International Journal of Machine Learning and Cybernetics (2020) 11:1319–1329
https://doi.org/10.1007/s13042-019-01042-0

ORIGINAL ARTICLE

# Graph-based label propagation algorithm for community detection

Gui Yang[1,2] · Wenping Zheng[1,2] · Chenhao Che[1,2] · Wenjian Wang[1,2]

Received: 12 March 2019 / Accepted: 6 December 2019 / Published online: 13 December 2019

**Abstract**

Community detection is one of the most important topics in complex network analysis. Among a variety of approaches for detecting communities, the label propagation algorithm (LPA) is the simplest and time-efficient approach. However, the original label propagation algorithm is not stable due to the randomness in its propagation process. In this paper, we propose a graph-based label propagation algorithm (GLPA) to detect communities incorporating the node similarity and connectivity information during the propagation of the labels. First, we define node similarity between adjacent nodes, and change each node's label to that of its most similar neighbor node. Based on the label propagation process, GLPA constructs a label propagation graph to get candidate communities. Then, GLPA calculates the connected components of the label propagation graph. Each connected component is treated as a candidate community in the next step. Second, GLPA constructs a weighted graph to obtain final communities, in which each connected component are treated as a super-node, and the number of edges lying between the corresponding components as the weight of edges. We compute the merging factor for each node in the weighted graph and merge super nodes with higher merging factor to its most similar node iteratively to reach the maximum complementary entropy. Compared with 8 other classical community detection algorithms on LFR artificial networks and 12 real world networks, the proposed algorithm GLPA shows preferable performance on stability, NMI, ARI, modularity.

**Keywords** Complex network · Community detection · Label propagation algorithm

## 1 Introduction

Complex network analysis has a wide range of from physical, technological, biological, to social sciences [1–9]. Community detection is one of the most important topics in complex network analysis. The purpose of community detection is to group nodes into different clusters, where nodes in the same cluster are more linked with each other than with nodes outside the cluster.

Many researchers have put forward various methods to detect clusters in complex networks. Label propagation based community detection algorithms such as LPA [10], require only local information and have shown highly efficient. However, label propagation algorithms might produce unstable results due to its random tie breaking strategy, which is highly undesirable in practice and prohibits its extension to other applications.

In this paper, we propose a graph-based label propagation algorithm called GLPA for non-overlapping community detection based on graph-based representation of label propagation process and community detection process. The main steps of GLPA proceeds as follows. First, we define the node similarity between a node and its neighbors, then we change each node's label to that of its most similar neighbor node. We construct graph-based representation for label propagation progress, and obtain connected components of the resulted graph. We call each connected component a candidate community. Second, we construct graph-based representation for community detection, in which each candidate community is contracted to a super node, and the edge weights between corresponding super nodes are defined as the number of edges between two communities in original network. We compute the merging factor of each node in the resulted network. Then, we get the final results of the communities by merging the small scale communities in decreasing order of their merging factors. The proposed algorithm

✉ Wenjian Wang
wjwang@sxu.edu.cn

[1] School of Computer and Information Technology, Shanxi University, Taiyuan 030006, Shanxi, China

[2] Key Laboratory of Computational Intelligence and Chinese Information Processing of Ministry of Education, Taiyuan 030006, Shanxi, China

GLPA reduce the instability of the original LPA. Compared with other classical community detection algorithm on some real networks and artificial networks, the proposed algorithm shows preferable performance on stability, NMI, ARI and modularity.

The organization of the rest of this paper is as follows. Section 2 gives a brief introduction of the basic concepts of graph theory and the definition of some existing nodes' similarity measures. In Sect. 3, the proposed graph based label propagation algorithm (GLPA) is introduced in detail. Section 4 investigates the effectiveness of the proposed method based on 12 real networks. Section 5 is the conclusion of this paper.

## 2 Preliminaries

### 2.1 Definitions and notations

Considering an unweighted and undirected simple network $G = (V, E)$ with node set $V$ and edge set $E$. Let $A = (a_{ij})_{|V|\times|V|}$ be the adjacency matrix of $G$. The open neighborhood of a node $v \in V$ is written as $N_v = \{u \in V | (u, v) \in E\}$. The degree of a node $v$ in $G$, denoted by $d_v$, is the number of nodes in $N_v$, i.e., $d_v = |N_v|$. The subgraph of $G$ induced by $U$, denoted by $G[U]$, is the subgraph with node set $U$ and edge set $\{(u, v) | u, v \in U \wedge (u, v) \in E\}$. So, $G[U]$ contains all nodes of $U$ and all edges of $G$ whose end nodes are both in $U$. We write $[U]$ to denote the induced subgraph by node subset $U$ when without causing confusion.

A connected subgraph, if it is unconnected to the other parts of the same graph, is called a (connected) component of the graph $G$, denoted as $C(G)$. We can also get the connected components of a subgraph.

The modularity $Q$ [11], defined as Eq. (1), compares the real density of intra-module links and the expected density in a random network without any community structure called a null model.

$$Q = \frac{1}{2|E|}\sum_{ij}\left(a_{ij} - \frac{d_i d_j}{2|E|}\right)\delta_{i,j} \tag{1}$$

where $a_{ij}$ is a component in the adjacency matrix, $\frac{d_i d_j}{2|E|}$ represents the total expected number of edges between nodes $i$ and $j$. $\delta_{i,j} = 1$ if node $v_i$ and $v_j$ are in the same community, otherwise $\delta_{i,j} = 0$. A high value of $Q$ means that the partition is markedly different from a random network, thus it has strong community structure.

The complementary entropy [12] is another index to measure the quality of communities in a complex network. Let $V_r (1 \le r \le k)$ denote community $r$ in the interesting network, $L_i(V_r)$ be the number of nodes in $V_r$ adjacent to node $v_i$, and $f_i(V_r) = \frac{L_i(V_r)}{|V_r|}$ is the fraction of nodes in $V_r$ adjacent to $v_i$. Then the the complementary entropy is defined as

$$F = \sum_{i=1}^{k} h_i I_i \tag{2}$$

where $h_i = \sqrt{\sum_{r=1}^{k}\left(\frac{f_i(V_r)}{\sum_{j=1}^{k} f_i(V_j)}\right)^2}$ represents the degree to which node $v_i$ is separated from other communities, and $I_i = \sum_{r=1}^{k} L_i(V_r) \times f_i(V_r)$ evaluates how close the node $v_i$ is to other communities.

For further details about concepts of graph theory, readers are referred to [6].

### 2.2 Related works

Researchers have proposed various methods for community detection recent years. Newman and Girvan defined a quality function modularity $Q$ and optimized it to detect communities [11, 13]. Since then, modularity-based methods have been widely used to detect communities by maximizing the modularity value. For example, the fast unfolding algorithm proposed by Blondel et al. [14] is a fast heuristic method for the modularity optimization.

Raghavan et al. [10] proposed the framework of label propagation algorithm (LPA). LPA initializes every node with unique labels and propagates each node's label to that of most of its neighbors belonging to. LPA is a simple and unsupervised near linear algorithm without any parameters, and does not need any information about the size and number of communities in advance. Hence, LPA may be suitable for partitioning large networks in real time.

Because a random factor exists in LPA when there is more than one most frequent label, one might obtain different results after multiple runs. This disadvantage may prevent LPA from being widely used in practice. In addition, LPA may produce a meaningless solution in which all nodes are assigned to one community. In order to solve this problem, Barber et al. [15] proposed a modularity-specialized LPA called LPAm. LPAm might stuck in a local optimum and lead to inaccurate partitions. LPAm+ [16] is an improved approach to obtain the highest modularity value and can effectively avoid local maxima. Li et al. proposed LPA-S [17], in which labels are propagated by similarity. Although LPA has distinct advantages for community detection, there have also been many superior LPA-based methods, and there is still room for partitioning real-world networks more accurately.

Raghavan et al. [10] pointed out that 95% of nodes or more are classified correctly in 5 iterations. However, determining the speed of LPA-based methods is still an open problem. Asynchronous LPA is not suitable for very large scale network, whilst updating labels synchronously might lead to oscillation of labels and prevent the update procedure from converging. Although there are some methods to avoid non-convergent behaviors, label oscillation still exists in some cases.

## 3 GLPA: a graph-based label propagation algorithm

Let $G$ be the complex network whose nodes to be clustered. In this section, we propose a Graph-based Label Propagation Algorithm (GLPA) for detecting communities in $G$ by graph-based representation of the label propagation process and the community detection process. The proposed GLPA modifies the label propagation pattern and is divided into two sections as follows.

(1) Constructing label propagation graph $L(G)$ to get candidate communities. We first define node similarity between adjacent node pairs and find the most similar neighbor node for each node. GLPA updates the labels of nodes based on the similarity measure. We construct a label propagation graph, in which the edges are designed to record the label propagating process. Then, GLPA calculates the connected components in the resulted label propagation graph. Each connected component is treated as a candidate community in the next step.

(2) Constructing a weighted graph $W(G)$ to obtain final communities. We treat each connected component of $L(G)$ as a super-node. If there are edges between two components, then there is an edge between two corresponding super-nodes, and the weight of the edge is assigned as the number of edges of $G$ lying between two components. For a given super-node $v \in W(G)$, we choose one of its neighbors $u \in W(G)$ as its most similar node, such that the weight on edge $(u, v)$ is the largest one among those of incident edges of $v$. We compute the *merging factor* for each node in $W(G)$. Then we merge super nodes with the highest merging factor to its most similar node iteratively until the maximum complementary entropy [as is shown in Eq. (2)] does not increase.

To illustrate how the GLPA detects communities in complex networks, we use Dolphin Social network [18] to show the detail steps of GLPA as an example. The Dolphin Social network has 62 nodes and 159 edges.

### 3.1 Node similarity

The basic LPA adopts the label belonging to the majority of its neighbors, which means that it treats all neighbors equally. However, in practice, a node plays different roles depending on its location in the networks. In the real world, entities with high similarity tend to be gathered in the same group. For this perspective, community detection methods using an appropriate similarity metric may discover some valuable results in practice. The number of common neighbors [19] between two nodes can reflect their similarity. The more common neighbors between two nodes, the more similar they are in structure. On the other hand, the node degrees also play an important role in detecting communities, especially for linked nodes without common neighbors. Hence, in our method we use a similarity $S_{i,j}$ between two adjacent nodes $v_i$ and $v_j$ based on their common neighbors $N_i \cap N_j$ and the degrees of $v_i$ and $v_j$. The formula of the similarity $S_{i,j}$ is given as Eq. (3).

$$S_{i,j} = \begin{cases} |N_i \cap N_j| + \frac{1}{d_i d_j}, & if\ (v_i, v_j) \in E; \\ 0, & otherwise \end{cases} \tag{3}$$

where $N_i \cap N_j$ denotes the set of common neighbors of adjacent nodes $v_i$ and $v_j$. The similarity of two adjacent nodes $S_{i,j}$ would be determined by the number of common neighbors and the node degrees. $S_{i,j}$ tends to be large when the two adjacent nodes share many common neighbors. And $S_{i,j}$ might be small when the degrees of $v_i$ and $v_j$ are large.

### 3.2 The steps of GLPA algorithm

#### 3.2.1 Construction of label propagation graph *L*(*G*)

At beginning, each node is identified by a unique label which implies different community identifier. We construct the graph representation $L(G)$ of the label propagation progress from a null graph, whose initial node set $V(L) = V(G)$ and initial edge set $E(L) = \emptyset$.

Then for each node $v_i \in V(G)$, we find the most similar node of $v_i$ according to Eq. (3). If there are more than one most similar neighbor node, we choose one randomly. Let node $v_j$ be the most similar node of $v_i$, then we add an edge $(v_i, v_j)$ between node $v_i$ and $v_j$ to $L$, i.e., $E(L) = E(L) \cup \{(v_i, v_j)\}$. We can find the most similar node for each node simultaneously. And we will obtain a label propagation graph $L$ with $|V(G)|$ nodes and no more than $|E(G)|$ edges.

Given the form of node similarity $S$ as Eq. (3), it is reasonable to expect that the number of neighbors with high similarity will be much lower than the degree of a node. In other words, there are only a few choices of neighbor node for each node. This would reduce the randomness in original

LPA algorithm for the reason that the number of most similar nodes is greatly reduced. And it can be easily seen that, if there is a path between two nodes in $L$, then the node pair should have a higher similarity.

To avoid the oscillation of the label propagation, we calculate the connected components of $L$. Let $\{C_1, \ldots, C_k\}$ $(k \leq |V(L)|)$ be the set of connected components of $L$. The nodes in the same connected component share the same initial label. Figure 1a shows the resulted label propagation graph $L(G)$ and connected components of Dolphin Social network. Figure 1b shows the initial communities in Dolphins network detected by GLPA.

For most of complex networks, the number of connected components might be much greater than the real number of communities in $G$. We need additional merging steps to deal with the over-segmentation of the network.

### 3.2.2 Merging initial clusters

Although, the number of nodes contained in a connected component is small, the similarity among these nodes might be high. In the second phase, we will perform merging process to maximize the complementary entropy $F$ of the partition by constructing a weighted graph $W(G)$ from $L(G)$. To obtain $W(G)$, we treat each connected component in $L(G)$ as a super-node.

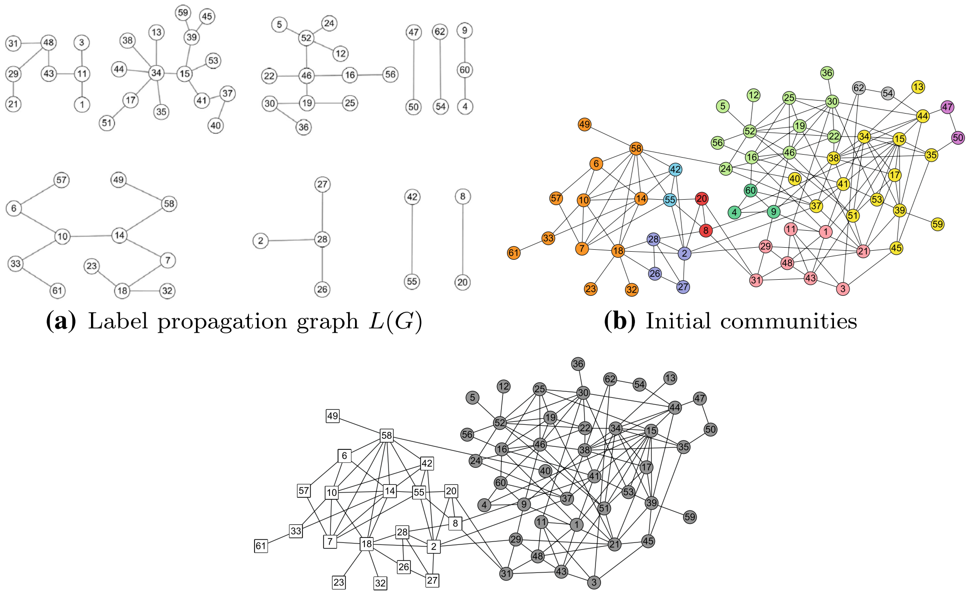
Let $c_i$ for $1 \leq i \leq k$ be the super-node in $W$ corresponding to the connected component $C_i$ of $L$, and $k$ is the number of connected components. For super-nodes $c_i$ and $c_j$, if there exist edges in $G$ that have one endpoint in $C_i$ and the other endpoint in $C_j$, then we add an edge $(c_i, c_j)$ to $W$ whose weight equals to the number of edges between the two components in $G$.

Then we compute the *merging factor* for each super-node $c_i$ in $W$ as follows:

$$PC(c_i) = \frac{\sum_{r=1}^{k} (\omega_{ir})^2}{\left(\sum_{j=1}^{k} \omega_{ij}\right)^2} \tag{4}$$

where $\omega_{ij}$ represents the edge weights between node $c_i$ and node $c_j$, thus $\omega_{ij} = \sum_{x \in V(C_i)} \sum_{y \in V(C_j)} a_{xy}$ and $\omega_{ii} = 1$.

We use merging factor to determine whether a component should be merged into another component to form a larger community. If a component is dispersedly connected with many other components, then the component would has a low merging factor and might be unlikely to be merged with others. On the contrary, if a component is intensively connected with a few other components, then the component would has a high merging factor and might be merged with another component. In the second step, we treat a component as a super-node. For a given super-node $v \in W(G)$, we choose one of its neighbors $u \in W(G)$ as its most similar node, such that the weight on edge $(u, v)$ is the largest one among those of incident edges of $v$. We compute the *merging factor* for each node in $W(G)$. We will merge super nodes with the highest merging factor to its most similar super-node iteratively to reach the maximum complementary entropy $F$ [as is shown in Eq. (2)]. Figure 1c shows the final communities in Dolphins network detected by GLPA.



(a) Label propagation graph $L(G)$

(b) Initial communities

(c) Final communities

**Fig. 1** Communities detection of Dolphins network

### 3.2.3 Time complexity analysis

In GLPA, suppose the node number of graph $G$ is $n$. In the first step, we need calculate similarities between adjacent node pairs. The computational complexity for computing nodes similarity is $O(n^2)$. Then we construct a label progress graph $L$ to imitate the label propagation process, and the computational complexity for obtaining label propagation graph is $O(n^2)$. It takes $O(n)$ time to obtain connected components of $L(G)$. In the second step, it takes $O(k^2)$ to determine the weights of edges in $W(G)$, where $k$ is the number of components in $L(G)$ and $k \ll n$. The computational complexity for updating labels in this phase is $O(t \times (n + k))$, where $t$ $(t \ll n)$ is the number of iterations. We compute the merging factors of each node of the resulted network $W$, which has a complexity $O(k^2)$. Therefore, the total time complexity of the proposed algorithm GLPA is $O(n^2)$.

## 4 Experiments and results

In this paper, we compare the performance of our method GLPA on both synthetic and real-world networks with 8 classical algorithms LPA, LPAm, LPAm+, LPA-S, BGLL, ISCD+, FMM and Infomap [20] in the literature.

### 4.1 Evaluation criteria

#### 4.1.1 NMI and ARI

To evaluate the performance of community detection algorithm, normalized mutual information(NMI) [21], adjusted rand index(ARI) [22] and modularity $Q$ are adopted.

Given a set $V$ with $n$ nodes, let $O = \{O_1, O_2, \dots, O_k\}$ represent the communities obtained by an algorithm, and $P = \{P_1, P_2, \dots, P_{k'}\}$ represent the ground truth communities. For $O_i$ and $P_j (1 \le i \le k, 1 \le j \le k')$, let $n_{ij} = |O_i \cap P_j|$, $b_i = \sum_{j=1}^{k'} n_{ij}$, $t_j = \sum_{j=1}^{k} n_{ij}$. NMI is defined as

$$NMI = \frac{-2\sum_i \sum_j n_{ij} \log(\frac{n_{ij}n}{b_i t_j})}{\sum_i b_i \log(\frac{b_i}{n}) + \sum_j t_j \log(\frac{t_j}{n})} \tag{5}$$

And ARI is defined as

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left(\sum_i \binom{b_i}{2} \sum_j \binom{t_j}{2}\right)/\binom{n}{2}}{\frac{1}{2}\left(\sum_i \binom{b_i}{2} + \sum_j \binom{t_j}{2}\right) - \left(\sum_i \binom{b_i}{2} + \sum_j \binom{t_j}{2}\right)/\binom{n}{2}} \tag{6}$$

The higher values of NMI and ARI indicate the better partition quality. If NMI and ARI reach their maximal value (which equals 1), then the algorithm obtains exactly the ground truth communities.

#### 4.1.2 Stability coefficient

In order to evaluate the stability of the obtained results, we adopt the stability coefficient $\sigma$ proposed in [23]. For network $G = (V, E)$ with $|V| = n$, we run an algorithm $T$ times to show the stability of the algorithm. For $1 \le t \le T$, we construct an $n \times n$ matrix $\Omega_t$, whose elements are defined as

$$\delta_{ij}^{(t)} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ in the same community,} \\ 0, & \text{otherwise.} \end{cases}$$

Then we calculate the variance matrix $\mathcal{S}$ of $\Omega_t$ $(1 \le t \le T)$. The element of $\mathcal{S}$ is defined as

$$\mathcal{S}_{i,j} = \frac{1}{T}\sum_{t=1}^{T}\left(\delta_{i,j}^{(t)} - \frac{1}{T}\sum_{x=1}^{T}\delta_{i,j}^{(x)}\right)^2$$

The *stability coefficient* $\sigma$ of the algorithm is defined as

$$\sigma = \frac{1}{n^2}\sum_{1 \le i,j \le n} \mathcal{S}_{i,j} \tag{7}$$

If an algorithm has a low stability coefficient $\sigma$ value, then the algorithm has good stability.

### 4.2 Experiments on computer-generated networks

We use the LFR benchmark networks [24] as synthetic networks to test the performance of our algorithm. LFR benchmark network can create networks by several parameters: the number of nodes $N$, average degree $\langle d \rangle$, maximum degree $d_{max}$, minimum community size $c_{min}$, maximum community size $c_{max}$, exponent $\gamma$ for the degree distribution, exponent $\beta$ for community size distribution, and mixing parameter $\mu$. We test the algorithms on the networks with $N \in \{1000, 5000\}$, community size varying in the range 10–50 and 20–100, $\langle d \rangle = 20$, $d_{max} = 50$, $\gamma = 2$ and $\beta = 1$. The mixing parameter $\mu$ denotes the fraction of inter-community edges of the network. When $\mu > 0.5$, there are more inter-community links than intra-community links, which indicates that there might be no obvious communities in the network. When $\mu < 0.05$, there is few inter-community links, which indicates that the network might be disconnected. Hence, the LFR networks used here were generated by setting the mixing parameter $\mu \in [0.05, 0.5]$. We compare our GLPA with 6 classical algorithms LPA, LPAm, LPAm+, LPA-S, BGLL and ISCD+ in this section.

We performed each algorithm 30 times on each dataset and obtained the average performance. Figures 2 and 3 show the comparison results of GLPA and comparing algorithms in term of NMI and ARI, respectively. When the mixing parameters were low ($\mu \in [0.05, 0.3]$), the community structure of these networks were relatively obvious, and the results of each algorithm were highly consistent with the ground truth communities in these networks. When $\mu$ grows, the community structure of the networks were not obvious. The proposed GLPA chooses communities with higher participation coefficient to merge, which ensures the quality of community detection and stability of the algorithm.

### 4.3 Experiments on real-world networks

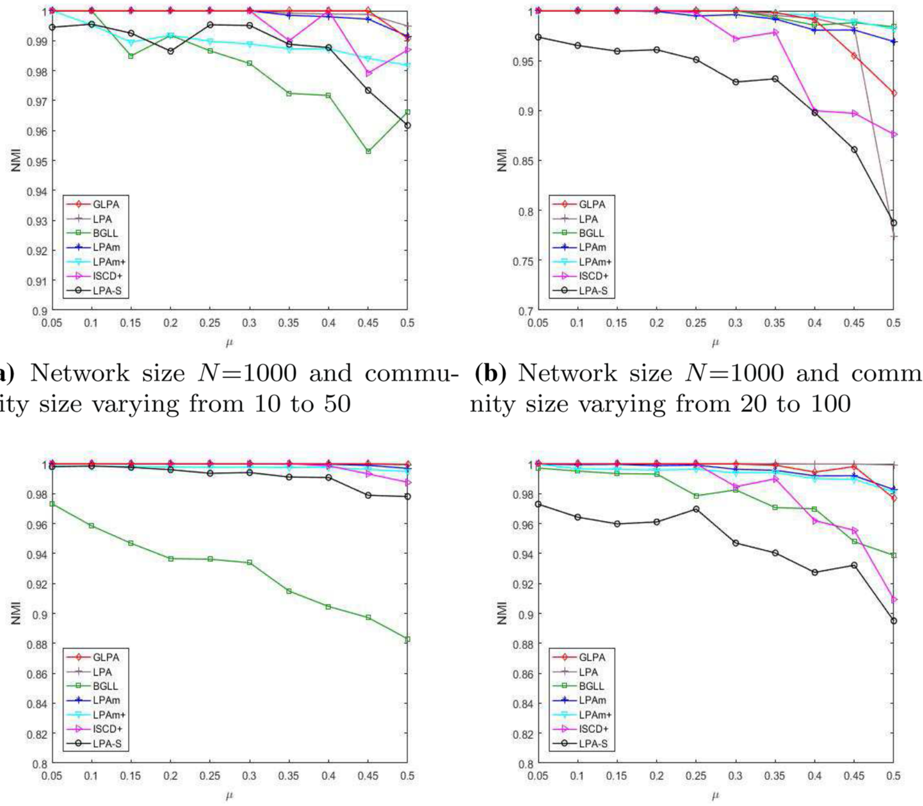
We test algorithms on 4 real world networks with ground truth communities: Zachary's Karate Club [25], Dolphins Social Network [18], Polbooks [26], and US College Football Network [27]; and we also run algorithms on 8 real-world networks without ground truth communities: Les Misérables [28], NetScience [29], Email [30], Yeast [31], Web_spam [32], Router [32], Bio_dmela [32] and PGP [32]. Table 1 displays the characteristics of these real world networks. We compared our GLPA with LPA, LPAm, LPAm+, LPA-S, FMM, BGLL, Infomap and ISCD+ on these networks.

For 4 real-world networks with ground truth communities, we use modularity, NMI, ARI and stability coefficient to show the performance of the algorithms. We perform 30 times for each algorithm on each dataset. The results are shown in Table 2, where $k$ is the number of communities obtained.

Zachary's Karate Club network is one of the most commonly used benchmark networks in community mining [25]. This network consists of 34 nodes and 78 links. Each member denotes a node in the network, and a link exists between two nodes if these two members interact consistently outside the activities of the club. However, the administrator (node 1) and instructor (node 33) fell out, and the members were split into two groups with either administrator or instructor. The proposed algorithm GLPA outputs two communities, as is shown in Fig. 4. Only node 10 is misclassified, and the value of NMI is higher than 0.8, which is much higher than the average of LPA, LPAm and LPAm+.

The second network is Dolphins Social Network which has been investigated by Lusseau [18] over several years. It is composed of 62 dolphins and 159 links. As shown in Fig. 5, GLPA detects two communities in Dolphins network, which is completely consistent with the ground truth. GLPA gets the best result on this network.

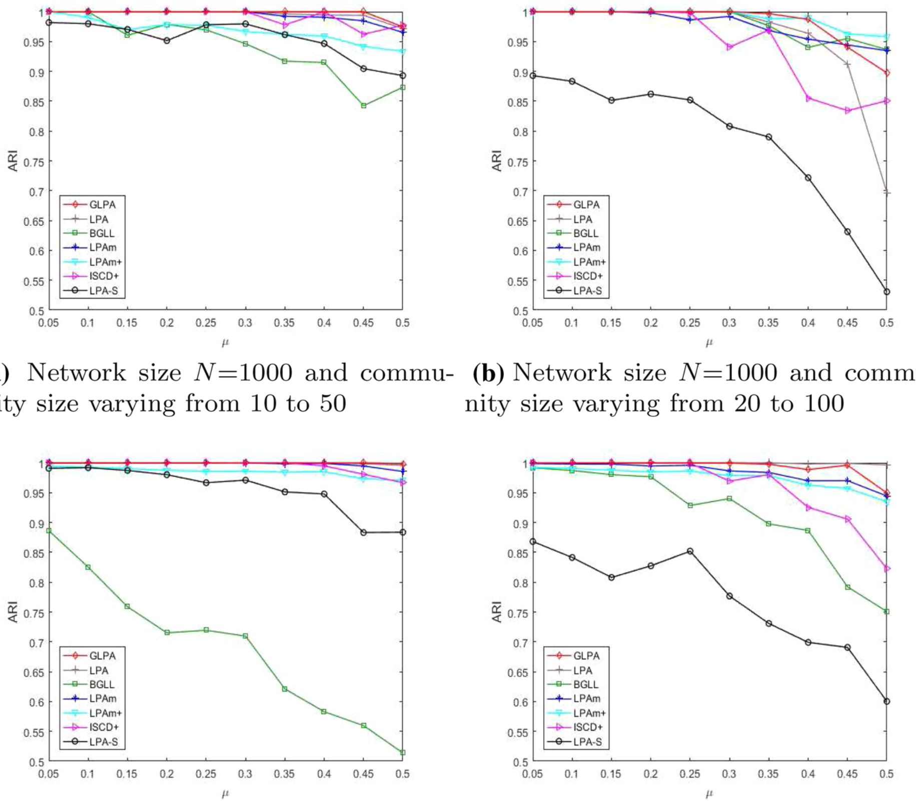
US College Football network [27] is the social network with 115 nodes and 613 edges. All nodes in the network are divided into 12 communities corresponding to the



**(a)** Network size $N$=1000 and community size varying from 10 to 50 **(b)** Network size $N$=1000 and community size varying from 20 to 100

**(c)** Network size $N$=5000 and community size varying from 10 to 50 **(d)** Network size $N$=5000 and community size varying from 20 to 100

**Fig. 2** Comparison of NMI on LFR benchmark networks



**(a)** Network size $N$=1000 and community size varying from 10 to 50

**(b)** Network size $N$=1000 and community size varying from 20 to 100

**(c)** Network size $N$=5000 and community size varying from 10 to 50

**(d)** Network size $N$=5000 and community size varying from 20 to 100

**Fig. 3** Comparison of ARI on LFR benchmark networks

**Table 1** Characteristics of 12 real networks

| Datasets | Nodes | Links | Communities |
|---|---|---|---|
| Karate | 34 | 78 | 2 |
| Dolphins | 62 | 159 | 2 |
| Football | 115 | 613 | 12 |
| Polbooks | 105 | 441 | 3 |
| Les Misérables | 77 | 254 | – |
| NetScience | 379 | 914 | – |
| Email | 1133 | 5451 | – |
| Yeast | 2375 | 11,693 | – |
| Web_spam | 4767 | 37,375 | – |
| Router | 5022 | 6258 | – |
| Bio_dmela | 7393 | 25,569 | – |
| PGP | 10,680 | 24,316 | – |

“conferences”. Our result is shown in Fig. 6, which contains 13 communities.

Another benchmark is commonly called Polbooks which is a network of books about American politics compiled by Valdis Krebs [26]. This network involves 105 nodes, each of which represents a book about US politics sold on Amazon.com. All nodes in the network were classified into three groups according to their political inclination: liberal, conservative, or centrist. If customers frequently bought two books at the same time, the two corresponding nodes are connected by an edge. The proposed GLPA detects two communities, as is shown in Fig. 7a. The value of the NMI via our algorithm are higher than other algorithms. Because in the original network division as shown in Fig. 7b, the nodes corresponding to the “centrist” books(blue nodes) are not densely connected due to the lack of obvious political inclination. GLPA divides the network into two communities and both communities were densely connected internally and sparsely connected externally. We think it might be reasonable to divide the network into two communities.

Table 3 shows the experimental results on 8 real-world networks without ground truth communities. We compare the performance of each algorithm from the modularity and stability. It can be seen that the stability coefficient of GLPA is low, which indicates that it has good stability.

## 5 Conclusion

In this paper, we propose a graph-based label propagation algorithm called GLPA for non-overlapping community detection based on neighbor influence and connectivity

**Table 2** Comparison of four real-world networks with ground truth communities

| Datasets | Index | FMM | LPA | BGLL | LPAm | LPAm+ | Infomap | ISCD+ | LPA-S | GLPA |
|---|---|---|---|---|---|---|---|---|---|---|
| Karate | $k$ | 2 | 1–3 | 3–4 | 4–7 | 4 | 3 | 2 | 2–3 | 2 |
| | Modilarity | 0.3718 | 0.3147 | 0.4047 | 0.3710 | 0.4164 | 0.4020 | 0.3715 | 0.3690 | 0.3716 |
| | NMI | 0.8372 | 0.6574 | 0.6537 | 0.5771 | 0.6521 | 0.6995 | 1.0000 | 0.9426 | 0.8372 |
| | ARI | 0.8823 | 0.6632 | 0.5445 | 0.4092 | 0.5245 | 0.7022 | 1.0000 | 0.9560 | 0.8823 |
| | $\sigma$ | – | 0.1214 | 0.0487 | 0.0438 | 0.0256 | – | – | 0.0131 | 0 |
| Dolphins | $k$ | 3 | 2–5 | 4–5 | 8–11 | 4–6 | 6 | 4 | 2–6 | 2 |
| | Modilarity | 0.4942 | 0.4920 | 0.5202 | 0.4988 | 0.5226 | 0.5158 | 0.4917 | 0.4373 | 0.3735 |
| | NMI | 0.6058 | 0.6349 | 0.5195 | 0.4437 | 0.5086 | 0.5270 | 0.4708 | 0.6820 | 1.0000 |
| | ARI | 0.4795 | 0.5106 | 0.4259 | 0.2308 | 0.3355 | 0.3614 | 0.3458 | 0.6442 | 1.0000 |
| | $\sigma$ | – | 0.0605 | 0.0310 | 0.0173 | 0.0308 | – | – | 0.0785 | 0 |
| Football | $k$ | 5 | 8–13 | 9–10 | 10–14 | 9–10 | 10 | 13 | 7–18 | 13 |
| | Modilarity | 0.5494 | 0.5819 | 0.6037 | 0.5821 | 0.6019 | 0.5902 | 0.5949 | 0.5291 | 0.5949 |
| | NMI | 0.6862 | 0.8725 | 0.8740 | 0.9023 | 0.8781 | 0.8801 | 0.9254 | 0.8406 | 0.9254 |
| | ARI | 0.4444 | 0.7390 | 0.7681 | 0.8232 | 0.7775 | 0.7601 | 0.8890 | 0.6801 | 0.8890 |
| | $\sigma$ | – | 0.0245 | 0.0103 | 0.0096 | 0.0081 | – | – | 0.0296 | 0 |
| Polbooks | $k$ | 3 | 1–4 | 3–6 | 8–12 | 4–8 | 5 | 3 | 2–6 | 2 |
| | Modilarity | 0.4993 | 0.3801 | 0.5210 | 0.4890 | 0.5197 | 0.5267 | 0.4973 | 0.4971 | 0.4569 |
| | NMI | 0.5566 | 0.4231 | 0.5234 | 0.4320 | 0.5063 | 0.5369 | 0.5245 | 0.5543 | 0.5979 |
| | ARI | 0.6563 | 0.4921 | 0.5966 | 0.3183 | 0.5366 | 0.6463 | 0.6390 | 0.6514 | 0.6671 |
| | $\sigma$ | – | 0.1228 | 0.0259 | 0.0367 | 0.0441 | – | – | 0.0372 | 0 |

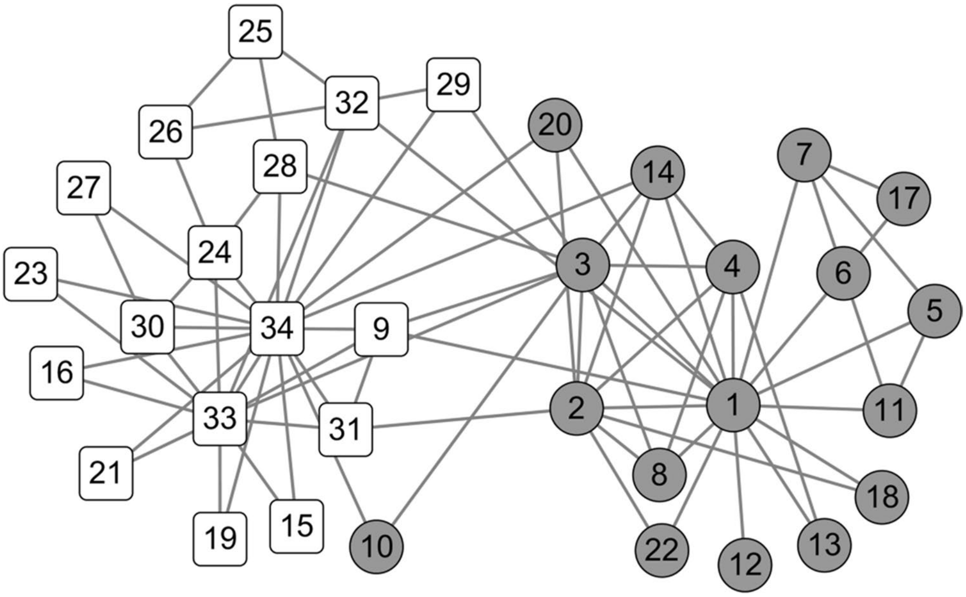


**Fig. 4** Result of GLPA on Karate network

information during the label propagation process. The main steps of GLPA proceed as follows. First, we define the node similarity between a node and its neighbors. Each node changes its label to that of its most similar neighbor node. We construct a label progress graph $L$ to imitate the label propagation process and calculate the connected components of $L$. Next, we construct a weighted graph $W$ by treating each connected component as a super node, and assign the number of edges between two communities in original network as their edge weights. We compute the merging factors of each node of the resulted network $W$. We get the final communities by merging the small scale communities with high merging factors. The proposed algorithm GLPA could reduce the instability of the original LPA. Compared with other classical community detection algorithms on some real networks and artificial networks, the proposed algorithm shows preferable performance on NMI, ARI and modularity.

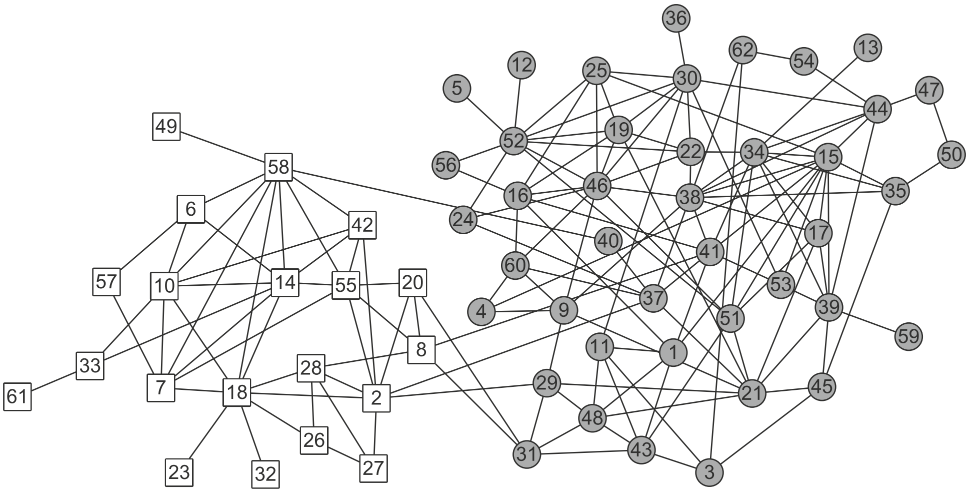


**Fig. 5** Result of GLPA on Dolphins network

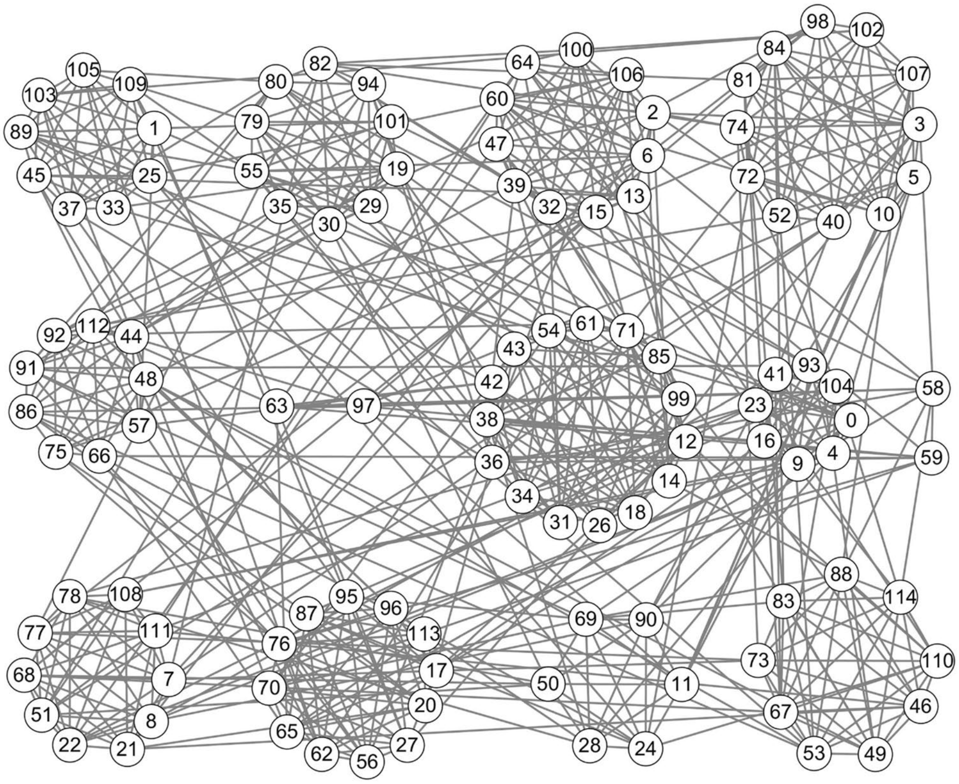
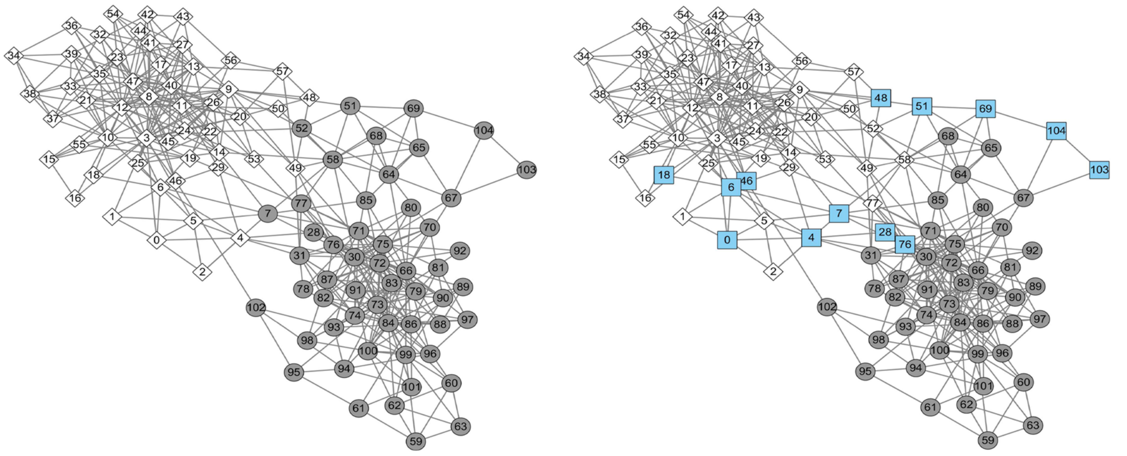


**Fig. 6** Result of GLPA on Football network

As future work, the authors intend to develop the graph-based representation to detect large amounts of small-scale communities, whose size distribution is imbalance. In this case, authors aim to employ the local version of measure to fit the imbalance size distribution. Moreover, we need to develop an effective community detection algorithm for detecting special community structures in complex networks.



**(a)** Result of GLPA on Polbooks network

**(b)** Original division on Polbooks network

**Fig. 7** Comparisons on the Polbooks network

**Table 3** Comparison of four real-world networks without ground truth communities

| Datasets | Index | FMM | LPA | BGLL | LPAm | LPAm+ | Infomap | ISCD+ | LPA-S | GLPA |
|---|---|---|---|---|---|---|---|---|---|---|
| Les Misérables | *k* | 4 | 2–5 | 5–6 | 10–14 | 6–10 | 8 | 2 | 3–23 | 5–6 |
| | Modilarity | 0.4961 | 0.2719 | 0.5461 | 0.5315 | 0.5499 | 0.5363 | 0.2540 | 0.4458 | 0.4746 |
| | $\sigma$ | – | 0.0833 | 0.0193 | 0.0103 | 0.0112 | – | – | 0.0534 | 0.0024 |
| NetScience | *k* | 18 | 31–40 | 18–20 | 68–74 | 64–70 | 7 | 6 | 4–60 | 14–29 |
| | Modilarity | 0.8385 | 0.8081 | 0.8464 | 0.7175 | 0.7299 | 0.7789 | 0.7546 | 0.7537 | 0.8058 |
| | $\sigma$ | – | 0.0130 | 0.0053 | 0.0020 | 0.0035 | – | – | 0.0573 | 0.0100 |
| Email | *k* | 8 | 1–3 | 8–13 | 92–127 | 88–123 | 50 | 5 | 177–361 | 36 |
| | Modilarity | 0.3461 | 0.0002 | 0.5522 | 0.4763 | 0.4841 | 0.5309 | 0.4563 | 0.3458 | 0.5151 |
| | $\sigma$ | – | 0.0015 | 0.0511 | 0.0157 | 0.0181 | – | – | 0.0031 | 0.0002 |
| Yeast | *k* | 30 | 104–131 | 16–24 | 329–354 | 325–350 | 10 | 4 | 199–656 | 87–118 |
| | Modilarity | 0.7021 | 0.6639 | 0.7291 | 0.6429 | 0.6450 | 0.5269 | 0.5192 | 0.5723 | 0.6556 |
| | $\sigma$ | – | 0.0258 | 0.0294 | 0.0021 | 0.0023 | – | – | 0.0045 | 0.0038 |
| Web_spam | *k* | 58 | 160–199 | 16–22 | 338–397 | 341–386 | 447 | 9 | 812–1581 | 169–175 |
| | Modilarity | 0.4700 | 0.1207 | 0.4767 | 0.4603 | 0.4616 | 0.2698 | 0.4297 | 0.3190 | 0.4014 |
| | $\sigma$ | – | 0.0447 | 0.0345 | 0.0233 | 0.0249 | – | – | 0.0065 | 0.0001 |
| Router | *k* | 72 | 1046–1106 | 33–49 | 1349–1402 | 1338–1389 | 749 | 14 | 44–253 | 129–214 |
| | Modilarity | 0.8779 | 0.6908 | 0.8937 | 0.6182 | 0.6199 | 0.7132 | 0.1720 | 0.6658 | 0.8019 |
| | $\sigma$ | – | 0.0033 | 0.0502 | 0.0002 | 0.0002 | – | – | 0.0172 | 0.0040 |
| Bio_dmela | *k* | 44 | 203–245 | 31–39 | 1347–1512 | 1324–1545 | 690 | 13 | 4065–4352 | 481–543 |
| | Modilarity | 0.4350 | 0.0548 | 0.4469 | 0.3470 | 0.3438 | 0.2905 | 0.0761 | 0.1070 | 0.3424 |
| | $\sigma$ | – | 0.0364 | 0.0252 | 0.0069 | 0.0068 | – | – | 0.0021 | 0.0067 |
| PGP | *k* | 215 | 1913–1994 | 64–77 | 2193–2271 | 2205–2263 | 1797 | 27 | 5233–6035 | 170–173 |
| | Modilarity | 0.8521 | 0.7317 | 0.8785 | 0.7113 | 0.7131 | 0.6878 | 0.7463 | 0.4861 | 0.8378 |
| | $\sigma$ | – | 0.0010 | 0.0102 | 0.0006 | 0.0005 | – | – | 0.0004 | 0.0004 |

**Acknowledgements** The authors of this paper are grateful to the National Natural Science Foundation of China (61673249, U1805263), the Natural Science Foundation of Shanxi (201801D121123)and the Research Project of Shanxi Scholarship Council of China (2017-014). The authors also gratefully acknowledge the anonymous referees for their constructive comments that improved this paper.

## References

1. Sara EG, Satu ES (2019) Community detection with the label propagation algorithm: a survey. Phys A 534:122058
2. Barabási AL, Albert R (1999) Emergence of scaling in random networks. Science 286(5439):509–512
3. Watts DJ, Strogatz SH (1998) Collective dynamics of 'small-world' networks. Science 393(6684):440–442
4. Fortunato S, Hric D (2016) Community detection in networks: a user guide. Phys Rep 659:1–44
5. Newman MEJ, Reinert G (2016) Estimating the number of communities in a network. Phys Rev Lett 117(7):078301
6. Chen Guanrong, Wang Xiaofan, Li Xiang (2015) Introduction to Complex Networks: Models, Structures and Dynamics. Higher Education Press, Beijing
7. Rolland T, Tasan M, Charloteaux B et al (2014) A proteome-scale map of the human interactome network. Cell 159(5):1212–1226
8. Bo Yang, Jiming Liu, Jianfeng Feng (2012) On the Spectral characterization and scalable mining of network communities. IEEE Trans Knowl Data Eng 24(2):326–337
9. Fortunato S, Barthélemy M (2007) Resolution limit in community detection. Proc Nat Acad Sci USA 104(1):36–41
10. Raghavan UN, Albert R, Kumara S (2007) Near linear time algorithm to detect community structures in large-scale networks. Phys Rev E 76(3):036106
11. Newman MEJ, Girvan M (2004) Finding and evaluating community structure in networks. Phys Rev E 69(2):026113
12. Liang Bai, Xueqi Cheng, Jiye Liang et al (2017) Fast graph clustering with a new description model for community detection. Inf Sci 388–389:37–47
13. Newman MEJ (2004) Fast algorithm for detecting community structure in networks. Phys Rev E 69(6):066133
14. Blondel VD, Guillaume J, Lambiotte R et al (2008) Fast unfolding of communities in large networks. J Stat Mech: Theory Exp 2008(10):P10008
15. Barber MJ, Clark JW (2009) Detecting network communities by propagating labels under constraints. Phys Rev E 80(2):026129
16. Xin Liu, Murata T (2010) Advanced modularity-specialized label propagation algorithm for detecting communities in networks. Phys A 389(7):1493–1500
17. Wei Li, Ce Huang, Miao Wang et al (2017) Stepping community detection algorithm based on label propagation and similarity. Phys A 472:145–155
18. Lusseau D, Newman MEJ (2004) Identifying the role that animals play in their social networks. Proc R Soc B Biol Sci 271(Suppl 6):S477–S481
19. Linyuan Lü, Tao Zhou (2011) Link prediction in complex networks: A survey. Phys A 390(6):1150–1170
20. Rosvall M, Bergstrom CT (2008) Maps of random walks on complex networks reveal community structure. Proc Natl Acad Sci USA 105(4):1118–1123
21. Danon L, Diazguilera A, Duch J et al (2005) Comparing community structure identification. J Stat Mech: Theory Exp 2005(9):P09008
22. Rand WM (1971) Objective criteria for the evaluation of clustering methods. J Am Stat Assoc 66(336):846–850
23. Wenping Zheng, Chenhao Che, Yuhua Qian et al (2018) A two-stage community detection algorithm based on label propagation. J Comput Res Dev 55(9):1959–1971
24. Lancichinetti A, Fortunato S, Radicchi F (2008) Benchmark graphs for testing community detection algorithms. Phys Rev E 78(4):046110
25. Zachary WW (1977) An information flow model for conflict and fission in small groups. J Anthropol Res 33(4):452–473
26. Newman MEJ (2006) Modularity and community structure in networks. Proc Natl Acad Sci USA 2006(103):8577–8582
27. Gavin M, Newman MEJ (2002) Community structure in social and biological networks. Proc Natl Acad Sci USA 99(12):7821–7826
28. Gui Yang, Wenping Zheng, Wenjian Wang et al (2017) Community detection algorithm based on weighted dense subgraphs. J Softw 28(11):3103–3114
29. Newman MEJ (2006) Finding community structure in networks using the eigenvectors of matrices. Phys Rev E 74(3):036104
30. Guimerà R, Danon L, Diazguilera A et al (2003) Self-similar community structure in a network of human interactions. Phys Rev E 68(6):065103
31. Yuhua Qian, Yebin Li, Min Zhang et al (2017) Quantifying edge significance on maintaining global connectivity. Sci Rep 7:45380
32. Rossi Ryan A, Ahmed Nesreen K (2015) The network data repository with interactive graph analytics and visualization. In: Proceedings of the twenty-ninth AAAI conference on artificial intelligence

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.