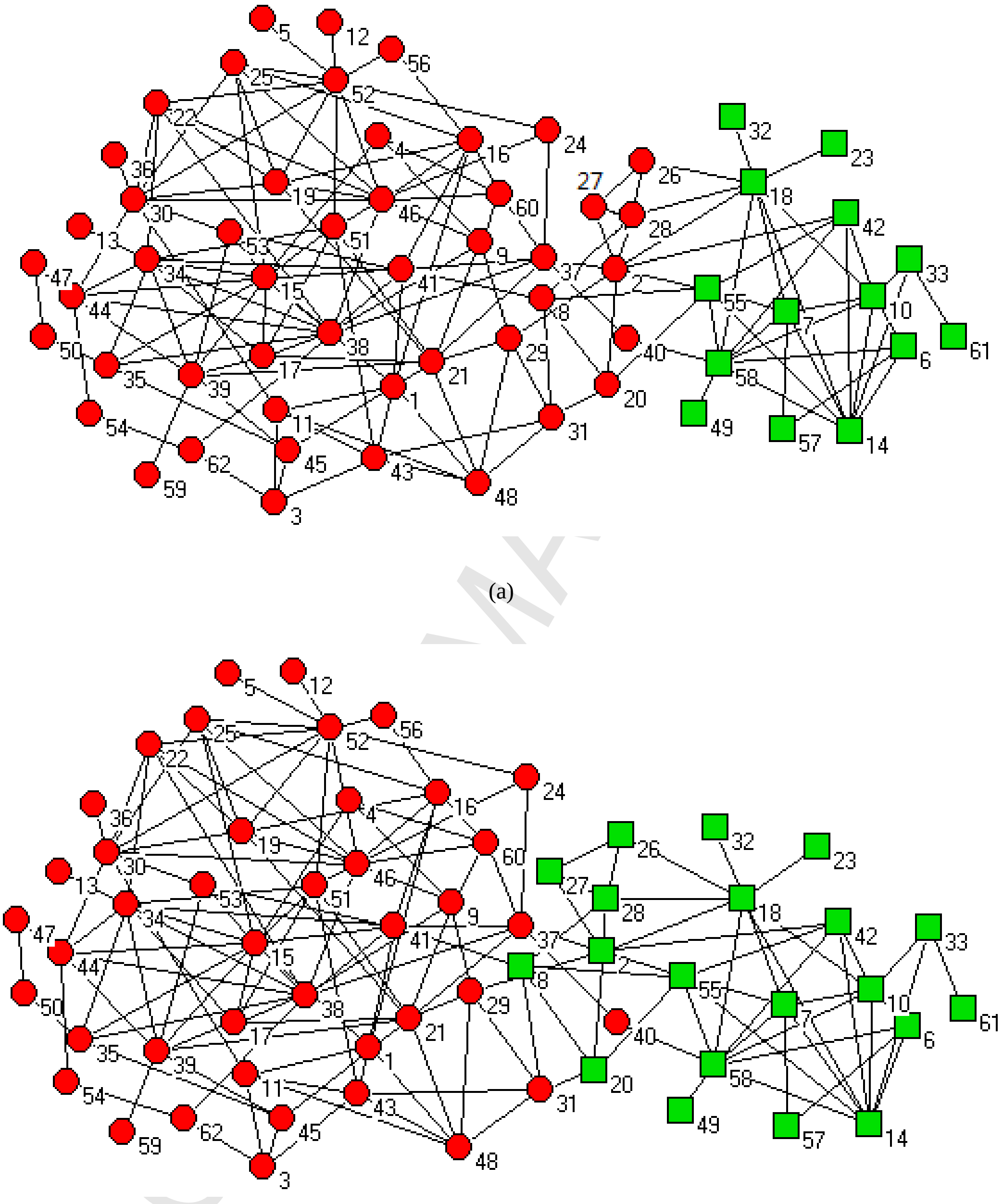
(a) (b)

Fig. 4 The karate network. (a) the final partition result of the karate network; (b) the standard division of karate network

From the perspective of communities accepting nodes, adjust the result of karate network community partition.

As can be seen from Fig. 4, the node 3 originally belonging to the community with the core node 34 is redivided into the community with the core node 1. The algorithm can get the true partition of the karate network.

Fig. 5 is the result of the final community partition of the dolphin network.



(a)

(b)

Fig. 5 The dolphin network. (a) the final partition result of the dolphin network; (b) the standard division of dolphin network

As shown in the Fig. 5, after the adjustment strategy, the nodes wrong divided before have been corrected into the right communities. Thus our algorithm can obtain the true partition of the dolphin network.

The flow of the adjustment strategy is as follows.

```
Algorithm 3：The adjustment strategy.

Input：The nodes number in the network $n$, the partition result $f$ after the label propagation algorithm, parameters $\alpha$, $\beta$;
Output：The detection result after the adjustment.
Step1: Initialization node $i$=1;
Step2: Find the neighbor community of the node $i$;
Step3: Calculate the adjustment function $R$ of the node $i$ and its every neighbor communities;
Step4: Adjust the label of the node $i$ into the community label that bring the function $R$ max value;
Step5: $i$=$i$+1; when $i$>$n$, end the adjustment, turn to Step6; otherwise, turn to Step2;
Step6: Output the detection result $f$.
```

3.5 The flow chart of the proposed algorithm

According to the introduction of above chapters, we can get the flow chart of this algorithm as shown in Fig. 6:

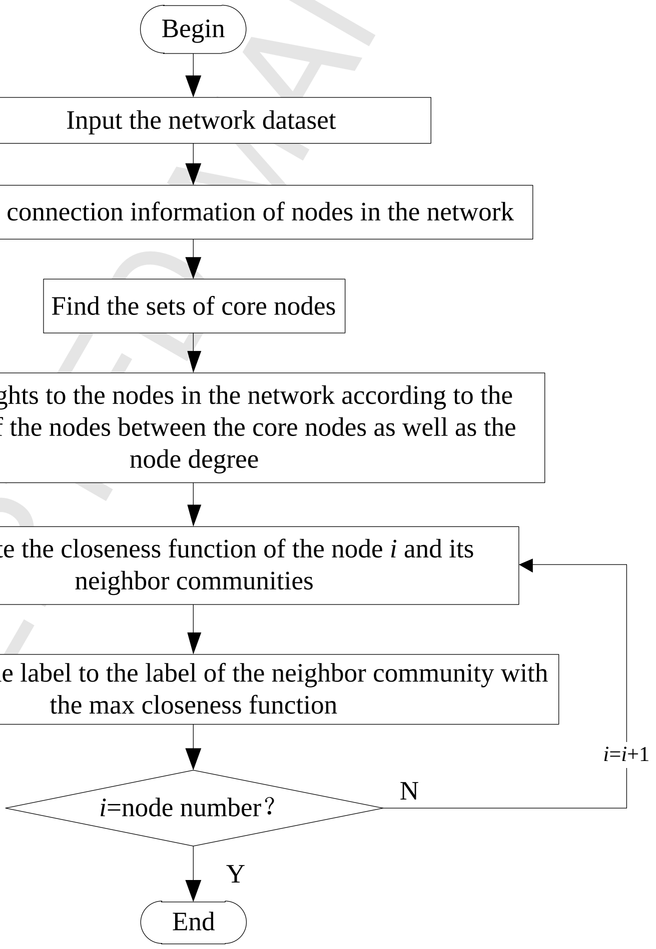

Begin
Input the network dataset
Obtain the connection information of nodes in the network
Find the sets of core nodes
Assign weights to the nodes in the network according to the similarity of the nodes between the core nodes as well as the node degree
Calculate the closeness function of the node i and its neighbor communities
Update the node label to the label of the neighbor community with the max closeness function
i=i+1
N
i=node number？
Y
End

Fig. 6 The flow chart of the proposed algorithm

4. Experimental results and analysis

4.1 Evaluation index

In 2004, Newman proposed the modularity function Q [1], which was used to evaluate the advantages and disadvantages of network community structure. This measure has been accepted and widely used by most scholars these years [16,26,43]. The more obvious the community structure in the network is, the greater the difference between the network and the random networks. The modularity function is defined as the expected difference between the number of edges within the communities in the network and the number of edges within the communities when randomly connected. The greater the value of modularity is, the more closely connected within the community structure in the network. For a network with no direction, the modular function can be reduced to:

$$\mathrm{Q} = \frac{1}{2e}\sum_{c=1}^{n}[2l_c - \frac{(d_c)^2}{2e}] \tag{6}$$

where *c* is the serial number of the community, *n* is the number of the communities, l_c is the sum of the edges of the community *c*, d_c is the sum of the node degree of in the community *c*, *e* is the total number of edges in the network.

Another common evaluation index of the network community structure is the normalized mutual information NMI. This index calculate the correct rate of community partition to test the gap of the community detection and the true partition of the network.

Danon [44] et al. confirmed the reliability of the normalized mutual information. For the two different categories of A and B, the normalized mutual information NMI is defined as follows:

$$\mathrm{NMI} = \frac{-2\sum_{u=1}^{M_A}\sum_{v=1}^{M_B} M_{uv}\cdot(\frac{M_{uv}\cdot n}{M_u \cdot M_v})}{\sum_{u=1}^{M_A} M_u \log(\frac{M_u}{n}) + \sum_{v=1}^{M_B} M_v \log(\frac{M_v}{n})} \tag{7}$$

where *n* is the node number, *M* is the confusion matrix, the element of the matrix M_{uv} represents the node number in the community *u* in the partition *A* also in the community *v* in the partition *B*, M_A is the community number in the partition *A*, M_B is the community number in the partition *B*, M_u is the sum of the elements in the row *u* in the matrix *M*, M_v is the sum of the elements in the row *v* in the matrix *M*. The greater the value of the NMI, the more similar between the partition *A* and the partition *B*. When the value of the NMI is 1, partition *A* and *B* are exactly the same.

4.2 The introduction of the test networks

First of all, we test an extended reference network with built-in community structure proposed by Lancichinetti [45]. This network is an extension of the classic GN benchmark proposed by Girvan and Newan [40]. The extended reference network consists of 128 nodes, divided into four communities. Each community has 32 nodes and the average node degree of each node in the network is 16. There is a mixed parameter *u*, when *u*<0.5, and the nodes in the communities have more neighbors than other communities in the network. That is to say, the community structure of the network is obvious. Otherwise, the community structure is fuzzy.

Then we will give the results of our algorithm and 6 contrast algorithms on three large-scale real network. Three large-scale real networks are: netscience network [46], Arxiv network [7] and Enron network [47].

The netscience network is a collection of review books on two networks. The nodes represent the 1589 co-authors networks, composed of scientists dedicated to network theory and experimentation. The network was compiled by Newman in 2006.

Arxiv network is the papers of general theory of relativity and quantum cosmology field. The network data set contains 5242 nodes and 14484 edges. Each node represents a author of a file, and each edge represents the relationship between the authors.

Enron network is a email communication network covers about 500 thousand of all e-mail communications within the data set. The data were initially public and posted into the network by the Federal Energy Regulatory Commission during the investigation. The 36692 node of the network is the e-mail address. There are 367662 edges in the network. If an address *i* sends at least one e-mail to the address *j*, there is at least one undirected edge between *i* and *j*.

4.3 Artificial network

We will compare our algorithm with 6 existing algorithms: GA algorithm [18], Infomap algorithm [11], MOEA/D algorithm [21], MODPSO algorithm [22], GDPSO algorithm [23], LPA algorithm [25], LPAm algorithm [26] and CKLPA algorithm [29].

The NMI value of each algorithm detected on the Extension of GN benchmark are shown in Fig. 7. As can be seen from Fig. 7, when the *u* is less than or equal to 0.1, all the algorithms can get the accurate community structures in the network. When the *u* is greater than 0.1, the accuracy of the GA algorithm starts to drop to minimum when the *u* is 0.35. when the *u* continue to increase, the accuracy of the MOEA/D algorithm, LPA algorithm and the CKLPA algorithm is droping. When the *u* is greater than 0.4, in addition to the MODPSO algorithm, the algorithm in this paper can get the highest NMI value. Which proves the accuracy of the algorithm in the network community structure detection.

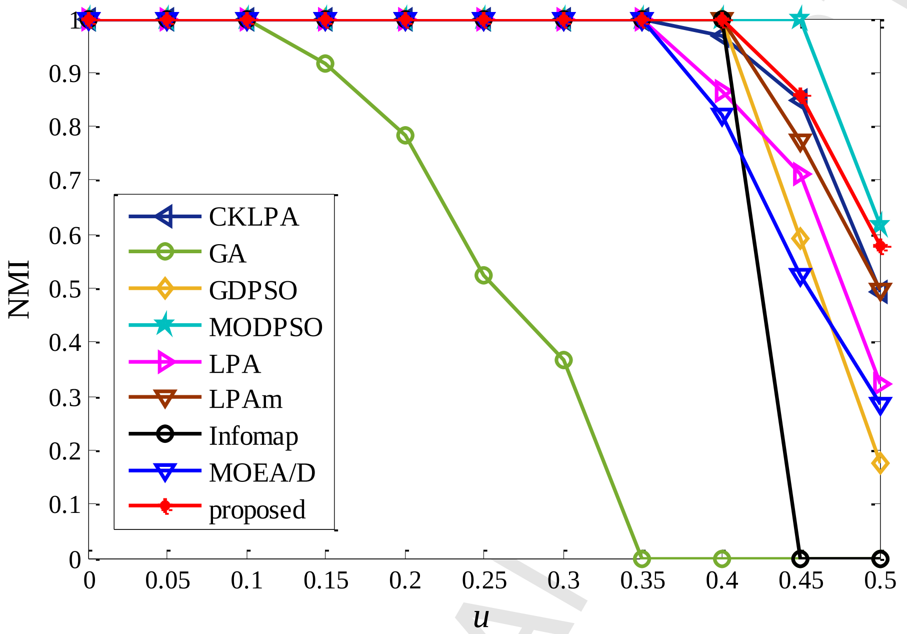


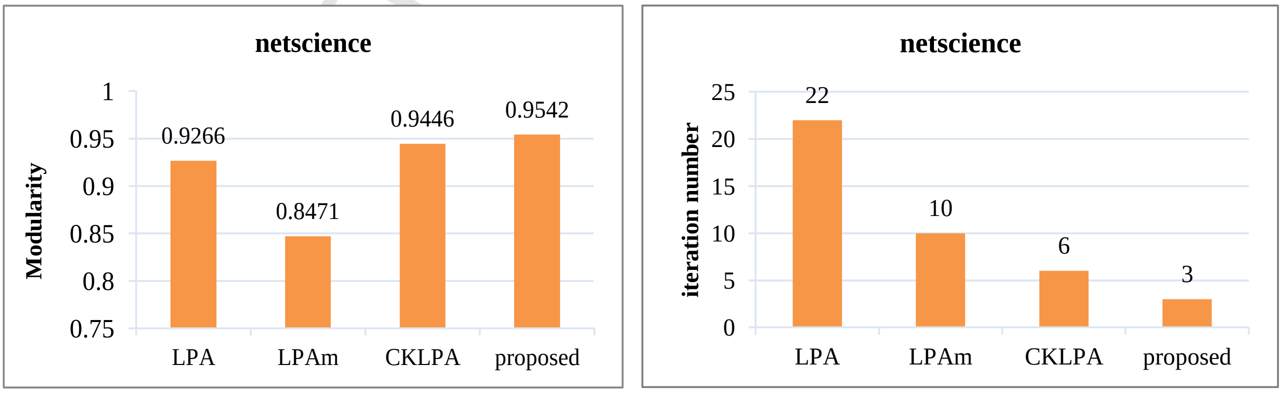
Fig. 7 The NMI value of each algorithm detected on the Extension of GN benchmark

4.4 Reality networks

In this paper, the 4 contrast algorithms including LPA algorithm, LPAm algorithm, CKLPA algorithm with the same algorithm are based on the traditional label propagation algorithm.

Below we will compare this algorithm with the three algorithms, in the section 4.2 of the introduction of the three large-scale real network test.

Fig. 8 shows the results of the four label propagation algorithms on the netscience network.



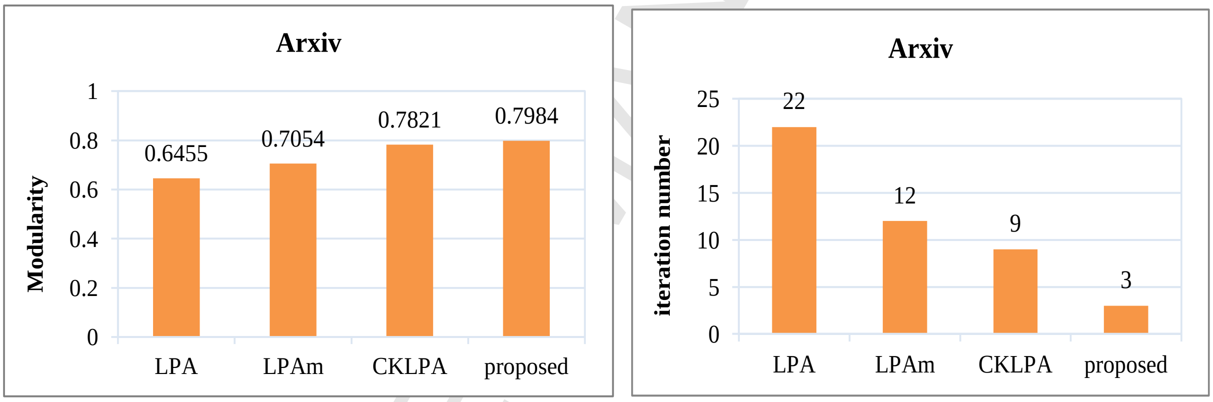
(a) (b)

Fig.8 The results of the four label propagation algorithms on the netscience network. (a) the modularity results; (b) the iteration number when the algorithms obtain the highest modularity value

As can be seen from Fig. 8, our algorithm can get the highest modularity value on the netscience network, which further proved the accuracy of the algorithm. Moreover, corresponding to the maximum number of iterations, the algorithm needs only 3 iterations to get the better results, while the other three kinds of label propagation algorithms require more iterations.

Fig. 9 shows the results of the four label propagation algorithms on the Arxiv network.

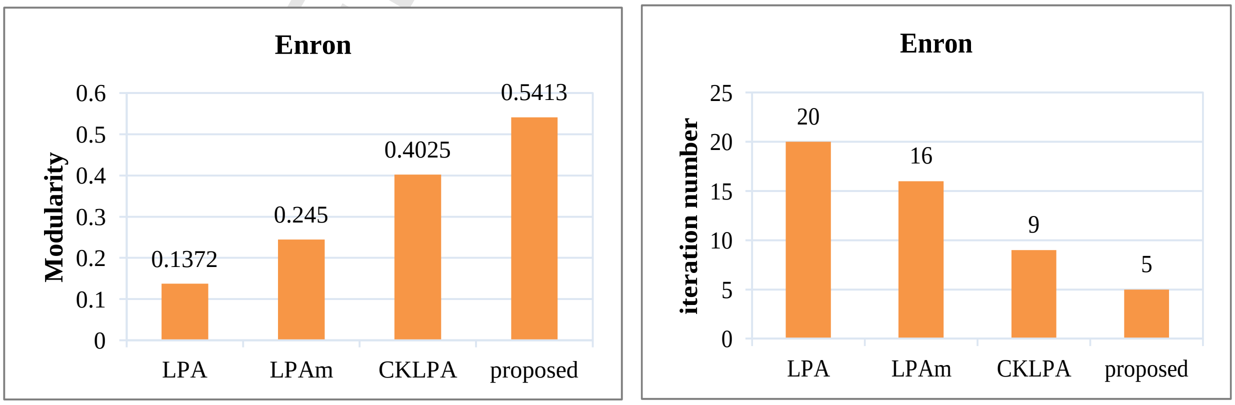
As can be seen from Fig. 9, the algorithm in this paper is compared with the other three kinds of label propagation algorithm, and gets the highest modularity value on the Arxiv network. The number of iterations required by the LPA algorithm corresponding to the highest modularity value is the same as that required on the netscience network. The number of iterations corresponding to the highest modularity value of the LPAm algorithm and CKLPA algorithm has increased compared to that on the netscience network. In this paper, the algorithm can still get the highest modularity value after 3 iterations, which further proves the validity of our algorithm.



(a) (b)

Fig.9 The results of the four label propagation algorithms on the Arxiv network. (a) The modularity results; (b) The iteration number when the algorithms obtain the highest modularity value

The results of the four label propagation algorithms on the Enron network are shown in Fig. 10:



(a) (b)

Fig.10 The results of the four label propagation algorithms on the Enron network. (a) the modularity results; (b) the iteration number when the algorithms obtain the highest modularity value

Fig. 10 shows that, in the Enron network, the modularity value obtained by the proposed algorithm is significantly higher than the other three kinds of label propagation algorithm. Moreover, the number of iterations required by our algorithm is still the least.

Below, we give the results of our algorithm and other 8 algorithms on the three large-scale networks, as shown in Table 2.

Table 2 The results of our algorithm and other 8 algorithms on the three large-scale networks

Q	GA	Infomap	MOEA/D	MODPSO	GDPSO	LPA	LPAm	CKLPA	proposed
netscience	0.8979	0.931	0.914	0.95	0.9521	0.9266	0.8471	0.9446	**0.9542**
Arxiv	0.6287	0.7628	0.6924	0.7851	0.7868	0.6455	0.7054	0.7821	**0.7984**
Enron	0.1071	0.245	-	-	-	0.1372	0.245	0.4025	**0.5413**

As can be seen from Table 2, on the three large-scale networks, our algorithm can always get the highest modularity value, which proves the accuracy of the algorithm. In addition, due to the high time complexity of MOEA/D algorithm and the GDPSO algorithm, when the network scale is more than 30 thousand, they can not get the valid community partition result in limited time. Because of its high spatial complexity, the MODPSO algorithm can not obtain the effective network community detection results when the network size is large. With the increase of the network size, the accuracy of the other 5 algorithms of community detection has decreased. Especially in the Enron network, the modularity value of our algorithm is significantly higher than the other 8 algorithms. Our algorithm inherits the low complexity of the traditional label propagation algorithm, which is beneficial to the community detection of large-scale networks. At the same time, we still ensure the effectiveness of the algorithm.

5. Conclusion

With the arrival of the era of big data, the problem of community detection for large scale networks has been attracted more and more attention. In this paper, we propose a node weight based label propagation strategy for large-scale complex network community detection. According to the node degree, we find the core node sets which has great influence in the network. Then we assign weight to the nodes in the network according to the similarity between the nodes and the core nodes as well as the node degree. We propose a weighted compactness function of the node and the neighbor community and use it as the objective function to implement the label propagation strategy. Finally, the adjustment strategy is used to correct the result of network partition. Experiments show that by looking for the core nodes to assign weighted value for the nodes, the importance of the influential nodes in the label propagation process is emphasized, which can effectively improve the accuracy of label propagation. Then we combine the connections between nodes and communities with the degree of the nodes belongs to the neighbor communities. The function effectively makes full use of the nodes and edges information in the network.

Many algorithms cannot guarantee the validity of the algorithm at the same time when the complexity is low. Our algorithm inherits the traditional label propagation algorithm with low complexity, which is very suitable for processing large-scale network community division problems. Besides, the algorithm keeps the detection accuracy of the network community. Although the algorithm has obtained better detection results, it still needs to be improved. For example, the adjustment strategy increases the complexity of the overall algorithm. The largest network the algorithm detected was 36692 nodes of the Enron network, but in reality, the network scales are often bigger. Our next work will be for larger scale network.

Acknowledgements

This work was partially supported by the National Natural Science Foundation of China, under Grant 61371201 and 61773304, the National Basic Research Program (973 Program) of China under Grant 2013CB329402, and the Program for Cheung Kong Scholars and Innovative Research Team in University under Grant IRT1170.

References

[1] M. E. J. Newman, M. Girvan, Finding and evaluating community structure in networks, Physical Review E. 2004, 69: 026113.

[2] P. G. Sun, L. Gao, Y. Yang, Maximizing modularity intensity for community partition and evolution. Information Sciences, 2013, 236: 83-92.

[3] M. Girvan, M. E. J. Newman, Community structure in social and biological networks, Proceedings of the National Academy of Sciences of the USA. 2002, 99: 7821-7826.

[4] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Defining and identifying communities in networks, Proc. Natl. Acad. Sci. USA. 2004, 101(9): 2658-2663.

[5] X. Huang, H. Cheng, J. X. Yu, Dense community detection in multi-valued attributed networks. Information Sciences, 2015, 314: 77-99.

[6] L. Bai, X. Cheng, J. Liang, Y. Guo, Fast graph clustering with a new description model for community detection. Information Sciences, 2017.

[7] B. W. Kernighan, S. Lin. An efficient heuristic procedure for partitioning graphs, Bell System Technical Journal, 1970, 49(2): 291-307.

[8] A. Pothen, H. D. Simon, K. P. Liou, Partitioning sparse matrices with eigenvectors of graphs, SIAM J. Matrix Anal. Appl., 1990, 11(3): 430-452.

[9] M. Fiedler, Algebraic connectivity of graphs, Czechoslovak Mathematical Journal, 1973, 23(2): 298-305.

[10] S. Vandongen, Graph clustering by flow simulation, PhD Thesis University of Utrecht, 2000.

[11] J. D. Noh, H. Rieger, Random walks on complex networks, Phys. Rev. Lett., 2004, 92(11): 118701.

[12] M. E. J. Newman, Fast algorithm for detecting community structure in networks, Physical Review E, 2004, 69(6): 066133.

[13] J. Mei, S. He, G. Shi, Revealing network communities through modularity maximization by a contraction dilation method, New Journal of Physics, 2009, 11: 043025.

[14] S. Kirkpatrick, D. G. Jr, M. P. Vecchi, Optimization by simmulated annealing, Science, 1983, 220(4598): 671-680.

[15] S. Boettcher, A. G. Percus, Optimization with extremal dynamics, Physical Review Letters, 2001, 86(23): 5211-5214.

[16] J. Duch, A. Arenas, Community detection in complex networks using extremal optimiza-tion, Physical Review E, 2005, 72(2): 027104.

[17] F. Glover, Future paths for integer programming and links to artificial intelligence, Computers & Operations Research, 1986, 13(5): 533-549.

[18] C. Pizzuti, GA-Net: A genetic algorithm for community detection in social networks, Parallel Problem Solving from Nature-ppsn X, 10th International Conference, 2008, 5199: 1081-1090.

[19] M. G. Gong, B. Fu, L. C. Jiao, H. F. Du, Memetic algorithm for community detection in networks, Physical Review E. 2011, 00: 006100.

[20] Z. Li, S. Zhang, R. S. Wang, X. S. Zhang, L. Chen, Quantitative function for community detection, Physical Review E. 2008, 77: 036109.

[21] M. G. Gong, L. J. Ma, Q. F. Zhang, L. C. Jiao, Community detection in networks by using multiobjective evolutionary algorithm with decomposition, Physica A. 2012, 391(15): 4050-4060.

[22] M. G. Gong, Q. Cai, X. Chen, L. Ma, Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition, IEEE Transactions on Evolutionary Computation. 2014, 18(1): 82-97.

[23] Q. Cai, M. Gong, L. Ma, S. Ruan, F. Yuan, L. Jiao, Greedy discrete particle swarm optimization for large-scale social network clustering. Information Sciences, 2015, 316: 503-516.

[24] J. Bagrow, E. Bollt, A local method for detecting communities. Physical Review E, vol. 2005, 72(2):

72-046108.

[25] U. N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Physical Review E. 2007, 76 (3): 1-11.

[26] M. J. Barber, J. W. Clark, Detecting network communities by propagating labels under constraints, Physical Review E, 2009, 80(2): 026129.

[27] X. Liu, T. Murata, Advanced modularity-specialized label propagation algorithm for detecting communities in networks, Physica A, 2010, 389(7): 1493-1500.

[28] P. Schuetz, A. Caflisch, Efficient modularity optimization by multistep greedy algorithm and vertex refinement, Physical Review E, 2008, 77: 046112.

[29] L. Zhen, X. L. Zheng, X. Nan, D. Chena, CK-LPA: Efficient community detection algorithm based on label propagation with community kernel. Physica A, 2014, 416: 386-399.

[30] R. R. Khorasgani, J. Chen, O. R. Zaïane. Top leaders community detection approach in information networks. 4th SNA-KDD Workshop on Social Network Mining and Analysis. 2010.

[31] W. Gao, W. Luo, C. Bu, Adapting the TopLeaders algorithm for dynamic social networks. The Journal of Supercomputing. 2017. https://doi.org/10.1007/s11227-017-2063-1.

[32] P. Kubat, Estimation of reliability for communication/ computer networks simulation/analytic approach. IEEE Trans. on Communication, 1989, 37(9): 927-9.

[33] Y. Chen, A. Q. Hu, J. Hu. A method for finding the most vital node in communication networks. High Technology Letters, 2004, 1: 573-575.

[34] R. Ghosh, K. Lerman. Predicting Influential Users in Online Social Networks, 2010, 1005-4882.

[35] Q. Chen, T. T. Wu. A Method for Local Community Detection By Finding Maximal-degree Nodes, ICMLC, 2011: 11-14.

[36] T. Zhou, L. Lü, Y. C. Zhang, Predicting missing links via local information. The European Physical Journal B, 2009, 71(4): 623-630.

[37] E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, A. -L. Barabási, Hierarchical organization of modularity in metabolic networks. Science, 2002, 297(5586): 1551-1555.

[38] E. A. Leicht, P. Holme, M. E. J. Newman, Vertex similarity in networks. Physical Review E, 2006, 73(2): 026120.

[39] L. Donetti, M. A. Munoz, Detecting network communities: a new systematic and efficient algorithm, Journal of Statistical Mechanics: Theory and Experiment, 2004, 2004: P10012.

[40] W. W. Zachary, An information flow model for conflict and fission in small groups, Journal of Anthropological Research, 1977, 33(4): 452-473.

[41] D. Lusseau. The emergent properties of a dolphin social network, Proceedings of the Royal Society of London. Series B: Biological Sciences, 2003, 270(Suppl 2): S186.

[42] R. H. Shang, S. Luo, Large-scale community detection based on node membership grade and sub-communities integration, Physica A. 2015, 428: 279-294.

[43] G. Agarwal and D. Kempe, “Modularity-maximizing graph communities via mathematical programming,” The European Physical Journal B, 2008, 66(3): 409-418.

[44] L. Danon, A. DÍaz-Guilera, J. Dych, Comparing community structure identification. Journal of Statistical Mechanics, 2005.

[45] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Physical Review E. 2008, 78: 046110.

[46] M. E. J. Newman, Finding community structure in networks using the eigenvectors of matrices, Physical Review E. 2006, 74: 036104.

[47] J. Leskovec, K. Lang, A. Dasgupta, M. Mahoney, Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. Internet Mathematics, 2009, 6(1): 29-123.