EAI Endorsed Transactions
on e-Learning

Research Article EAI.EU

A Community Detection Algorithm Based on Balanced Label Propagation

H.J. Jia[1], T. Liu[2,]* and X.H. Zhang[1]

[1]School of Software, Henan Polytechnic University, Jiaozuo, Henan 454000, China
[2]College of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, Henan 454000, China

Abstract

OBJECTIVES: In conventional label propagation algorithms, the randomness inherent in the selection order of nodes and subsequent label propagation frequently leads to instability and reduces the accuracy of community detection outcomes.
METHODS: First, select the initial node according to the node importance and assign different labels to each initial node, aiming to reduce the number of iterations of the algorithm and improve the efficiency and stability of the algorithm; second, identify the neighbor node with the largest connection to each initial node for the pre-propagation of the labels; then, the algorithm traverses the nodes in descending order of the node importance for the propagation of labels to reduce the randomness of the label propagation process; finally, the final community is formed through the rapid merging of small communities.
RESULTS: The experimental results on multiple real datasets and artificially generated networks show that the stability and accuracy are all improved.
CONCLUSION: The proposed community detection algorithm based on balanced label propagation is better than the other four advanced algorithms on Q and NMI values of community division results.

Keywords: Community Detection, Node Importance, Community Merging, Balanced Label Propagation

Received on 04 April 2024, accepted on 07 May 2024, published on 16 July 2024

doi: 10.4108/eetel.5881

*Corresponding author. Email: 447768907@qq.com

1. Introduction

There are many complex systems in the real world that can be represented by complex networks [1]. Nodes in complex networks represent entities in the system and edges represent relationships between entities [2]. Complex networks usually have a community structure, which is characterized by tight connections between nodes in the same community, while sparse connections between nodes in different communities. Analyzing the community structure in networks plays a very important role in understanding and revealing the organizational principles and functions of complex networks [3,4]. Community detection has been widely used in protein. Structure and interaction analysis [5], product recommendation [6] and core drug discovery [7].

Label propagation algorithm [8] (LPA) has attracted much attention because of its linear time complexity in dealing with complex networks. The fundamental principle of the algorithm is to initially assign different initial labels to each node in the network and then update the node --- its own label --- according to the label that appears most frequently in the neighbor. When a label that appears most frequently in the neighbor is multiple, it is randomly selected for update. Finally the nodes with the same label are merged into the same community. However, the randomness inherent in the label propagation process reduces the stability and efficiency of the algorithm.

In order to reduce the randomness of label propagation, Zhang et al [9] incorporated the node importance into LPA and selected node labels based on the descending order of the node importance and label influence. Saeid et al [10] proposed a method to extend the community based on the node importance and local similarity. Kong et al [11] proposed a method to propagate labels based on the importance of nodes and label influence and introduced a new tightening function to propagate labels. Yue et al [12] defined a new label selection mechanism to update the labels of nodes. Deng et al [13] proposed constructing the node importance model based on K-shell algorithm and formulated new label updating strategy for label propagation. Zarezade et al [14] proposed hybrid node scoring and boundary node synchronized label updating method for selecting node labels. Thakare et al [15] proposed Skip-LPA algorithm. This algorithm initialized only some nodes for label propagation, which effectively reduced the number of iterations of the algorithm. Yuan et al [16] proposed the CDIC algorithm. The algorithm proposed an approach based on core node influence and label propagation, and merged nodes through the community's attraction to the nodes. Li et al [17] and Lin et al [18] combined the modularity function and the community core initialization to enhance the stability of the algorithmic results, respectively. Zhao et al [19] performed the algorithm's large-scale community detection by graph compression and label propagation in order to reduce the algorithm's time complexity. Roghani et al [20] proposed a new label propagation algorithm with local similarity metric to measure the importance of nodes. Bouyer et al [21] proposed an algorithm based on the community expansion of low degree nodes. Zhang et al [22] and Zhai et al [23] both proposed methods based on core nodes and their layer-by-layer label propagation.

Although the above work improves the label propagation algorithm through various methods, the problem of stability and accuracy of community detection due to its inherent randomness is still an open problem. To address the above issues, this paper proposes a community detection algorithm based on balanced label propagation. The algorithm first selects initial nodes based on node importance and assigns different labels to each initial node; second, in the neighborhood of each initial node, selects the neighbor node with the closest connection which is carried out for the pre-propagation of labels; then traverses the nodes in descending order of the node importance and updates the labels of the nodes based on the balanced label propagation rule; finally, in order to solve the problem that small communities may be wrongly divided in the process of community division and improve the accuracy of community detection, the algorithm merges the generated small communities that meet the merging conditions to form the final community. The main contributions of this paper are as follows.

(i) A new measure of the node importance and initial node selection rules are defined. By considering the number of connections of nodes and the closeness among its neighbors, combined with the local search strategy, effective initial nodes are selected to improve the rationality of community detection.

(ii) A new balanced label propagation method is proposed. The method prevents incorrect label assignment, effectively reduces the number of iterations and improves the stability of the algorithm.

(iii) Two community merging strategies are designed, Whereby the smaller-scale communities are merged to improve the accuracy of the community detection algorithm.

2. Related work

In this paper, an undirected unweighted graph $G = (V, E)$ is used to represent complex network. V is the set of nodes in the network, denoted as $V = \{v_i | i=1, 2, 3, ..., n\}$, while n is the total number of nodes in the network, and E is the set of edges, denoted as $E = \{(v_i, v_j) | v_i \in V, v_j \in E$ and $i \neq j\}$. $A(v_i, v_j)$ is 1 when node v_i is connected to node v_j, otherwise it is 0. The partitioning result of this network can be denoted as $C=\{C_i| i=1,2, \cdots, r\}$, r denotes the number of communities and C_r denotes the rth community.

Raghavan et al is the first to use the label propagation algorithm in the community detection problem in 2007, and the algorithm can be simply summarized as follows:

(i) Assign each node in the network a different label whose total number of labels is equal to the total number of nodes n;

(ii) Iteratively update the label for each node. In the iterative process, according to the random sequence of nodes, each node will receive the label information propagated by its neighbor nodes, and then update its own label to the one with the highest number of occurrences in the received label information; if there is more than one label with the highest number of occurrences, a label will be randomly selected as the label of the node; after a number of iterations, the change of labels in the neighbors of each node tends to stabilize;

(iii) Divide all nodes with the same label into a community.

The label propagation algorithm has linear time complexity, but the randomness of node selection and label update leads to the instability and low accuracy of community division results. For example, in the process of community division in Figure 1, no matter the update order of v_7-v_8-v_9 or v_9-v_8-v_7 is chosen, when the update order of v_7-v_8-v_9 is selected, if the label of node v_7 is updated to the label of node v_6 or v_9, the final community results are different. This indicates that the randomness of the update node order and the label selection process during label propagation process have very important impact on the community detection results.

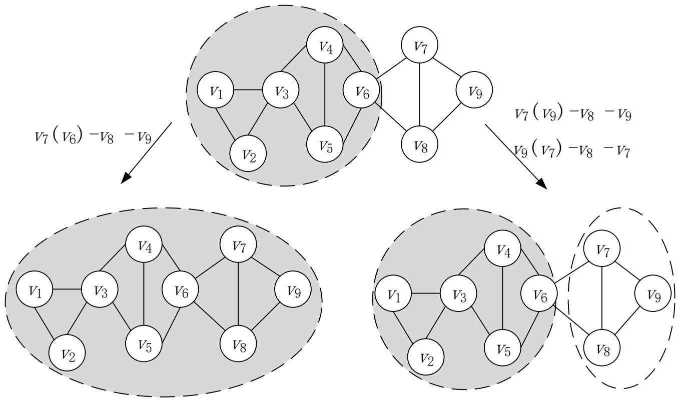

Figure 1. Community division

3. Proposed methods

Aiming at the problems of randomness in the propagation process and unstable community detection results of label propagation algorithm, this paper proposes a community detection algorithm based on balanced label propagation. During the initialization of node labels, the initial nodes are selected according to the node importance and only different labels are assigned to the initial nodes to improve the stability of the algorithm. During label propagation, node labels are updated according to different propagation rules. Finally, communities that meet the criteria of the small community merging strategy are merged to improve the overall quality, resulting in the formation of the final community structure.

3.1. Relevant definitions

To more clearly introduce the method proposed by this study, the following concepts are initially defined:

Definition 1. Neighbor nodes. $\forall v_i \in V$, $\forall v_j \in V$, if $(v_i, v_j) \in E$, then node v_i and node v_j are neighbor nodes to each other.

Definition 2. The set of neighbor nodes. $\forall v_i \in V$, the set of neighbor nodes of v_i is denoted as $N(v_i)$ $N(v_i) = \{v_j | v_j \in V, (v_i, v_j) \in E\}$.

Definition 3. Node degree. $\forall v_i \in V$, the size of the set of neighbor nodes of v_i is the degree of v_i, denoted as $D(v_i)$.

$$D(v_i) = |N(v_i)| \tag{1}$$

Definition 4. Node Connectivity tightness. Describe the extent of tightness of the connection between the two nodes, denoted as T.

$$T(v_i, v_j) = \sum_{k=1}^{m} \frac{1}{|p_k(v_i, v_j)|} \tag{2}$$

$\forall v_i \in V$, $\forall v_j \in N(v_i)$, $i \neq j$, the connectivity tightness between v_i and v_j is denoted as $T(v_i, v_j)$, which can be calculated according to formula (2). In this formula, m represents the number of all connection paths from v_i to v_j within three hops. $P_k(v_i, v_j)$ is the k-th connection path from node v_i to v_j. $|P_k(v_i, v_j)|$ represents the path length of $P_k(v_i, v_j)$, which the number of hops passed from node v_i to node v_j.

Definition 5. Node Importance. Describe the importance of a node within its neighborhood, denoted as I.

$$I(v_i) = D(v_i) + \frac{2 \sum_{v_j, v_k \in N(v_i)} A(v_j, v_k)}{D(v_i) \times (D(v_i) - 1)} \tag{3}$$

For $\forall v_i \in V$, the importance of node v_i is denoted as $I(v_i)$, which can be calculated according to equation (3).

Definition 6. Label Weight. Describes the strength of a label attributed to a node, denoted as W.

$$W(v_i, L_{v_j}) = \prod_{v_k \in N_{v_i}^{L_{v_j}}} I(v_k) \tag{4}$$

For $\forall v_i \in V$, $\forall v_j \in V$, the label weight of node v_j's label L_{v_j} on node v_i is denoted as $W(v_i, L_{v_j})$, which is calculated according to equation (4). In this equation, $N_{v_i}^{L_{v_j}}$ represents the set of nodes within the neighborhood of node vi that possess the label L_{v_j}.

3.2. Label initialization

Label initialization is the process of selecting several initial nodes in the network and assigning different labels to them, then start label pre-propagation in the neighborhood of each initial node. The steps to select initial nodes are as follows, where $V_{\text{candidate}}$ represents initial node candidate set and V_{initial} represents the initial node set.

Step1. For each node in V, if $I(v_j) > \frac{1}{n} \times \sum_{v_i \in V} I(v_i)$ is satisfied, v_j will be added to the $V_{\text{candidate}}$, where v_i is an arbitrary node in V, v_j is a node whose importance is greater than the average importance of all nodes in V, and n is number of nodes in V.

Step2. For each node v_i in $V_{\text{candidate}}$, if $I(v_i) \geq I(v_j)$ is satisfied, v_i will be added to the V_{initial}; otherwise, it will be removed from $V_{\text{candidate}}$, where v_i represents a node in $V_{\text{candidate}}$, and v_j represents any neighbor node of v_i.

Taking the initial node set in Figure 2 as an example, this paper describes the proposed label initialization process. Firstly, based on Step 1, the initial candidate set of nodes $V_{\text{candidate}}$, whose importance is greater than the average importance of all nodes in the network, is obtained. Then, according to Step 2, the initial node set $V_{\text{initial}} = \{v_1, v_{34}\}$, with the highest importance in their neighborhoods, is further selected from $V_{\text{candidate}} = \{v_1, v_2, v_3, v_4, v_9, v_{14}, v_{24}, v_{32}, v_{33}, v_{34}\}$. The grey nodes represent the initial nodes. The initial nodes selected through the above steps not only have the advantage of greater-than-average importance, which effectively enhances the propagation efficiency between network nodes, but also avoid excessively concentrated local connections, thereby enhance the robustness of the entire network.

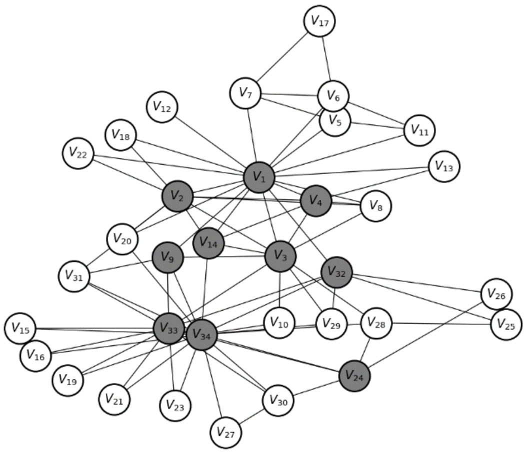

Figure 2. Initial node set

After the initial nodes is selected, the label pre-propagation starts. Assign different labels to each of initial nodes, find out the neighbor node with largest connection tightness in the neighborhood of each initial node and mark these neighbor nodes as the same label as the corresponding initial node. The label pre-propagation algorithm is as follows.

The label pre-propagation algorithm is shown as follows:

```
Input: Neighbor list of the nodes, network $G=(V, E)$;
Output: Pre-propagation node labels
1.Compute I(average) values for all nodes in network G.
2. For $v_i \in V$
3. $I(v_i) > I(\text{average})$, add $v_i$ to $V_{\text{candidate}}$
4. If $V_{\text{candidate}} \neq \emptyset$ and for $v_i \in V_{\text{candidate}}$:
5. If $I(v_i)$ is greater than that of all its neighbors:
6. Add $v_i$ to $V_{\text{initial}}$
7. Else:
8. Delete node $v_i$ from $V_{\text{candidate}}$
9. Assign different labels to each node in $V_{\text{initial}}$
10. For each $v_i \in V_{\text{initial}}$
11.Identify the neighbor $v_j$ with the highest connectivity tightness. The above label propagation algorithm can avoid the problem of inaccurate label propagation or excessive community expansion caused by relying solely on high-importance nodes, and ensure that the influence of nodes of different importance on label propagation can be balanced and add to $V_{\text{initial}}$
12. Propagate label of node $v_i$ to $v_j$
13. Return $V_{\text{initial}}$
```

3.3. Label propagation

During the process of label propagation, nodes are selected in a random order for label propagation. When a node to be updated has more than one label with the highest frequency in its neighborhood, the traditional LPA algorithm randomly selects one of them. This random selection of labels often leads to inconsistent community detection results on the same dataset. To address this issue, the algorithm in this paper no longer updates node labels in a random order, but instead updates them according to the importance of the nodes.

Initially, nodes are sorted in descending order of importance and stored in a list. If the node is unlabelled, the label is assigned to the node according to the label propagation rule 1; otherwise, the label is updated according to label propagation rule 2. If all the neighbor nodes of the node are unassigned labels, these marked neighbor node numbers are used as their temporary labels.

Label Propagation Rule1. If node v_i is unlabelled, first score the neighbor nodes, then calculate the scores for the labels that appear on the neighbor nodes based on these scores, and the label with the highest score is selected as the label of the node v_i.

If $I(v_i)$ of node *vi* is greater than the average importance of all nodes in the network is satisfied, each of its neighbor nodes will be scored according to equation(5), where $S(v_j)$ represents the score of neighbor node v_j;

$$S(v_j) = I(v_j) \times T(v_j, v_i), v_j \in N(v_i) \tag{5}$$

If $I(v_i)$ of node v_i is less than the average importance of all nodes in the network is satisfied, its individual neighbor nodes will be scored according to equation(6);

$$S(v_j) = D(v_j) \times D(v_i), v_j \in N(v_i) \tag{6}$$

Label scores of neighbor node are calculated for each label according to equation(7), where $S(L_{v_j})$ is the score of label L_{v_j}, L_{v_j} is the label of v_j, $N_{v_i}^{L_{v_j}}$ representing the set of nodes with L_{v_j} in the neighbor nodes of node v_i;

$$S(L_{v_j}) = \sum\nolimits_{v_k \in N_{v_i}^{L_{v_j}}} S(v_k) \tag{7}$$

Label Propagation Rule2. If node v_i is labelled, the most frequently occurring label among its neighbor nodes is selected to update the label of v_i; If the most frequently occurring label among the neighbors is not unique , then the label with the largest label weight is chosen to update the label of node v_i.

In this algorithm, nodes with a degree of 1 do not participate in the calculation process. After label propagation is completed, to improve the efficiency of the algorithm, these nodes will adopt the label of their sole neighbor as their own label.

The above label propagation algorithm can avoid the problem of inaccuracy in label propagation or excessive community expansion caused by solely relying on high-importance nodes, and ensure that the influence of nodes of different importance on label propagation can be balanced.

3.4. Community generation

When dividing communities based on the results of label propagation, some smaller communities are identified, which should actually be part of a larger community. To address this issue, the algorithm in this paper performs community merging on small communities that meet the merging conditions.

The algorithm firstly calculates the average size of all communities excluding the largest community, defines any community smaller than this average as a small community. This paper proposes two different strategies for community merging. By testing on multiple datasets, the most optimal result is selected. Where $\forall C_i \in C_{small}$, C_i represents the ith community, C_{v_i} represents the community where node v_i is located, and v_{r1} and v_{r2} are the nodes with and the largest importance ies, respectively.

Merging strategy 1. When v_{r1} is the node with the largest degree in the current small community, select the node with different labels and the largest degree from its neighbors. If degree of this neighbor node is greater than $D(v_{r1})$ is satisfied, the labels of all nodes in the small community in which v_{r1} is located will be updated to the label of this neighbor node;

Merging strategy 2. When v_{r2} is the node with the largest importance in the current small community, select the neighbor node v_j with different labels and the largest importance. If the difference between the internal edge density of $C_{v_{r2}}$ and twice the number of edges connecting $C_{v_{r2}}$ and C_{v_j} is less than 1 is satisfied, the labels of all nodes of the small community where v_{r2} is located will be updated to the label of neighbor node $v_{j.}$

In the merging of small communities, the two merging strategies mentioned above are used. If the conditions are satisfied, the small communities are merged and all the node labels in these communities are updated; otherwise, the community is not merged until all the communities in C_{small} have been filtered, and the final community detection is completed.

3.5. Algorithm description

The algorithm in this paper goes through label initialization, label pre-propagation, label propagation, community generation, and the merging of small communities to achieve the community division results. The pseudo-code of this algorithm is shown as follows:

```
Input: node list $V_{list}$, network $G = (V, E)$
Output: community set $C$
1. According to Algorithm 1, get the pre-propagated node labels.
2. Nodes are arranged within node list $V_{list}$ in descending order of node importance;
3. Initialize high pointer to point to the top node of the list
4. Initialize bottom pointer to point to the tail node of the list
5. While high and bottom do not point to the same node:
6. If (top and bottom point to nodes $v_i$ without labels)
7. According to Label Propagation Rule 1
8. Else:
9. According to Label Propagation Rule 2
10. Update top to point to the next node
11. Update bottom to point to the previous node
12. Assign the label of the node with degree 1 to its neighbor nodes
13. Nodes with the same label are grouped into the same community
14. If merge:
15. Merge small communities according to the two strategies for merging small communities
16. If merge strategy is satisfied:
17. Update small community node labels
18. Return community detection $C$
```

For initial node selection, the time complexity of computing N nodes is $O(N \times d^2)$, where d is the average degree of the nodes. The time complexity of the sorting operation during label propagation is $O(k \times \log(k))$, where k is the number of neighbors. The list of nodes with node degree 1 needs to be traversed and the time complexity of assigning labels to nodes with degree 1 is $O(m)$, where m is the number of nodes with degree 1. Small community merging needs to traverse all the communities, and the number of communities may be as large as N. Also limited by the number of communities C, the time complexity can be estimated as $O(C \times N)$. In summary, for dense networks the time complexity is $O(N \times d^2)$, while most real-world networks are usually sparse, so the actual time complexity may be lower than $O(N \times d^2)$.

4. Experiments

In order to verify the correctness and effectiveness of the algorithm proposed in this paper, comparative experiments are conducted with the LPA [8] algorithm, LSMD [21] algorithm, FSLD [24] algorithm and LBLD [20] algorithm on several real and artificial datasets. The operating system used in the experiments is Win10 and the processor is Intel (R) Core (TM) i7-7700HQ configured with 8GB of RAM.

4.1. Evaluation indicators

In order to evaluate the performance of each algorithm more accurately, this paper chooses two classical community evaluation indicators: NMI and Q. They are defined as follows:

(1) Normalized Mutual Information (NMI), which is used to measure the similarity between the real community structure A and the community structure obtained by the algorithm B. NMI takes the value in the range of [0, 1], in which the closer the value is to 1, the better the community segmentation is; on the contrary, the closer the value is to 0, the worse segmentation effect is indicated. The NMI metric is used to evaluate the similarity of the network community detection results with the real community detection, as shown in equation (8):

$$NMI(X,Y)=\frac{-2\sum_{i=1}^{C_X}\sum_{j=1}^{C_Y}C_{ij}\log\left(\frac{C_{ij}N}{C_iC_j}\right)}{\sum_{i=1}^{C_X}C_i\log\left(\frac{C_i}{N}\right)+\sum_{j=1}^{C_Y}C_j\log\left(\frac{C_j}{N}\right)} \qquad (8)$$

Where C is a matrix, rows represent real communities, columns represent communities obtained by running the algorithm, C_{ij} is the number of identical nodes of real community i and community j obtained by running the algorithm, C_i is the sum of the elements of row i, C_j is the sum of the elements of column j, C_X is the number of real communities in the network, and C_Y is the number of communities obtained by the algorithm.

(2) The modularity degree Q is used as an assessment of the delineation quality and density of the detected communities. Q takes values usually in [-1/2, 1], with larger values representing higher quality of community delineation, as shown in equation (9):

$$Q = \frac{1}{2|E|}\sum_{v_i,v_j}\left(A(v_i,v_j) - \frac{D(v_i)D(v_j)}{2|E|}\right)\delta\left(C_{v_i},C_{v_j}\right) \quad (9)$$

Where $|E|$ is the total number of edges in the network, $D(v_i)$, $D(v_j)$ represent the degree of v_i and v_j, respectively. if node v_i and node v_j have the same label, then $\delta(C_{v_i}, C_{v_j}) = 1$, otherwise, $\delta(C_{v_i}, C_{v_j})=0$.

4.2. Experimental datasets

This paper conducts comparative experiments on 9 real datasets and 9 synthetic networks. The specific information of the real datasets used in the experiments is shown in Table 1.

(1) Real datasets

Table 1. Data set information table

Networking	Nodal	Edge	Number of communities
Karate [25]	34	78	2
Dolphins [26]	62	159	2
Pollbooks [27]	105	441	3
Football [28]	115	613	12
Netscience [29]	1461	2742	-
PGP [30]	10680	24316	-
Condmat2003 [31]	31163	120029	-
DBLP [32]	317080	1049866	13477
Amazon	334863	925872	75149

(2) Artificial datasets

The artificial datasets used in the experiments were generated based on the LFR [33] benchmark generator with the following parameter configurations: the total number of nodes N=10000, the average degree of nodes k=20, the maximum degree max k=50, the minimum number of nodes in the community min C=20, and the maximum number of nodes max C=100, μ stands for the mixing parameter in the network, which takes the values in the range of [0.1, 0.5]. Each time it is increased by 0.05, a total of 9 artificial dataset networks are obtained. As μ increases, the structure of the network becomes fuzzier and community detection becomes more and more difficult.

4.3. Experimental datasets

In order to better verify the accuracy of this paper's proposed algorithm, the experiment selects real dataset networks of different sizes and compares them with four algorithms: LPA, LSMD, FSLD, and LBLD.

Table 2. Number of communities obtained by the algorithm on the real dataset

C	LPA	LSMD	FSLD	LBLD	Our_ Method
Karate	3	2	2	2	2
Dolphins	4	2	2	2	2
Polbooks	6	3	3	2	2
Football	8	12	8	8	10
Netscience	336	275	473	303	197
PGP	2495	643	302	358	330
Condmat_ 2003	5610	3502	2068	2314	1682
DBLP	58795	17280	14663	6846	15217
Amazon	63566	34303	49128	15501	17972

Table 2 presents the comparison of the number of communities (C) achieved by different algorithms on real datasets. From Table 2, it is evident that the algorithm proposed in this paper produces a number of communities that perfectly matches the actual communities in the Karate and Dolphins datasets. On the DBLP dataset, the C value of the community division results from this algorithm is the next closest to the actual community division compared to those of the other comparison algorithms.

Table 3. Comparison results of Q-values of algorithms on the real datasets

Q	LPA	LSMD	FSLD	LBLD	Our_ Method
Karate	0.34	0.37	0.37	0.37	**0.371**
Dolphins	**0.50**	0.378	0.37	0.378	0.378
Polbooks	0.45	0.44	0.44	0.45	**0.456**
Football	0.56	0.58	0.49	0.547	**0.60**
Netscience	0.901	0.940	0.802	0.935	**0.944**
PGP	0.664	0.585	**0.875**	0.815	0.843
Condmat_ 2003	0.543	0.565	0.634	0.695	**0.714**
DBLP	0.56	0.65	0.624	0.728	**0.736**
Amazon	0.59	0.68	0.725	0.80	**0.803**

Table 3 shows the comparison results of each algorithm for Q on the real dataset, where the bold represents the highest value of Q. According to Table 3, it can be seen that the algorithm in this paper is second only to the FSLD algorithm on the PGP dataset and to the LPA algorithm on the Dolphins dataset, but it performs equally well or better than the comparative algorithms on the other datasets.

Table 4. Comparison results of NMI-values of algorithms on the real datasets

NMI	LPA	LSMD	FSLD	LBLD	Our_Method
Karate	0.629	**1**	**1**	**1**	**1**
Dolphins	0.51	**1**	**1**	**1**	**1**
Polbooks	0.50	**0.59**	0.52	**0.59**	0.57
Football	0.81	**0.93**	0.89	0.825	0.90
DBLP	0.74	0.50	0.735	0.703	**0.753**
Amazon	0.93	0.72	0.945	**0.967**	0.963

Table 4 illustrates the comparison results of NMI values of the algorithms on real datasets, where the bold represents the highest value of NMI. According to Table 4, the NMI of the proposed algorithm on Dolphins is 1, indicating that its community detection results are completely consistent with the real community structure; the NMI value of this paper's algorithm is equal to that of LSMD algorithm, FSLD algorithm, and LBLD algorithm on the first two datasets. On the Polbooks and Amazon datasets, its NMI value is second only to the LBLD algorithm, and on the Football dataset, its NMI value is second to the LSMD algorithm. Moreover, on the DBLP dataset, its NMI value is higher than all comparison algorithms. Compared with other comparison algorithms, the results are improved by 1.8%, 50.6%, 2.4% and 7.1% respectively. Combined with the experimental data from Table 3, which proves that the number of communities identified by this paper's algorithm is closer to the actual situation, while the LPA algorithm tends to divide into too many smaller communities.

In order to compare the stability of LPA and the proposed algorithm, each is executed 30 times on the Dolphins dataset. The number of communities C and the modularity generated by the results of each run are shown in Figure 3 and 4.

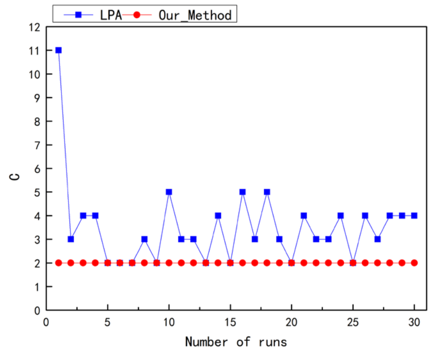

Figure 3. Change in the number of communities for multiple runs of the LPA and the algorithms of paper

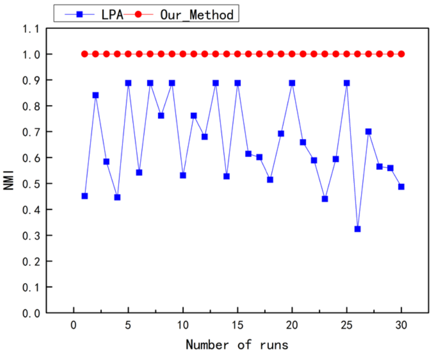

Figure 4. Change in NMI values calculated by LPA and the algorithm in this paper

As can be seen from Figure 3, the algorithm in this paper not only accurately detects two real communities, but also shows high stability and accuracy in multiple runs; while the LPA algorithm shows greater fluctuation, and generates some small communities during the operation.

As can be seen from Figure 4, the LPA algorithm produces large uncertainty and variability due to its own randomness. This randomness is reflected in the fact that node label updates tend to rely on the labels of neighbor nodes and when multiple labels with the same frequency appear in the neighborhood, the algorithm randomly selects a label to update. Although LPA is able to detect communities quickly, it does not always accurately identify the community structure in the network. In contrast, the algorithm proposed in this study is able to detect communities consistently and efficiently and the division results on the Dolphins dataset are consistent with the ground truth data.

In conclusion, the algorithm in this paper has high accuracy and greater stability on different real datasets with known and unknown communities, regardless of the size of the network.

4.4. Comparative experiments and analysis on artificial datasets

In order to further verify the performance of the algorithm proposed in this paper on artificial datasets, experiments are conducted using LFR to generate 9 artificial datasets. The NMI values of the algorithm in comparison with the 4 algorithms on 9 artificial datasets containing 10,000 nodes are shown in Figure 5.

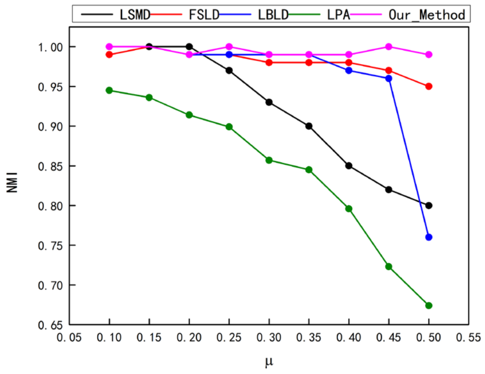

Figure 5. NMI comparison on artificial data set

As can be seen from Figure 5, when the value of μ is 0.2, the NMI value of this paper's algorithm is slightly lower than that of the LSMD algorithm; for other values of μ, the NMI values obtained by this paper's algorithm on the artificial dataset are greater than those obtained by the comparison algorithms. In the best case, this paper's algorithm obtains NMI values that are 46.88%, 23.75%, 4.2% and 30.26% higher than those obtained by the LPA algorithm, LSMD algorithm, FSLD algorithm and LBLD algorithm, respectively.

4. Conclusion

In this paper, we proposed a community detection algorithm based on balanced label propagation. The algorithm first selects initial nodes and assigns different labels only to the initial nodes to reduce meaningless labels in the subsequent label propagation process. During the label propagation process, corresponding propagation rules are used to propagate labels according to the node importance ranking to reduce the randomness of the algorithm. Finally, the rationality of community detection is enhanced by merging small-scale communities. Experimental results on several real and artificial datasets verify the effectiveness and accuracy of the method. However, when detecting communities in large-scale complex networks, the time required to compute node information or label weights becomes a significant challenge. Big data processing platforms such as Spark and Hadoop have demonstrated significant advantages in handling large-scale data. Therefore, exploring how to effectively utilize these platforms to improve the efficiency of community detection will be a key direction for future research.

References

[1] Shang R, Zhang W, Zhang J, et al. Local community detection based on higher-order structure and edge information[J]. Physica A: Statistical Mechanics and its Applications, 2022, 587: 126513.

[2] Teng X, Liu J, Li M. Overlapping community detection in directed and undirected attributed networks using a multiobjective evolutionary algorithm[J]. IEEE transactions on cybernetics, 2019, 51(1): 138-150.

[3] N. Papadopoulos A, Tzortzidis G. Distributed time-based local community detection[C]//Proceedings of the 24th Pan-Hellenic Conference on Informatics. 2020: 390-393.

[4] Midoun M A, Wang X, Talhaoui M Z. A jungle community detection algorithm based on new weighted similarity[J]. Arabian Journal for Science and Engineering, 2021, 46: 8493-8507.

[5] Zhang, Wei Tong. Complex network community detection and its application based on graph representation and label propagation [D]. Xi'an University of Electronic Science and Technology, 2021.

[6] Liu Wanjun. Research on the Application of Artificial Intelligence in Digital Archive Service [D]. Heilongjiang University, 2020.

[7] Zhang Y, Liu Y, Li Q, et al. LILPA: A label importance based label propagation algorithm for community detection with application to core drug discovery[J]. Neurocomputing, 2020, 413: 107-133.

[8] Raghavan U N, Albert R, Kumara S. Near linear time algorithm to detect community structures in large-scale networks[J]. Physical review E, 2007, 76(3): 036106.

[9] Zhang X, Ren J, Song C, et al. Label propagation algorithm for community detection based on node importance and label influence[J]. Physics Letters A, 2017, 381(33):2691-2698.

[10] Saeid A, Taghavi S A, Asgarali B, et al. A three-stage algorithm for local community detection based on the high node importance ranking in social networks[J]. Physica A: Statistical Mechanics and its Applications, 2021, 563125420-.

[11] Kong H, Kang Q, Liu C, et al. An improved label propagation algorithm based on node intimacy for community detection in networks[J]. International Journal of Modern Physics B, 2018, 32(25): 1850279.

[12] Yue Y, Wang G, Hu J, et al. An improved label propagation algorithm based on community core node and label importance for community detection in sparse network[J]. Applied Intelligence, 2023: 1-17.

[13] Deng Kaixuan, Chen Hongchang, Huang Ruiyang. Improved LPA algorithm based on label propagation capability[J]. Computer Engineering, 2018, 44(3): 60-64.

[14] Zarezade M, Nourani E, Bouyer A. Community detection using a new node scoring and synchronous label updating of boundary nodes in social networks[J]. Journal of AI and Data Mining, 2020, 8(2): 201-212.

[15] Thakare S B, Kiwelekar A W. Skiplpa: An efficient label propagation algorithm for community detection in sparse network[C]//Proceedings of the 9th Annual ACM India Conference. 2016: 97-106.

[16] YUAN Huilin, HAN Zhen, FENG Chong et al. A community discovery method based on core node influence[J]. Computer Science,2022,49(S2):240-246.

[17] Li H, Zhang R, Zhao Z, et al. LPA-MNI: an improved label propagation algorithm based on modularity and node importance for community detection[J]. Entropy, 2021, 23(5): 497.

[18] Lin Z, Zheng X, Xin N, et al. CK-LPA: Efficient community detection algorithm based on label propagation with community kernel[J]. Physica A: Statistical Mechanics and its Applications, 2014, 416: 386-399.

[19] Zhao X, Liang J, Wang J. A community detection algorithm based on graph compression for large-scale social networks[J]. Information Sciences, 2021, 551: 358-372.

[20] Roghani H, Bouyer A. A fast local balanced label diffusion algorithm for community detection in social networks[J]. IEEE Transactions on Knowledge and Data Engineering, 2022.

[21] Bouyer A, Roghani H. LSMD: A fast and robust local community detection starting from low degree nodes in social networks[J]. Future Generation Computer Systems, 2020, 113: 41-57.

[22] Zhang W, Shang R, Jiao L. Large-scale community detection based on core node and layer-by-layer label propagation[J]. Information Sciences, 2023, 632: 1-18.

[23] Zhai Zhenxin, Yu Yecheng, Gu Yu et al. A label propagation community discovery method based on layer-by-layer expansion of core nodes[J]. Computer and Digital Engineering, 2022, 50(06): 1327-1333+1346.

[24] Bouyer A, Azad K, Rouhi A. A fast community detection algorithm using a local and multi-level label diffusion method in social networks[J]. International Journal of General Systems, 2022, 51(4): 352-385.

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

[26] Lusseau D, Schneider K, Boisseau O J, et al. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations: can geographic isolation explain this unique trait[J]. Behavioral Ecology and Sociobiology, 2003, 54: 396-405.

[27] Newman M E J. Modularity and community structure in networks[J]. Proceedings of the national academy of sciences, 2006, 103(23): 8577-8582.

[28] Girvan M, Newman M E J. Community structure in social and biological networks[J]. Proceedings of the national academy of sciences, 2002, 99(12): 7821-7826.

[29] Newman M E J. Finding community structure in networks using the eigenvectors of matrices[J]. Physical review E, 2006, 74(3): 036104.

[30] Boguná M, Pastor-Satorras R, Díaz-Guilera A, et al. Models of social networks based on social distance attachment[J]. Physical review E, 2004, 70(5): 056122.

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

[32] Yang J, Leskovec J. Defining and evaluating network communities based on ground-truth[C]//Proceedings of the ACM SIGKDD Workshop on Mining Data Semantics. 2012: 1-8.

[33] Lancichinetti A, Fortunato S, Radicchi F. Benchmark graphs for testing community detection algorithms[J]. Physical review E, 2008, 78(4): 046110.