Physica A 551 (2020) 124137

Contents lists available at ScienceDirect

Physica A

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

Label propagation algorithm based on node importance

Tao Wang [a,*], Shanshan Chen [a], Xiaoxia Wang [b], Jinfang Wang [a]

[a] School of Mathematics and Physics, North China Electric Power University, Baoding, HeBei, 071003, China
[b] School of Control and Computer Engineering, North China Electric Power University, Baoding, HeBei, 071003, China

ARTICLE INFO

Article history:
Received 7 April 2019
Received in revised form 31 December 2019
Available online 3 January 2020

Keywords:
Label propagation
Community detection
Node importance
Complex networks

ABSTRACT

Label propagation algorithm (LPA) has attracted much attention due to its linear time complexity. However, there are disadvantages of uncertainty and randomness in the label propagation process, which may affect the stability and accuracy of community detection. In order to solve this problem, this paper proposes a novel label propagation algorithm based on node importance (NI-LPA). In the algorithm, a new index of node importance is presented which integrates the signal propagation of nodes, ***ks*** value of nodes themselves and *Jaccard* distance between adjacent nodes. The signal propagation considers the node importance from the perspective of network locality, the index reflects the position of nodes in the entire network, and *Jaccard* distance embodies the connection between nodes. The proposed index can fully reflect the node importance in the entire network. In the label propagation process, when the nodes with the maximum number of neighboring nodes are not unique, their labels are updated in terms of node importance. The proposed algorithm can avoid the instability caused by random selection in LPA algorithm. Experiments on real and synthetic networks show that NI-LPA can significantly improve the modularity of community and reduce the number of iterations. NI-LPA has better stability and accuracy than LPA.

1. Introduction

In the era of big data, complex systems such as biology [1], social science [2], and engineering [3] can be represented as networks with a common feature, i.e., community structure [4]. A community is defined as a connected subgraph whose nodes are densely connected with each other. In addition, nodes in each community have sparse connections to the rest nodes of network [5]. At the same time, communities in different types of networks reflect different meanings [6]. For example, communities in social networks are usually mapped to common interest, region, occupation, or background of real society. A community structure in social networks is obtained by analyzing users, marketing accurately, recommending via friends, etc. In biochemical networks, a community can be viewed as a functional unit in which community detection helps to better understand and develop the unit [7] such as gene bank analysis, food chain analysis in nature. Therefore, it is of great significance for structure analysis of large-scale complex network to detect community [8].

In recent years, many algorithms [9] have been presented to detect community. These methods can be roughly divided into two categories: heuristic methods and optimization methods [10]. The heuristic methods find the optimal partition by setting heuristic rules. The representative algorithms include GN algorithm [11] and hierarchical clustering algorithm [12,13]. The former calculates the betweenness of each edge in the network, removes the edge with the maximum number of edge betweenness, and then repeats until there are no edges in the network. The later calculates

[*] Corresponding author.
E-mail address: wtwxx@126.com (T. Wang).

https://doi.org/10.1016/j.physa.2020.124137
0378-4371/© 2020 Elsevier B.V. All rights reserved.

the similarity between nodes by similarity measure, gradually aggregates, and divides node set according to similarity in descending order. The optimization methods set objective functions and obtains optimal values of objective functions by iterations to detect community. In [14], genetic algorithm is employed and the modularity function is used as objective function to detect community structure.

GN algorithm, evolutionary algorithm, and hierarchical clustering algorithm are effective algorithms for dealing with community division. However, owing to high space and time complexity, these algorithms are not suitable for community detection in large-scale networks. Raghavan et al. [15] proposed a label propagation algorithm (LPA), whose time complexity is almost linear. LPA has good performance in terms of efficiency for dealing with large-scale networks. In addition, neither known scale and number of communities in advance nor a predefined objective function is needed in LPA algorithm, which makes it attract much attention from many scholars. In order to make it suitable for overlapping communities, Steve et al. [16] proposed COPRA algorithm, as an extension of LPA algorithm, for overlapping community detection. COPRA allows each node to retain multiple community labels, but still cannot solve the problem of randomness during the process of node label update. In order to improve the accuracy of LPA algorithm, Zhang et al. [17] proposed the LPAc algorithm based on edge clustering coefficients to improve label update process by calculating edge clustering coefficients. Barber et al. [18] proposed the constrained label propagation algorithm, which employs corresponding objective function to maximize modularity value. However, these algorithms cannot solve the problems of strong randomness and poor robustness in LPA.

The above algorithms ignore the influence of important nodes on label propagation, which may cause some unimportant nodes to overly affect some important nodes. Zhang et al. [19] calculated node importance based on Bayesian network and updated labels according to node importance to reduce the randomness of the algorithm. Jokar et al. [20] proposed the BLDLP algorithm considering weights for each edge, instead of random selection in LPA algorithm, to reduce uncertainty in the output of the algorithm. Xie et al. [21] used the distribution of position probability of random walker to measure the importance of nodes to obtain stable results. Xing et al. [22] proposed the NIBLPA algorithm, which improves node orders of label updating and the mechanism of label choosing to enhance the performance of LPA. However, most of these methods need to determine their parameters in advance. Li et al. [23] proposed the Stepping LPA-S algorithm, in which the label with the largest modularity is selected to propagate when the labels of the most similar neighbors are not unique. As a single index, the evaluation of the similarity index may be one-sided and cannot completely reflect node importance in the whole network.

This paper proposes a novel label propagation algorithm for community detection based on node importance (NI-LPA) in networks. In NI-LPA, a new index is proposed to measure node importance, which integrates signal propagation of nodes, *ks* value of nodes themselves and *Jaccard* distance between adjacent nodes. The signal propagation considers the node importance from the perspective of network locality, *ks* index reflects the position of nodes in the entire network, and *Jaccard* distance embodies the connection between nodes. The proposed index can fully reflect node importance in the entire network. As the maximum number of labels is not unique among the labels of neighbor nodes, all the neighboring nodes are sorted according to the importance index. Moreover, the label of neighboring node with the maximum importance index is selected to update the current node. Experiments on real and synthetic networks show that NI-LPA significantly improves the community modularity, reduces the iteration number, and enhances the accuracy and stability of the algorithm.

The rest of the paper is organized as follows. Section 2 introduces *Jaccard* distance, *ks* index, signal propagation amount, and node importance index. The main ideas and detailed procedure of the proposed algorithm are given in Section 3. Results on real and synthetic networks are illustrated in detail in Section 4. Finally, conclusion is given in Section 5.

2. Definition

Assuming that $G = (V, E)$ denotes an unweighted and undirected network graph, where V is the set of nodes, $|V| = n$ is the number of nodes, and E is the set of edges, $|E| = m$ is the number of edges. In the graph G, the adjacency matrix is denoted as $\boldsymbol{A}$, where the element a_{ij} is 1 if nodes i and j are linked; otherwise, a_{ij} is 0.

2.1. Neighborhood Jaccard distance

When a network is unweighted, the ability to propagate information is considered to be the same for all edges. However, in real networks, the ability to propagate information between nodes is different, and the potential influence between adjacent nodes is related to the coincidence degree of neighborhood corresponding to these nodes. Therefore, *Jaccard* distance is used to define the potential influence between two nodes.

$$d_{ij} = \frac{\left|N_i \cup N_j\right| - \left|N_i \cap N_j\right|}{\left|N_i \cup N_j\right|} \tag{1}$$

where d_{ij} denotes the *Jaccard* distance between adjacent nodes, that is, the coincidence degree of neighborhood corresponding to adjacent nodes. N_i and N_j represent neighbor node sets of nodes i and j, respectively. The larger *Jaccard* distance of adjacent nodes' neighborhood is, the more the number of non-common neighbor nodes between adjacent nodes is. It makes the information more likely to affect other nodes during the propagation process.

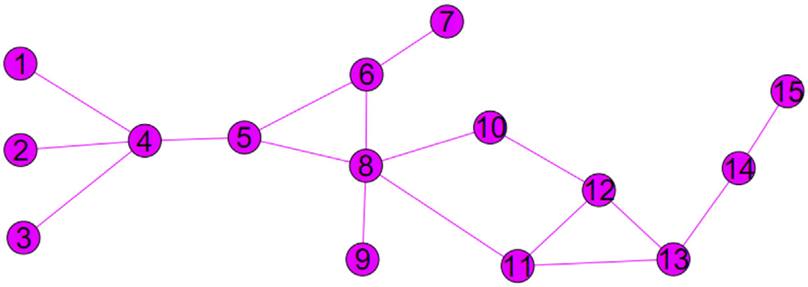


Fig. 1. *k*-shell decomposition example.

2.2. ks index

ks index is an important concept in graph theory, which reflects the position of a node in the whole network and is regarded as an improvement of node degree index. *ks* index is calculated by *k*-shell decomposition [24]. Nodes are divided hierarchically by *k*-shell decomposition according to the degrees and locations of nodes [25]. The method continuously strips nodes with degrees less than or equal to *k* in the network, and finally obtains a *k*-shell value, i.e., *ks* value, for each node. The specific process is as follows: firstly, delete the nodes and their edges whose node degree is 1 in the network, then re-check to determine if a new node appears whose node degree is 1. If it occurs, return to the previous step until nodes with degree 1 no longer appear in the new network. Namely, the *ks* value of the deleted nodes is 1 in the above deletion process. Similar to the above steps, continue to delete the nodes with the degree 2 in the remaining network, that is, the *ks* value of the deleted nodes is 2. After repeating the whole process, all nodes can obtain their *ks* values in the network. For example, it can be seen from Fig. 1 that, the *ks* value is 1 for nodes 1, 2, 3, 4, 7, 9, 14, and 15, the *ks* value is 2 for nodes 5, 6, and 13, and the *ks* value is 3 for nodes 8, 10, 11, and 12.

2.3. Signal propagation amount

The amount of signal propagation measures node importance from the perspective of network locality [26]. Each node in the propagation is used as a signal source to affect the entire network, and it sends the signal to its neighbor and itself. After τ steps, the amount of signal propagation can be used as the influence value of source node on the entire network.

$$\boldsymbol{S} = (\boldsymbol{A} + \boldsymbol{I})^{\tau} \tag{2}$$

where $\boldsymbol{I}$ is a n-dimensional identity matrix and τ is the total number of propagation steps. The element $s_{i,i}$ on the main diagonal of the matrix $\boldsymbol{S}$ represents the signal propagation amount of the ith node. In general, the amount of signal propagation between source node and nodes in its own community is greater than that between source node and nodes of other communities.

2.4. Node importance index

Complex networks contain a large number of nodes and their topology is complex [27]. Existing methods to measure node importance or influence show certain defects, such as limitations of single index, high computational complexity, and ignorance of relationship between nodes. In order to solve these problems, a new node importance index is defined in terms of its signal propagation amount, *ks* value and *Jaccard* distance among adjacent nodes.

$$h_{ms}(i) = s_{ii} * ks(i) * \sum_{j=1, j\neq i}^{n} a_{ij} d_{ij} ks(j) \tag{3}$$

where s_{ii} represents the signal propagation amount of node i, $ks(i)$ represents the k-shell value of node i, and d_{ij} represents the *Jaccard* distance between adjacent nodes. If nodes i and j are linked, the element a_{ij} is 1, otherwise 0. The larger the value h_{ms} of a node is, the greater the importance of the node is.

2.5. Label update method

Asynchronous updates can be a good solution to label oscillating problem [19]. Asynchronous update formula is shown as

$$c_j(t) = f\left(c_{j_1}(t-1), \ldots, c_{j_m}(t-1), c_{j_{m+1}}(t), \ldots, c_{j_k}(t)\right), j_i \in N_j \tag{4}$$

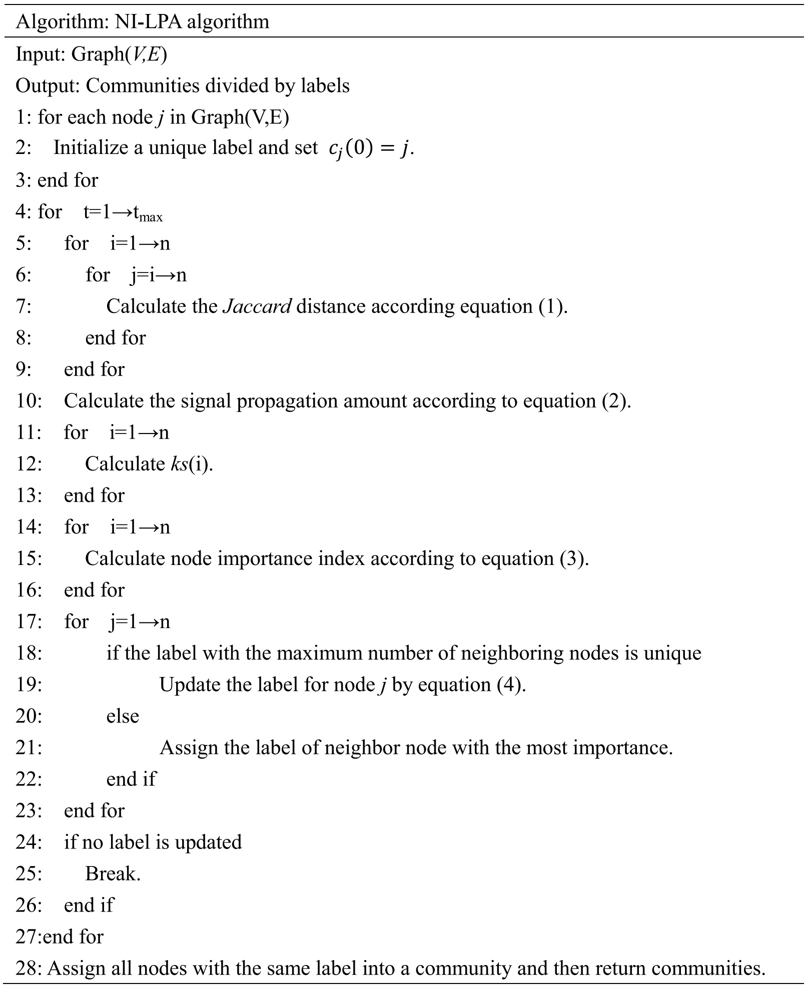
where f function denotes the most frequently occurring label among neighbor nodes of node j. $c_j(t)$ indicates the label of node j at the tth iteration, and $c_{j_m}(t-1)$ represents the label of the mth neighbor node of node j at the $(t-1)$th iteration. m is the label number of non-updated neighbor nodes of node j in this iteration.

3. Proposed method

In this section, a novel label propagation algorithm NI-LPA is presented in detail. In NI-LPA, the node's importance index is used to update node labels, which can greatly improve the accuracy and reduce the randomness of the algorithm. When the labels with the maximum number are not unique among neighbor nodes of one node during the label propagation process, the importance of each neighbor node is measured and then the node label is updated in terms of node importance, instead of random selection. The importance measure of neighbor nodes integrates signal propagation amount of each node, *Jaccard* distance between adjacent nodes, and the position of each node in the network.

3.1. Algorithm procedure

The proposed NI-LPA algorithm can be described as follows.
Input: undirected connected network $G = (V, E)$, the maximum number of iterations t_{max}
Output: a group of non-overlapping communities C.
Step 1: Initialize a unique label for each node j in the network, and set $c_j(0) = j$.
Step 2: Set the initial iteration number $t = 1$.
Step 3: Calculate the *ks* index of each node, the *Jaccard* distance of each pair of adjacent nodes, and the signal propagation amount of each node by Eqs. (1) and (2), respectively.
Step 4: Calculate the node importance h_{ms} for each node according to Eq. (3).
Step 5: Update the label of each node in the network according to Eq. (4). When the labels with the maximum number of neighboring nodes are unique, the node is updated by this label. Otherwise, the node is assigned the label of neighbor node with the most importance.
Step 6: Set $t = t + 1$.
Step 7: Repeat steps (5)–(6) until $t = t_{max}$ or the label of each node does not change anymore. Then the nodes with the same label are divided into a community, and the algorithm ends.



Algorithm: NI-LPA algorithm

```
Input: Graph(V,E)
Output: Communities divided by labels
for each node j in Graph(V,E)
    Initialize a unique label and set $c_j(0) = j$.
end for
for  t=1→t_max
    for  i=1→n
        for  j=i→n
            Calculate the Jaccard distance according equation (1).
        end for
    end for
    Calculate the signal propagation amount according to equation (2).
    for  i=1→n
        Calculate ks(i).
    end for
    for  i=1→n
        Calculate node importance index according to equation (3).
    end for
    for  j=1→n
        if the label with the maximum number of neighboring nodes is unique
            Update the label for node j by equation (4).
        else
            Assign the label of neighbor node with the most importance.
        end if
    end for
    if no label is updated
        Break.
    end if
end for
Assign all nodes with the same label into a community and then return communities.
```

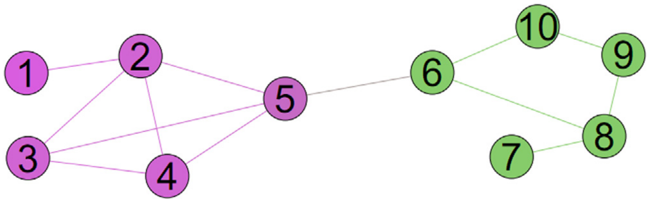


Fig. 2. A schematic representation of a network with community structure.

3.2. Example

In this section, NI-LPA is used to extract communities from a given network. An example network with 10 nodes and 13 edges is shown in Fig. 2.

Firstly, initial label is assigned for each node, $c_j(0) = j$.

Next, the *ks* index of each node is calculated. For nodes 1 and 7, their *ks* values are 1. For nodes 6, 8, 9, and 10, their *ks* values are 2. For nodes 2, 3, 4, and 5, their *ks* values are 3. The *Jaccard* distance between nodes and the signal propagation amount of each node are calculated according to Eqs. (1) and (2), respectively. For example, $d_{12} = 0.75$, $s_{11} = 4$.

Moreover, the importance index of each node is calculated using Eq. (3), the importance index values of ten nodes are 12, 376.2, 244.8, 244.8, 433.2, 150.4, 8, 185.87, 55.8 and 120.4, respectively

For node 1, the adjacent node label is only 2. So we choose label 2 as new label of node 1. For node 2, the adjacent node labels are 2, 3, 4 and 5, respectively, and the corresponding node importance indexes are 376.2, 244.8, 244.8 and 433.2, respectively. Because the labels with the maximum number of neighboring nodes are not unique, the node is assigned the label of neighbor node with the most importance. So we choose label 5 as new label of node 2. After repeating this process, all nodes can obtain their labels in the network.

When $t = 2$, the label of each node does not change anymore. The results of node labels are 5, 5, 5, 5, 5, 8, 8, 8, 8 and 8, respectively. The obtained communities are $C_1 = \{1, 2, 3, 4, 5\}$, $C_2 = \{6, 7, 8, 9, 10\}$.

4. Experimental results

In order to evaluate the effectiveness and efficiency of the proposed algorithm, NI-LPA is compared with traditional LPA on several real networks. These networks include Dolphins network [28], Polbooks network [29], and College football network [30]. Furthermore, NI-LPA is compared with five existing algorithms on a synthetic network [31]. In order to verify the rationality of *Jaccard* distance in NI-LPA algorithm, we compare resistance distance [32], weighted distance [33], novel distance [33] with *Jaccard* distance on Polbooks network. The parameters τ, t_{max} in the experiment are set to 3, 100, respectively. All the experiments are programmed and compiled under MATLAB on an Intel(R) Core(TM) CPU i5-3470 @ 3.20 GHz and 4 GB RAM computer running Microsoft Windows 7.

4.1. Dolphins network

Dolphins network is undirected which contains 62 nodes and 159 edges [27]. Fig. 3 shows Dolphins network divided into four communities by NI-LPA. Table 2 gives modularity values and corresponding iteration number of Dolphins network by LPA and NI-LPA.

To verify the effectiveness of NI-LPA, the modularity value Q [4] is utilized to measure the community division results. The greater the value of Q is, the more accurate the community detection results are. Simultaneously, the modularity variance indicates the fluctuation of Q value. The smaller the fluctuation range of Q, the smaller the iteration number corresponding to the modularity or the smaller the fluctuation of iteration number, the more stable community detection results. It can be seen from Table 1 that average value of modularity is 0.4633 for LPA, and 0.5176 for NI-LPA. Average modularity value of NI-LPA is larger than that of LPA, so the community detection results by NI-LPA are more accurate. Moreover, average iteration number of NI-LPA is less than that of LPA. The maximum modularity is 0.5069, and the minimum modularity is 0.4111 for LPA algorithm. There is a big difference among modularity values. On the contrary, the modularity value of NI-LPA is around 0.51, which is relatively stable. At the same time, the iteration number of corresponding modularity is 8 or 9 for NI-LPA, while that of LPA is unstable. Therefore, NI-LPA is more stable than LPA.

4.2. College football network

College football network consists of 115 nodes and 613 edges. Two nodes connected to each edge represent two teams playing together in this league [30]. Table 2 shows the modularity values and iteration number in the football dataset. Fig. 4 indicates college football network divided into five communities by NI-LPA. From Table 2, it can be observed that the modularity variance and iteration number of NI-LPA are smaller than that of LPA algorithm. Average iteration number of NI-LPA and LPA are 4.4 and 7, respectively. Compared with LPA, the stability of NI-LPA is improved. At the same time, average modularity values of NI-LPA and LPA are 0.6009 and 0.5653, respectively, so NI-LPA is more accurate than LPA.

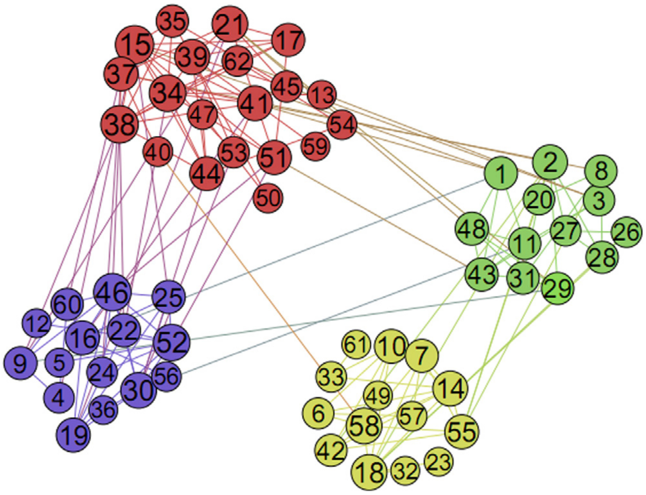


Fig. 3. Dolphins network detected by NI-LPA algorithm.

Table 1
Modularity and iteration number of Dolphins dataset.

Algorithms	Modularity	Iteration number	Modularity variance	Average modularity	Iteration number variance	Average iteration number
	0.4779	6				
	0.4907	16				
LPA	0.4111	10	0.0017	0.4633	16.8	9.4
	0.4301	6				
	0.5069	9				
	0.5079	8				
	0.5156	9				
NI-LPA	0.5223	9	4.7343×10^{-5}	0.5176	0.3	8.6
	0.5162	8				
	0.5258	9				

Table 2
Modularity and iteration number of Football dataset.

Algorithms	Modularity	Iteration number	Modularity variance	Average modularity	Iteration number variance	Average iteration number
	0.5507	7				
	0.5378	6				
LPA	0.5902	9	4.4171×10^{-4}	0.5653	1.5	7
	0.5778	7				
	0.5699	6				
	0.6043	4				
	0.5998	6				
NI-LPA	0.5969	4	1.1823×10^{-5}	0.6009	1.3	4.4
	0.5987	5				
	0.6046	3				

4.3. Polbooks network

Polbooks network includes 105 books (nodes) and 441 edges [29]. Nodes represent books about US politics sold by the online bookseller Amazon.com. Edges represent frequent co-purchasing of books by the same buyers. Table 3 displays the modularity values and iteration number of Polbooks network by LPA and NI-LPA. From Fig. 5, Polbooks network is divided into four communities by NI-LPA.

From Table 3, average modularity value of NI-LPA is larger than that of LPA, which highlights the advantage of NI-LPA in the community division. The modularity variance of NI-LPA and LPA are 1.8082×10^{-5} and 4.5327×10^{-4}, respectively. Compared with LPA, NI-LPA is more stable. Moreover, average iteration number of NI-LPA is smaller than that of LPA. At the same time, the iteration number variance of NI-LPA and LPA are 0.8 and 4.7, respectively. Obviously, NI-LPA has better stability.

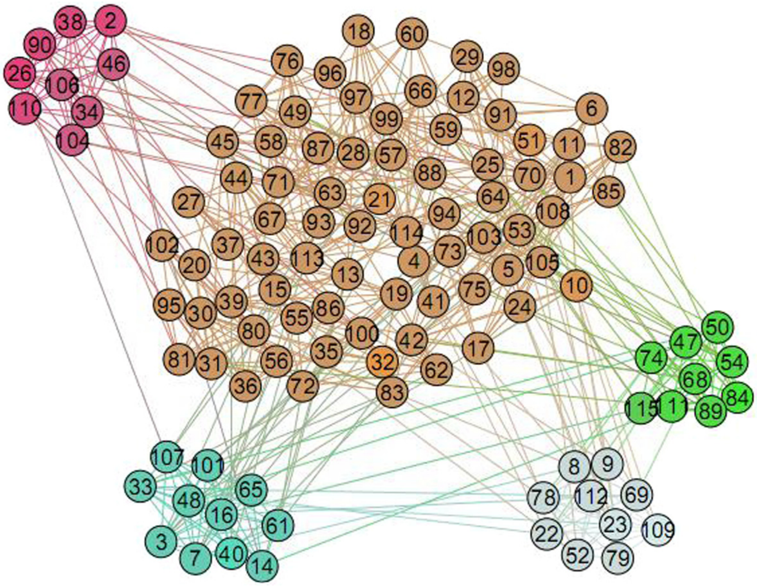


Fig. 4. Football network detected by NI-LPA algorithm.

Table 3
Modularity and iteration number of Polbooks dataset.

Algorithms	Modularity	Iteration number	Modularity variance	Average modularity	Iteration number variance	Average iteration number
	0.4725	4				
	0.4946	10				
LPA	0.4556	7	4.5327×10^{-4}	0.4625	4.7	6.8
	0.4424	6				
	0.4473	7				
	0.4882	4				
	0.4946	6				
NI-LPA	0.4986	6	1.8082×10^{-5}	0.4950	0.8	5.6
	0.4986	6				
	0.4949	6				

Table 4 gives the modularity values and iteration number under different distances on Polbooks network. NI-LPA algorithm with resistance distance is named as NI-LPA(1), NI-LPA algorithm with weighted distance as NI-LPA(2), and NI-LPA algorithm with novelty distance as NI-LPA(3). Average modularity value of NI-LPA is larger than those of other algorithms on Polbooks dataset, which highlights the advantage of NI-LPA in the community division. The modularity variance of NI-LPA, NI-LPA(1), NI-LPA(2), NI-LPA(3) and LPA are 1.8082×10^{-5}, 0, 5.0876×10^{-4}, 4.0909×10^{-4} and 4.5327×10^{-4}. The difference in modularity variance of different algorithms is much small. Moreover, the mean and variance of iteration number of NI-LPA is relatively low among 4 algorithms. In summary, *Jaccard* distance is more effective than other comparison distances.

4.4. Experiments on artificial network

The LFR benchmark network [31] was proposed by Lancichinetti, which is a widely used manual data. The degree of nodes and the size of community follow a power-law distribution. The importance of community structure is determined by the mixing parameter μ. The smaller μ is, the more prominent community structure is. The parameters used to constrain the topology of this network contains the number of nodes N, average degree $\langle k \rangle$, maximum degree k_{max}, mixing parameters μ, maximum size of community C_{max}, and minimum size of community C_{min}. The power-law distributions of node degree and community size are set to 2 and 1, respectively. In addition, normalized mutual information (NMI), as the standard for community division [34], is employed to measure the similarity between detected community and real community, ranging from 0 to 1. The closer the value of NMI is to 1, the closer the community is to the real community.

Table 5 gives the parameter settings for LFR benchmark. Fig. 6 displays the community division results on LFR generation network for GN [11], FN [35], LPA [15], BLDLP [20], NIBLPA [22], Stepping LPA-S [23], and the proposed NI-LPA at $N = 1000$, $N = 2000$, $N = 5000$ and $N = 10\,000$, respectively. As can be seen from Fig. 6, when $N = 1000$, the

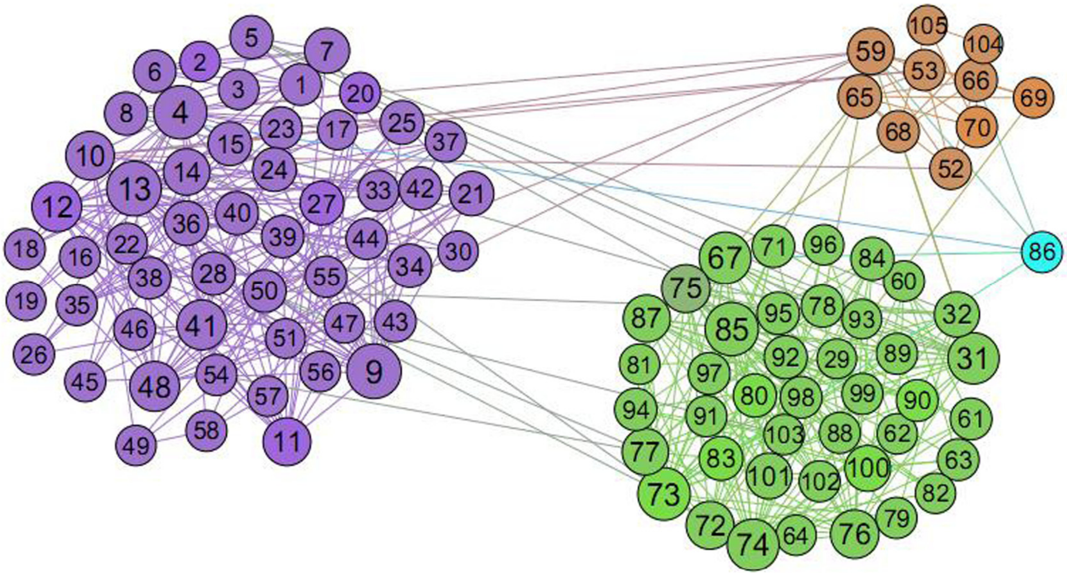


Fig. 5. Polbooks' network detected by NI-LPA algorithm.

Table 4
Comparison of different distances on Polbooks dataset.

Algorithms	Modularity	Iteration number	Modularity variance	Average modularity	Iteration number variance	Average iteration number
NI-LPA	0.4882 0.4946 0.4986 0.4986 0.4949	4 6 6 6 6	1.8082×10^{-5}	0.4950	0.8	5.6
NI-LPA(1)	0.4569 0.4569 0.4569 0.4569 0.4569	4 4 4 4 4	0	0.4569	0	4
NI-LPA(2)	0.4437 0.4424 0.4949 0.4568 0.4424	4 4 12 12 4	5.0876×10^{-4}	0.4560	19.2	7.2
NI-LPA(3)	0.4569 0.4946 0.4424 0.4569 0.4759	5 8 5 6 7	4.0909×10^{-4}	0.4653	1.7	6.2

NMI value of NI-LPA with $\mu \in [0, 0.5]$ is 1, and the community division result obtained by NI-LPA is consistent with real community. When $\mu > 0.5$, the NMI value obtained by NI-LPA is greater than those by GN, FN, LPA, BLDLP, NIBLPA, and Stepping LPA-S. When $N = 2000$ and $N = 5000$, the NMI value of NI-LPA with $\mu \in [0, 0.6]$ is 1. When $\mu > 0.6$, the NMI value of NI-LPA is greater than those of other comparison algorithms. When $N = 10\,000$, the NMI value of NI-LPA with $\mu \in [0, 0.7]$ is 1. When $\mu > 0.7$, the NMI value is significantly greater than those of other algorithms. In summary, NI-LPA can obtain much better performance than GN, FN, LPA, BLDLP, NIBLPA, and Stepping LPA-S. Therefore, NI-LPA algorithm is more suitable for community detection of large-scale networks.

Tables 6 to 9 show the iteration number for convergence for seven algorithms at $N = 1000$, $N = 2000$, $N = 5000$ and $N = 10\,000$, respectively. As can be seen from Table 6, NI-LPA algorithm can obtain the least number of iteration, i.e., about 3, at N = 1000, while the iterations number fluctuates greatly for other comparison algorithms. Table 7 gives the iteration number for seven algorithms at $N = 2000$. The iterations number is between 3 and 5 for NI-LPA algorithm, between 4 and 8 for GN algorithm and NIBLP algorithm, between 4 and 7 for FN algorithm, between 3 and 8 for LPA algorithm and BLDLP algorithm, between 3 and 6 for Stepping LPA-S algorithm, respectively. NI-LPA algorithm can obtain

Table 5
Parameter settings of LFR benchmark network.

N	k_{max}	τ_1	τ_2	C_{min}	C_{max}	$\langle k \rangle$	μ
1000	50	2	1	15	50	20	0.1–0.8
2000	50	2	1	15	50	20	0.1–0.8
5000	50	2	1	15	50	20	0.1–0.8
10 000	50	2	1	15	50	20	0.1–0.8

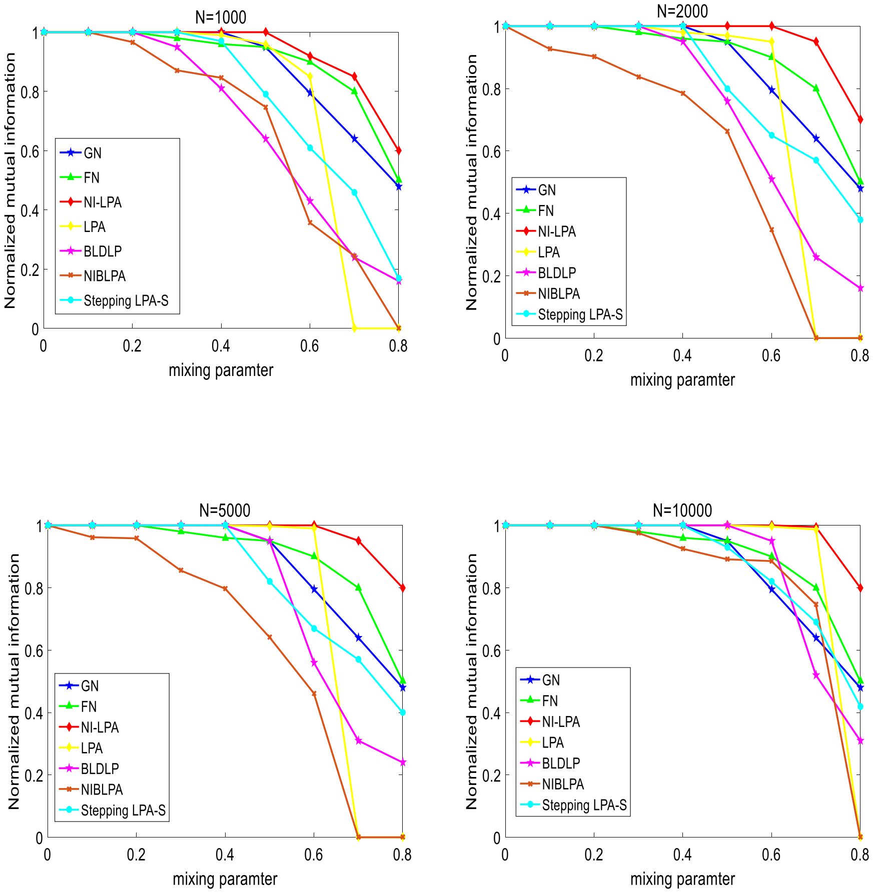


Fig. 6. Comparison of different methods on LFR benchmark.

more stable performance than other comparison algorithms. Similar results can be found when $N = 5000$ and $N = 10\,000$ from Tables 8 and 9. In summary, NI-LPA algorithm can obtain more stable performance.

4.5. Time complexity analysis

Suppose that n is the number of nodes and m is the number of edges. The time complexity of NI-LPA algorithm is estimated as follows.

(1) The time complexity of node initialization in step (1) is $O(n)$.

(2) The time complexity of calculating the *Jaccard* distance in step (3) is $O\left(n^2\right)$.

(3) The time complexity of calculating node importance in step (4) is $O(n)$.

(4) The time complexity of label update in step (5) is $O(m)$.

(5) The time complexity of assigning nodes with the same label to the same community in step (7) is $O(n)$.

Finally, the overall time complexity of NI-LPA is $O\left(n^2\right)$.

For BLDLP algorithm, the time complexity for calculating the weight of each node is $O(nd)$, while the time complexity is $2 \times O(n) + (2 \times t + 1) \times O(m) + O(n \log(n))$ for NIBLPA algorithm. Although it has slightly higher time complexity than NIBLPA and BLDLP, NI-LPA can obtain more stable and accurate results.

Table 6
Iterations number of seven algorithms with different mixing parameters at N = 1000.

Algorithm	μ							
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
GN	4	5	5	6	7	4	6	5
FN	4	3	5	4	5	5	4	4
NI-LPA	3	3	3	4	4	4	4	3
LPA	3	3	4	4	6	8	4	3
BLDLP	4	5	8	7	6	5	5	4
NIBLP	4	6	9	4	5	5	6	4
Stepping LPA-S	3	4	4	5	4	3	4	5

Table 7
Iterations number of seven algorithms with different mixing parameters at N = 2000.

Algorithm	μ							
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
GN	6	5	5	8	7	4	6	4
FN	5	4	5	4	7	5	4	6
NI-LPA	3	3	3	4	5	4	4	3
LPA	3	3	4	4	6	8	4	3
BLDLP	3	8	6	4	6	4	5	4
NIBLP	4	5	7	8	5	6	6	4
Stepping LPA-S	3	3	4	4	5	6	4	5

Table 8
Iterations number of seven algorithms with different mixing parameters at N = 5000.

Algorithm	μ							
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
GN	4	5	7	6	5	4	6	5
FN	6	4	8	4	5	5	4	4
NI-LPA	3	3	3	3	3	5	4	4
LPA	3	3	4	4	4	7	7	5
BLDLP	4	6	4	5	9	7	3	4
NIBLP	6	3	3	7	8	4	5	5
Stepping LPA-S	3	3	4	7	4	4	4	5

Table 9
Iterations number of seven algorithms with different mixing parameters at N = 10 000.

Algorithm	μ							
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
GN	6	8	10	9	7	7	18	6
FN	6	6	7	8	10	11	17	5
NI-LPA	3	3	3	3	3	4	12	4
LPA	3	3	4	4	7	9	28	5
BLDLP	4	5	8	7	6	5	19	4
NIBLP	4	6	6	4	9	9	23	4
Stepping LPA-S	3	4	4	5	4	3	15	5

5. Conclusions

This paper proposes a label propagation algorithm based on node importance. The node importance index is introduced to measure the importance of neighbor nodes. When the labels with the maximum number of neighbor nodes are not unique, the node label is assigned that of the most important node instead of random selection, which improves the stability of the algorithm. Experiments on real networks and synthetic networks show that the proposed algorithm has better performance in terms of stability and accuracy than existing algorithms.

Declaration of competing interest

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

References

[1] P. Sun, X. Sun, Complete graph model for community detection, Physica A 471 (2017) 88–97.
[2] X. Chen, J. Li, Community detection in complex networks using edge-deleting with restrictions, Physica A 519 (2019) 181–194.
[3] H. Jin, W. Yu, S. Li, A clustering algorithm for determining community structure in complex networks, Physica A 492 (2018) 980–993.
[4] Z. Yakoubi, R. Kanawati, LICOD: A leader-driven algorithm for community detection in complex networks, Vietnam J. Comput. Sci. 1 (4) (2014) 241–256.
[5] Y. Liu, J. Moser, S. Aviyente, Network community structure detection for directional neural networks inferred from multichannel multisubject EEG data, IEEE Trans. Biomed. Eng. 61 (7) (2014) 1919–1930.
[6] Q. Gui, R. Deng, P. Xue, X. Cheng, A community discovery algorithm based on boundary nodes and label propagation, Pattern Recognit. Lett. 109 (2018) 103–109.
[7] J. Reichardt, S. Bornholdt, Statistical mechanics of community detection, Phys. Rev. E 74 (2006) 016110.
[8] X. Yu, J. Yang, Z. Xie, A semantic overlapping community detection algorithm based on field sampling, Expert Syst. Appl. 42 (1) (2015) 366–375.
[9] S. Fortunato, Community detection in graphs, Phys. Rep. 486 (3–5) (2010) 75–174.
[10] D. Liu, D. Jin, D. He, J. Huang, J. Yang, B. Yang, Community mining in complex networks, J. Comput. Res. Dev. 50 (10) (2013) 2140–2154.
[11] M. Girvan, M.E.J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci. 99 (12) (2002) 7821–7826.
[12] V.D. Blondel, J.L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. Theory Exp. 2008 (2008) 10008.
[13] J. Eustace, X. Wang, Y. Cui, Community detection using local neighborhood in complex networks, Physica A 436 (2015) 665–677.
[14] L. Ma, M. Gong, J. Liu, Q. Cai, L. Jiao, Multi-level learning based memetic algorithm for community detection, Appl. Soft Comput. 19 (2014) 121–133.
[15] U.N. Raghavan, R. Albert, S. Kumara, Near linear-time algorithm to detect community structures in large-scale networks, Phys. Rev. E 76 (3) (2007) 036106.
[16] G. Steve, Finding overlapping communities in networks by label propagation, New J. Phys. 12 (10) (2010) 2011–2024.
[17] X. Zhang, X. Tian, Y. Li, C. Song, Label propagation algorithm based on edge clustering coefficient for community detection in complex networks, Internat. J. Modern Phys. B 28 (30) (2014) 15.
[18] M.J. Barber, J.W. Clark, Detecting network communities by propagating labels under constraints, Phys. Rev. E 80 (2) (2009) 026129.
[19] X. Zhang, J. Ren, C. Song, J. Jia, Q. Zhang, Label propagation algorithm for community detection based on node importance and label influence, Phys. Lett. A 381 (33) (2017) 2691–2698.
[20] E. Jokar, M. Mosleh, Community detection in social networks based on improved label propagation algorithm and balanced link density, Phys. Lett. A 383 (8) (2019) 718–727.
[21] C. Su, X. Jia, X. Xie, Y. Yu, A new random-walk based label propagation community detection algorithm, in: 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), Vol. 1, 2015, pp. 137–140.
[22] Y. Xing, F. Meng, Y. Zhou, M. Zhu, M. Shi, G. Sun, A node influence based label propagation algorithm for community detection in networks, Sci. World J. 2014 (2014) 1–13.
[23] W. Li, C. Huang, M. Wang, X. Chen, Stepping community detection algorithm based on label propagation and similarity, Physica A 472 (2017) 145–155.
[24] M. Kitsak, L.K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H.E. Stanley, H.A. Makse, Identification of influential spreaders in complex networks, Nat. Phys. 6 (2010) 888–893.
[25] J. Bae, S. Kim, Identifying and ranking influential spreaders in complex networks by neighborhood coreness, Physica A 395 (2014) 549–559.
[26] Rachel A. Hillmer, F. Katagiri, Toward predictive modeling of large and complex biological signaling networks, Physiol. Mol. Plant Pathol. 95 (2016) 77–83.
[27] X. Zhu, X. Li, S. Zhang, Z. Xu, L. Yu, C. Wang, Graph PCA hashing for similarity search, IEEE Trans. Multimed. 9 (9) (2017) 2033–2044.
[28] T. Wang, L. Yin, X. Wang, A community detection method based on local similarity and degree clustering information, Physica A 490 (2018) 1344–1354.
[29] F.D. Zarandi, M.K. Rafsanjani, Community detection in complex networks using structural similarity, Physica A 503 (2018) 882–891.
[30] X. Wang, J. Li, Detecting communities by the core-vertex and intimate degree in complex networks, Physica A 392 (10) (2013) 2555–2563.
[31] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (2008) 046110.
[32] T. Zhang, C. Bu, Detecting community structure in complex networks via resistance distance, Physica A 526 (2019) 120782.
[33] D. Wang, Y. Zhao, Network community detection from the perspective of time series, Physica A 522 (2019) 205–214.
[34] R. Guimera, M. Sales-Pardo, L.A. Amaral, Modularity from fluctuations in random graphs and complex networks, Phys. Rev. E 70 (2) (2004) 188.
[35] S.A. Shah, V. Koltun, Robust continuous clustering, Proc. Natl. Acad. Sci. 114 (37) (2017) 9814–9819.