A Community Detection Algorithm Based on Jaccard Similarity Label Propagation

Meng Wang, Xiaodong Cai(✉), Yan Zeng, and Xiaoxi Liang

School of Information and Communication,
Guilin University of Electronic Technology, Guilin 541004, Guangxi, China
caixiaodong@guet.edu.cn

Abstract. Due to the randomness of label propagation algorithm, the stability is poor and the accuracy is low for community detection results in complex networks. In order to solve the problem, this paper proposes a novel community detection algorithm based on Jaccard similarity label propagation. Firstly, the Jaccard similarity is used to measure nodes importance. Then, the importance of nodes is utilized to reduce the randomness in label selection. Finally, nodes with the highest importance are selected to update labels in iteration, which improves the stability for community detection. Stability and accuracy of data sets of real networks are measured by modularity and normalized mutual information, respectively. Experimental results show that, the proposed algorithm for community detection results is more stable and more accurate than LPA, LPA_SI and KLPA algorithms in the cases of near linear time complexity.

Keywords: Complex network · Community detection · Label propagation · Jaccard similarity

1 Introduction

In nature and social activities, there are many universal connections and interactive relationships. This phenomenon is expressed with complex network models. Network nodes represent specific objects, while edges stand for links between objects, such as biology network [1], citation network [2] and social network [3]. It was found that community structures are a common feature of these networks. The community is composed of nodes with similar characteristics. Furthermore, the internal connection of communities is dense, the external correlation is relatively sparse [4]. Besides, it is important for research not only on the community structure in terms of biology, sociology and e-commerce, but also on applications in network security and criminal organizations recognition.

A large number of algorithms have been proposed with the widespread concern for community detection in complex networks. The classical community detection algorithms proposed graph segmentation, hierarchical clustering and modularity optimization method. In 2007, a fast label community detection algorithm is proposed based on label propagation [5]. The algorithm does not require any input parameters, such as the number and size of the community, and performs a linear time complexity. The efficiency of community detection results is improved extensively. It is suitable for

H. Yin et al. (Eds.): IDEAL 2017, LNCS 10585, pp. 45–52, 2017.
https://doi.org/10.1007/978-3-319-68935-7_6

large-scale complex networks. However, the label propagation algorithm is random in iterative update process, which leads to instability of the result for community detection. In order to solve the problem, the optimization and improvement methods were provided in many papers. In [6], a semi-synchronous label propagation algorithm overcoming the node oscillation problem in complex networks is proposed, but the stability is not improved for community detection. In LPA_SI [7], the stability of community detection is improved by integrating node importance and label influence. However, the computational complexity is large, which makes it difficult for large-scale complex networks to divide. In KLPA [8], the author combined the K value and local influence to achieve community detection, the transmission of label nodes is reduced. Nevertheless, the time complexity is increased, and the efficiency of large-scale networks for community detection is affected.

In this paper, a novel community detection method is proposed to solve the problem of randomness in label propagation algorithm. It is inspired by Jaccard similarity measure [9]. Firstly, the similarity is used to measure nodes importance. Then, the importance of node is utilized to reduce the randomness in label selection. Finally, the nodes with the highest importance are selected to update labels, which improves the stability for community detection.

2 Label Propagation Algorithm for Community Detection Based on Jaccard Similarity

2.1 Label Propagation Algorithm

In the literature [5], a community detection method is proposed based on Label Propagation Algorithm (LPA). Each node chooses the label with the largest number of neighbors as its own label by sending the label information between the node and its neighbors. After numerous iterations, node labels belong to the same community tend to be consistent. The main process of LPA is described as follows:

Input: Network model is $G = (V, E)$, where $V = \{v_i, i = 1, 2, 3 \ldots n\}$ represents the collection of nodes, set $C = \{C_1, C_2, \ldots, C_k\}$ denotes individual; where $E = \{e_i, i = 1, 2, 3, \ldots, n\}$ represents the collection of edges, $e_{ij} = \{(v_i, v_j), v_i \in V, v_j \in V\}$ denotes relationship.

Output: Set $C = \{C_1, C_2, \ldots, C_k\}$, where k represents the number of community.

(1) Each node v is assigned a unique label as the identity which stands for the community.
(2) An ordered list is generated by the random ordering of node sets.
(3) Following the ordered list, the node is updated. The label of each node v is updated by the label with the largest number of neighbors in the network. When there are multiple labels with the same maximum number, a label is chosen randomly as the label of the node.
(4) After several iterations, the label tends to be stable in each node neighbors.
(5) Nodes with the same label are classified to the same community.

In LPA, the order of each iteration is random. When a neighbor node has the same number of labels, a label is randomly selected to update, so that the stability of the community detection result is greatly affected. In the process of label propagation, the importance of the node itself is not considered, which leads to some less important nodes affect some of the more important nodes in turn, resulting in "counter current" phenomenon. In addition, the nearby small communities are swallowed by the first formation of community labels. Finally, a very large community is formed and even the labels of nodes in the entire network are the same.

2.2 Label Propagation Algorithm Based on Jaccard Similarity

In order to solve the "counter current" phenomenon in LPA, this paper considers the influence of the node importance and proposes an algorithm based on Jaccard similarity label propagation for community detection.

Related Definitions. The proposed algorithm is based on Jaccard similarity of label propagation. The related definitions are as follows:

Definition 1: Node Neighborhood.

$$T_{(v_i)} = \{v_i\} \cup \{(v_i, v_j) \in V\} \tag{1}$$

$T_{(v_i)}$ denotes the set of neighbors of node i.

Definition 2: Jaccard Similarity between Nodes.

$$S(v_i, v_j) = \frac{\left|T_{(v_i)} \cap T_{(v_j)}\right|}{\left|T_{(v_i)} \cup T_{(v_j)}\right|} \tag{2}$$

The degree of similarity is used to measure the tightness of any two neighboring nodes. In formula (2), $\left|T_{(v_i)} \cap T_{(v_j)}\right|$ is the number of communal neighbors between node v_i and v_j in networks. $\left|T_{(v_i)} \cup T_{(v_j)}\right|$ denotes the number of all neighbors between node v_i and v_j. When node v_i and v_j have more jointed neighbors, while the similarity is bigger, the node is more important to others. The range of $S(v_i, v_j)$ is $[0, 1]$.

Definition 3: Similarity between Nodes and Communities.

$$S(v_i, C_j) = \frac{\left|T_{(v_i)} \cap C_j\right|}{\left|T_{(v_i)} \cup C_j\right|} \tag{3}$$

The similarity between the node and community is used to measure the degree of the connection between nodes and adjacent communities. Where $\left|T_{(v_i)} \cap C_j\right|$ represents the

number of common nodes, $\left|T_{(v_i)} \cup C_j\right|$ denotes the number of all neighbors between node v_i and community C_j.

Definition 4: Node Importance.

$$\text{important}(v_i) = argmax\left\{S\left(v_i, C_j\right)\delta\left(v_i, C_j\right)\right\} \tag{4}$$

Where $\delta\left(v_i, C_j\right)$ is the Kronecker function, if v_i is identical to C_j, $\delta\left(v_i, C_j\right)$ equals to 1, otherwise equals to 0. When the node importance is bigger, the influence to others is larger. The label of node spreads more easily.

In the process of updating labels, the Jaccard similarity of adjacent nodes is firstly calculated according to formula (2). Based on the similarity descending sort, node labels are updated in order and the randomness is avoided. After several iterations, the node is updated when there are multiple adjacent label collections. The similarity between adjacent label collection is calculated according to formula (3). It represents the tightness of the association between label nodes and adjacent label sets. A higher tightness indicates that the node label is more important for the adjacent label sets. The probability is greater to be the label of the node updated. The most important label set is selected to update the node label by the formula (4) to avoid the "reverse flow" phenomenon in the process of label propagation. The formula of node label updated is as follows:

$$\text{LabelNew}(v_i) = \arg\max_{l}\{important(v_i)\} \tag{5}$$

Where l is the adjacent label collection of the node.

Algorithm Description. The steps of the algorithm for community detection are as follows:

Input: Complex network $\mathrm{G} = (\mathrm{V}, \mathrm{E})$, set the maximum number of iterations $t = maxIter$. Node v_i has n neighbors. $\mathrm{L}v_{i(t)}$ is the label of node v_i after the t^{th} iteration.

Output: $\mathrm{SetC} = \{C_1, C_2, \ldots, C_k\}$, where k represents the number of communities.

1. The label of each node $\mathrm{v} \in \mathrm{V}$ is initialized. Each node is assigned a unique label. The iteration time is initialized, $\mathrm{t} = 1$.
2. The similarity between the label nodes is calculated by formula (2). According to the similarity from high to low sort, an ordered list $V' = \{v_1, v_2, \ldots, v_n\}$ is generated.
3. According to the ordered list V', the label of node is updated. According to formula (3), the node is updated to the most important label.
4. If iteration $t = maxIter$ and $\mathrm{L}v_{i(t)}$ no longer changes, the nodes with the same label are grouped into same community. If $\mathrm{t} = \mathrm{t} + 1$, return to step 3.

Compared with LPA for community detection, in the proposed algorithm, the randomness of label propagation is reduced by considering the transmission characteristics of node labels and the tightness of connection among nodes. In addition, the influence of node is integrated, which makes it more accurate in label selection. The stability is improved for community detection.

3 Experimental Results and Analysis

To evaluate the proposed algorithm, real network data sets by Mark Newman [10] and the information of data sets is shown in Table 1 are used. The modularity and normalized mutual information are used as the measurement standard for community detection of these structure known data sets. For unknown structure data sets Hep-th and Internet, the modularity is used to test. The stability and accuracy of the proposed algorithm are analyzed, and compared by using the LPA [5], LPA_SI [7] and KLPA [8]. The time complexity is also analyzed. The experimental platform configuration consists of Intel (R) Core (TM) i5-4460 CPU @ 3.20 GHz processor, 8 GB memory, Microsoft Windows 7 operating system and JDK1.8, and Java programming language. In the testing process, to reduce the random caused by different operations, the average of 50 runs is taken. The maximum number iteration of the algorithm is set to 100.

Table 1. Data sets information.

Datasets	Node (N)	Edge (E)	Description
Karate	N = 34	E = 78	Zarchary's karate club network
Dolphins	N = 62	E = 159	Dolphin social network
Polbooks	N = 105	E = 441	Books about US politics network
Footballs	N = 115	E = 616	NCAA College-Football network
Hep-th	N = 8361	E = 15751	High-energy theory collaborations network
Internet	N = 22963	E = 48436	Internet at the level of autonomous systems network

3.1 Modularity

Modularity (Q) is the measure of quality for community structure proposed by Girvan and Newman [4], which can be used to measure the stability for community detection. It is defined as follows:

$$Q = \sum_{1}^{k} \left[\frac{link(C_j)}{m} - \left(\frac{\deg(C_j)}{2m} \right)^2 \right] \tag{6}$$

Where k is the number of community, $link(C_j)$ is the number of C_j, and $\deg(C_j)$ is the sum of all nodes in the community C_j, and m is the number of edges in the network. The value range of Q is [−1, 1]. If Q is larger, the quality is better for community detection and the result is more stable.

The result of Q with the proposed algorithm, LPA [5], LPA_SI [7] and KLPA [8] in different data sets are shown in Table 2. It can be seen that the value of Q is obtained by using the Karate, Dolphins, Polbooks, Footballs, Hep-th and Internet data sets. Since the algorithm considers the importance of the node itself, the value of Q is obviously higher than that of the LPA [5], and the stability of the algorithm is improved significantly. Compared with LPA_SI [7], it considers importance of nodes and influence of labels together, there is a slightly higher Q value of the three small data

Table 2. Comparison the modularity Q of the algorithm.

Datasets	Algorithm			
	LPA [5]	LPA_SI [7]	KLPA [8]	Proposed
Karate	0.338	0.395	0.382	0.397
Dolphins	0.465	0.512	0.506	0.524
Polbooks	0.474	0.521	0.583	0.601
Footballs	0.465	0.612	0.607	0.632
Hep-th	0.604	0.645	0.625	0.634
Internet	0.365	0.497	0.505	0.528

sets: Dolphins, Polbooks and Footballs. In addition, Karate data set of the Q value is similar and Hep-th is lower. However, the Q of the big data set Internet is higher. Compared with KLPA [8], the value of Q is higher than that of the LPA [5].

3.2 Normalized Mutual Information

The accuracy for community detection is evaluated in NMI [11, 12] by comparing the similarity between prediction and the true community structure. It is defined as follows:

$$NMI(p,r) = \frac{2MI(p,r)}{H(p)+H(r)} \tag{7}$$

Where $MI(p,r)$ denotes the mutual information between prediction community p and the real community r, and the formula is:

$$\mathrm{MI(p,r)} = \sum_i \sum_j P\left(C_{p_i} \cap C_{r_j}\right) log \frac{P\left(C_{p_i} \cap C_{r_j}\right)}{P\left(C_{p_i}\right)P\left(C_{r_j}\right)} \tag{8}$$

H(p) represents the entropy of the community p, the formula is:

$$\mathrm{H(p)} = -\sum_j P\left(C_{p_i}\right) \log\left(P\left(C_{p_i}\right)\right) \tag{9}$$

Where $P\left(C_{p_i}\right)$ and $P(C_{r_i})$ respectively represent the probability of a node in the network divided into the community p and in the real community r. The range of NMI is [0, 1]. The NMI is greater and the result is more accurate.

The result of NMI with the proposed algorithm, LPA [5], LPA_SI [7] and KLPA [8] in different data sets are shown in Fig. 1. The values of NMI is obtained by using the Karate, Dolphins and Footballs data sets. The NMI of the proposed are close to 1, which is larger than that of LPA, LPA_SI and KLPA algorithm. For the Dolphins data set, NMI does not reach 1. However, it is also higher than LPA, LPA_SI and KLPA. Experimental results show that, the proposed algorithm improves the accuracy of community detection.

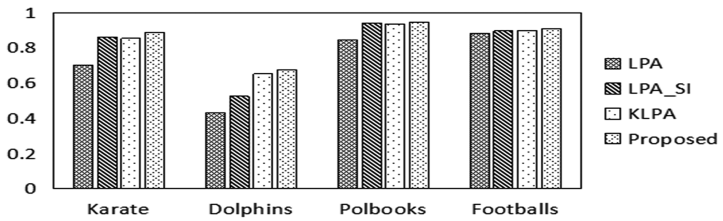

Fig. 1. The normalized mutual information (NMI).

3.3 Computational Complexity

If in a given network, n represents the number of nodes, and m represents the number of edges. All nodes in the network initialize the label, all nodes traverse once, the time complexity is $O(n)$. When calculating the similarity between nodes, all edges traverse once and the time complexity is $O(m)$. Similarity computing node according to importance and sorting node update, the time is required for most $O(n+m)$. The time complexity of each node is divided into $O(n)$, and the time complexity is $O(ms)$ after iteration s times. Thus, the total time complexity is $O(n+ms)$. Compared with the LPA algorithm, the time complexity increases, mainly on the ordering of nodes, but still close to the linearity. Compared with LPA_SI and KLPA, the time complexity is similar (Table 3).

Table 3. Computational complexity.

Algorithm	LPA [5]	LPA_SI [7]	KLPA [8]	Proposed
Computational complexity	$O(n+m)$	$(n+ms)$	$O(n+ms)$	$O(n+ms)$

4 Conclusion

The proposed algorithm integrates label propagation and the importance of nodes, and adopts the new node importance measurement for label iteration process. The improvement is shown by comparing the modularity and normalized mutual information of the LPA, LPA_SI and KLPA algorithms in small and large data sets of real networks. It can be confirmed that the stability and accuracy of the community detection are improved when the importance of node itself and the impact of labels in the label propagation algorithm are considered. With the increase of the number of nodes in networks, the performance of the proposed algorithm is further improved.

Acknowledgment. This work was supported by the 2016 Key Laboratory of Cognitive Radio and Information Processing, Ministry of Education Project No. CRKL160102 and 2016 Guangxi Science and Technology Project No. AB16380264.

References

1. Mering, V.C., Krause, R., Snel, B., et al.: Automated inference of highly conserved protein interaction networks. J. Nat. **417**(6998), 399–403 (2002)
2. Xiao, X., Chen, Y., Deng, Y.: Development of community discovery in citation networks. J. Inf. **35**(4), 125–130 (2016)
3. Yuta, K., Ono, N., Fujiwara, Y.: A gap in the community-size distribution of a large-scale social networking site. Physics (2007)
4. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. J. Phys. Rev. **E69**(2), 026113 (2004)
5. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large - scale networks. J. Phys. Rev. **E76**(3), 036106 (2007)
6. Cordasco, G., Gargano, I.: Community detection via semi synchronous label propagation algorithms (2011)
7. Huang, J., Guo, K., Guo, H.: Label propagation algorithm for community detection based on vertex significance and label influence. J. Chin. Comput. Syst. **36**(6), 1171–1175 (2015)
8. Deng, G.: Community detection base on mixed k-influence. J. Inf. Commun. **2**, 61–63 (2016)
9. Pan, L., Lei, Y., Wang, C., et al.: Method on entity identification using similarity measure based on weight of Jaccard. J. Beijing Jiaotong Univ. **33**(6), 141–145 (2009)
10. Mark Newman network data. http://www-personal.umich.edu/∼mejn/netdata
11. Danon, L., Diaz-Guilera, A., Duch, J., et al.: Comparing community structure identification. J. Stat. Mech. Theor. Exp. **2005**(09), P09008 (2005)
12. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure of complex networks. New J. Phys. **11**(3), 033015 (2009)