International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6, August, 2019

A Method To Improve The Time Of Computing For Detecting Community Structure In Social Network Graph

Nguyen Xuan Dung, Doan Van Ban, Truong Tien Tung

Abstract: ***Identifying communities has always been a fundamental task in analysis of complex networks. Currently used algorithms that identify the community structures in large-scale real-world networks require a priori information such as the number and sizes of communities or are computationally expensive. Amongst them, the label propagation algorithm (LPA) brings great scaslability together with high accuracy but which is not accurate enough because of its randomness. In this paper, we study the equivalence properties of nodes on social network graphs according to the labeling criteria to shorten social network graphs and develop label propagation algorithms on shortened graphs to discover effective social networking communities without requiring optimization of the objective function as well as advanced information about communities. Test results on sample data sets show that the proposed algorithm execution time is significantly reduced compared to the published algorithms. The proposed algorithm takes an almost linear time and improves the overall quality of the identified community in complex networks with a clear community structure.***

Keywords: ***Class of equivalent nodes, community structure, identical nodes, label propagation.***

I. INTRODUCTION

A social network is one collection of different subjects linked together through relations or links. Social network analysis is to consider social networks based on graph theory comprising nodes and links or edges or connections. A node is individual community on Internet and links are the relations among these communities. Social network structure is one collection of individuals or objects interacting to each other, thanks to specific means of communication capable of surpassing geographical and political boundaries in order to chase one common goal or benefits [1]. Currently, analyzing and finding community on the social network graph is one main research focus in mining social network. Algorithms have been proposed and developed with executive time reduction in finding community on the social network graph [2]-[9], [11], [12]. One latest and most effective algorithm is label propagation algorithm. Here is how LPA functions: initially, every node in the network is attached one unique label. After each repeatition, each node will be updated its personal label into the most frequent with the highest appearance usage in nearby zones; in case more nodes as the most frequent label with highest appearance is selected randomly. LPA variations or improvements result from original label assignment, random choosing in broken links and whether one node is comprised of itself in calculating the most frequent node in the nearby zones [10].

Social network graph are complicated with high number of nodes and edges; therefore, finding structure community of social network takes a lot of time. On the other hand, social network graph is comprised of many nodes in relation with other rather similar nodes, like the nodes in nearby zones are similar to each other. This is the similar relation based on betweenness structure or lable transmission criteria and so on. Many similar nodes with the same betweenness, same label make up class of similar nodes, and these can be combined together into one representative node so as to help significantly reduce the number of nodes and sides of one graph. This paper concentrates on researching similar properties of nodes for graph reduction based on label propagation algorithm presented as below. Following the introduction is the presentation of LPA. The following section is the introduction of the properties of class of similar nodes with the same labeling criteria and social network graph reduction based on classes of similar nodes. The final part is to present the development of label propagation algorithm on social network graph reduction as well as the assessment and efficiency evaluation of the proposed algorithm.

II. LABEL PROPAGATION ALGORITHM

LPA is paid the most attention because of the advantage of chronological complexity near linear and the unnecessariness of target function identification as well as community number prediction. LPA concentrates on the main functions as below. First is the chronological complexity near linear. For a network with n nodes and m sides, LPA chronological complexity is O (m+n). Second, LPA ability of finding community is independent on network scale and size. This function is not influenced by decomposition limitation like other model-based methods. Third,

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
Nguyen Xuan Dung*, Hanoi Open University, Hanoi, Vietnam.
Doan Van Ban, Vietnam Academy of Science and Technology, Hanoi, Vietnam.
Truong Tien Tung, Hanoi Open University, Hanoi, Vietnam.

LPA randomness includes initial random label, random updated-label order and the random selection of one most optimal label as node label while this optimal label is not the unique one. Due to label propagation randomness, when LPA is used to discover community of network graph, any information related to this network, except the unneccessariness of providing its nodes and sides, more structure communities are collected [10]. After Newman introduces model features in network division measurement, Barber & Clark propose one specialized LAP model (LPAm) to control the label propaganda process [13]-[15]. Liu et al. introduce one specilaized algorithm based on the advanced model LPAm+ [16], [17]. Moreover, Zhang et al. continue to ameriolate into LPAp [18]. However, the researches above mostly solve the issue of discovering the direct community structure on graph whereas very few researches focus on minimizing the calculating space regarding to node and edge number in a graph so as to shorten analysis time as well as quick discovery of community social network and increasing efficiency in community discovery in social networks.

According to structural viewpoint, social network modeling can be equivalent to graph G = (V, E), in which V as n node collection (peak) and E as m edge collection (correlation). V collection represents social network members (agents, entities) whereas E collection implies social network among members. Based on graph theory, social network structure can be expressed through the contiguous matrix $A = (a_{ij}) \in \{0, 1\}^{n \times n}$, in which, n = |V|, a_{ij} = 1 if both i and j nodes have an edge between them (correlative and direct connection); in contrast, $a_{ij} = 0$. L(i) signal is the label of i node, i ∈ V.

Label propagation Agorithm

Input: Social network G = (V, E)

Output: Community structure in social network

Step 1. Initiating the unique label for all network nodes, L(i) = i, i ∈ V.

Step 2. Naming X as the list of nodes areanged in a random order.

Step 3. For each v ∈ X selected in a random order, re-updating L(v) as the label of the neighboring node with the most frequent appearance.

Step 4. If one node has label with the maximum amount, stop the agorithm, move into step 5; vice versa, conduct step 2.

Step 5. Those nodes with the same label result in one community in social network.

LPA complexity is O(m+n); for those rarefield graphs, it is O(n), in which n = |V|, m = |E|; in other words, LPA complexity is near linear [10].

III. ABRIDGING EQUIVALENT NODES IN SOCIAL NETWORK GRAPHS

Social network graphs consist of many nodes with labels similar to those neighboring in one same network and their labels are always updated during the label propagation. These nodes are structurally equivalent, with the same nodes in propagation steps; therefore, they can be combined nto one representative node for the whole class (normally more than two equivalent nodes) in order to minimize the number of nodes and sides of a social network.

A. Class Of Similar Nodes

Social network is expressed by undirected and connected graph, G = (V, E), V as the collection of nodes and E as the collection of edges. V node is side by side with w if (v, w) ∈ E (or (w, v) ∈ E). Supposed that v node has no side nodes, signaled as $N(v) = \{v_1, v_2, ..., v_k\}$. Each side node v_j with label $L(v_j)$ signifying the community that v_j belongs to.

LPA conducts the update label of v node based on the most frequent label appearing of the side node. According to format style, v label is also updated based on the labels of other side nodes as below:

$$L(v) = \underset{l}{argmax} \sum_{u \in N(v)} [L(u) == l] \quad (1)$$

In which, L(u) signals u label and [P] = 1 if P expression is right; in contrast, [P] = 0.

Therefore, if both nodes u, v have the same edge nodes (N(u) = N(v)), their labels L(u), L(v) are reupdated, based on the label of the most frequent node, for example node w ∈ N(u). In other words, L(u) = L(v) = L(w), with w mostly appears in the side nodes of u and also the node in side node collection of v (N(u) = N(v)).

Undirected graph given in advance, connected G = (V, E). Both nodes u, v ∈ V are considered as identical in G, symbolized as u ≈ v once N(u) = N(v).

Example 1. Consider one social network in the graph from figure 1.

It si easy to recognize that nodes 1, 2 and 3 are esimilar to each other because N(1) = N(2) = N(3) = {4}. Similarly, node 6 ≈ 7 for N(6) = N(7) = {4, 5, 8, 9} or 14 ≈ 15 as N(14) = N(15) = {10}.

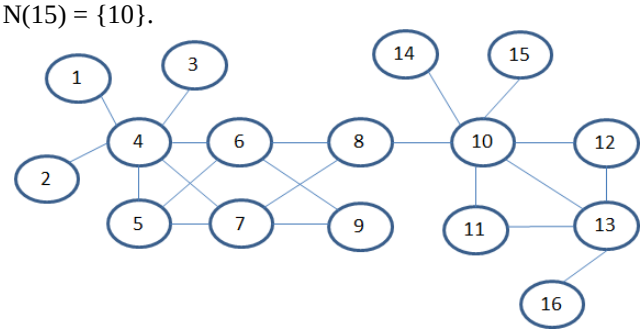

Fig. 1. Social network G

Relation ≈ obviously means the similar or equivalent relation; therefore, identical nodes make a class of similar /equivalence nodes. All nodes in one class of similar /equivalence nodes can be combined for a representative node for a reduction in nodes and edges of network graphs.

Undirect graph given in advanced, connected G = (V, E) and the relation ≈ identifies q class of similar /equivalence nodes D_i, i = 1..q. Combining identical nodes in class D_i, $|D_i| > 2$, i = 1..q into one representative node D_i' for a reduced graph $G_1 = (V_1, E_1)$, in which:

- $V_1 = V - V_2 \cup \{D'_1, D'_2, \ldots, D'_q\}$, in which $V_2 = D_1 \cup D_2 \cup \ldots \cup D_q$.
- $E_1 = E - \{(u, v) \mid u \in V_2, v \in N(u)\} \cup \{(v, D'_i) \mid i = 1..q, v \in N(u)$, in which $u \in D_i\}$

According to LPA, label of the nodes in one similar class will be re-updated based on the label of the representative node when the propagation finishes.

Example 2. Combining identical nodes of graph G from figure 1 is as below: Class of similar nodes including nodes 1, 2, 3 will be represented by node 3', the equivalent class including nodes 6, 7 combine together and will be represented by 7'. Similarly, equivalent classes of nodes 11, 12 will be represented by 12' as well as the equivalent class of nodes 14, 15 will be represented by 15'. Graph G from Figure 1 will be reduced into:

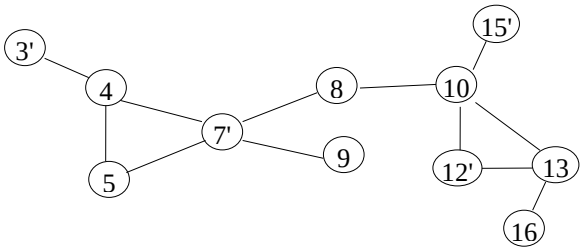

Fig. 2. Graph G_1 abridging the similar nodes from graph G
Graph G has 16 nodes, 21 sides are abridged into G_1 with 11 nodes and 12 sides. Obviously, L(1) = L(2) = L(3) = L(3'), L(6) = L(7) = L(7'), L(14) = L(15) = L(15') and L(11) = L(12) = L(12').

B. Reduce Equivalence Nodes Algorithm Abridging The Similar Nodes In Graph

Algorithm Reduce Equivalence Nodes (REN) combines similar nodes into representative nodes in order to reduce the calculation space in the graph.

Algorithm REN (G)

Input: G = (V, E) - original graph
Output: $G_1 = (V_1, E_1)$ - achieved graph after the combination of similar nodes.

$V_1 = V$; $E_1 = E$; $S = \varnothing$;
//Step 1. Find side nodes and save into Stack S
for u ∈ V do {
  N[u] = Neighbor(G, u);
  S.push(u, N[u]); //Save (u, N[u]) into S
}
//Step 2. Find class of similar nodes of identical nodes
h = 1;
(u, M) = S.pop();
D[h] = {u} //Class of identical nodes
N[h] = M;
while(S != ∅) do {
  (u, M) = S.pop();
  j = 1;
  loop = true;
  while (j <= h && loop) do
    if(N[j] == M) then //Check identical nodes
      D[j] = D[j] ∪ {u}; //Put u into class D[j]
      $V_1 = V_1 - \{u\}$; //Delete u node
      for v∈ N[j] do
        $E_1 = E_1 - \{(u, v)\}$;
      loop = false;
    } else j = j + 1;
  if (loop) then {
    h = h + 1; //Consider the successive classes
    D[h] = {u};
    N[h] = M;
  }
}
//Step 3. Combine similar nodes into peak representation D'_j
$V_D = \varnothing$;
for j ∈ 1… h do {
  $V_D = V_D \cup \{D'_j\}$; //Add representative node for identical class
  for v ∈ N[j] do
    $E_1 = E_1 \cup \{(D'_j, v)\}$
} $V_1 = V_1 \cup V_D$;

Algorithm Neighbor (G, u): Find the side nodes of u in graph G

Input: Graph G = (V, E) and peak u ∈ V
Output: N – collection of continuous peaks u in graph G
N = ∅;
for v ∈ V do
  if ((u, v) ∈ E) then
    N = N ∪ {v};
return N;

C. Calculated Complexity Of Algorithm

Algorithm REN is comprised of 3 steps: Step 1. Calculated complexity is O (n * d), in which n = |V| and d is the calculated complexity of Neighbor (G, u), finding the side nodes of u. Step 2. Examine the pairs (nodes, side node collections) from S so as to find the class of similar nodes with the calculated complexity as O(n * k), in which k as the degree of the nodes in the graph (k = d). Step 3. Abridge h similar classes for calculated complexity as O(h * k), normally h << n. For social network graph with charts with the number of side nodes as d (or k) in limited state; therefore, algorithm REN consists of chronological complexity almost as linear O(n).

IV. LABEL PROPAGATION ALGORITHM DEVELOPMENT ON SOCIAL NETWORK

LPA algorithm on abridged social network is comprised of two phases:
Phase 1. Use algorithm REN to find equivalent nodes of graph G = (V, E) and abridge similar nodes into graph $G_1 = (V_1, E_1)$.
Phase 2. SPLPA in abridged graph G_1 discovers nodes with the same label to create social network.
LPA on abridged graph is conducted repeatedly with many steps. Each step of the nodes on graph will be updated based on the side nodes with the most frequent appearance use from the calculation formula (1).

The simplest ceasing condition of the algorithm is to examine the label at the current time compared to that of the previous step; if no change happens, cease the algorithm (no change in label happens in the following step).

L(i, v) symbolizes node v in the i repeated step, in which i = 0, 1, … and ∀ v ∈ V.
SPLPA is conducted in the following steps:

Input: Undirected graph, connected G = (V, E)
Output: Communities on social network
REN(G); //Outcome is the abridged graph $G_1 = (V_1, E_1)$
i = 0; //Repetition of i times
for v ∈ V_1 do
| L(i, v) = v;
while (Dissatisfying the ceasing condition of algorithm)
| as {
| i = i + 1; //Propagating label in the successive step
| for v ∈ V_1 do{
| | N(v) = Neighbor (G, v);
| | A[k] = N(v)[k];
| | j = 0; max = 1;
| | for (j = 1; j < k; j++){
| | | dem = 0;
| | | for (l = 0; l < k; l++)
| | | | if(L(i-1, A[l]) == L(i-1, A[j]))
| | | | | dem = dem +1;
| | | if(dem > max)
| | | | max = dem;
| | | | // Label v following the nodes in N(v)
| | | | most frequently appearing as A[j]
| | | }
| | | L(i, v) = L(i-1, A[j]);
| | } // Satisfying (Ceasing condition for label propagation)
} // return L(i, v) with all nodes v ∈ V_1 at the final i times of repetition.

When the algorithm completes, those nodes with the same label in one social network community. Those nodes in one similar class are identified in phase 1 have labels overlapping in the nodes of the representative label; therefore, they share the same community as that of the representative node.

Algorithm REN with complexity in the recent time is almost near linear (O(n)) and the developed algorithm on social network is almost linear; therefore, algorithm SPLPA also has complexity as near linear O(n), in which n = |v|.

Example 3. Apply algorithm SPLPA for abridged graph in figure 2. Divide into two communities: the first community is comprised of nodes {3', 4, 5, 7', 8, 9} whereas the second community is comprised of nodes {10, 12', 13, 15', 16}. Based on the labeling feature of the nodes in one similar class overlapping with the node of the representative node, graph in figure 1 will be divided into two communities: first including nodes {1, 2, 3, 4, 5, 7', 8, 9} whereas second including nodes {10, 11, 12, 13, 14, 15, 16}.

One advantage of algorithm SPLPA is simplicity and easiness in parallel performance in order to analyze and discover social networks in an effective way.

V. RESULT AND DISCUSSION

This research concentrates on performing experiments and evaluating efficiency of the suggested algorithm SPLPA compared to LPA regarding to performing length.

The program is set by language R and the experiment is conducted on computers with CPU Intel Core i5 4200U, RAM 4G. Due to the limit in computer configuration, our algorithm is conducted optimally on 1000-peak and 5000-peak graphs.

- In experiment 1, SPLPA is conducted to show the necessary time to calculate the graphs produced randomly from 100 peaks to 1000 peaks graphs as well as from 500 edges to 5000 edges graphs.

Table- I: Performance time of LPA and SPLPA with random graph

(Timing unit: Second)

No	Number of graph peak	Number of graph edge	Performance time for LPA	Performance time for SPLPA
1	100	500	0.77	0.62
2	200	1000	3.32	2.76
3	300	1500	6.9	5.8
4	400	2000	8.8	7.5
5	500	2500	11.5	9.8
6	600	3000	13.6	11.5
7	700	3500	17.1	14.6
8	800	4000	19.7	16.9
9	900	4500	22.5	19.4
10	1000	5000	25.6	22.2

- In experiment 2, SPLPA is conducted to show the necessary time to calculate large-scale social networks known by huge user community (Wikipedia vote network with 7115 peaks và 103689 edges as well as Email-Eu-Core network with 1005 peaks và 25571 edges) based on the date publicly informed from Stanford large network dataset collection [19].

Table- II: Performance time of LPA and SPLPA on large-scale social networks

(Calculating unit: Second)

No	Large scale social network	Number of graph peaks	Number of graph edges	Perfor-mance time for LPA	Perfor-mance time for SPLPA
1	Wikipedia vote network	7115	103689	1550.1	1435.2
2	Email-Eu-Core network	1005	25571	101.1	89.2

Experiment result shows that the suggested algorithm to discover community structure SPLPA presents the substantially faster result compared to that of the LPA.

VI. CONCLUSION

Social networks have many nodes that have similar structure, with the same label according to the label propagation method. Therefore, the combination of nodes equivalent to the representative node will help reduce the number of nodes and edges of the graph quite a lot, to reduce the calculation time of algorithms to detect the community structure above Social Network. The SPLPA algorithm developed on a shortened social network graph is quite effective, with computational complexity that is close to linear. Especially SPLPA is easy to implement in parallel to analyze, quickly and effectively detect community structures on large and complex social networks. As future work, this algorithm can be extended to be used for overlapping community detection where each node may belong to several different communities.

REFERENCES

1. H. T. Thanh, and D. Phuc, "Social Network Analysis Based on Topic Model with Temporal Factor", International Journal of Knowledge and Systems Science (IJKSS), ESCI Journal, Indexed by Web of Science, Web of Science Emerging Sources Citation Index (ESCI), SCOPUS, INSPEC, ACM, 2018, pp. 82-97.
2. M. Girvan, and M. E. J. Newman, "Community structure in social and biological networks", Proceedings of the National Academy of Sciences of the United States of America, Vol.99, No.12, 2002, pp. 7821-7826.
3. M. E. J. Newman, "Analysis of weighted networks". Physical Review E., Vol.70, No.5, Article ID 056131, 2004.
4. M. E. J. Newman, "Finding community structure in networks using the eigenvectors of matrices". Physical Review E., Vol.74, Article ID 036104, 2006.
5. J. Reichardt, and S. Bornholdt, "Statistical mechanics of community detection". Physical Review E., Vol.74, Article ID 016110, 2006.
6. M. E. J. Newman, "Fast algorithm for detecting community structure in networks". Physical Review E., Vol.69, Article ID 066133, 2004.
7. A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks". Physical Review E., Vol.70, Article ID 066111, 2004.
8. H. J. Li, B. Y. Xu, L. Zheng, and J. Yan, "Integrating attributes of nodes solves the community structure partition effectively". Modern Physics Letters B., Vol.28, No.5, Article ID 1450037, 2014.
9. J. Duch, and A. Arenas, "Community detection in complex networks using extremal optimization". Physical Review E., Vol.72, No.2, Article ID 027104, 2005.
10. U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks". Physical Review E-Statistical, Nonlinear, and Soft Matter Physics., Vol.76, No.3, Article ID 036106, 2007.
11. A. P. Zhang, G. Ren, B. Z. Jia, H. Cao, and S. B. Zhang. "Generalization of label propagation algorithm in complex networks". Proceedings of the 25th IEEE Chinese Control and Decision Conference, 2013, Guiyang, China. pp. 1306-1309.
12. I. X. Y. Leung, P. Hui, P. Lio, and J. Crowcroft, "Towards real-time community detection in large networks". Physical Review E-Statistical, Nonlinear, and Soft Matter Physics., Vol.79, No.6, Article ID 066107, 2009.
13. M. E. J. Newman, and M. Girvan, "Finding and evaluating community structure in networks". Physical Review E: Statistical, Nonlinear, and Soft Matter Physics., Vol.69, No.2, Article ID 026113, 2004.
14. M. E. J. Newman, "Modularity and community structure in networks". Proceedings of the National Academy of Sciences of the United States of America., Vol.103, No.23, pp. 8577-8582, 2006.
15. M. J. Barber, and J. W. Clark, "Detecting network communities by propagating labels under constraints". Physical Review E-Statistical, Nonlinear, and Soft Matter Physics., Vol 80, No.2, Article 026129, 2009.
16. X. Liu, and T. Murata, "Advanced modularity-specialized label propagation algorithm for detecting communities in networks". Physical A., Vol.389, No.7, pp. 1493-1500, 2010.
17. P. Schuetz, and A. Caflisch, "Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement". Physical Review E., Vol.77, No.4, Article ID 046112, 2008.
18. A. Zhang, G. Ren, Y. Lin, B. Jia, H. Cao, J. Zhang, and S. Zhang, "Detecting Community Structures in Networks by Label Propagation with Prediction of Percolation Transition", Hindawi Publishing Corporation, the Scientific World Journal, Vol.2014, Article ID 148686, 2014.
19. J. Leskovec, and A. Krevl, 2014, SNAP Datasets: Stanford large network dataset collection. Avaiable: https://snap.stanford.edu.

AUTHORS PROFILE

Nguyen Xuan Dung's Educational Profile is as below: In 2010, achieving bachelor in information technology, Open university, Hanoi, Vietnam; In 2013, achieving bachelor in English lingusitics, Hanoi university, Hanoi, Vietnam; In 2014, being rewarded master in information system, Hanoi national university of education, Hanoi, Vietnam. Currently, he is a PhD researcher in information technology, institute of post and telecommunication, Hanoi, Vietnam. He is a lecturer in information technology, Open university, Hanoi with lecturing subjects as database and data mining. His interested fields are database, data mining, artificial intelligence and intelligence systems. His second language is English. Concerning to his presented scientific researches, two resaerches are as: General planning information entrance of Open university (2014 - 2020 period) and Electronic guidance on mobile phone model construction.

Doan Van Ban's Educational Profile is as below: Graduating from Warszawa university, majoring in mathematics informations, Poland; Achieving PhD in information technology, Warszawa university, Poland. He is currently working as Associate Professor in Institute of Information Technology, Vietnam Academy of Science and Technology, Hanoi, Vietnam. His interested fields are: high performance computing, software engineering, data mining and machine learning. His lecturing subjects are: System analysis and designing and subject - targeted analysis, data development and information discovery, subject - targeted database and decentralization. His second languages is English. His performed researches are: models, solutions and developed technology for information management systems integrated and applied in governmetal administration; model and tool construction in developing information systems serving administrative management; construction of information models serving managing hospitals.

Truong Tien Tung's Educational Profile is as below: Learning a Bachelor in electric automatics, Transport university, Moscow, Russia; Being a Bachelor in Electronic Technology, Multi-Science University, Hanoi, Vietnam; Bieng a Bachelor in information technology, Open university, Hanoi, Vietnam; Achieving PhD in Information Technology, Multi-Science University, Sanit-Petersburg, Russia. Currently, he is a lecturer in information technology, Open university, Hanoi, Vietnam. His interested fields are database, data mining, artificial intelligence, intelligence systems and E-commerce. His lecturing subjects are Database, Information system analysis and planning, programming technology, E-commerce and E-commerce administration. His two second languages are Russian and English. Concerning to his performed scientific researches, there are two: General planning Information Entrance of Open University, Hanoi, Vietnam (2014 - 2020 period) and Electronic guidance on mobile phone model construction.