Physics Letters A 383 (2019) 2481–2487

Contents lists available at ScienceDirect

Physics Letters A

www.elsevier.com/locate/pla

Growth curve based label propagation algorithm for community detection

Xue Li

Computer Science, Northeast Forestry University, Harbin, 150040, China

ARTICLE INFO

Article history:
Received 10 January 2019
Received in revised form 27 April 2019
Accepted 2 May 2019
Available online 8 May 2019
Communicated by F. Porcelli

Keywords:
Complex network
Community detection
Label propagation
Growth curve

ABSTRACT

How to better and faster identify the community structure is a hot issue in complex networks. During the past decades, various attempts have been made to solve this issue. Amongst them, without doubt, label propagation algorithm (LPA) is one of the most satisfying answers, especially for large-scale networks. However, it has one major flaw that when the community structure is not clear enough, a monster community tends to form. To address this issue, we set a growth curve for communities, gradually increasing from a low capacity to a higher capacity over time. Further, we improve the mechanism of label choosing for small communities to escape from local maximum. The experimental results on both synthetic and real networks demonstrate that our algorithm not only enhances the detection ability of the traditional label propagation algorithm, but also improves the quality of the identified communities.

1. Introduction

Any collection of entities and their reciprocal relationships in real life systems, such as medical systems [1] [2], traffic systems [3], and society [4], could be abstracted as a network of nodes and edges. One of the most important properties appeared in networks is community structures which are also called groups, clusters, cohesive subgroups or modules in different contexts. As a result, the exploration of the community structure in complex networks has aroused a great amount of attention throughout the recent years. However, the precise definition of the community still doesn't reach an agreement yet in the literature, but like many imperfect defined terms, it is suggestive rather than a settled notion [5]. Generally speaking, individuals in a network tend to form closely-knit groups and interact more frequently with members within the group than those outside the group.

To measure the strength of a community partition, Newman et al. [6] put forward a useful method called modularity. It quantifies the quality of a partition for a certain network by measuring how much denser the edges are within the groups compared to what they would be in a random graph with the same distribution of degrees. With the emerge of modularity, many algorithms for community detection set it as their objective function, which makes community detection become a modularity optimization problem. The definition of modularity can be written as below:

$$Q = \sum_{t=1}^{N_c} \left(\frac{I_t}{2m} - \left(\frac{D_t}{2m}\right)^2\right) \tag{1}$$

where I_t is the number of edges inside partition t, D_t is the total degree of nodes inside t, $2m$ is the total number of edges, N_c is the set of all communities.

Over the past decades, various kinds of community detection algorithms have been proposed. Roughly speaking, these approaches can be separated into four categories: node-centric, group-centric, network-centric, and hierarchy-centric [7]. Node-centric criteria, commonly used in traditional social network analysis, requires each node in a group to satisfy certain properties. For instance, considering the reachability, any node in a group should be reachable in k hops. A k-clique algorithm [4] tries to find a maximal subgraph in which the largest geodesic distance between any bodes $\leq$ k. It is still a challenge to generalize node-centric algorithms to large-scale networks. A group-centric criterion requires the whole group to satisfy a certain condition. One such example is density-based communities [8]. A subgraph can be seen as a cohesive subgroup when its density is greater than or equal to a given threshold. The network-centric community detection considers the global topology of a network and aims to partition nodes of a network into a number of disjoint sets. The approaches include clustering based on vertex similarity [9], latent space models [10], block model approximation [11], spectral clustering [12], and modularity maximization [13]. The algorithms presented above can

E-mail address: nefu_education@126.com.

https://doi.org/10.1016/j.physleta.2019.05.005
0375-9601/

normally be applied to medium-size networks except the latent space model. The high computational complexity hinders its application. Another line of community detection research is to build a hierarchical structure of communities based on network topology. This facilitates the examination of communities at different granularity. There are mainly two types of hierarchical clustering: divisive [14], and agglomerative [15].

Among all the approaches and techniques described above, a network-centric based famous approach called label propagation algorithm (LPA) [16] is without doubt one of the most efficient detection methods. However, the original LPA has some drawbacks, such as the occurrence of monster community, high randomness, and weak robustness. A standard approach for eliminating the monster community phenomenon is to add constraints to the objective function of the method. For example, Barber et al. [17] set a penalty term to Hamiltonian function and tried to get communities with the same total degree. Zhang et al. [18] integrated maximum belonging coefficient and edge probability into LPA. Chen et al. [19] introduced information entropy to describe the relation between direct and indirect neighbors. Another popular approach to attenuate the instability of LPA is to set an update order by calculating the node importance. Berti at el. [20] showed that ordering nodes according to centrality measures, such as betweenness centrality, closeness centrality, and page rank centrality can improve the quality of groups detected. Moreover, some scholars think the reason for high randomness is that every node in the network is initialized with a unique label. Some of them combined the above approaches and proposed a core based label propagation, in which only core members have labels at the initial phase [21].

Different from the above approaches, we propose a new way to avoid the occurrence of monster community and increase the stability of the method. Without too many changes, we set a growth curve for LPA to balance the community growth rate in case that some small communities are too easy to be swallowed. Further, the mechanism of label choosing for propagation conflicts in the original LPA will produce fake groups, which are sparse and hard to grow up or to be swallowed. We slightly change the mechanism by keeping randomly choosing labels for small subgroups when propagation conflicts, until they are big enough. We validate the proposed method on both synthetic and real networks, which demonstrates that our algorithm not only improves the detection ability of the traditional label propagation algorithm but also improves the quality of the identified communities.

The structure of the remainder of this paper is as follows. In Sec. 2, we briefly introduce the original LPA, growth curve, and the definition of the smallest group, which are the bases of next section. In Sec. 3, the proposed algorithm is presented and we validate it in Sec. 4. Finally, we conclude with a summary in the last section.

2. Related work

In this section, we will introduce several relevant definitions that are used in this paper, mainly including: original label propagation, growth curve, and definition of the smallest group.

2.1. The label propagation

Label propagation algorithm was proposed by Raghavan et al. [16], which only considers topology, requiring less extra information. It is famous for its simplicity and time-efficiency. The main idea of LPA is simple; it works as follows:

Step 1: Each node is initialized with its node id (i.e. with a unique label).

Step 2: Each node updates its label to a new one which occurs with the highest frequency among the neighbors.

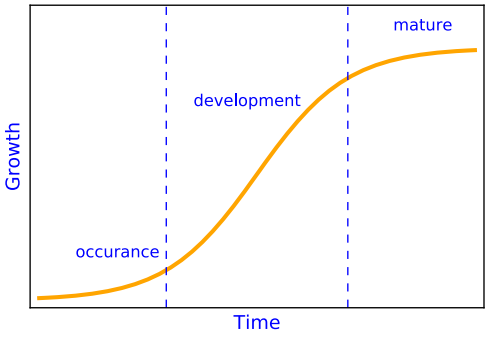


Fig. 1. The growth curve.

Step 3: If all the labels are no longer changed, then stop the algorithm, or back to Step 2. Finally, each community is determined by the identical label.

The label updating principle is denoted as follows [22]:

$$l_v^{new} = \underset{l}{argmax} |N^l(v)| \tag{2}$$

where v is a node, l denotes the label of a node, N_v is the neighbors of v.

2.2. The growth curve

Here, we borrow the term, growth curve, from statistics to describe the growth pattern of the communities in a network. Values for the community size can be plotted on a graph as an S-shaped curve function of time; see Fig. 1 for an example.

Generally speaking, things go through three stages of occurrence, development, and maturity, and the growth rate varies in different stages [23]. Usually, in the stage of occurrence, the speed of growth is relatively slow; in the development stage, the speed of change is accelerating; in the mature stage, the growth rate is slowing down. The level beyond which no major increase can occur is referred to as saturation level or carrying capacity.

2.3. The smallest group based on modularity resolution

Modularity is still one of the most widely used metrics to quantify community structures, although it may fail to identify some smaller modules that contain fewer links than $\sqrt{2m}$, which is known as resolution limit (RL) [24].

Here, we give the definition of the smallest group size based on the edge resolution limit. Let ns and m denote the size of the smallest group and the total edges of a network respectively. With the assumption that there is a connection between any two nodes, then the relation of ns and m can be defined as follows:

$$C_2^{ns} = \frac{ns \times (ns-1)}{2} = \sqrt{2m} \tag{3}$$

Then the size of the smallest group in a certain network can be written as below:

$$ns \approx \sqrt{2\sqrt{2m}} \tag{4}$$

The real-world networks are sparse and there is no clear relationship between the number of the nodes and the edges, so ns is the lower limit of the size of groups containing $\sqrt{2m}$ links. We can define ns for any network. For example, the smallest group size for a network with 30000 edges is 16. If the size of a group is less than ns, it can be regarded as a small piece which hinders the community detection. It is necessary to prevent unnecessarily large amounts of small pieces from hiding in the networks.

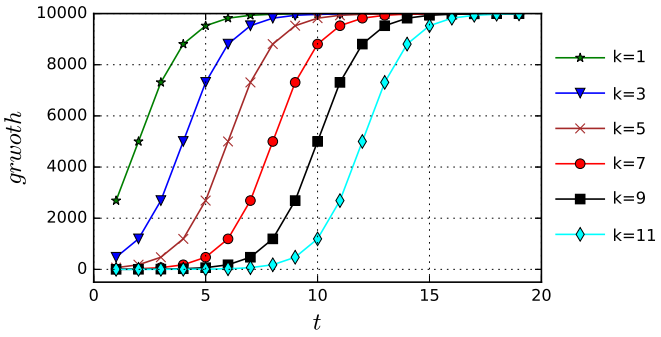


Fig. 2. The growth capacity curve.

3. Proposed method

For original LPA, some scholars observed that roughly 70% of the nodes do not change their labels after the second iteration [25] and 95% of the nodes reach their final state in 5 iterations [16]. Thus the phenomenon that many potential small groups are swallowed already happened in the early stage. The growth rate, which varies significantly for different groups, is a noticeable factor for the monster community. Based on above considerations, to prevent over propagation, we integrate growth curve into LPA and present a growth-constrained LPA. The growth curve function is defined as follows:

$$growth(t) = \frac{N}{1 + e^{-t+1+k}} \tag{5}$$

where N is the number of nodes in networks, t is the number of iterations, k is relative with the number of times we increase the size of communities. The relationship among N, t, and k can be observed in Fig. 2. When t is bigger enough, there is no constraint on the size of communities. Generally, in the first iteration, we require the group capacity is bigger than 1, which means that we need $growth(t = 1) \geq 1$, so the maximum k for the network of ten thousand nodes in Fig. 2 is 9 ($growth(t = 1) = 1.23$).

Growth curve gives the weak cores a chance to grow up. However, not all of them can grow into a community. To recognize and release these invalid small pieces, we define the size of the smallest group and change the mechanism of label choosing. In our algorithm, after the second iteration, we will increase the instability of these small pieces. If the size of the structure identified by the existing label is less than the size of the smallest group, the existing label will randomly change. All the steps in our algorithm are shown as follows:

Algorithm 1 GCLPA.

for $v \in V$ **do**
  $L_v \Leftarrow id(v)$
end for
for t : 1 to T **do**
  $com\_size(t) \Leftarrow growth(t)$
  for $v \in V$ **do**
    $A \Leftarrow \{L | S(L) < com\_size(t), Nei_v(L) > 0\}$
    $C \Leftarrow \{L | L \in A, Nei_v(L) = max(Nei_v(L' | L' \in A))\}$
    $candidate \Leftarrow$ random select from C
    $L_v \Leftarrow candidate$
    if $S(L_v) < ns$ and size of $C > 1$ **then**
      $L_v \Leftarrow$ random select from C
    end if
  end for
end for

There $S(L)$ is the number of nodes with a label L, $Nei_v(L)$ denotes the number of nodes having a label L among the neighbors of v, and ns is the smallest group size for a network.

In the next section, we show the results of our tests on various datasets and examine the effectivity of our algorithm.

4. Experiments

In this section, we verify the performance of our algorithm in different ways. Firstly, we validate that the prevention of both over propagation and invalid small pieces will not reduce the quality of communities compared to traditional LPA. To validate that, we compare our algorithm's performance with that of LPA on real networks. Then we compare the performance of LPA and GCLPA on synthetic networks with different group sizes. After that we apply powerful state-of-the-art algorithms and our algorithm to synthetic networks.

4.1. Measurements

We use Eq. (1) to calculate the modularity for tests of real-world networks which are without the ground truth.

To evaluate the performance of community detection in synthetic networks, we use another commonly adopted measure, normalized mutual information (NMI), proposed by Danon [26]. The NMI compares the difference between real communities and the found communities by measuring the amount of their mutual information. It's defined as:

$$NMI(X|Y) = (-2 \times \sum_{i=1}^{|X|}\sum_{j=1}^{|Y|} |X_i \cap Y_i| \times log(\frac{n \times |X_i \cap Y_i|}{|X| \times |Y|})) \times (\sum_{i=1}^{|X|} |X_i log(\frac{|C_i|}{n})|) + \sum_{j=1}^{|Y|} |Y_j| log(\frac{|C_i|}{n})|)^{-1} \tag{6}$$

where n is the number of nodes in the network, X denotes a partition generated by the algorithm, and Y is the corresponding real communities.

4.2. Tests on real-world networks

We have tested GCLPA against LPA on twelve well-known networks which are widely used in literature. Table 1 lists the details of these networks. They are involved in different fields such as biology, physics, politics and society. We ran our algorithm with different k (the number of times to increase the size of the group) such as 1, 3, 5 on all networks and select one of the best as the final result. For the sake of simplicity, all the networks are treated as unweighted and undirected. Table 2 lists modularity scores for 12 datasets averaged on 100 or 3 realizations based on the size of networks.

To have a better understanding of the output from the detection algorithms, we calculate minimal modularity, average modularity and maximal modularity in Table 2. In almost all the test networks, GCLPA achieves a higher modularity, meaning that let the weak communities grow can improve the overall quality of detected communities. Moreover, the K-value choosing is not so difficult because of its upper limit for finding a positive initial capacity.

4.3. Tests on synthetic networks with different sizes

To explore the performance of LPA and GCLPA, we generate 6 synthetic networks with different group sizes. The artificial networks employed in our experiment are generated by LFR benchmark. By construction, we can use LFR model to produce networks considered as realistic [27]. The LFR parameters are as below:

Table 1
The real networks.

Network	Description	Nodes	Edges
Karate	Zachary's karate club	34	78
Dolphins	Lusseau's bottlenose dolphins	62	159
Lesmis	Characters in the novel Les Miserables	77	254
Polbooks	Co-purchased political books	105	441
Football	American football league	115	613
Jazz	Jazz musicians network	198	5484
Netscience	Scientists working on network theory	1461	2742
Polblogs	U.S. political weblogs	1490	19090
Power	Western U.S. power grid	4941	6594
Hep-th	High Energy Physics Archive	7610	15751
CondMat	Collaboration network of Arxiv Condensed Matter	23133	93497
DBLP	DBLP collaboration network	317080	1049866

Table 2
The modularity comparison on real-world networks.

Datasets	LPA			GCLPA				
	Ave_Q	Min_Q	Max_Q	Ave_Q	Min_Q	Max_Q	K	maxK
Karate	0.3426	0.0	0.4033	0.3907	0.3160	0.4156	3	4
Dolphins	0.4802	0.2209	0.5265	0.5077	0.4435	0.5285	3	5
Lesmis	0.4967	0.2089	0.5510	0.5368	0.5089	0.5532	3	5
Polbooks	0.4945	0.4509	0.5222	0.5135	0.4645	0.5262	3	5
Football	0.5920	0.5432	0.6052	0.5937	0.5448	0.6054	1	5
Jazz	0.3400	0.0	0.4426	0.4040	0.2787	0.4428	5	6
NetScience	0.9098	0.8950	0.9220	0.9062	0.8907	0.9160	3	8
Polblogs	0.3962	0.0005	0.4261	0.4228	0.0008	0.4316	1	8
Power	0.8036	0.7923	0.8151	0.8026	0.7924	0.8106	3	9
Hep-th	0.7696	0.7518	0.7778	0.7603	0.7453	0.7700	1	10
CondMat	0.6181	0.4286	0.6399	0.6261	0.6068	0.6359	1	12
Dblp	0.6887	0.6714	0.6975	0.6907	0.6871	0.6934	1	14

Net1: $N = 1000, < k >= 20, k_{max} = 50, t1 = 2, t2 = 1, c_{min} = 5, c_{max} = 16, \mu = 0.6$

Net2: $N = 1000, < k >= 20, k_{max} = 50, t1 = 2, t2 = 1, c_{min} = 25, c_{max} = 100, \mu = 0.6$

Net3: $N = 1000, < k >= 20, k_{max} = 50, t1 = 2, t2 = 1, c_{min} = 10, c_{max} = 100, \mu = 0.6$

Net4: $N = 1000, < k >= 50, k_{max} = 100, t1 = 2, t2 = 1, c_{min} = 5, c_{max} = 21, \mu = 0.6$

Net5: $N = 1000, < k >= 50, k_{max} = 100, t1 = 2, t2 = 1, c_{min} = 25, c_{max} = 100, \mu = 0.6$

Net6: $N = 1000, < k >= 50, k_{max} = 100, t1 = 2, t2 = 1, c_{min} = 10, c_{max} = 100, \mu = 0.6$

The scales of the size such as 5-16 and 5-21 are designed according to the smallest group size definition to show the community resolution of the algorithms.

Table 3 lists NMI and modularity scores for those synthetic datasets averaged on 20 realizations.

As the table shows that in both NMI and modularity, GCLPA is more stable and can get a better result. However, LPA easily fails to detect the communities. The extreme monster community phenomenon of LPA can be observed in Net2 and Net3 and obviously, the whole graph is regarded as one community. We demonstrate the details of the optimal modularity outputs for Net4, Net5, Net6 in Fig. 3.

The Max size and Min size in Fig. 3 are corresponding to the maximal and minimal group size of the true network partitions. It is obvious that GCLPA can prevent the over propagation and recognize the communities in proper size scales. The application of growth curve greatly improves the performance of GCLPA. By contrast, LPA can easily form some giant communities, which hinders the detection.

Table 3
The comparison between LPA and GCLPA.

Networks	Algorithms	NMI	MaxQ	MeanQ	MinQ
Net1	LPA	0.7996	0.3775	0.3156	0
	GCLPA	0.9994	0.3773	0.3767	0.3753
Net2	LPA	0	0	0	0
	GCLPA	0.4577	0.3306	0.1611	0
Net3	LPA	0	0	0	0
	GCLPA	0.8666	0.3456	0.3040	0
Net4	LPA	0.5008	0.3620	0.1907	0
	GCLPA	1.0	0.3652	0.3652	0.3652
Net5	LPA	0.1786	0.3327	0.0639	0
	GCLPA	0.9992	0.3462	0.3453	0.3377
Net6	LPA	0.1349	0.3290	0.0477	0
	GCLPA	1.0	0.3408	0.3408	0.3408

As mentioned earlier, we set the smallest group size, *ns*, for every network to decrease the smaller communities in each iteration. To show its importance, we define the algorithm without monitoring *ns* as GCLPA* and compare it with GCLPA on Net1.

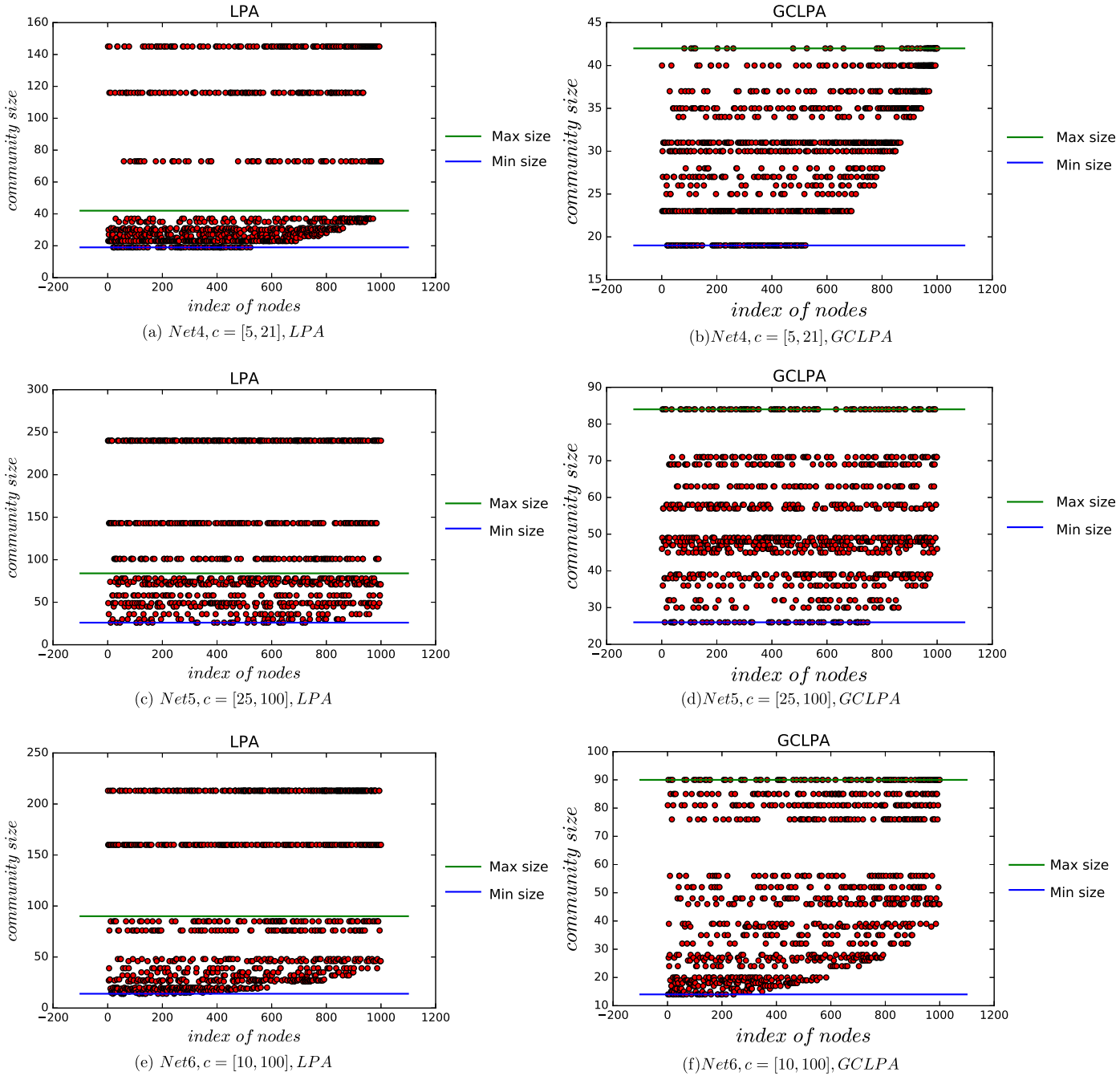
From Fig. 4, we can see that GCLPA detects smaller communities than GCLPA*, meaning that monitoring the smallest group size is helpful to filter the invalid small pieces.

4.4. Comparison with traditional algorithms

In this part, we compare GCLPA with original LPA, Louvain [15] and CNM [28], which are the state-of-the-art algorithms. We generate 4 LFR benchmark networks with the following parameters:

$N = 1000,\ < k >= 20,\ k_{max} = 50,\ t1 = 2,\ t2 = 1,\ c_{min} = 10,\ c_{max} = 50,\ \mu \in [0.1 - 0.6]$

$N = 1000,\ < k >= 20,\ k_{max} = 50,\ t1 = 2,\ t2 = 1,\ c_{min} = 20,\ c_{max} = 100,\ \mu \in [0.1 - 0.6]$



(a) $Net4, c = [5, 21], LPA$ (b) $Net4, c = [5, 21], GCLPA$

(c) $Net5, c = [25, 100], LPA$ (d) $Net5, c = [25, 100], GCLPA$

(e) $Net6, c = [10, 100], LPA$ (f) $Net6, c = [10, 100], GCLPA$

Fig. 3. The detected community sizes.

$N = 5000,\ < k >= 20,\ k_{max} = 50,\ t1 = 2,\ t2 = 1,\ c_{min} = 10,\ c_{max} = 50,\ \mu \in [0.1 - 0.6]$

$N = 5000,\ < k >= 20,\ k_{max} = 50,\ t1 = 2,\ t2 = 1,\ c_{min} = 20,\ c_{max} = 100,\ \mu \in [0.1 - 0.6]$

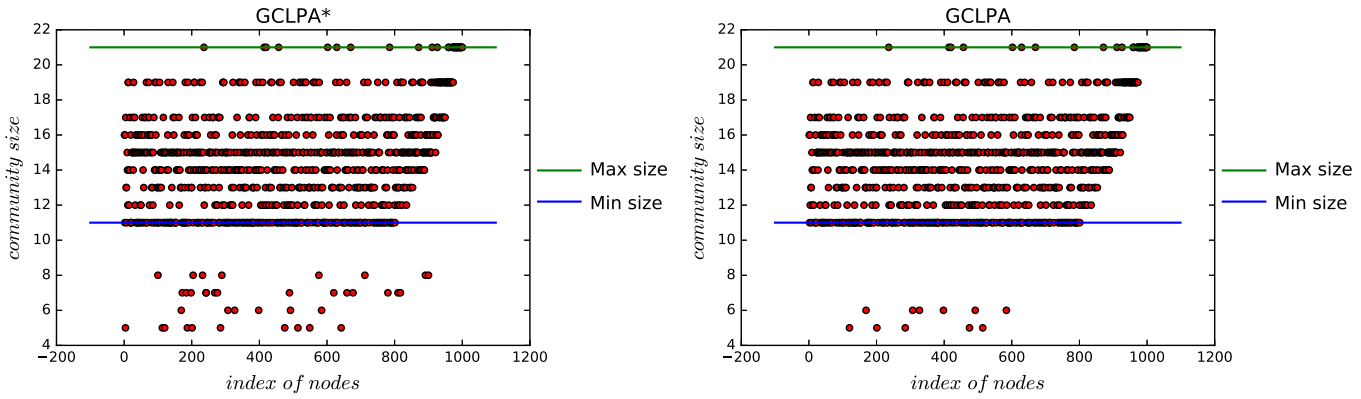
The properties of LFR networks, such as transitivity and degree correlation, are significantly affected by changes in μ. The larger it is, the less clear the community structure is.

The experiment results are displayed in Fig. 5, where each point in curves is obtained by running an algorithm 100 times for each value of the mixing coefficient.

As shown in Fig. 5(c), (d), both LPA and GCLPA achieve superior accuracy over CNM and Louvain even up to a mixing parameter of 0.6. Interestingly, the original LPA shows signs of failure at about $\mu = 0.5$ in the N = 1000 benchmark networks, meaning that some monster communities are formed. Within the detectable range of the original LPA, GCLPA performs better. The application of growth curve greatly improves the overall performance of LPA.

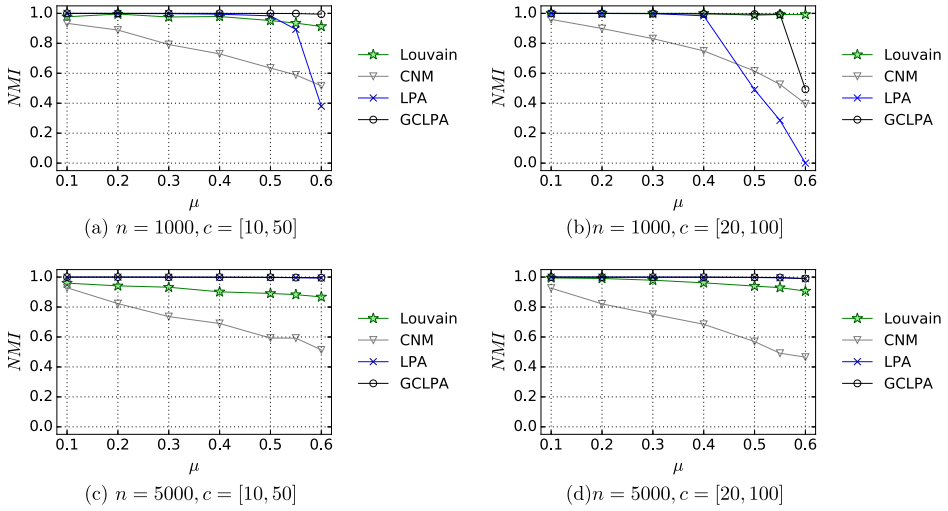
5. Conclusion

This paper presents a label propagation algorithm for community detection based on the growth curve. To ensure the weak cores of communities have a chance to grow, we adopt a strategy that constrains the size of community in each iteration to balance the growth rate. We also define the size of the smallest community based on the resolution of modularity and invalidate small pieces, of which the size is smaller than the smallest community size. By experiments on real and synthetic networks, we demon-



(a) $Net1, c = [5, 16], GCLPA*$ (b)$Net1, c = [5, 16], GCLPA$

Fig. 4. The comparison between GCLPA* and GCLPA.



(a) $n = 1000, c = [10, 50]$ (b)$n = 1000, c = [20, 100]$

(c) $n = 5000, c = [10, 50]$ (d)$n = 5000, c = [20, 100]$

Fig. 5. Average NMI comparison on synthetic networks.

strate that the proposed algorithm has a better performance than some of the current representative algorithms. In the future work, we will try to combine the techniques used in segmentation such as low-rank decomposition [29] and hierarchical local region [30] with the label propagation algorithm.

References

[1] Yong Chang, Yongfeng Yuan, Changyong Guo, Yadong Wang, Yuanzhi Cheng, Shinichi Tamura, Accurate pelvis and femur segmentation in hip ct with a novel patch-based refinement, IEEE J. Biomed. Health 23 (2019) 1192–1204.

[2] Jinke Wang, Yuanzhi Cheng, Changyong Guo, Yadong Wang, Shinichi Tamura, Shape-intensity prior level set: combining probabilistic atlas and probability map constrains for automatic liver segmentation from abdominal CT images, IJCARS 11 (2016) 817–826.

[3] Hai Yang, Hai-Jun Huang, The multi-class, multi-criteria traffic network equilibrium and systems optimum problem, Transp. Res., Part B, Methodol. 38 (1) (2004) 1–15.

[4] G. Palla, I. Derényi, I. Farkas, T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature 435 (7043) (2005) 814.

[5] R.F. Potthoff, S. Roy, A generalized multivariate analysis of variance model useful especially for growth curve problems, Biometrika 51 (3–4) (1964) 313–326.

[6] M.E. Newman, Modularity and community structure in networks, Proc. Natl. Acad. Sci. 103 (23) (2006) 8577–8582.

[7] L. Tang, H. Liu, J. Zhang, Z. Nazeri, Community evolution in dynamic multi-mode networks, in: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2008, pp. 677–685.

[8] J. Abello, M.G. Resende, S. Sudarsky, Massive quasi-clique detection, in: Latin American Symposium on Theoretical Informatics, Springer, 2002, pp. 598–612.

[9] E.A. Leicht, P. Holme, M.E. Newman, Vertex similarity in networks, Phys. Rev. E 73 (2) (2006) 026120.

[10] P.D. Hoff, A.E. Raftery, M.S. Handcock, Latent space approaches to social network analysis, J. Am. Stat. Assoc. 97 (460) (2002) 1090–1098.

[11] A. Decelle, F. Krzakala, C. Moore, L. Zdeborová, Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications, Phys. Rev. E 84 (6) (2011) 066106.

[12] U. Von Luxburg, A tutorial on spectral clustering, Stat. Comput. 17 (4) (2007) 395–416.

[13] X.-S. Zhang, R.-S. Wang, Y. Wang, J. Wang, Y. Qiu, L. Wang, L. Chen, Modularity optimization in community detection of complex networks, Europhys. Lett. 87 (3) (2009) 38002.

[14] M.E. Newman, M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E 69 (2) (2004) 026113.

[15] V.D. Blondel, J.-L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech. Theory Exp. 2008 (10) (2008) P10008.

[16] U.N. Raghavan, R. Albert, S. Kumara, Near linear time algorithm to detect community structures in large-scale networks, Phys. Rev. E 76 (3) (2007) 036106.

[17] M.J. Barber, J.W. Clark, Detecting network communities by propagating labels under constraints, Phys. Rev. E 80 (2) (2009) 026129.

[18] X. Zhang, Y. Li, S. Jiang, B. Xie, X. Li, Q. Zhang, M. Lu, Efficient community detection based on label propagation with belonging coefficient and edge probability, in: Chinese National Conference on Social Media Processing, Springer, 2016, pp. 54–72.

[19] N. Chen, Y. Liu, H. Chen, J. Cheng, Detecting communities in social networks using label propagation with information entropy, Phys. A, Stat. Mech. Appl. 471 (2017) 788–798.

[20] A. Berti, An improved node ranking for label propagation and modularity based clustering, preprint, arXiv:1409.7878.

[21] J.J. Whang, D.F. Gleich, I.S. Dhillon, Overlapping community detection using seed set expansion, in: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, ACM, 2013, pp. 2099–2108.

[22] X. Liu, T. Murata, Advanced modularity-specialized label propagation algorithm for detecting communities in networks, Phys. A, Stat. Mech. Appl. 389 (7) (2010) 1493–1500.

[23] D. Von Rosen, The growth curve model: a review, Commun. Stat., Theory Methods 20 (9) (1991) 2791–2822.
[24] S. Fortunato, M. Barthelemy, Resolution limit in community detection, Proc. Natl. Acad. Sci. USA 104 (1) (2007) 36–41.
[25] L. Subelj, M. Bajec, Unfolding communities in large complex networks: combining defensive and offensive label propagation for core extraction, Phys. Rev. E 83 (3) (2011) 036103.
[26] L. Danon, A. Diazguilera, J. Duch, A. Arenas, Comparing community structure identification, J. Stat. Mech. Theory Exp. 2005 (09) (2005) 09008.
[27] G.K. Orman, V. Labatut, H. Cherifi, Qualitative comparison of community detection algorithms, digital information and communication technology and its applications 167 (2011) 265–279.
[28] M.E.J. Newman, Fast algorithm for detecting community structure in networks, Phys. Rev. E 69 (6) (2004) 066133.
[29] Changfa Shi, Yuanzhi Cheng, Jinke Wang, Yadong Wang, Kensaku Mori, Shinichi Tamura, Low-rank and sparse decomposition based shape model and probabilistic atlas for automatic pathological organ segmentation, Med. Image Anal. 38 (2017) 30–49.
[30] Changfa Shi, Yuanzhi Cheng, Fei Liu, Yadong Wang, Jing Bai, Shinichi Tamura, A hierarchical local region-based sparse shape composition for liver segmentation in CT scans, Pattern Recognit. 50 (2016) 88–106.