2013 IEEE Congress on Evolutionary Computation
June 20-23, Cancún, México

MLPA: Detecting Overlapping Communities by Multi-Label Propagation Approach

Qiguo Dai, Maozu Guo*, Yang Liu, Xiaoyan Liu, Ling Chen
School of Computer Science and Technology
Harbin Institute of Technology, HIT
Harbin, P.R. China
maozuguo@hit.edu.cn

Abstract—**The identification of communities is an important step in understanding of the complex network. Comparative studies suggest that the development of accurate and efficient methods to infer the communities is still in its early stages. Label propagation algorithm (LPA) that detects communities by propagating labels among vertices, attracts a great deal of attention recently. However, the communities detected by most LPAs are disjointed. Due to communities are often overlapping in real world networks, we show a multi-label propagation algorithm (MLPA) to detect overlapping communities. The inspiration is that the more people are familiar, the more they trust each other. To simulate the confidence of human communication, propagating intensity (PI) is defined to describe the confidence extent of the label propagated by neighboring vertices. The PI is then used to guide the propagation, with the purpose to make the detection more accurate. The results of extensive experiments both on synthetic and real networks show that the proposed MLPA outperforms many other methods. The effectiveness of MLPA can be attributed to its multi-label propagating strategy.**

Keywords- overlapping community; multi-label propagation; label propagation; complex network

I. INTRODUCTION

Community is a major common structure in most real complex networks, which is usually considered to be a dense group of vertices that are similar to each other and dissimilar from others. Community structure may play a certain corresponding role or specific function in real system. Due to its significant meaning, detecting communities in network has attracted much interest and a great of methods have been proposed in the past decade. Many of them [1] identify disjointed communities, in which each vertex belongs to one community at most.

In real network, however, communities are often overlapping. Vertices may belong to multiple communities, e.g. a person usually associates to several social groups such as the family; an interdisciplinary researcher may be active in several scientific areas; a protein often participates in several complexes in the biological system. Identification of overlapping communities in real network is currently a challenge problem, because there is not a precise formula of the overlapping community. In recent years, many detecting methods have been put forward, which can be generally categorized as: clique percolation method (CPM) [2], link-partition method (Link) [3], optimization-based expansion (OE) [4] etc. Although much progress has been made, many of them have some limitations, e.g. CPM depends on the distribution of clique in the network; Link relies on partition density D; and OE depends on seed position and fitness function definition. Moreover, as the networks in the real world become larger, even with millions of vertices, the algorithms of community detection must have a low time complexity.

Label propagation algorithm (LPA), proposed by Raghavan in 2007 [5], is currently one of the fastest algorithms for community detection, with a near-linear time complexity on sparse networks. In LPA, each vertex holds a label (community identifier) indicating the community it belongs to. The algorithm identifies communities by iteratively propagating label among neighboring vertices. The fast speed of LPA can be attributed to the mechanism of local propagation that uses the local topological structure alone as its guide, independent of any optimization with respect to a predefined function or any prior knowledge about the communities. Due to the excellent performance, the LPA has attracted a great deal of attention recently. Many improvement and theory analysis of LPA have been made. Lovro et al. [6] present an advanced LPA that combines two strategies of defensive preservation and offensive expansion of communities, to extract the core of the network and to identify whisker communities. Tibely and Kertesz [7] have proven that LPA is equivalent to a Potts model. Barber and Clark [8] have refined the approach into a modularity optimization algorithm. Liu and Murata [9] have further combined the modularity optimization version of LPA with a greedy agglomeration.

However, the communities detected by many label propagation algorithms are disjointed. The reason is that LPA is originally designed for disjointed communities, in which each vertex is assigned only one label indicating the community that it belongs to. To detect overlapping communities, some extensions of LPA have been proposed by allowing vertex possess multiple labels. Gregory proposed an improved version of LPA to detect overlapping communities, namely COPRA [10]. In COPRA, each vertex iteratively updates its labels as well as their belonging coefficients with top v labels in all its neighbors. Inspired by human speaker-listener process, Xie et al. propose SLPA [11], which provides each vertex with a memory to save all the labels it received in the history. Although they have successfully adapted LPA to the detection of overlapping community, the detecting accuracy of these extensions is not satisfied. The reason may be that the propagation procedures of them do not consider the heterogeneous acquaintances of neighbors, which may cause the detecting accuracy loss.

In this paper, we show a multi-label propagation algorithm (MLPA) inspired from the confidence of human communication, to detect overlapping communities in complex networks. In MLPA, the propagation is guided by the propagating intensity (PI), in order to simulate the information communication among people. The PI is defined to describe the confidence of the label propagated by neighbors, which is on the basis of the closeness between neighboring vertices. Based on PI, the novel procedures of propagating and updating multi-label are proposed to make the propagation more effective, and thus to make the detecting more accurately.

A series of extensive experiments on synthetic and real networks are conducted to test the effectiveness of the MLPA. In the experiments on synthetic networks, the results show that the proposed MLPA algorithm can get a more accurate detecting performance on overlapping communities, especially on overlapping vertices of different communities. The carefully comparisons have been made between MLPA and other state-of-art algorithms, and the results suggest that the quality of communities detected by MLPA superiors to those of others. The remainder of this paper is organized as follows. Section 2 illustrates the basic idea of label propagation. Section 3 describes the proposed MLPA algorithm. The experimental results are shown and discussed in Section 4. Finally, Section 5 presents some concluding remarks.

II. Related Works

Label propagation algorithm (LPA) is a self-adaptive iterative approach to detect communities in networks. In the original version of LPA, each vertex takes a single label denoting the community it may belong to. At the beginning, each vertex is initialized with a distinct label. In the process of iteration, all vertices update their labels using the most popular one in its neighbors, as:

$$l_v^{'} = \arg\max_{l} \sum_{u\in\sigma(v)} \delta(l_u, l)$$

where l is a label, l_u is the label of vertex u, $l_v^{'}$ is new label of vertex v, $\sigma(v)$ is the set of all the vertices neighboring to v, and

$$\delta(l_u, l) = \begin{cases} 1, & if\ l_u = l \\ 0, & \text{otherwise} \end{cases}$$

After the iteration terminates, vertices in a common community will be assigned with the same label. LPA does not require any prior information or assumption about communities. Moreover, it has near-linear complexity over edge number of the network. However, as many other classic community detection methods, LPA cannot detects overlapping communities, due to each vertex just keeps only one label.

A feasible way to extend LPA for overlapping community detection is that vertices are allowed to possess multiple labels. Thence, vertex with multiple labels are overlapping vertex of different communities, denoting that it belongs to more than one community. Following this idea, LPA is recently improved proposed to detect overlapping communities, such as COPRA, SLPA. In the propagation of COPRA [10], each vertex first summarizes and normalizes labels of all its neighbors and deletes the labels whose coefficient lower than threshold *1/v*, where *v* is the parameter of the algorithm, controlling the maximum of communities a vertex can belong. Xie et al. proposed the SLPA [11] that is inspired from human speak-listen process. It substitutes belonging coefficient of labels with the received frequency during propagation history. However, because each vertex just considers the label information of its neighbors when updates its own labels, the detecting accuracy of above methods for overlapping communities is still not satisfactory.

III. MLPA: Multiple-Label Propagation Algorithm

In this section, a multi-label propagation algorithm (MLPA) is described to detect by simulating the confidence in human communication. The motivation is that the communication between people depends on their relationship significantly and the more familiar they are, the more they trust each other.

In light of this, propagating intensity (PI) is defined here to describe the confidence extent of the received label when propagating, in order to guide the multi-label propagation in MLPA, and thus to enhance the propagation efficiency. The PI combines two aspects of information. In addition to the belonging coefficient of the propagated label, the vertex similarity is also considered in PI. The vertex similarity, which has been widely employed as local structural information in network mining, can measure the closeness of two neighboring vertices, with the assumption that two vertices are more familiar if they have more common neighbors. To make effective use of PI, multi-label propagating process and MLPA algorithm is also proposed.

In MLPA, each vertex in a network can take multiple labels, not single one as in classic LPA. And each vertex plays the role of receiver according to a specific propagating order. Given vertex *r* as a receiver, all its neighbors are the transmitters of *r*. They propagate labels to the receiver *r* with different intensities of PI, and then r updates its own label memory with the received labels with high PI. The details of MLPA including the definition of label memory of vertices, the procedures of label-propagating and label-receiving, are as follows.

A. Label Memory of Vertices

At the beginning, we first illustrate the label memory of vertex saving multiple labels. A memory L_i is distributed for each vertex, which consists of a set of pair (l_{ij}, c_{ij}) as in

$$L_i = \{(l_{i1}, c_{i1}), (l_{i2}, c_{i2}), \ldots, (l_{ij}, c_{ij}), \ldots, (l_{ih}, c_{ih})\} \ \sum_{j=1}^{h} c_{ij} = 1 \quad (1)$$

where l_{ij} is a label (community identifier), c_{ij} is the membership strength of l_{ij}, and h is the size of memory indicating the number of communities the vertex belonging to.

B. Label-Propagating Based on PI

At first, transmitter t randomly selects a pair in L_t with the probability of membership strength, denoting by (l_t, c_t). Then, t propagates the selected identifier l_t to receiver r with intensity of PI_{tr}. To facilitate, we use (l_t, PI_{tr}) to denote the information propagated from vertex t to r. The propagating intensity is defined as:

Definition 1 (Propagating Intensity) Given the selected pair (l_t, c_t) of vertex t, the propagating intensity (PI) of label l_t from vertex t to r is

$$PI_{tr} = \sqrt{S_{tr} \cdot c_t} \quad (2)$$

where S_{tr} the vertex similarity between t and r. For S_{tr}, here we use a cosine similarity as

$$S_{tr} = \frac{|\Gamma(t) \cap \Gamma(r)|}{\sqrt{|\Gamma(t)| \cdot |\Gamma(r)|}}$$

where $\Gamma(t)$ is vertex set including t and all its neighbors. The cosine similarity is widely used in the field of complex network to indicate the clossness of topological structure between nodes in the network.

We can see that the propagating intensity is the geometric mean of local closseness information (S_{tr}) and membership strength (c_t) of the propagated label in L_t, which can make a good balance between them.

C. Label-Receiving Step

Receiver r selects the representative label pairs from them it received and updates its own memory with representative pairs. The detail of the receiving procedure is as follows:

1) Vertex r receives the pairs propagated by all its neighbors:

$$L'_r = \{(l_1, PI_{1r}), (l_2, PI_{2r}), \ldots, (l_j, PI_{jr}), \ldots, (l_k, PI_{kr})\} \quad (3)$$

where l_j and PI_{jr} are the label and its intensity propagated by neighbor j, and k is the number of neighbors of receiver r.

2) Merge pairs with same label in L_r' and sum up their intensity, then,

$$L'_r = \{(l_1, c_1), (l_2, c_2), \ldots, (l_j, c_j), \ldots, (l_{k'}, c_{k'})\} \quad (4)$$

where k' is the number of labels remained.

3) To keep the representative pairs, delete pairs in which intensity is lower than a threshold of *p-value*, as:

$$p\text{-}value = c_{\max} \cdot p \quad (5)$$

where c_{max} is the max value in $\{c_1, \ldots, c_{k'}\}$, and p is the parameter of MLPA algorithm, determining the number of memberships of receiver r. Note that, MLPA can be used to detect disjointed communities, when p=1.

4) Update the memory of receiver r, $L_r \leftarrow L'_r$.

D. MLPA Algorithm

We have described the main procedures of MLPA, including propagating and memory updating. Based on them, pseudo code of MLPA is shown in Algorithm 1.

At the beginning of MLPA, memories of all vertices are initialized by (lb_i,1.0), where lb_i is the name of vertex i. The main body of MLPA is an iteration, which consists of propagating and memory-updating steps. In each iteration, every vertex in the network is as a receiver in turn according to shuffled order. When vertex r serves a receiver, all neighbors of r are its corresponding transmitters. Each transmitter t propagates a label to r using the *propagating method* in (B). Then receiver r updates its memory by the method of *receiving method* in (C). When the *termination criterion* in (E) of iteration is met, a cover C_{ov} of overlapping communities is obtained by carrying out a *post-processing* in (F). It can be found that the presented MLPA inherits the basic self-adaptive propagation process of other LPAs. The main difference is that propagations are guided by the defined propagating intensity, which captures the heterogeneous neighborhood by exploiting the closeness between vertices.

E. Termination

MLPA is an iterative algorithm and it is important to determine when the algorithm is convergent. To this end, an adaptive termination is defined for MLPA here.

Initially, the total number of labels in all vertex memories is same as the number of vertices. It will decrease during iteration and get the minimum when MLPA is approximately convergent. Therefore, it might be reasonable to assume that the number of labels is strongly related to the convergence. Based on this, MLPA terminates the iteration when the total number of labels is unchanged from the previous iteration, i.e. $i_t = i_{t-1}$, where i_t is the number of labels in iteration t. The testing result of this termination is reported in following experiments section.

F. Post-Processing

After the iteration terminates, a cover of overlapping communities can be obtained based on the memories of all vertices, which save underlying information of community structure within the input network.

Specifically, each pair in memory L_t indicates a membership of the vertex *t*, and the memory size is the maximum of communities the vertex may belong to. Each vertex is allocated to one or multiple communities according to labels saved in its memory. By means of this, an elementary cover of communities C_{ov} is obtained. As other LPAs, optional refining steps are employed to refine C_{ov}, which include removing the communities that are a subset of others and splitting the disconnected communities into connected one. Then the final cover C_{ov} of overlapping communities on the network is detected.

G. Complexity

MLPA consists of three stages: initialization, iteration and post-processing.

- At initialization, similarity-computing is $O(N\cdot k^2)$ and memory initialization is $O(N)$, where N is the number of vertices in network and k is the average degree of vertices. Therefore, the complexity of initialization step is $O(N\cdot k^2)$.

```
Algorithm 1 MLPA (p)
Input: Network net
Output: C_ov
1. Initialization:
   Vertices[] = getVertex(net);
   Sim = getSimilarity (net);
   for each vertex i
       L[i] ← (lb_i,1.0);
   end for
2. Iteration:
   while (!Termination)
       Vertices.shuffleOrder();
       for each r
           Receiver ← Vertices [r];
           Transmitters[] ←Vertices[r].Neighbors();
           L'[r] ← Φ;
           for each t
               (l_t,PI_tr) ←Transmitters[t].label-propagating();
               L'[r].add(l_t, PI_tr);
           end for
           L[t] ← label-receiving (L'[t], p);
       end for
   end while
3. Post-processing:
   C_ov ← Post-Processing(L);
```

- In each iteration, all N vertices are traversed once. For each vertex v, propagations from all its neighbors to v take $O(k\cdot h)$, where h is the average size of memory of vertices; and memory updating of vertex v is $O(h)$. Thence, the total complexity of iteration part of MLPA is $O(T\cdot N\cdot k\cdot h)$, in which T is the iteration numbers.
- In the post-process step, obtaining of overlapping community need $O(N\cdot h)$.

Note that, average membership h is far less than vertex number N in most real network ($h \ll N$); and MLPA usually terminates after only few iterations on most networks ($T \ll N$).

Therefore, the complexity of the MLPA algorithm is approximate $O(N\cdot k^2)$, which is similar to the original LPA. It is near linear to edges number $N\cdot k$ in the network, because most real networks are sparse that ($k \ll N$).

IV. EXPERIMENTS

To study the performance of MLPA on detecting overlapping community, extensive experiments are conducted on both synthetic and real networks. We test its detecting accuracies on synthetic networks with known communities as well as known overlapping vertices, and also examine its effectiveness on real networks. Note that, due to random, the results of MLPA in the following are average values of 30 times running. And the parameter p is the one (0,1) with an interval of 0.1 that get the best result of detecting. The following experiments is conduct in the environment of HP servers with Linux operating system.

A. Synthetic Benchmark

We use the synthetic networks generated by LFR benchmarks [12] to test MLPA. LFR is widely used in testing overlapping community detection methods, because networks generated by LFR have some key properties in common with real network, such as power-law distributions on vertex degree and community size.

The parameters of LFR include number of vertices, N; average degree, k; maximum degree, *maxk*; mixing parameter μ; the minus exponents for power-law distribution of vertex degree and community size, τ_1, τ_2; minimum and maximum community size, *minc, maxc*; number of overlapping nodes, *on*; number of membership of overlapping nodes, *om*. Unless otherwise stated, default parameters of LFR we used are: $N = 5000$, $k = 10$, $maxk = 3\cdot k$, $minc = 10$, $maxc = 50$, $\tau_1 = 2$, $\tau_2 = 1$, $\mu = 0.3$, $on = 500$, $om = 5$.

In experiments on synthetic networks, extended normalized mutual information (NMI) and F-score are employed to evaluate the detecting accuracies of overlapping communities and overlapping nodes separately.

NMI is widely used for overlapping community detection, which is proposed by Lancichinetti [13], and the value of NMI is between 0 and 1. For a detected cover of overlapping communities, the value of NMI closer to 1 indicates a better matching to the real cover.

Identification of overlapping nodes is also very important to overlapping community detection. Xie *et al.* define the identification as a binary-classification problem and use F-score to measure its accuracy [11]. F-score is widely used in data-mining and machine learning field as following:

$$F\text{-}score = \frac{2 \cdot recall \cdot precision}{recall + precision}$$

where precision is the number of overlapping nodes detected correctly (on^d) divided by the total number of detected overlapping vertex; recall is on^d divided by the number of real overlapping nodes (on) in networks.

B. Properties of MLPA

We first study some properties of MLPA on networks generated by LFR, including time consuming, termination criteria, as well as the effectiveness of propagating intensity (*PI*). Fig. 1 depicts time-consuming of MLPA on synthetic networks, in which edge number (E) ranges from 0 to 1,000,000 (N ranges from 0 to 100,000 and average degree is 10). The results shown in Fig. 1 confirm that the execution times scale near linear to edge numbers. It is consistent with the complexity analysis in the previous section.

As aforementioned, *PI* incorporates both structural similarity between vertices and belonging coefficient of propagated label. To verify its effectiveness, we compare MLPA using PI, with MLPA1 and MLPA2 which separately use PI^1, PI^2 as: $PI^1_{tr} = S_{tr}, PI^2_{tr} = c_t$, where S_{tr} and c_t are same as in (2). PI^1 and PI^2 mean that propagations between vertices are weighted just by structural similarity and belonging coefficient of propagated label separately. The result shown in Fig. 2, demonstrate that MLPA using PI, get the best detecting accuracy, both on community and vertex levels.

Fig. 3 illustrates the convergence process of MLPA (without termination check) on synthetic networks with om between 2 to 8. Fig. 3 (a) and Fig. 3 (b) show NMI and F-score values of detected covers on the networks in each iteration. Solid circles on each curve mark the point when the defined termination is met. Although there are minor oscillations, it can be found that MLPA can approximately convergent after several iterations both with respect to NMI and F-score. The defined termination can always be met soon after converge. It means that the proposed termination for MLPA is effective, at least in this experiment.

C. Comparison with Other Methods on Sythettic Networks

We compare our algorithm with SLPA, COPRA, CFinder, Link and OSLOM. SLPA and COPRA are two versions of MLPA; CFinder [14] is popular software based on the CPM method [2] and Link is a link-communities based algorithm; OSLOM is a recently proposed local optimization method. Table 1 lists parameters settings (ranges), and resources of these methods. Taking into account random factor, the result of COPRA and SLPA are also average values of 10 times of running.

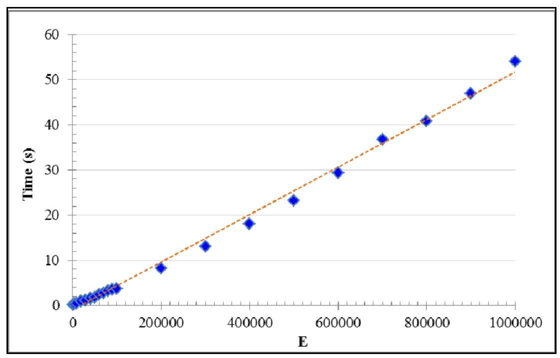
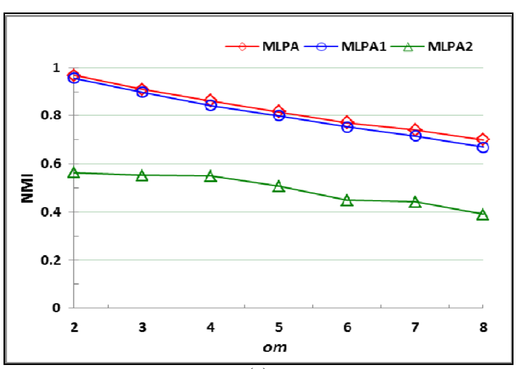
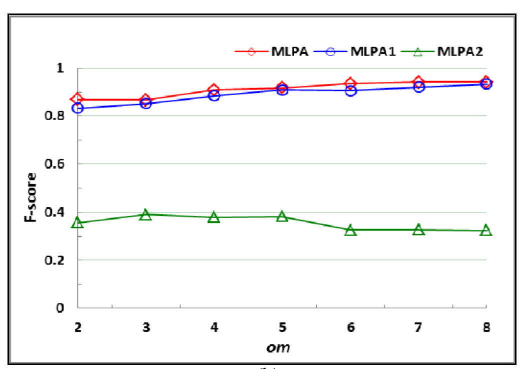


Figure 1. Near-linear execution time of MLPA on synthetic networks, with k=10 and N ranging from 0 to 100000

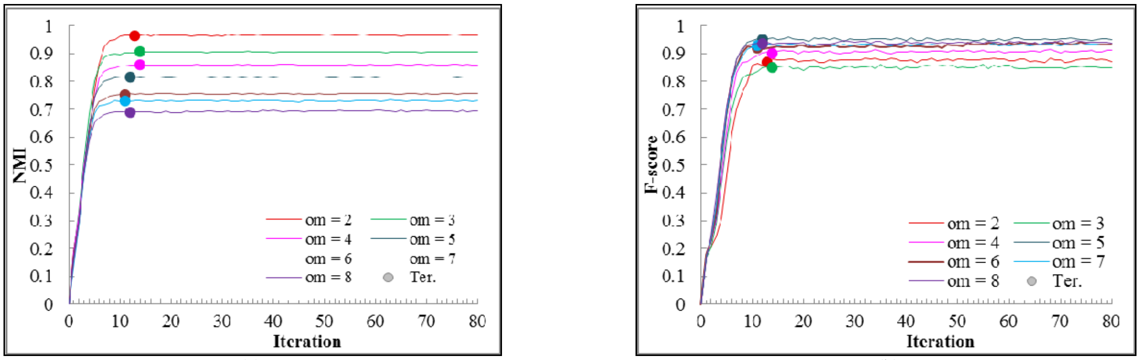


(a)



(b)

Figure 2. Comparisons of MLPA with different *PI* on synthetic networks, with *om* ranging from 2 to 8



(a) (b)

Figure 3. Convergence and termination point of MLPA on synthetic networks, with *om* between 2 and 8

TABLE I. PARAMETERS SETTING AND SOURCES OF TESTED METHODS

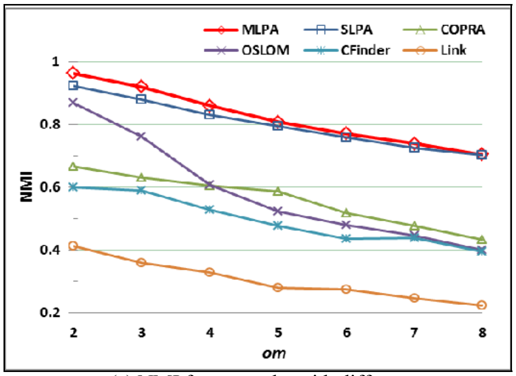
Methods	Parameters	Value (Range)	Reference	Source
SLPA	*r* (Threshold)	(0,0.45] (Interval: 0.05)	[11]	https://sites.google.com/site/communitydetectionslpa/ (C++ version 1.3)
	T (number of iterations)	100		
COPRA	*v* (maximum of membership)	(0,10]	[10]	http://www.cs.bris.ac.uk/~steve/networks/software/copra.html (Version 1.25)
CFinder	*k* (size of k-clique)	[3,9]	[14]	http://www.cfinder.org/ (Version 2.0.5)
Link	□	□	[3]	https://github.com/bagrow/linkcomm (Python version)
OSLOM	□	□	[4]	http://www.oslom.org/ (Version 2.4)

The results of comparisons are shown in Fig. 4. We test them on networks with various membership of overlapping nodes (*om*), mixing parameters (μ) and number of overlapping nodes (*on*) separately. The data shown in Fig. 4 are of the covers detected by methods using the best parameter in term of community level (NMI).

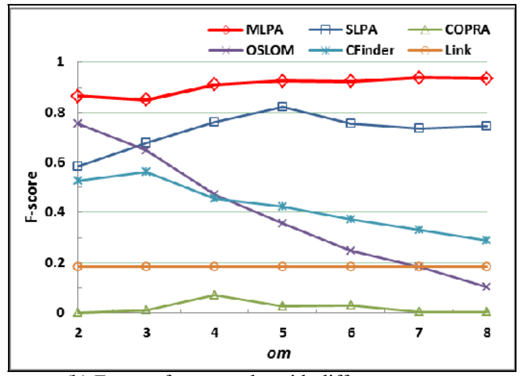
In Fig. 4, (a) and (b) show NMI and F-score values of detected covers as functions of the number of memberships (*om*) ranging from 2 to 8; (c) and (d) show NMI and F-score values of detected covers as functions of the mixture parameter (μ) ranging in [0,0.5); (e) and (f) show NMI and F-score values of detected covers related to proportion of overlapping vertices (*on/N*), [0,0.5). From these results, it can be found that the proposed MLPA not only exhibits favorable detecting performance on overlapping communities, but also can find the overlapping nodes with higher accuracy.

TABLE II. COMPARISONS OF MLPA WITH OTHER METHODS ON Q_{ov} IN REAL NETWORKS

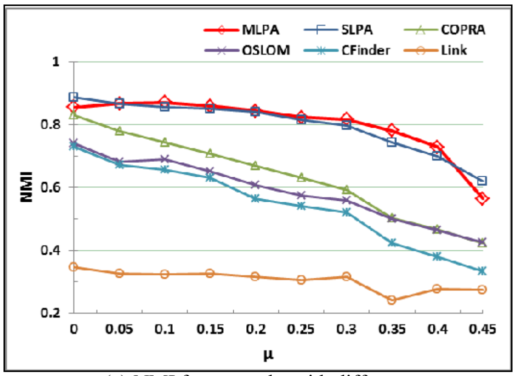
Network	Reference	*N*	*E*	MLPA		SLPA		COPRA		OSLOM	CFinder	Link
				Q_{ov}	*std.*	Q_{ov}	*std.*	Q_{ov}	*std.*	Q_{ov}	Q_{ov}	Q_{ov}
Karate	[16]	34	78	**0.744**	0.004	0.660	0.210	0.478	0.080	0.723	0.515	0.108
Dolphins	[17]	62	159	**0.773**	0.010	0.765	0.022	0.687	0.008	0.729	0.662	0.042
Lesmis	[18]	77	254	**0.787**	0.000	0.782	0.016	0.716	0.047	0.740	0.634	0.223
Football	[19]	115	613	0.702	0.016	**0.704**	0.013	0.698	0.031	0.665	0.641	0.000
Polbooks	[6]	105	441	**0.840**	0.006	0.831	0.023	0.835	0.011	0.839	0.786	0.014
Power	[20]	4941	6594	**0.798**	0.009	0.660	0.006	0.628	0.013	0.501	0.152	0.097
Co-author.	[21]	16662	22446	**0.769**	0.002	0.727	0.001	0.651	0.025	0.374	0.273	0.187



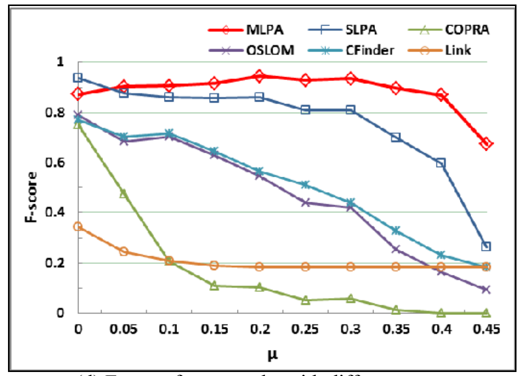
(a) NMI for networks with different *om*



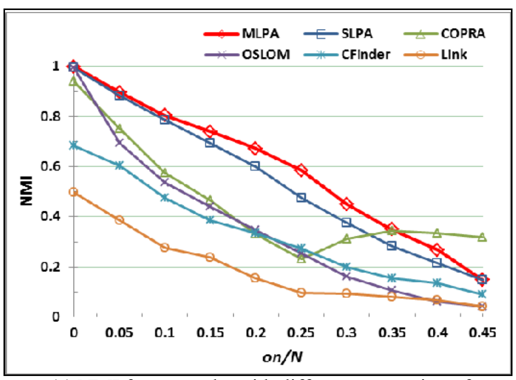
(b) F-score for networks with different *om*



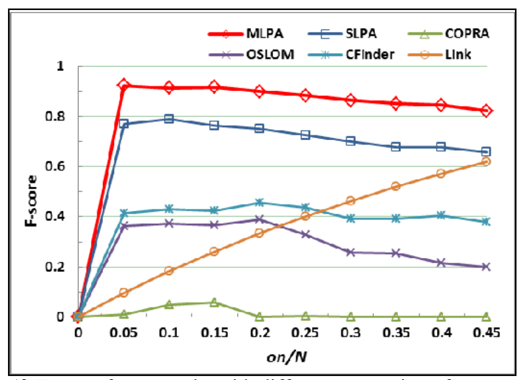
(c) NMI for networks with different μ



(d) F-score for networks with different μ



(e) NMI for networks with different proportion of *on*



(f) F-score for networks with different proportion of *on*

Figure 4. Comparisons of MLPA with other methods on synthetic networks generated by LFR

V. CONCLUSION

In this paper, we present a multi-label propagation algorithm (MLPA) to detect overlapping communities in networks. The MLPA is inspired from the confidence of communication in human society. In MLPA, propagating intensity (PI), which combines both the closeness of vertices and information of propagated label, is proposed to guide the propagation. Experimental results on synthetic and real networks show that the MLPA outperforms many other methods, including existing label propagation algorithms for the detection of overlapping communities. It confirms that the introduction of the closeness between vertices into label propagation can significantly improve the detecting performance on overlapping communities.

Nonetheless, the proposed method is not perfect. Its main current drawback is that the detected communities depends on the parameter of algorithm. And the determination of the value of parameter is a complicated problem. In the future, we would like to give a theortical method to determine the parameter.

ACKNOWLEDGMENT

The work was supported by the Natural Science Foundation of China under Grant No. 60932008, 61172098 and 61271346, and the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant No. 20112302110040.

REFERENCES

[1] M. E. J. Newman, “Communities, modules and large-scale structure in networks,” Nature physics, vol. 8, Jan. 2012, pp. 25-31.

[2] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, “Uncovering the overlapping community structure of complex networks in nature and society,” Nature, vol. 435, Jun. 2005, pp. 814-818.

[3] Y. Ahn, J. Bagrow, and S. Lehmann, “Link communities reveal multiscale complexity in networks,” Nature, vol. 466, Aug. 2010, pp. 761–764.

[4] A. Lancichinetti, F. Radicchi, J. Ramasco, and S. Fortunato, “Finding statistically significant communities in networks,” Plos one, vol. 6, Apr. 2011, p. e18961.

[5] U. Raghavan, R. Albert, and S. Kumara, “Near linear time algorithm to detect community structures in large-scale networks,” Physical Review E, vol. 76, Sep. 2007, p. 036106.

[6] L. Subelj and M. Bajec, "Unfolding communities in large complex networks: Combining defensive and offensive label propagation for core extracxtion," Physical Review E, vol. 83, 2011, pp. 036106.

[7] G. Tibely and J. Kertesz, "On the equivalence of the label propagation method of community detection and a Potts model approach," Physica A, vol. 387, 2008, pp. 4982-4984.

[8] M. Barber and J. Clark, "Detecting network communities by propagating labels under constraints," Physcial Review E, vol. 80, 2009, pp. 026129.

[9] X. Liu and T. Murata, "Advanced modularity-specialized label propagation algorithm for detecting communities in networks", Physical A, vol. 389, 2010, pp, 1493-1500.

[10] G. Steven, “Finding overlapping communities in networks by label propagation,” New Journal of Physics, vol. 12, Oct. 2010, p. 103018.

[11] J. Xie, B. Szymanski, and X. Liu, “SLPA: uncovering overlapping Ccommunities in social networks via a speaker-listener interaction dynamic process,” Proc. IEEE 11th International Conference on Data Mining Workshops (ICDMW), IEEE Press, Dec. 2011, pp. 344 - 349.

[12] A. Lancichinetti, S. Fortunato, and F. Radicchi, “Benchmark graphs for testing community detection algorithms”, Physical Review E, vol. 78, Oct. 2008, p. 046110.

[13] A. Lancichinetti, A. Fortunato, and J. Kertesz, “Detecting the overlapping and hierarchical community structure of complex networks,” New Journal of Physics, vol. 11, Mar. 2009, p. 033015.

[14] B. Adamcsek, G. Palla, I. Farkas, I. Derenyi and T. Vicsek, “CFinder: locating cliques and overlapping modules in biological networks,” Bioinformatics, vol. 12, Feb. 2006, pp. 1021-1023.

[15] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri, “Extending the definition of modularity to directed graphs with overlapping communities,” Journal of Statistical Mechanics: Theory and Experiment, vol. 2009, Mar. 2009, p. 03024.

[16] W. Zachary, “An information flow model for conflict and fission in small groups,” Journal of Anthropological Research, vol. 33, 1977, pp. 452-473.

[17] D. Lusseau, K. Schneider, O. Boisseau, “The bottlenose dolphin community of doubtful sound features a large proportion of long – lasting associations – Can geographic isolation explain this unique trait?” Behavioral Ecology and Sociobiology, vol. 54, 2003, pp. 396-405.

[18] D. E. Knuth, The Stanford GraphBase: A Platform for Combinatorial Computing, Addison-Wesley, Reading, MA, 1993

[19] M. Girvan, M. Newman, “Community structure in social and biological networks, ” Proceeding of the National Academy of sciences of the U. S. A., vol. 99, Dec. 2002, pp. 7821-7826.

[20] D. Watts and S. Strogatz, “Collective dynamics of 'small-world' networks,” Nature, vol. 393, Jun. 1998, pp. 440-442.

[21] M. Newman, “The structure of science collaboration networks,” Proceedings of the National Academy of Sciences of the U. S. A., vol. 98, Jan. 2001, pp. 404-409.