Research on the Influence Maximization Based on Community Detection

Kai Sheng[1, 2], Zhi Zhang[1, 2]

[1]Collage of Computer Science and Technology, Wuhan University of Science and Technology,

[2]Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System

Wuhan, China, 430065

*Corresponding email:345157979@qq.com

Abstract: The research of influence maximization aims to select the k most influential nodes in social network, these nodes are a group of initial seed nodes for information dissemination and make the ultimate scope of influence maximum. Considering that Community-based greedy algorithm has two problems. One is that community division is unstable, and the other is that the time complexity of selecting seed nodes on the basis of community division is too high , this paper proposes an influence maximization algorithm (LPIMA) based on community detection. The algorithm selects optimized label propagation algorithm for community detection. Firstly, we use LeaderRank algorithm to quantify the influence of community nodes. And then assign candidate seed node set according to these quantified values. Finally, we use the submodel characteristic to improve greedy algorithm and mining community seed nodes from candidate seed sets. The results show that the proposed algorithm guarantees the scope of influence and improves the time efficiency in the large-scale network.

Keywords: Label Propagation; Community Detection; Greedy Algorithm; Influence Maximization

I. INTRODUCTION

The study of influence maximization[1] began at word of mouth effect and viral marketing, it has become one hotspots of social network research in recent years. It has important practical significance in advertising marketing, public opinion monitoring, virus transmission and information recommendation and other fields. The research of influence maximization aims to select the k most influential nodes in social network, these nodes are a group of initial seed nodes for information dissemination and make the ultimate scope of influence maximum. With the rapid development of social network such as Facebook, Flickr, Weibo, wechat and Douban. It is challenging to maximum influence in complicated large scale network.

The issue of influence maximization has proved to be NP-hard, a non-deterministic problem of the complexity of polynomials. Since the model of influence maximization has been proposed, many scholars have done relevant research. A number of influence maximization algorithms are proposed according to the independent cascade [2],linear threshold [3] and other improved models. However, all the algorithms can be classified into the following categories, greedy algorithm [4] and its improvement strategy, heuristic algorithm [5] and the influence maximization algorithm based on community structure [6-8]. The main idea of greedy algorithm is make best decision under current circumstances. In a sense, the result of the greedy strategy solution can be regarded as the local optimal solution. The algorithm of CELF(Cost-Effective Lazy Forward- selection) [9] use submodel characteristic of influence to improve greedy strategy, which reduces the unnecessary calculation of marginal influence increment of some nodes. The CGA algorithm(Community-based greedy algorithm) [10] reduces the network processing size by community partitioning, which reduces the time complexity by one order of magnitude. But this method still needs to simulate the diffusion of all nodes to obtain the marginal influence increment when selects the core node in the community. Due to the ignorance of the influence differences between nodes, it is unable to reduce the execution time of selecting core nodes in community.

Aiming at the problem of high time complexity of existing greedy algorithms, this paper proposes an influence maximization algorithm(LPIMA) based on label propagation community detection. This method uses LeaderRank algorithm to quantify the influence of community nodes, assigns candidate seed node set according to these quantified values. And improved greedy strategy by submodel characteristic of influence.

II. Community Detection Algorithm and LeaderRank Sorting Algorithm

In this section, we will introduce the Community Detection Algorithm and LeaderRank sorting algorithm.

A. Community Detection Algorithm

This paper refers to an Multi-label propagation algorithm for overlapping community detection based on LeaderRank (LRMLPA) [13] in the community detection phase. The algorithm is divided into five stages, which are calculating the importance of nodes, the generation of rough clique, the initialization of node labels, label propagation phase and result processing phase. Firstly, the algorithm quantifies the importance of nodes in the network by LeaderRank algorithm, then expands the nodes according to their importance to obtain a plurality of overlapped rough clique. Each rough clique is assigned a unique label. The nodes in the same clique have the same label, isolated nodes not in a clique are assigned a unique label, which propagates from the node in the rough clique to the outer layer. The label update adopts the synchronous update mode, so after the first round of iteration, the labels of all the isolated nodes are updated, and then the next iteration is performed according to the result of the random ordering of the nodes, and the iteration process is ended until the termination condition of the label propagation is reached. Classify all the nodes with the same label into one community, de-weight and merge the resulting community, reduce the meaningless community, and finally output the community findings.

978-1-5386-3758-6/18/$31.00 ©2018 IEEE

LRMLPA algorithm pseudo-code described in Table 1 below:

Table. 1 LRMLPA Algorithm

Input: Figure $G(V,E)$, the most influential rough clique LRRC

Output: The corresponding node set for each community: communites

```
oldLabelMap = InitLable();
newLabelMap ← ∅;
rcNode ← ∅;
foreach RCi in LRRC:
    rcNode ← rcNode ∪ RCi;
outRcNode ← V − rcNode;
rcNode ← rand_sort(rcNode);
outRcNode ← rand_sort(outRcNode);
sortNode ← outRcNode ∪ rcNode;
foreach vertex v in sortNode:
    Propagate(v, oldLabelMap, newLabelMap);
sortNode ← rand_sort(V);
foreach vertex v in sortNode:
    Propagate(v, oldLabelMap, newLabelMap);
if not stopCond(oldLabelMap, newLabelMap):
    oldLabelMap ← newLabelMap;
    goto line 12;
communities ← removeSub(oldLabelMap);
splitDiscontinuous(communities);
return communities;
```

B. LeaderRank Sorting Algorithm

The LeaderRank algorithm is an improvement of the PageRank algorithm [14]. It adds a ground node g to the network and connects it with all the nodes in the network to change the network into a new strong connection network.

The algorithm first assigns one unit LR value (ie, LeaderRank value) to all nodes except node g in the graph, and then updates the LR values of all nodes according to formula (1). The process iterates until convergent. Finally, Fixed LR value according to formula(2)

$$s_i(t+1) = \sum_{j \in N(x)} \frac{s_j(t)}{k_j} \tag{1}$$

$$S_i = s_i(t_c) + \frac{s_g(t_c)}{N} \tag{2}$$

Where N(i) represents the set of adjacent nodes of node i, k_j represents the degree of node j, $s_j(t)$ represents the LR value of node j at iteration t, and t_c represents the number of convergences, $s_g(t_c)$ is the LR value of node g in the convergent state.

LeaderRank has no parameters, thus avoiding the complexity consumption and accuracy impact caused by parameter selection. Moreover, the addition of ground node g reduces the radius of the entire network and increases the convergence speed. In addition, the LeaderRank algorithm is more robust against noise and malicious attacks than other sort algorithms. These advantages make the LeaderRank algorithm more suitable for complex social networks.

III. Influence Maximization Algorithm Based on Community Detection

The algorithm in this paper is divided into four stages, which are the stage of label propagation community detection, LeaderRank, seed node candidate set generation and improved greedy algorithm.

A. Seed node candidate set generation

It is known from the power law distribution of node influence [15], there are a large number of nodes that have less influence and a few more influential nodes on the network. However, the seed nodes to be excavated mainly refer to those nodes with higher influence, so as to maximize the diffusion of information as much as possible. If you do not consider the distribution of the node's influence, we need to simulate the diffusion of each node to calculate the marginal influence increment of the node. Although the ultimate influence scope can be guaranteed, it will obviously be very time-consuming. Therefore, on the basis of community detection, we first evaluate the influence of each node, then according to a certain distribution strategy to select some influential nodes to form a candidate set of seed nodes.

As for how to evaluate the influence of a node [16], the LeaderRank centrality will be used as the node influence measure.

As for how to reasonably select the most influential nodes to form a candidate set of nodes without losing the scope of influence. This chapter introduces the scaling parameter p for adjustment, where p ∈ (0,1]. The larger the p, the more nodes in the candidate set. When p = 1, each collection will contain all the nodes in the network. When p = 1, the candidate set will contain all the nodes in the network. The selection of p value is generally determined by the number of network nodes n and the number k of core nodes to be excavated.

This chapter uses the method of proportionally distribution [17] to allocate the number of core nodes. According to the size of the number of nodes k to be excavated, if the number of divided communities is greater than k and the number of community nodes with community sizes lower than k is always calculated as zero, then only the first k large communities need to calculate the core node allocation. Given network G and the number of nodes to be excavated k, the community set obtained by the LRMLPA community detection algorithm is $C=\{C_1, C_2, \ldots, C_q\}$, where q is the number of communities contained in community set C,then the distribution of core nodes will be divided into two situations:

1) If the number of communities contained in the community set C is greater than k, that is, q> k,then take the

first k largest communities, and update the C set as $C = \{C_1, C_2, \ldots, C_n\}$, where r = k, then the number of nodes k_i to be excavated in community C_i is as shown in the formula.

$$k_i = \frac{k \times |C_i|}{\sum_{j=1}^{r} |C_j|} \quad (3)$$

2) If the number of communities contained in the community set C is less than or equal to k, that is, $q \leq k$, then community set C can be expressed as $C = \{C_1, C_2, \ldots, C_r\}$, where r=q, then the number of nodes k_i to be excavated in community C_i is as shown in the formula.

$$k_i = \frac{|C_i| \times k}{n} \quad (4)$$

Where n is the total number of nodes of the network G.

B. Improved greedy algorithm

The influence objective function has been proved to have submodel characteristic [18], the submodel characteristic means that when the set S becomes larger, the marginal influence increment of the current inactive node obtained by the pre-simulation diffusion is non-increasing. The greedy algorithm does not take into account the submodel characteristic, and therefore each time you select a seed node, it is necessary to pre-diffuse the simulation of all inactive nodes in the network to calculate the marginal influence increment, then select the node with the largest incremental margin as the seed node, resulting in a very high time consumption. However, the recalculation of the increment of marginal influence on most of the nodes may not be meaningful. According to submodel characteristic, in each round of selecting the seed node, only the selective incremental update of marginal influence on some nodes is needed, making it possible to save a large amount of unnecessary simulation diffusion time while not affecting the normal selection of seed nodes. Therefore, based on the high-influence greedy strategy, the submodel optimization strategy adopted in this chapter is divided into two steps:

1) When selecting the seed node in the first round, firstly, the marginal influence increment of all the nodes in the candidate set is calculated, and the node with the largest increment of marginal influence is used as the seed node. Then removed it from the candidate set.

2) From the new candidate set, select the node with the largest increment in the last round of marginal influence and recalculate its marginal influence increment,then use the temporary variables aNode and gMax respectively record the node and its marginal influence increment. Next, select the node with the next larger increment of marginal influence and compare its marginal influence increment with gMax, if it is not greater than gMax, it will not calculate the increment of marginal influence of updating this node in the current round; if it is greater than gMax, calculate the marginal influence increment of this node in the current round and compare again with gMax, if it is still greater than gMax, update aNode and gMax separately for the new node and its marginal influence increment. And so on until all nodes in the candidate set have been traversed, find the node that maximizes the gMax value, take it as the seed node for the current round and remove it from the candidate set.

LPIMA algorithm pseudo-code described in Table 2 below.

Table. 2 LPIMA Algorithm

Input： Community C_i, Number of seeds to be mining k_i, proportion of selection p

Output： Initial communication set S

1: The influence value LRI of each node in community according to LeaderRank;

2: Select the p% nodes with largest LRI as alternative set U_i of seed nodes;

3: Initial S as null, $S_0 = \emptyset$. Calculate scope of influence $\delta(v|S_0)$ of the all the nodes v in U_i;

4: Select the node v with the largest $\delta(v|S_0)$ to S_1, remove v from U_i.

5: Let j=1;

6: Select the node u with the largest LRI(influence value) among U_i, and calculate the marginal influence increment $\delta(u|S_j)$ of u, $\delta(u|S_j) = \delta(S_j \cup \{u\}) - \delta(S_j)$;

7: Let gMax= $\delta(u|S_j)$, aNode=u, record the current maximum increment of marginal influence and its corresponding node respectively.

8: for $\forall z \in U_i$, if $\delta(z|S_{j-1}) >$ gMax, calculate the current marginal influence increment $\delta(z|S_j)$ of z, $\delta(z|S_j) = \delta(S_j \cup \{z\}) - \delta(S_j)$. If $\delta(z|S_j)$>gMax again, let gMax=$\delta(z|S_j)$, aNode=z;

9: Add aNode into S_j, and remove it from U_i;

10: Let j=j+1, if $j > k_i - 1$, end. Otherwise, step 6.

IV. Experiments

In order to verify that the algorithm proposed in this paper has significantly improved the time complexity of the existing social network influence maximization algorithm, and also have a good scope of influence. The experiment compares the Greedy algorithm, the CELF algorithm, the CGA algorithm and the LPIMA algorithm proposed in this paper on the real network data set.

A. Experimental data

This paper selects two large-scale real data sets, namely Internet and SinaWeibo. The specific parameters of these data sets are shown in Table 3 below.

TABLE 3 Data set basic information

Network	Number of nodes	Number of edges
Internet	22963	48436
Sinaweibo	305162	2357240

The first is a classic social network dataset, and the second is a dataset of concerns and concerns between some of the users crawling from SinaWeibo.

B. Experimental results and analysis

During the experiment, the number of seed nodes selected is set as 50, the influence probability is taken as 0.01, and the influence propagation model is an independent cascade model. Considering the choice of the parameter p $\in(0,1)$, when the value of p is too small, the influence range of the seed node is too low, and when the value of p is too large, the utility of the algorithm is greatly reduced. Therefore, based on these factors, p= 0.03.

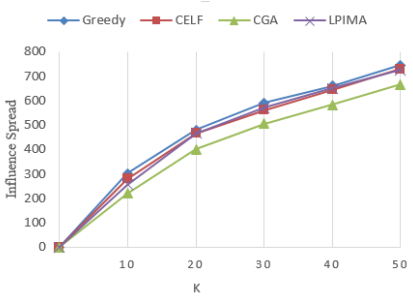

Fig.1. Comparison of the influence scope of the algorithm on Internet

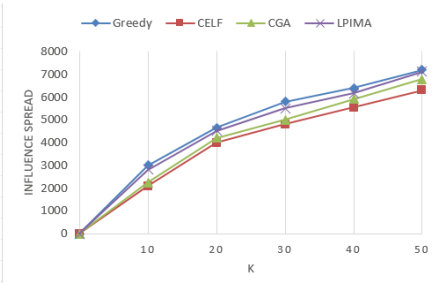

Fig.2. Comparison of the influence scope of the algorithm on Sinaweibo

It can be seen from Figure 1 that the Greedy algorithm has the best scope of influence ，and the curve of influence scope of LPIMA and CELF close to Greedy. However, when the k value is less than 20, the LPIMA algorithm is less effective than the CELF algorithm and the CGA algorithm has the worst influence scope. Figure 2 shows that the LPIMA algorithm is better than the CELF in any case. It can be seen from the comparison that for the same k value, the influence range of the LPIMA algorithm in larger networks is better.

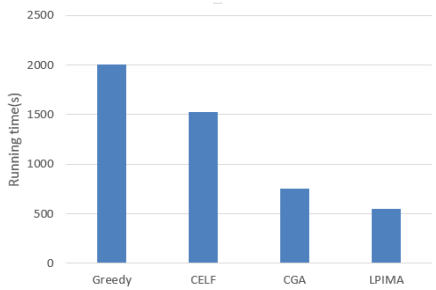

Fig.3. Comparison of the running time of the algorithm on Internet

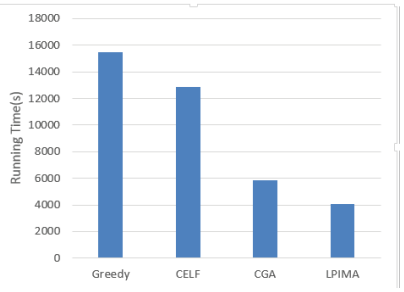

Fig.4. Comparison of the running time of the algorithm on Sinaweibo

It can be seen from Figure 3 and Figure 4 that the running time of various algorithms tends to be the same in both scale networks. The greedy algorithm has the highest time complexity and the LPIMA algorithm has the lowest time complexity. CELF compared to the first relatively small dataset in the second dataset Run time is high, LPIMA relative to the Greedy algorithm Sinaweibo dataset percentage of running time relative to the Internet dataset also decreased by 1.2%. The more visible the more practical the large-scale network LPIMA algorithm is.

Based on the above analysis, we can see that the proposed social network influence algorithm based on label propagation has a close scope of the greedy algorithm in large-scale networks and at the same time reduces the running time of the algorithm, and proves the effectiveness of the algorithm.

CONCLUSIONS

On the basis of studying and analyzing the influence maximization problem, aiming at the disadvantage that the greedy algorithm is too time-consuming and not suitable for large-scale network, this paper proposes a community influence-based maximization algorithm based on community detection LPIMA. The algorithm firstly uses the optimized label propagation algorithm for community detection and guarantees the quality of the community. Then, the candidate set of seed nodes is generated by calculating the LeaderRank value, and finally the strategy of improving the greedy algorithm is implemented in the set of candidate seed nodes. Experiments on real datasets show that the LPIMA algorithm proposed in this paper not only reduces the time consumption but also improves the influence scope, which proves the effectiveness of the algorithm.

In the following work, we can study from the following aspects: The current improved community discovery algorithm is to divide the network into independent sub-communities without considering the phenomenon of community overlap, so the community partitioning algorithm can be extended to overlap Community division.

REFERENCE

[1] Richardson, Matthew, Domingos, et al. Mining knowledge-sharing sites for viral marketing[J]. 2002.

[2] Li Lei. Research on social network influence model and its algorithm [D]. Beijing Jiaotong University, 2010.

[3] Zhou Shengfu.Study on the algorithm of maximizing influence under the linear threshold model [D] .Yunnan University, 2014.

[4] Kempe D, Kleinberg J. Maximizing the spread of influence through a social network[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2003:137-146.

[5] WAN Xue-fei, CHEN Duan-bing, FU Yan.Heuristic algorithm for overlapped community discovery [J] .Computer Engineering and Applications, 2010, 46 (3): 36-38.

[6] Galstyan A. Distributed online localization in sensor networks using a moving target[C]// International Symposium on Information Processing in Sensor Networks. IEEE, 2004:61-70.

[7] Cao T, Wu X, Wang S, et al. OASNET: an optimal allocation approach to influence maximization in modular social networks[C]// ACM Symposium on Applied Computing. ACM, 2010:1088-1094.

[8] Wang Y, Cong G, Song G, et al. Community-based greedy algorithm for mining top-K influential nodes in mobile social networks[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2010:1039-1048.

[9] Leskovec J, Krause A, Guestrin C, et al. Cost-effective outbreak detection in networks[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2007:420-429.

[10] Wang Y, Cong G, Song G, et al. Community-based greedy algorithm for mining top-K influential nodes in mobile social networks[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2010:1039-1048.

[11] Kang Xu-Bin, Jia Cai-Yan.An Improved Method for Rapid Community Discovery of Label Spread [J] .Journal of Hefei University of Technology: 2013,36 (1): 43-47.

[12] Shi Meng-yu, Zhou Yong, Xing Yan.Reader-based tag discovery community discovery algorithm [J] .journal of Computer Applications, 2015,35 (2): 448-451.

[13] HUANG Jia-xin, GUO Kun, GUO Hong. A Label Distribution Community Discovery Algorithm Incorporating Node Importance and Label Influence [J]. Microcomputer Systems, 2015, 36 (6): 1171-1175.

[14] Gregory S. An algorithm to find overlapping community structure in networks[C]//European Conference on Principles of Data Mining and Knowledge Discovery. Springer Berlin Heidelberg, 2007: 91-102.

[15] Si A L B, Albert R. Emergence of Scaling in Random Networks[M]// The Structure and Dynamics of Networks. 2011.

[16] SUN Rui, LUO Wanbo.A Review of Methods for Evaluating Node Importance in Internet Consensus [J] .Application Research of Computers, 2012, 29 (10): 3606-3608.

[17] JIN Jin-chao, HUANG Lan, WANG Zhe, et al.A new method of maximizing influence based on community structure [J]. Journal of Jilin University: 2011, 49 (1): 93-97.

[18] Tang Daping.Study on Maximization of Social Network Based on Changes in User Preference [D] .Yunnan University, 2015.

[19] TIAN Jia-Tang, WANG Yi-Tong, FENG Xiao-Jun.A Novel Algorithm for Maximizing the Impact of Social Networks [J] .Journal of Computers, 2011, 34 (10): 1956-1965.

[20] Chen W, Wang Y, Yang S. Efficient influence maximization in social networks[C]// ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2009:199-208.