International Journal of Future Generation Communication and Networking
Vol. 9, No. 8 (2016), pp. 317-326
http://dx.doi.org/10.14257/ijfgcn.2016.9.8.31

Adaptive Label Propagation Algorithm to Detect Overlapping Community in Complex Networks

Chunying Li [1,2,a], Yonghang Huang[1,b], Zhikang Tang[3,c], Yong Tang[1*,d] and Jiandong Zhao[2]

([1] School of Computer science, South China Normal University, Guangzhou 510631, China;
[2] Computer Network Center, GuangDong Polytechnic Normal University, Guangzhou 510665, China;
[3] School of Computer science, GuangDong Polytechnic Normal University, Guangzhou 510665, China.)
[a]zqxylcy@163.com,[b]yongh701@163.com,[c]fzutang@126.com,[d]ytang@scnu.edu.cn

Abstract

According to the defects that community detection algorithm in unknown complex networks has a pre-parameter. We propose Adaptive Label Propagation Algorithm (ALPA) to detect community structures in complex networks. The ALPA algorithm find out all disjoint Maximal Clique (MC) and let each MC share the identical weight and unique label so as to reduce the redundant labels and uncontrollable factors. The stability of ALPA algorithm is enhanced by synchronous update during iterations. Meanwhile it will converge easily due to the termination condition that all of the vertexes have the label. During iterations we use the adaptive threshold method to overcome the pre-parameter limitation. Compared with other community detection algorithms in synthetic networks and real networks, our experiments show that ALPA algorithm not only improves the tolerance of mixing parameter, but also enhances its robustness.

Keywords: *Complex Networks, Community Detection, Overlapping Community, Label Propagation, Adaptive Threshold*

1. Introduction

With the rapid developments of intelligent terminals and the extensive applications of computing and communications technology, the amount of network data has proliferated in recent years. These networks can be abstracted as the graph in data structure. Furthermore, individuals and relationships can be regarded as vertexes and edges respectively. Similar to the network of relationships among individual in real life, it can be divided into some sub graphs, the relationship of vertexes in these sub graphs is very close. We defined the subgraph as community.

Community is the most meaningful property of network. That is, the entire network is made up of several communities. Generally, community is a set of vertexes with similar properties. Such as the same belief, resources, preferences, requirements in social network. If a vertex belongs to at least two communities, we regard it as overlapping vertex, while the network as a complex network of overlapping communities. Community overlapping is one of the most important features of complex networks [1-2], and overlapping vertexes plays a special role in it.

We detect community structure in complex networks and each vertex belongs to community by the topology of the network and vertex information. Community represents

*Corresponding Author.

ISSN: 2233-7857 IJFGCN

real group, members in it have the same interest or similar behavior. Community detection can reveal some hidden relationships and further explain some phenomenon. And it can also provide support for users with accurate personalized service. Therefore, community detection algorithm has great theoretical significance and practical application value in areas such as network analysis, functional evolution and forecast.

In recent years the study of community detection have received wide attention in scholars, the related research results can be illustrated from different angles. Firstly, Palla *et al*. proposed CPM algorithm [3] based on complete subgraph. They allow vertexes belong to multiple communities at the same time, so overlapping community detection began to get wide attention and quickly becomes research hotspot. CPM algorithm thinks that community structure is made up of adjacent clique, a vertex can belong to several clique, so it can detect overlapping community. But some experience has shown that the CPM algorithm has a high time complexity and difficult to reach effective results when dealing with large-scale and higher-density complex networks. Secondly, Ahn *et al*. proposed LINK algorithm [4] based on link clustering. In the LINK algorithm, the Edge set is first divided into subsets, and then the results of division are translated into community structures of corresponding vertex. Thirdly, GN algorithm [7] based on betweenness. The GN algorithm deletes edges that have the largest betweenness by repeating to detect communities. GN algorithm need to recalculate betweenness after removing an edge. So it's time complexity is higher. Fourthly, algorithms based label propagation. Steve extended LPA algorithm [8], presented multi-label propagation algorithm called COPRA[9], to find overlapping community in large scale complex networks. In COPRA every vertex is given a unique label when the algorithm is initialized and each vertex can belong to up to v communities. Like LPA, time complexity of COPRA is linear. But in unknown complex networks we have no methods to evaluate number of communities that a vertex belongs to up to. Moreover, if the number of communities a vertex belongs to have a large difference, COPRA has difficult to detect a more accurate community structure by adjusting the parameters *v*.

2. Adaptive Label Propagation Algorithm

We propose **A**daptive **L**abel **P**ropagation **A**lgorithm (ALPA) to detect community structures in complex networks (ALPA).Like COPRA, the multi-label propagation was used to find overlapping communities. But the ALPA algorithm does not need to consider the problem of the parameters in any complex networks, so it has an excellent adaptability to the unknown complex networks. The ALPA algorithm has three main procedures: initialization, label propagation and post processing.

Definition 1 (Maximal Clique). The complex network is defined as an undirected graph G= {V, E}, and V denote vertex set, E⊆V×V, denote edge set in complex network. We find out induced complete graph with e (v_i, v_j) as the initial edge that v_i and v_j are maximal degree adjacent vertexes and their label sets are empty set. By iteration, their adjacent maximal degree vertex is constantly joined. This induced complete graph is called G_m, it's also called clique. If $G_m \subseteq$ G and no other complete graph $G_t \subseteq$ G and $G_m \subset G_t$, we call G_m Maximal Clique(MC).

Definition 2 (overlapping vertex). C_i and C_j are two different communities, if vertexes $v_i \in C_i \cap C_j$, we call v_i overlapping vertex.

Definition 3 (merge community). If exists community C_i and C_j ($i \neq j$), $C_i \subseteq C_j$. Delete C_i and save C_j.

2.1. Initialization

In COPRA algorithm each vertex is given a unique label in initialization, but during initial iterations vertexes receive different label that possible affect community detection quality, so the COPRA is unstable. To further enhance stability of the algorithm, we first consider reducing the number of labels in complex networks. Therefore we find out all disjoint MC in complex network and initialize vertexes of each MC with a weight and unique label. The approach can reduce redundant labels significantly, and enhance its stability. CPM has difficult to detect community in large-scale and higher-density complex networks due to its all clique tactic. In this paper we find out all disjoint MC to avoid consuming a lot of time in large-scale and higher-density complex networks. MC is the core unit of community through analysis of complex networks topology structure. A community at least contains one MC. In other words, vertexes belong to MC, must belong to the same community. So the community consists of MC and vertexes connected to MC. To sum up, initialization of the ALPA algorithm is shown below.

(1) Set $C_i = \emptyset$, C_i is label set of vertex v_i;
(2) Set id=1;
(3) By definition 1 find out one MC, and set $(\mathrm{id}, 1) \in C_i$, where id is label number and 1 is the label weight;
(4) id=id+1;
(5) Repeat (3)-(4), until no vertexes meet the requirements, initialization terminate.

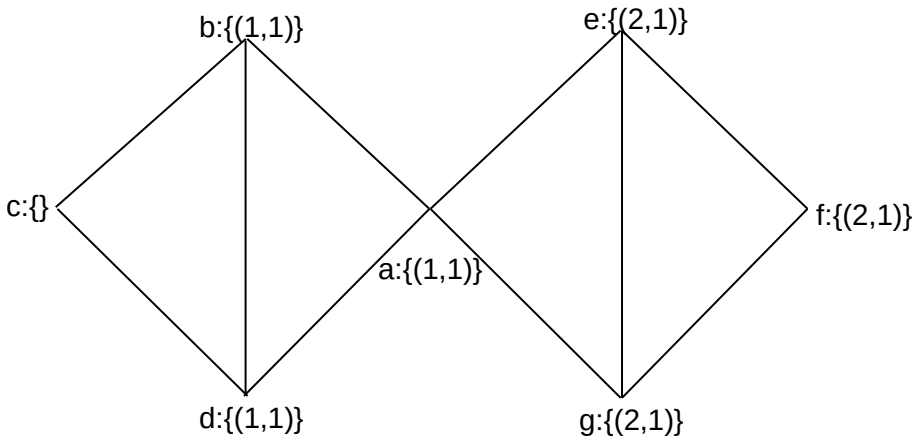

Figure 1. Result of After Initialization

We find out two MC in the Figure 1 by label initialization process, (a,b,d) and (e,f,g) respectively . Labels and weights are 1 for all vertexes respectively of the first MC, but in the second MC, vertex labels are 2 respectively, but the weights are still 1, result as shown in Figure 1. After initialization, vertexes of MC have label and weight, while the vertexes out of MC are not. By the model of small world in complex networks, if the vertex has neighbors, it would get a label in the algorithm iterations.

2.2. Design Alternatives

We describe alternatives briefly in this section. After initialization some vertexes get labels and weights, which would become seed vertex in propagation process. Each vertex label is a set of pair (c,x), where c is label number and x is the label c weight. The function $\mathrm{b}_t(c,x)$ is the weight value of label c in vertex x at the tth iteration. We use the following definition which is shown in formula (1). The vertex will inherit its neighbor vertexes' labels and weights which according to formula (1) at each iteration.

$$b_t(c,x) = b_{t-1}(c,x) + \frac{\sum_{y \in N(x)} b_{t-1}(c,y)}{|N(x)|} \quad (1)$$

Where $N(x)$ denotes the neighbor set of vertex x. Comparing to the asynchronous update mode, synchronous update label disseminate the results more stability [10]. Therefore we use synchronous update mode: the vertex label update in the tth iteration depends on the results in the (t-1)th iteration. It will take into account that no matter the vertex belong to a community or not, which is related to the distribution of vertex degree. So a vertex has L labels, weight of each label should not less than 1/L. Each vertex has a different number of labels, so L is variable. We define 1/ L as adaptive threshold. After each iteration the label pair will be deleted which is less than the adaptive threshold. Finally, normalize them so that weights of all labels for x vertex sum to 1. The specific iteration process is shown as follows.

(1). Set iteration number t = 1;
(2). Disrupt the order of vertexes, get a random sequence X;
(3). According to random sequence X update the vertex labels by formula (1);
(4). Delete the label pair whose weight less than 1 / L after each iteration, where L is the length of vertex label set. If all label weights of one vertex are less than 1 / L, we retain the largest label pair. When the largest label pair isn't unique, one of them is chosen randomly.
(5). Normalized the label weights of vertex, which are to be retained.
(6). If each vertex has at least one label, the algorithm terminates.
(7). Otherwise let t = t + 1, repeat (2) - (6).
(8). Algorithm terminated, vertexes that have the same label belong to the same community.

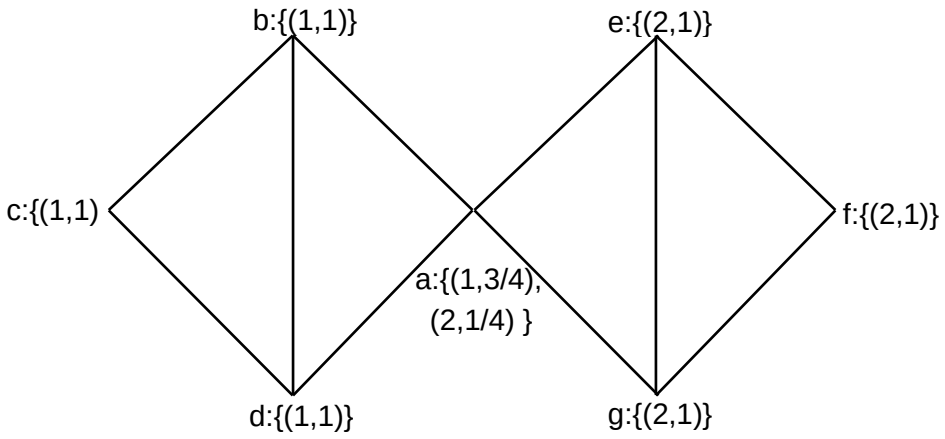

Figure 2. Result After Label Propagation

Figure 2 shows the results produced by the label update rules. After the first iteration, labels and weights of the vertexes as follows: *c*:{(1,1)}, *b*:{(1,2)}, *d*:{(1,2)}, *e*:{(1,1/3),(2,5/3)}, *f*:{(2,2)}, *g*:{(1,1/3),(2,5/3)}, *a*:{(1,3/2),(2,1/2)}. If the label spread farther that might affect the quality of the community detection. We avoid it by the fourth step of iteration rule. Therefore, the results of the first iteration are processed as follows. The vertex *e* has two labels, and their corresponding threshold is 1/2 respectively. According to our rule, delete label pair of the label weight below 1/2. So the vertex *e* reserved label is *2*. That is e:{(2,5/3)}. For vertex *g*, the solutions are almost the same, delete label weight below 1/2, final result is the label *2, that is g*:{ (2,5/3). Then the label weights of all vertexes are normalized. The final results are as follows: *c:*{(1,1)}, *b*:{(1,1)},*d*:{(1,1)}, *e*:{(2,1)}, *f*:{(2,1)}, *g*:{(2,1)}, *a*: {(1,3/4),(2,1/4)}. After first iteration, all vertexes have the labels, the algorithm achieves termination condition. Therefore, the network contains two communities: {a,b,c,d} and {a,e,f,g} respectively. According to

definition 2, vertex a is an overlapping vertex.

2.3. Termination

Like COPRA, simultaneous multi-label propagation cause our algorithm iteration will not converge to the vertex label remains unchanged state. We try to follow COPRA algorithm termination conditions, but did not get a good quality of community detection. We also try to limit the number of iterations, the results are still unsatisfactory. Therefore, we need a new termination condition.

In our algorithm, the number of community identifiers equal to the number of MC which has no intersection in complex networks, and with the continuously iteration the number of labels will gradually reduce, even reduced to one label. In order to prevent the spread of community identifiers too far which lead to the formation of a super-large community, even one community. We use all of the vertexes in complex network having the label as the algorithm termination condition. However, few relationships between vertexes in the network maybe have not been calculated when the ALPA algorithm terminates. The experience shows that these vertexes are usually not overlapping vertexes and they have a label that is sufficient. So when the algorithm terminates even if losing some relationships, the quality of community detection is not affected.

2.4. Post Processing

When the algorithm terminates, some vertexes contain more than one label, each vertex whose label contains community identifier *c* is simply allocated to community *c*. In order to improve the quality of community detection, we need post-processing to detect the generated community. If the length of vertex v_i label set in community is greater than 1 (L>1), it indicates that the vertex is overlapping vertex. So the vertex v_i is placed in the corresponding multiple communities. However, some sub-communities have formed. It will be deleted according to Definition3.

2.5. Complexity

The time complexity of each step is estimated as below. *n* is the number of vertices in the complex network and *k* is average degree of all vertices, *H* is the number of MC and *m* is the average number of vertexes in each MC.

(1). Set $C_i = \emptyset$, takes time $O(n)$. To find out all disjoint MC takes time $O(\frac{n(n+1)}{2} + H(m-1)k)$. The time for the whole phase is therefore $O(\frac{n(n+3)}{2} + H(m-1)k)$.

(2). Labels update phase, *T* is the number of iterations when the algorithm terminates. Each vertex label update have four steps, it's shown as follows.

(a) Receive new label of neighboring vertexes;
(b) Summing the same label with the neighbor vertexes;
(c) Delete the noise label according to adaptive threshold;
(d) Normalized the vertex label weights.

So the time for the whole phase is $O(Tkn)$. And post-processing, the total time is $O(n)$.

For a complex network, the whole time complexity is therefore $O(\frac{n^2}{2} + (\frac{5}{2} + Tk)n + H(m-1)k)$.

3. Experiments

3.1. Methodology

Now，there are two ways to evaluate the performance of the ALPA algorithm. One is to run the algorithm on real-world network dataset. A problem is how to judge the community detection quality because we usually do not know the real communities that are present in the original data. The other method is to use randomly generated synthetic networks based on LFR[11] network generator and compare the known communities with those found by the ALPA algorithm. A benefit of this method is that we can analyze the algorithm's performance in detail. The drawback is that synthetic networks might not share the properties of real networks.

For synthetic networks, LFR network generator produces benchmark networks that are claimed to possess properties found in real networks, such as number of vertex, heterogeneous distributions of degree and community size. Although not described in [11], the generator also allows communities to overlap, with the restriction that every overlapping vertex belongs to a fixed number of communities. To evaluate overlapping communities on the synthetic networks, we use the Normalized Mutual Information (NMI) measure of Lancichinetti *et al* [12] in the experiments reported in this paper. $NMI \in [0,1]$, it measures the closeness of the found communities to the real communities.

The benchmark networks parameters of our experiments are: the mixing parameter ($u \in [0.1, 0.8]$), each overlapping vertex belongs to two communities (O_m=4). The exponents of the power-law distribution of vertex degrees (t_1=2) and community sizes (t_2=2). The remaining parameters are: the number of vertices (*n*), the average degree (*k*), and the maximum degree (k_{max}), the minimum community size (C_{min}), the maximum community size (C_{max}), and the number of overlapping vertices (O_n). They are shown in table 1.

Table 1. LFR Benchmark Network Remaining Parameters

Network ID	*n*	*k*	k_{max}	C_{min}	C_{max}	O_n
S_1	1000	10	20	10	20	20
S_2	1000	10	20	20	50	20
S_3	5000	20	100	20	100	100
S_4	5000	40	100	40	100	500

For real networks, the most common measure is modularity [13-14]. Nicosia *et al* [15] designed a variant that is defined for overlapping communities. So we use this *overlap modularity* (Q_{ov}) measure for the experiments in this paper. The Q_{ov} is defined as formula (2).

$$Q_{\mathrm{ov}} = \frac{1}{2m}\sum_{ij}(A_{ij} - \frac{k_i k_j}{2m})\delta(C_i, C_j) \quad (2)$$

Where *m* denotes the number of edges, and k_i, k_j is degree of vertex i and j. A denotes the adjacency matrix of the network, the vertexes i and j are adjacent, A_{ij} = 1, otherwise A_{ij} = 0. If the vertexes i and j in the same community，$\delta(C_i, C_j)$=1，else $\delta(C_i, C_j)$=0. When i=1, j=2, and j=1, i=2, the set of vertexes is the same contribution to Q_{ov}. For ease of calculation we evolved formula (2) into the formula (3).

$$Q_{ov} = \frac{1}{m}\sum_{ij}^{i<j}(A_{ij} - \frac{k_i k_j}{2m})\delta(C_i, C_j) \quad (3)$$

3.2. Comparison with Other Algorithms

In this section, we use synthetic networks to compare the performance of ALPA with some other algorithms. We use benchmarks network as shown table 1 for comparison. The network size is either 1000 or 5000. The vertex degree is 10, 20 or 40. The vertex max degree is either 20 or 100. The overlapping vertex is 20,100 or 500. Community sizes are in the range 10–40 or 20–100, the mixing parameter μ varies from 0.1 to 0.8 and other parameters are fixed.

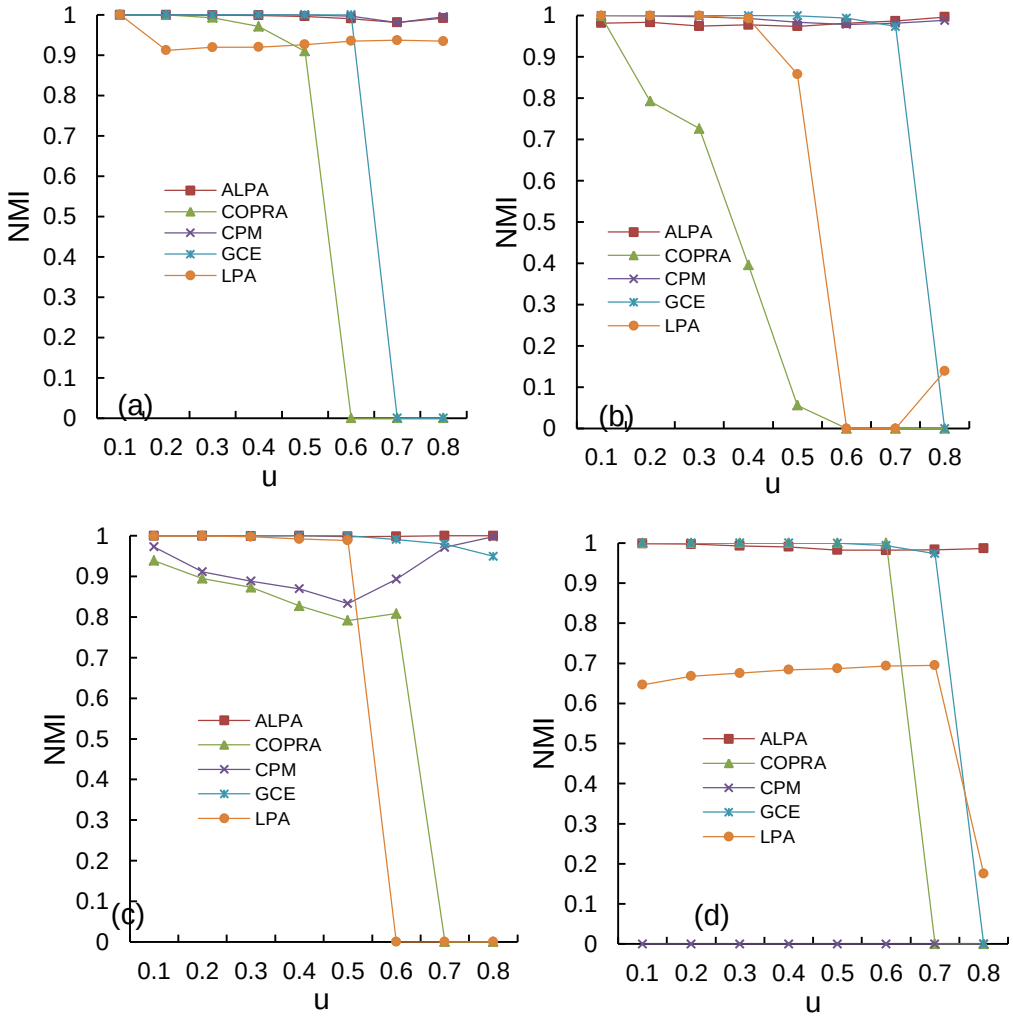

Figure 3. NMI of ALPA and Some Other Algorithms on S_1 (a), S_2 (b), S_3 (c) and S_4 (d)

In Figure 3, the algorithm parameters about:(1)COPRA,v=4.(2)CPM,k=3.(3)GCE, α=1.0.About COPRA and ALPA, NMI is average result of running 10 times on every benchmark network dataset. These results show that LPA is more no effective for overlapping community detection. Parameter v is equal to 4 in COPRA algorithm, and this parameter fits overlapping community number of our synthetic network dataset. But the figure 3 shows that NMI of experiment is not satisfactory. In small networks, CPM and ALPA have the best quality of community detection, but CPM is instability in bigger networks, and it can't detect community in dense network. Meanwhile, GCE algorithm can't detect community in small networks when community structure is not obvious.

In addition, the COPRA algorithm is unstable because of excessive redundancy label.

The ALPA algorithm at least three vertexes have a unique label, moreover experiments show that label number of ALPA is about 5% of the number of network vertexes. So ALPA reduced the risk of label selected during iterations, and therefore ALPA has more stability than COPRA. It is worth mentioning that ALPA algorithm still remains good community detection quality when the community structure is not obvious, which is related to our definition about MC. Therefore, MC is core of the community. It is also the key factor that the ALPA algorithm achieves better quality and stability.

The real networks that we use and their sizes are listed in table 2. We have also compared four other classic community detection algorithms on our real networks using formula (3). Table 2 shows information of five real network datasets and modularity Q_{ov} of these algorithms running on them. GCE algorithm has better community detection quality when community structures on synthetic networks are obvious. But Q_{ov} of GCE is zero on real network "CA-hepPH". ALPA gives the best modularity than the other four algorithms for network tested, except "football". On football dataset, Q_{ov} of ALPA is 0.63, but Q_{ov} of COPRA is 0.65. The difference is very small too.

Table 2. Comparison of ALPA with Other Algorithms on Real Networks

Name	Vertices	Edges	Q_{ov}				
			ALPA	GCE	COPRA	CPM	LPA
Karate	34	78	0.43	0.35	0.05	0.24	0.31
Dolphins	62	159	0.56	0.49	0.47	0.44	0.33
Football	115	613	0.63	0.60	0.65	0.20	0.50
Power	4941	6594	0.95	0.08	0.15	0.21	0.23
CA-hepPH	11204	117649	0.51	0.00	0.42	0.00	0.46

Therefore the experimental results on the synthetic network and real network show ALPA algorithm improves the tolerance of the parameters *u*, it can obtain better community detection quality when community structures are not obvious. In other words, ALPA algorithm is not affected by the network size and community structure, more suitable for community detection on various types of unknown complex networks.

4 Conclusions

4.1. Contributions

We propose a new algorithm called ALPA, to detect overlapping communities in complex networks by label propagation. Found MC and used unique label to initialize vertex in MC. compared with COPRA and LPA, our method decrease redundant label, and therefore dramatically decreasing the random factors. MC strategy also improves community detection quality, especially in complex networks with less obvious community structures. Adaptive threshold method delete redundant labels avoiding all labels owned all vertices, and this strategy overcome pre-parameter limitations in unknown complex networks.

So ALPA algorithm has ability to detect community structure in any complex networks, it is particularly suitable for community detection in real commercial complex network, and its result can provide support for accurate personalized service.

4.2. Future Work

There are at least two directions in which this work could be extended. One is extending definition of MC. When there are multiple vertices simultaneously eligible to join MC, whether we should consider allowing them to join at the same time? In general, they should belong to the same community. Another is to develop faster implementations. The algorithm is highly amenable to parallel implementation because each vertex can be updated independently during each propagation step, as a result of its use of synchronous updating. Parallelization implementation can make the ALPA algorithm more quickly detect community in complex networks with millions of vertices.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant numbers 61272067 and 61502180, by the Natural Science Foundation of Guangdong Province, China under grant numbers S2012030006242, 2014J4300033 and 2014A030310238, by the Research Fund of Educational Commission of Guangdong Province of China under grant number 2014WTSCX078.

References

[1] M. T. Thai and P. M. Pardalos, "Handbook of Optimization in Complex Networks", Theory and Applications, Springer-Verlag New York Publishers, Berlin, **(2011)**.
[2] F. Reid, A. McDaid and N. Hurley, "Partitioning breaks communities", Proceedings of International Conference on Advances in Social Networks Analysis and Mining, Kaohsiung, Taiwan, **(2011)**.
[3] G. Palla, I. Derényi, I. Farkas and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society", Nature, vol. 435, no. 7043, **(2005**), pp. 814-818.
[4] Y. Y. Ahn, J. P. Bagrow and S. Lehmann, "Link communities reveal multiscale complexity in networks", Nature, vol. 466, no. 7307, **(2010)**, pp. 761-764.
[5] B. Ball, B. Karrer and M. E. J. Newman, "An efficient and principled method for detecting communities in networks", Physical Review E, vol. 84, no. 3, **(2011)**, pp.036103.
[6] Y. Kim and H. Jeong, "Map equation for link community", Physical Review E, vol. 84, no. 2, **(2011)**, pp. 026110.
[7] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks", Proceedings of the National Academy of Sciences, vol. 99, no.12, **(2002)**, pp. 7821-7826.
[8] U. N. Raghavan, R. Albert and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks", Physical Review E, vol. 76, no. 3, **(2007)**, pp. 036106.
[9] S. Gregory, "Finding overlapping communities in networks by label propagation", New Journal of Physics, vol. 12, no.10, **(2010)**, pp. 103018.
[10] I. X. Y. Leung, P. Hui, P. Liò and J. Crowcroft, "Towards real-time community detection in large networks", Physical Review E, vol. 79, no. 6, pp. 853-857.
[11] A. Lancichinetti, S. Fortunato and F. Radicchi, "Benchmark graphs for testing community detection algorithms", Physical Review E, vol. 78, no. 4, **(2008)**, pp. 046110.
[12] A. Lancichinetti, S. Fortunato and J. Kertész, "Detecting the overlapping and hierarchical community structure of complex networks", New Journal of Physics, vol. 11, no. 15, **(2008)**, pp. 19-44.
[13] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks", Physical Review E, vol. 69, no. 2, **(2004)**, pp. 026113.
[14] M. E. J. Newman, "Modularity and community structure in networks", Aps March Meeting, vol. 103, no. 23, **(2006)**, pp. 8577-8582.
[15] V. Nicosia, G. Mangioni, V. Carchiolo and M. Malgeri, "Extending modularity definition for directed graphs with overlapping communities", J. Stat. Mech., **(2009)**, pp. P03024

Authors

Chunying Li, she was born in 1978. She received his master degree from Beijing Institute of technology,Beijing city, in 2007. Now she is pursing the Ph.D. degree from South China Normal University, Guangzhou, China. Currently, she is an associate professor of Guangdong Polytechnic Normal University,

Guangzhou,China. Her research interests include social network and cooperative computing.

Yong Tang, he was born in 1964. He received the Ph.D. degree from University of Science and Technology of China, Beijing city, in 2001. Currently, he is a professor and dean of School of Computer Science, South China Normal University. His research interests include temporal database, cooperative computing, cloud computing and social network service. He is associate director of Technical Committee on Collaborative Computing of CCF. He has also served as general or program committee co-chair of more than 10 international/national conferences.