A Label Propagation Algorithm based on Circular Spread

Yong Wang, Xinzhen Fang, Jiahao Shi, and Jing Yang*

College of Computer Science and Technology, Harbin Engineering University, Harbin, 150001, China

Abstract

A label propagation algorithm has attracted widespread attention in community detection due to its linear time complexity. However, the traditional label propagation algorithm has a strong problem of randomness and may bring in backtracking during the process of label propagation; the result of finding the community is unstable and of low quality. This essay proposes a circular spread label propagation algorithm (CS-LPA), which takes full account of the structural characteristics of the community, introduces node influence measures, and discovers the potential community through the proliferation of labels that integrate the cyclic structure of social network. Finally, experimental results of real datasets show that CS-LPA not only enhances the stability of community detection results, but also effectively improves the quality of community detection.

Keywords: social network; community detection; label propagation; circular spread

(Submitted on July 10, 2018; Revised on August 14, 2018; Accepted on September 16, 2018)

© 2018 Totem Publisher, Inc. All rights reserved.

1. Introduction

With the rapid development of the Internet and the rapid spread of mobile smart devices, social network applications represented by WeChat, QQ, blogs, posts, and forums have become important carriers for maintaining social relationship and information dissemination. Social network has become the current social interaction and one of the main ways of communication that has greatly changed people’s lives and social habits [1]. The research on social network has important theoretical value and practical significance. Virtual community detection is one of the important contents of social network research. Through the analysis on the social network structure, the potential community in the network can further understand the structure and characteristics of the social network, help to better maintain the stability of social network and perceive user behavior therein [2].

In recent years, scholars have proposed many community detection algorithms by methods such as graph segmentation [3], hierarchical clustering, segmentation clustering, and spectral clustering [4-5]. For example, the GN algorithm [6], proposed by Newman et al, continuously removes the edge with the largest number of network mediations. However, the algorithm lacks the end condition and the time complexity is high. Kerninghan and Lin proposed a K-L algorithm [7] that constantly exchanges nodes to maximize the gain function by defining the gain function in the network. However, the algorithm needs to know the scale of the community in advance and is impractical in the practical network; Clauset proposed the CNM [8] algorithm through greedy selection and aggregation hierarchical clustering, which determines the community division of the network by adding the edges that increase the degree of modularity in order. There is also the Louvain algorithm [9] based on modularity optimization and the label propagation (LPA) algorithm [10]. The label propagation algorithm has good time complexity and can converge in near-linear time, but its random selection in node selection and label updating results in unstable results of community division and poor robustness of the algorithm are disadvantageous. In response to these problems, many scholars have made some improvements to the label propagation algorithm [11-13]. For example, Zhao et al proposed the concept of label entropy, and proposed an LPA-E algorithm to update nodes according to the order of label entropy [14]. Zhang et al uses node similarity to deal with the problem of random selection of labels, and Deng et al uses k-shell decomposition algorithm to introduce node influence to handle label propagation process. Barber designed the objective function of label propagation and proposed the LPAM algorithm [15].

* Corresponding author.
E-mail address: paperworkharbeu@163.com
Although the above algorithms solve the problems in the label propagation algorithm to a certain extent, it does not fully consider the structural characteristics of the community and the potential links with the node properties. Retrospective phenomena occur from time to time, and there are still problems with unstable division results and low quality.

For this reason, this essay proposes a label propagation algorithm based on circular spread. It fully considers the characteristics of the circular structure of the community, adopts pruning strategies for pre-processing, introduces node influences to measure, sort and group nodes, and selects the center of the network. Some (the more influential point) spread the label, and then spread out to the circular and integrate the importance of the node into the label's update strategy. This reduces the randomness of the algorithm and improves the quality of the community. Experiments show that the Label Propagation Algorithm based on Circular Spread proposed in this essay has greater improvement in the quality and stability of community detection than the traditional LPA algorithm and other improved LPA algorithms.

2. Label Propagation Algorithm based on Circular Spread

2.1. The Circular Structure of the Community

By studying the structural characteristics of social network, in a complex community under real conditions, the most influential nodes usually have higher degree of nodes, and have higher degree of node neighbors. These nodes are often distributed in the center of each community. The influence of other nodes in the community is inversely proportional to the distance to the central node; that is, the nodes in the network are usually distributed in a circle based on the size of influence. The circular distribution of nodes in the community is shown in Figure 1.

![Figure 1. The circular distribution of nodes in the community](image)

Using the circular distribution characteristics of community nodes, in the process of label propagation, if the label of the inner layer is determined earlier, the label replacement efficiency of the edge nodes in the network can be accelerated, thereby improving the efficiency of the algorithm. This essay uses the LeaderRank algorithm as a measure of the influence of nodes in the network [16]. LeaderRank algorithm is an improved PageRank algorithm, which can be used to calculate the degree of importance of nodes in the social network [17]. When the algorithm is running, we set a threshold value \( \eta \). According to the size of the node's LeaderRank, the nodes are divided into \( \eta \) groups. When the labels are propagated, the group with the greatest influence is first operated so that the central part of each community is first determined, and then the rings are successively Outward spread.

In the label propagation algorithm, the node label is replaced with the label with the largest label number in the neighbor node, which can be considered as a process in which a node integrates into its neighbor node community. According to the concept of node influence, it can be understood that the nodes in the network are affected by the nodes of the node with greater influence, and then integrated into their communities. Therefore, in the process of label propagation, we first select the nodes with low influence to carry out label propagation. When a neighbor node encounters the same number of label in the label update process, it does not select a label randomly, but selects a label with greater influence. This not only meets the objective rule, but also avoids repeated changes to the label of the node in the network. The “backtracking” phenomenon of node label loop change is effectively avoided.

As shown in Figure 2, the Karate Club dataset records the relationships among the 34 members of the American University’s karate club. We use the traditional label propagation algorithm to divide the community and assign node 2 to the community centered on node 1. For example, if the subsequent node update is selected in a poor order, nodes 3 and 20
were first assigned to communities with node 33 and 34 as the core. This will cause the label to update node 2 and assigned it to the community centered on node 33 and 34. However, with the continuous iteration of the label propagation algorithm, node 2 will eventually be assigned to the community centered on node 1. It can be considered that the continuous “backtracking” changes of the node labels will cause the labels of the nodes to break out of the correct division for a time, which affects the efficiency of the algorithm.

Figure 2. The “backtracking” phenomenon in the process of label propagation

2.2. Algorithm

Based on the above idea, this essay presents a label propagation algorithm based on circular spread algorithm (CS-LPA). The algorithm is divided into two parts. First, we preprocess the nodes in the network, filter out leaf nodes with a degree of one in the network. We also calculate the LeaderRank of the remaining nodes while the nodes are divided into η groups according to the LeaderRank value. The first group is the collection of the most influential nodes, which basically includes the core parts of the community. Then, according to the influence of the groups from large to small, the influence of the nodes within the group has a small to large order to update the labels. The detailed description of the algorithm is as follows:

Algorithm 1 CS-LPA Algorithm

Input: Social network \( G = (V, E) \) Number of groups \( \eta \)

Output: Community division results

1. Prune the network, Get network \( G_1 \) without leaf nodes, Leaf node collection \( T \)
2. Each node is grouped \( C_1, C_2, \ldots, C_\eta \) according to the size of the LeaderRank value;
3. while the Algorithm does not converge to do
4. Calculate the number of times different labels appear on the neighbour nodes of node \( i \)
5. \( C_i = \{ C_{i1}, C_{i2}, \ldots \} \) // \( C_i \) is the label with the most occurrences of the neighbor node of node \( i \)
6. \( \text{for } i \in \eta \text{ do} \)
   \( L_i = \arg \max \sum_{j \in C} LR(j) \) // The label of the leaf node is replaced with the label of its neighbor node
7. \end while
8. \( L_T = L_{\text{neighbor}} \) // LAB is the label of each node, and the nodes with the same label are the same community
9. \( \text{LAB} = \{ L_1, L_2, \ldots \} \)
10. return LAB

The main time consumption of the algorithm is respectively the calculation of the influence of the node and the label propagation process. When using the LeaderRank to calculate the influence of the node, the time complexity of this step is \( O(tN) \), Where \( t \) is the number of iterations of the algorithm, the time complexity of each traversal of the label propagation process is \( O(N) \), and the Algorithm usually converges within less than 4-5 cycles. Therefore, the time complexity of the process is less than \( O((5+\tau)N) \), so CS-LPA algorithm is a near-linear time complexity algorithm.

3. Experiments and Analysis

The following experiment compares and analyses CS-LPA, classic LPA, LPA-E and LPAM algorithms proposed in this essay. Four real network datasets and artificially generated datasets are run, compared and analyzed. (Random algorithm takes average over multiple experiments).

In this essay, the performance of the algorithm is evaluated by the two indicators of modularity [18] and normalized mutual information (NMI) [19]. The modularity reflects the closeness of the community and the NMI indicates the degree
of similarity between the results of the community detection algorithm and the real community.

3.1. Experiments on Real Network Datasets

Four real web datasets (Karate, Dolphins, Polbooks, and Football) are selected in this experiment, which are widely used to verify the quality of community detection. The specific information of the data set is shown in Table 1.

<table>
<thead>
<tr>
<th>Network</th>
<th>Number of nodes</th>
<th>Number of edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Karate</td>
<td>34</td>
<td>78</td>
</tr>
<tr>
<td>Dolphins</td>
<td>62</td>
<td>159</td>
</tr>
<tr>
<td>Polbooks</td>
<td>105</td>
<td>441</td>
</tr>
<tr>
<td>Football</td>
<td>115</td>
<td>616</td>
</tr>
</tbody>
</table>

3.2. Experimental Results and Analysis

3.2.1. Partitioning Effect of Real Network Data Set

Use CS-LPA to classify 4 real network datasets; the results are shown in Figure 3 to Figure 6. The Karate dataset is a network of members of the Zachary Karate Club. The dataset represents the social relationship of the club members. The CS-LPA algorithm divides the karate club dataset into two communities, representing groups with different coaches as the core. The Dolphins dataset is a dolphin network of 62 bottlenose dolphins of two families recorded by Lusseau et al. The ties of dolphins of the same family are closer together, forming a community structure in the network. Polbooks is a network dataset abstracted from Amazon's book sales records. Its nodes represent American politics-related books sold on Amazon's online bookstore. They also purchase these books on behalf of a certain number of readers. The Football dataset is a complex social network created by Newman based on the American College Football League. The network contains 115 nodes and 616 sides. The 115 participating college teams are divided into 12 leagues. The teams of the same league contact each other closely; the result of community division shows the results of the division of the alliance.

Figure 3. Comparison before and after Karate dataset partitioning

Figure 4. Comparison before and after Dolphins dataset partitioning
3.2.2. Circular Spread Update

Selecting the algorithm and traditional LPA algorithm to compare the number of label updates on a real network dataset, as shown in Table 2, we can see that the proposed CS-LPA algorithm compared to the traditional label propagation algorithm, due to the use of circular diffusion for label updates, greatly reduces the “backtracking” phenomenon, which decreases the number of label updates. The karate club network has a small number of nodes and a simple structure. Only one traversal can make all nodes obtain the correct community label. The circular diffusion update method used in this algorithm can effectively improve the efficiency of the algorithm.

As shown in Table 3, when the traditional label propagation algorithm is used to divide the community of the karate dataset, the algorithm usually requires three iterations. The degree of modularity and NMI after each iteration are shown in Table 3. The CS-LPA proposed in this essay has adopted a circular diffusion label. The method of propagation can achieve convergence with only one iteration, and the quality of the community's findings are superior to traditional label propagation algorithms.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>LPA</th>
<th>CS-LPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Karate</td>
<td>45</td>
<td>32</td>
</tr>
<tr>
<td>Dolphins</td>
<td>82</td>
<td>59</td>
</tr>
<tr>
<td>Polbooks</td>
<td>141</td>
<td>130</td>
</tr>
<tr>
<td>Football</td>
<td>147</td>
<td>132</td>
</tr>
</tbody>
</table>
Due to the instability of the original label propagation algorithm at runtime, the division result of each community of the algorithm will vary greatly. The results of the original label propagation algorithm and the CS-LPA algorithm are shown in Table 4. The results show that the effect of the LPA algorithm is very different for each branch of the operation. The second time is the invalid division of the modularity of 0.133. Because the CS-LPA adopts the circular spread label propagation method, the community division results are relatively stable.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Modularity</th>
<th>NMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>LPA</td>
<td>0.362/0.479</td>
<td>0.133/0.233</td>
</tr>
<tr>
<td>CS-LPA</td>
<td>0.372/1.00</td>
<td></td>
</tr>
</tbody>
</table>

3.2.3. Algorithm Runtime Comparison

The experiment compares the time required for the algorithm LPA, CS-LPA, LPA-E, Walktrap, Leading Eigenvector and GN, to divide the community between the Karate dataset and the Dolphins dataset. All algorithms are implemented in Python and the experimental environment is Windows 10, Intel Core i5-3210M CPU @ 2.50 GHz, and 8 GB of memory. In Figure 7, we can see that the traditional LPA algorithm requires the least time and has the fastest speed. The CS-LPA and LPA-E algorithms increase the node influence calculation and grouping and label entropy calculation respectively. However, this only requires linear time. The time required for the random walk algorithm mainly depends on the number of edges in the network. In these two sparse real data sets, the time complexity is less than the Leading Eigenvector. The time complexity of the GN algorithm is $O(mn^2)$. Community division takes the most time. With the increased number of data points and edges, it is not obvious that the time required for the community classification of LPA, CS-LPA, LPA-E is a near-linear algorithm. The time required for the algorithms such as Walktrap, Leading Eigenvector and GN increases significantly. The time required by the GN algorithm with the highest time complexity increases most; that is, as the time complexity of the algorithm increases, the running time of the algorithm increases more and more significantly. All the LPA, CS-LPA, and LPA-E algorithms have near-linear time complexity, and the community finds higher efficiency.
Compare the results of community partitioning on the four real network datasets of the CS-LPA, traditional LPA, LPA-E, and LPAM algorithms; the results are shown in Table 5 and Figure 8.

Compared with the traditional LPA algorithm, we can see that CS-LPA algorithm divides the community modularity and standardized mutual information more. The result of the division of the karate club is exactly the same as the real division result, LPA-E and LPAM. The algorithm is also greatly improved compared to the LPA algorithm, but the effect is slightly worse than the CS-LPA algorithm.

Table 5. Comparison of Algorithms on Real Network Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>LPA Modularity/NMI</th>
<th>LPA-E Modularity/NMI</th>
<th>LPAM Modularity/NMI</th>
<th>CS-LPA Modularity/NMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Karate</td>
<td>0.351/0.575</td>
<td>0.370/0.611</td>
<td>0.347/0.995</td>
<td>0.372/1.00</td>
</tr>
<tr>
<td>Dolphins</td>
<td>0.425/0.381</td>
<td>0.446/0.385</td>
<td>0.480/0.675</td>
<td>0.483/0.697</td>
</tr>
<tr>
<td>Polbooks</td>
<td>0.471/0.327</td>
<td>0.477/0.435</td>
<td>0.483/0.577</td>
<td>0.483/0.577</td>
</tr>
<tr>
<td>Football</td>
<td>0.455/0.720</td>
<td>0.472/0.764</td>
<td>0.532/0.890</td>
<td>0.547/0.920</td>
</tr>
</tbody>
</table>

Figure 8. (a) Comparison of real dataset module degree; (b) NMI Comparison of Real dataset

3.2.5. Artificial Network Dataset Test Results

LFR Benchmark procedure [20] is a program proposed by Lancichinetti et al. to generate an artificial simulation network. The program generates a community result with a landmark; at the same time, the baseline network is generated. The LFR reference network is widely used to test the performance of community discovery algorithms. It can generate networks that conform to user-specified distributions to be more realistic and more flexible to test the performance of community discovery algorithms.

Table 6. LFR Benchmark network parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Number of nodes</td>
</tr>
<tr>
<td>MinN</td>
<td>Minimum node degree</td>
</tr>
<tr>
<td>MaxN</td>
<td>Maximum node degree</td>
</tr>
<tr>
<td>MinS</td>
<td>Minimum number of nodes</td>
</tr>
<tr>
<td>MaxS</td>
<td>Maximum number of nodes</td>
</tr>
<tr>
<td>mu</td>
<td>Obvious degree</td>
</tr>
</tbody>
</table>

The algorithm of this paper is verified and compared in the four networks. The NMI index of the community is compared under different mu-valued conditions. The network specific parameter settings are shown in Table 7.

Table 7. Artificial network dataset parameter settings

<table>
<thead>
<tr>
<th>Network</th>
<th>parameter settings</th>
<th>N</th>
<th>MinN</th>
<th>MaxN</th>
<th>MinS</th>
<th>MaxS</th>
<th>mu</th>
</tr>
</thead>
<tbody>
<tr>
<td>N₁</td>
<td></td>
<td>2000</td>
<td>20</td>
<td>50</td>
<td>20</td>
<td>100</td>
<td>0.20-0.60</td>
</tr>
<tr>
<td>N₂</td>
<td></td>
<td>4000</td>
<td>20</td>
<td>50</td>
<td>20</td>
<td>100</td>
<td>0.20-0.60</td>
</tr>
</tbody>
</table>
The NMI curves of the four algorithms varying with the mu value in the four networks are shown in the figures below:

As shown in Figure 9 and Figure 10, the four algorithms can better define the community structure when the mu value is low. But, as the mu value increases, the community structure gradually becomes indistinct and the quality of the algorithm divides into communities. In the network with \(N=2000\), due to the poor stability of the traditional LPA algorithm, the traditional LPA algorithm cannot be divided into community results as the mu value increases, indicating that the traditional LPA algorithm is difficult to apply to the community with a relatively vague community structure. In the other three networks, although the traditional LPA algorithm can identify the community structure of the network, it results in a lower NMI index and a poor quality of division when the mu value is high. The LPA-E, LPAM and CS-LPA algorithms can identify the community structure of the network under different mu values of the four networks. When the mu value is low and the community structure is relatively obvious, all three algorithms can identify the community structure of the network with high quality. However, with the increase of mu value, the advantages of LPAM and CS-LPA algorithm are more obvious than LPA-E algorithm. When the effect of LPA-E algorithm is obviously decreased, LPAM and CS-LPA algorithm still have high division results, and CS-LPA algorithm is superior to LPAM algorithm in dividing quality. It can be seen that the CS-LPA algorithm guarantees the quality of the community division based on the elimination of the randomness of the algorithm.

4. Conclusion

With an aim to solve the problems of randomness and backtracking in traditional label propagation algorithms, a new
circular spread-based label propagation algorithm (CS-LPA) is proposed. The algorithm effectively integrates the structural characteristics of the social network with the attributes of the nodes, effectively solves the problem of poor community detection quality and weak stability of the label propagation algorithm, and significantly improves the quality of community detection. The work of this essay is as follows: 1) The LeaderRank value is introduced to measure the influence of the node. After the packet is sorted, the network center node is extracted and the randomness of the label propagation is eliminated; 2) Combined with the characteristics of social network structure, a Circular spread strategy to spread labels is adopted. The strategy fully considers the influence of nodes in the process, and solves the problems of community division quality; 3) Experiments and analysis show that the CS-LPA algorithm enhances the stability of label propagation and improves the quality of community classification while retaining the near-linear time complexity of the label propagation algorithm.

Acknowledgement

The research work was supported by The Youth Foundation of Heilongjiang Province of China under Grant No. QC2016083, the Fundamental Research Funds for the Central Universities under Grant No. HEUCF180606, the Innovative Talents Research Special Funds of Harbin Science and Technology Bureau under Grant No. 2016RQQXJ128, and the National Natural Science Foundation of China under Grant No. 61672179, 61370083 and 61402126.

References

Yong Wang was born in 1983. Now he is a lecturer at College of Computer Science and Technology, Harbin Engineering University, and the member of CCF. His research interests include social network Analysis, data mining, etc.

Xinzhen Fang was born in 1994. He is an M.S. candidate at College of Computer Science and Technology, Harbin Engineering University. His research interests include data mining and Social Network Analyzing, etc.

Jiahao Shi was born in 1997. He is an undergraduate at College of Computer Science and Technology, Harbin Engineering University. His research interests include social network Analysis, etc.

Jing Yang was born in 1962. Now she is a professor and Ph.D. supervisor at College of Computer Science and Technology, Harbin Engineering University, and the senior member of CCF. Her research interests include privacy protection, social network Analysis and data mining, etc.