ICIC Express Letters
Volume **10**, Number **12**, December **2016**
 ISSN 1881-803X
pp. **2995–3002**

RESEARCH ON COMMUNITY DETECTION ALGORITHM BASED ON MODULARITY MAXIMUM

Jing Chen[1,2] and Yun Wan[1]

[1]College of Information Science and Engineering
Yanshan University
[2]Key Laboratory for Computer Virtual Technology and System Integration of Hebei Province
No. 438, Hebei Avenue, Qinhuangdao 066004, P. R. China
{ xychenjing; wanyun }@ysu.edu.cn

Received July 2016; accepted October 2016

Abstract. *It is well known that the drawbacks of label propagation algorithm are high randomness, weak robustness and easy to form monster community. Many improved methods are proposed constantly in order to avoid these problems. However, the related label propagation algorithm is seldom concerned with the stability of community structure and the results of community detection. In order to obtain the high value communities, modularity and community structure are involved in detecting communities. In this paper, the algorithm CDMMCS (Community Detection based on Modularity Maximum and Community Structure) is proposed. The main idea is based on label propagation, but CDMMCS begins with some small communities, and the process of label updating is in consideration of the modularity maximum. When the label stops propagating, the communities with the same label are merged. Check the community structure of merged communities, and perform different operations according to different community structures. Experiments show that CDMMCS successfully detects communities with higher modularity values than LPA (Label Propagation Algorithm) and LPAm+ in some real-world networks. And the running time of CDMMCS is less than LPAm+.*
Keywords: Community detection, Label propagation algorithm, Modularity maximum, Community structure

1. **Introduction.** Community detection in networks has caused a great deal of attention recently. Various methods are proposed to detect communities, and among these algorithms the label propagation algorithm (LPA) proposed by Raghavan attracts a lot of attention with the advantages of its practicability, implementation and its near-liner time complexity. And the label propagation algorithm is widely used in many fields, such as text information classification, information retrieval in multimedia, and community detection [1]. Certainly, LPA also has many drawbacks. The high randomness and vibration in bipartite network are the typical problems which result from the randomness of label updating order. Many improved algorithms are proposed to solve these problems. However, many algorithms settle these problems at the expense of time. In addition, community structure is also a hot topic in network treated as one of the important attributes in network. Finding community structures in networks is another step towards understanding the complex networks. It is also helpful to get stable communities. Therefore, community structure is also an important factor in research of community detection.

The organization of the study is summarized as follows. In Section 2, the related research works and the goal of this paper are introduced. In Section 3, the relative parameters are defined, the implementation steps of the algorithm are described in detail, and time complexity of CDMMCS algorithm is analyzed. In Section 4, in different social networks, CDMMCS algorithm is verified and compared. In Section 5, the whole paper is summarized, and the future research work is pointed out.

2. **Related Works.** In this paper, two aspects of label propagation and community structure which affects community detection are discussed in the following.

Raghavan et al. proposed label propagation algorithm (LPA), and assigned a unique label for every node at the beginning of algorithm [2]. The advantages of this algorithm are its simplicity and time efficiency; however, some drawbacks, such as the formation of a monster community, weak robustness, and high randomness, remain in LPA [3]. [4] proposed an algorithm (COPRA) for finding overlapping community structure in very large networks. The contribution of this algorithm is to extend the label and propagation step to include information about more than one community. [5] proposed a modularity-specialized label propagation algorithm (LPAm) for detecting network communities, and it is succeeded by the raise of advanced modularity-specialized label propagation algorithm (LPAm+) [6]. The latter is proposed in order to escape local maxima combining multistep greed agglomerative algorithm (MSG) and LPAm. Ugander and Backstrom introduce an efficient algorithm, balanced label propagation, for precisely partitioning massive graphs while greedily maximizing edge locality, the number of edges that are assigned to the same shared of a partition [7]. [8] proposed a community detection method (CK-LPA) based on the label propagation algorithm with community kernel. The main idea is to assign a corresponding weight to each node based on node importance in the whole network. Then the node label is updated in sequence according to the weight.

The research of community structures in networks is an important issue in many fields. Girvan and Newman proposed a new method for detecting community structure [9]. However, the definition of community structure taken from relative research is mostly marked a tree which is treated as a procedure of the formation of communities and it is difficult to understand which branches of the tree have real significance for non-topological information. So the definitions of community in a strong sense and in a weak sense are proposed [10], and Liu et al. also define community structure property [11].

In this paper, the motivation of this paper is to obtain a stable community structure, and community structure is defined on the basis of the above-mentioned documents, which is introduced in the following. Therefore, combining label propagation algorithm and community structure, CDMMCS is proposed. Because the update rule is based on the modularity maximum, CDMMCS begins with some small communities and labels propagate between communities. The communities are not merged until the propagation stops. Checking the merged community is to avoid the appearance of weak sense community.

3. Algorithm Descriptions.

3.1. Relative definitions.

Given complex network, $G(V, E)$, among which $V = (v_1, v_2, \ldots, v_m)$ is the node set and $E = (e_1, e_2, \ldots, e_n)$ is the edge set. In this paper, network is considered as undirected, unweighted and unsigned.

3.1.1. *Modularity.* Modularity is treated as a criterion to evaluate the detected communities, which is introduced by Newman and Girvan [12]. It implies the structure of community and it is defined as follows:

$$Q = \sum e_{ii} - (a_i)^2 \tag{1}$$

where e_{ii} is the fraction of edges that connects two nodes within the community i, and a_i represents the fraction of edges that connects with community i.

Formula (1) is rewritten according to the specific situation. E_i is defined as the number of the internal edges of the community i and m is defined as the number of the edges of network. E_{xt_i} represents the number of all external edges connecting to community i. So the formula of modularity is written as follows in this paper.

$$Q = \sum \frac{E_i}{m} - \frac{2E_i + E_{xt_i}}{2m} \tag{2}$$

3.1.2. *The change of modularity.* It is the assumption that there are some sub-communities including two sub-communities c_i and c_j. If c_i and c_j are merged forming c_m, and Q_i, Q_j, Q_m are their modularity respectively, the change between c_i, c_j and c_m is described as Formula (3).

$$\Delta Q = Q_m - (Q_i + Q_j) \tag{3}$$

And in more detail from Formula (1):

$$\Delta Q = e_{mm} - a_m^2 - \left(e_{ii} - a_i^2 + e_{jj} - a_j^2\right) \tag{4}$$

Combining Formula (2) and Formula (4), we could have

$$\Delta Q = \frac{E_{ij}}{m} - 2 \times \frac{2E_i + E_{xt_i}}{2m} \times \frac{2E_j + E_{xt_j}}{2m} \tag{5}$$

where E_{ij} is the number of edges between community i and community j.

3.1.3. *Community structure.* Here two types of community structure are defined.

Strong sense community is described as $k_{in} > k_{out}$. k_{in} are the internal connections of a community, and k_{out} are the external connections. Strong sense community has more connections within the community than with the rest of the graph.

Weak sense community is described as $k_{in} < k_{out}$. k_{in} are the internal connections of a community, and k_{out} are the external connections. In weak sense community, the connections within the community are smaller than its external connections. However, the weak sense community should be divided into two cases.

The internal connections are larger than the connections between this community and any other communities. This case is called temporarily stable states. The opposite situation is the second case. And in the second case, the community tends to merge with a strong sense one.

3.2. **Algorithm process.** A novel community detection algorithm is proposed based on label propagation and community structure, considering with modularity maximum in the phase of updating label.

3.2.1. *Algorithm steps.* The algorithm steps are described as follows.

Step 1: The given network is divided into some small communities subCom based on the weight of edges. The weight of edge is assigned at first according to Formula (6) [13].

$$W_{ij} = A_{ij} + \sum_{k \in N} \left(\frac{A_{ik}}{D_i - A_{ij}} \times \frac{A_{kj}}{D_k} \right) \tag{6}$$

If there is a link between node i and node j, A_{ij} equals 1. Otherwise it will be 0 when there is no connection between them. D_i is the degree of node i, and D_k is the degree of node j. This weight involves with common neighbors between two nodes, so it denotes similarity to some extent.

Step 2: Every edge is sorted descending by their weight. Every node is divided into community according to the sequence that edges are picked from the array. Finally, the network is divided into some small communities. Every small community has its own label.

Step 3: Among small communities, label is propagated through modularity maximum. Calculating Formula (5) between a community and its neighbor communities, check the largest result of ΔQ which is the value of change. So the community changes its label into the label of its neighbor.

Step 4: In above step, the small communities are only propagating their labels, and they are not merged at the same time. So when the label stops propagating, small communities with the same label are put into one big community. And check community structure of every community on the basis of the definition of community structure. If there are weak

sense communities, repeat Step 4. If there is no weak sense community, the algorithm ends.

Step 5: If there are still weak sense communities after repeating Step 4, analyze the weak sense community. Merge it with its strong sense neighbor community if their own modularity has tendency to increase after merging. Repeat Step 5 till no weak sense community exists.

3.2.2. *Pseudo-code of algorithm.* List the main pseudo-code of algorithm.

Algorithm 1. CDMMCS

/*the formation of subCom according to Formula (6)*/
Input: subCom
Output: strong sense communities

```
While (no weak sense community exists)
    For each sub_j ∈ subCom do
        Find neighbors of sub_j;
        Label propagation by maximum modularity among neighbors according
        to Formula (5)
    End for
    Check community structure for every forming community;
End while
Merge the communities with the same label
If (merged communities still has weak sense community)
Merge the weak sense community with its most strong sense neighbor
community;
Return strong sense communities;
```

In Algorithm 1, the subCom is the small communities formed via Formula (6). When the label propagation finishes, subCom with the same labels is put together. Some small and explicit structure networks are usually done in this phase and do not need to propagate labels again. If weak sense communities are checked, it needs to propagate labels again till no weak sense communities exist. However, in this process some subCom are ineligible for maximum modularity and these subCom are also weak sense community, and they propagate labels via maximum modularity in idle work. In these circumstances, merge weak sense community with strong sense neighbor and their own modularity must have tendency to increase. So Algorithm 1 has the only target which is to avoid the appearance of weak sense community and ensure stability of detected communities.

In summary, the time complexity of the CDMMCS algorithm is as follows: the analysis of the running time of CDMMCS algorithm is mainly consumed in two aspects, one is the process of modularity maximization, which needs to calculate the inter community modularity change value; the other is to check the process of community structure. However, merging between communities after label propagation is completed, and the implementation steps of CDMMCS algorithm can be seen, the community structure obviously networks usually form the stable community after in several rounds of label propagation. Therefore, running time of CDMMCS algorithm on this kind of network will be significantly reduced.

4. **Experimental Results.**

4.1. **Experimental data.** We choose five classic datasets, namely, Zachary karate club dataset, dolphins social network, American football network, polbooks, and email network.

4.1.1. *Experiment on Zachary karate club dataset.* The experiment result on Zachary karate club dataset is shown in Figure 1. It is divided into three parts. Although there is discrepancy between partitions by CDMMCS and real partitions, their community structures are all strong sense and their states are stable.

4.1.2. *Experiment on Dolphins social network.* The final experiment result on Dolphins social network is shown in Figure 2. And the interim results after executing the first phase label propagation are $\{\{47, 50\}, \{7, 58, 20, 55, 8, 42, 10, 14, 6, 33, 40, 49, 57, 61\}, \{1, 41, 37\}, \{19, 46, 22, 30, 52, 24, 16, 25, 5, 12, 36, 56\}, \{17, 51, 34, 38, 35, 44, 15, 39, 45, 53, 13, 59\}, \{31, 48, 4, 9, 60, 11, 43, 3, 21, 29\}, \{27, 28, 26, 2, 18, 23, 32\}, \{54, 62\}\}$.

Among the results, there are weak sense communities. So it will go on executing label propagation and checking community structure till no weak sense communities exist. The final communities detected by CDMMCS are a little different from the real communities. In consideration of community structure, it is reasonable to divide it into four parts as Figure 2 shows.

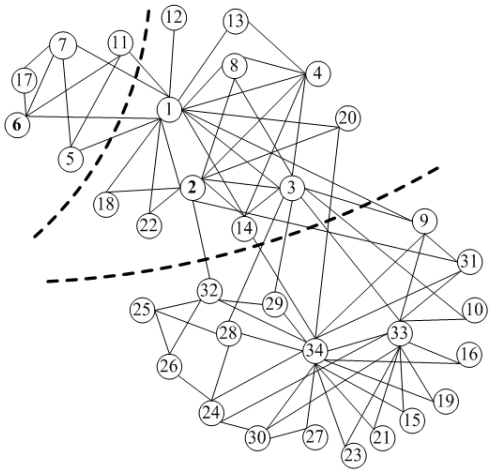


FIGURE 1. Result of karate by CDMMCS

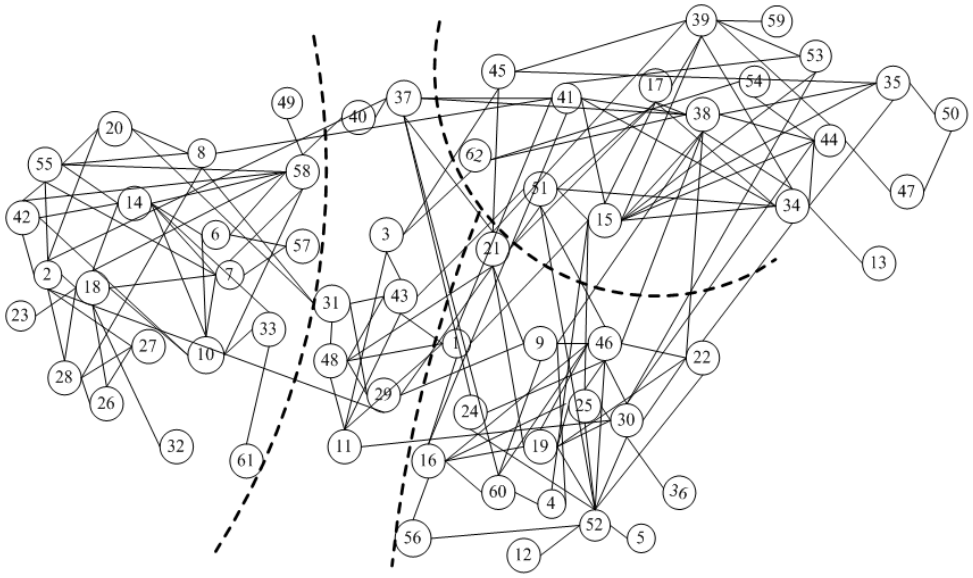


FIGURE 2. Results of dolphins by CDMMCS

TABLE 1. Real communities of American football

group	Nodes in community
1	2, 26, 34, 38, 46, 90, 104, 106, 110
2	3, 7, 14, 16 ,33, 40, 48, 61, 65, 101, 107
3	20 ,30, 31, 36, 56, 80, 95, 102
4	4, 6, 11, 41, 53, 73, 75, 82, 85, 99, 103, 108
5	45, 49, 58, 67, 76, 87, 92, 93, 111, 113
6	37, 43, 81, 83, 91
7	13, 15, 19, 27, 32, 35, 39, 44, 55, 62, 72, 86, 100
8	1, 5, 10, 17, 24, 42, 94, 105
9	8, 9, 22, 23, 52, 69, 78, 79, 109, 112
10	18, 21, 28, 57, 63, 66, 71, 77, 88, 96, 97, 114
11	12, 25, 51, 60, 64, 70, 98
12	29, 47, 50, 54, 59, 68, 74, 84, 89, 115

TABLE 2. Communities detected by CDMMCS

group	Nodes in community
1	46, 106, 104, 110, 34, 90, 2, 26, 38
2	14, 16, 61, 107, 3 ,40, 101, 7, 65, 33, 48
3	30, 95, 31, 36, 56, 80, 20, 102, 81, 83
4	53, 75, 4, 41, 11, 108, 6, 85, 73, 103, 82, 99
5	84, 89, 50, 54, 47, 68, 115, 74, 111
6	18, 88, 71, 77, 28, 57, 63, 96, 66, 97, 21, 114
7	8, 78, 22, 112, 52, 79, 23, 109, 9, 69
8	7, 76, 58, 93, 113, 64, 98, 59, 60, 49, 87, 45, 92
9	55, 72, 19, 35, 15, 39, 32, 100, 62, 27, 44, 43, 13, 86, 37
10	46, 106, 104, 110, 34, 90, 2, 26, 38

4.1.3. *Experiment on American football network.* Table 1 shows the real communities of football network, and the experiment result is shown in Table 2.

Try to find some explanations for the above partitions in the comparative analysis of Table 1 and Table 2. In initial phase of CDMMCS, $\{74, 111\}, \{59, 60\}, \{64, 98\}$ these three small communities lead to the result as Table 2 while they are not in the same community in real groups. However, these three pairs have the largest similarity among their own neighbors. So treating them as the one community respectively is reasonable.

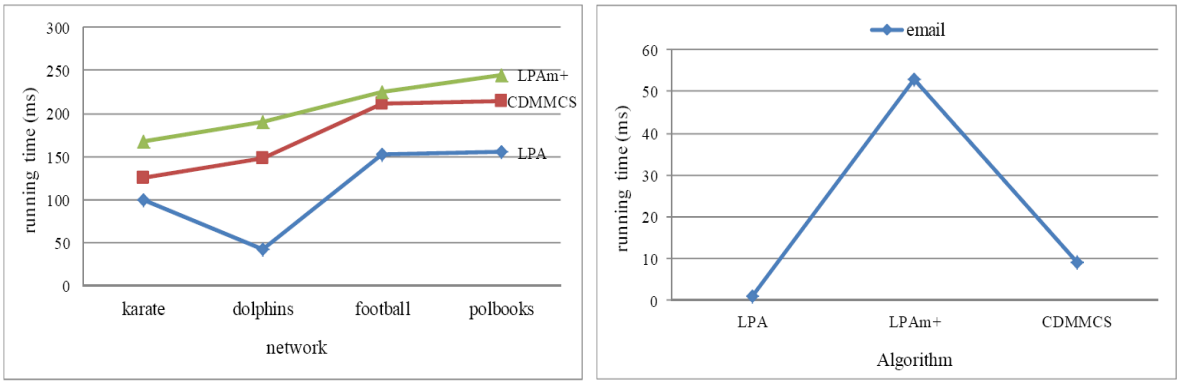
4.2. **Comparisons among algorithms.** The traditional label propagation algorithms LPA and LPAm+ are realized based on their main ideas of algorithms. We compare the CDMMCS with LPAm and LPAm+ in two aspects of modularity and running time.

4.2.1. *Comparison of modularity.* From the 20 experimental results, the maximal modularity is chosen in LPA. Compute the modularity of CDMMCS and LPAm+ respectively and the result of comparisons is shown in Table 3.

Because the LPA tends to forming one community in dolphins network and email network, the two values of modularity are null in Table 3. From Table 3, we can see that the modularity of CDMMCS is usually larger than the LPAm+ excluding the email network. Based on the modularity maximum in both LPAm+ and CDMMCS, the value of modularity is nearly between these two algorithms. However, CDMMCS algorithm checks community structure from two different aspects, therefore, the value of modularity has more advantages than LPAm+ algorithm.

TABLE 3. Comparison of modularity

Networks	LPA (Max)	LPAm+	CDMMCS
Karate	0.3717	0.3397	0.4020
Dolphins	*	0.4811	0.5123
Football	0.5931	0.5563	0.6044
Polbooks	0.5131	0.5122	0.5144
Email	*	0.5213	0.4980



(a) Running time in four networks (b) Running time in email network

FIGURE 3. Comparisons of running time

4.2.2. *Comparison of running time.* Comparison of running time is shown in Figure 3. Figure 3(a) is the result of comparison in four networks and Figure 3(b) is the result in email network. LPA has the best time efficiency. CDMMCS ranks second. LPAm+ is the last one in this paper. The time of traditional algorithm LPA is the best. CDMMCS spends most time in checking the community structure, so the network with specific structure costs a little time. While the time of LPAm+ costs in the label propagation and the process of merging. So in all kinds of networks, the running time of LPAm+ is the most.

5. **Conclusions.** In this paper, CDMMCS is proposed in order to gain higher modularity and avoid the appearance of weak sense community. Firstly, the change of modularity is listed as Formula (5). In process of maximum modularity, the case that ΔQ is below zero is divided into different situations. When need-to-merge community is a weak sense community and its modularity can increase merging with one strong sense community, it merges with this strong sense community because merged community becomes one stronger sense community. Secondly, definition of community structure also has a little difference in order to obtain strong sense community. Weak sense community has two different cases. Finally, small communities create a good condition to use modularity maximum because modularity implies the community structure not the single node. Label propagation between communities decreases the times of propagation. When communities are merged, the inspections of community structure increase the possibilities of obtaining the strong sense communities and larger modularity values. So from the experiments results, we can see that CDMMCS has the time efficiency which is the best advantage of label propagation algorithm and the modularity is also larger than the LPAm+. In fact, the social network is an ongoing dynamic network, new nodes and edges to join or form; therefore, how to effectively detect the dynamic social network community structure and track the evolution of community in dynamic social networks is our future work.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (No. 61472340, No. 61602401).

REFERENCES

[1] J. L. Zhang, Y. L. Chang and W. Shi, Overview on label propagation algorithm and applications, *Application Research of Computers*, vol.30, no.1, pp.21-25, 2013.

[2] 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 & Soft Matter Physics*, vol.76, no.3, 2007.

[3] I. X. Leung, P. Hui and P. Liò, Towards real-time community detection in large networks, *Physical Review E Statistical Nonlinear & Soft Matter Physics*, vol.79, no.6, pp.853-857, 2008.

[4] S. Gregory, Finding overlapping communities in networks by label propagation, *New Journal of Physics*, vol.12, no.10, pp.2011-2024, 2009.

[5] M. J. Barber and J. W. Clark, Detecting network communities by propagating labels under constraints, *Physical Review E Statistical Nonlinear & Soft Matter Physics*, vol.80, no.2, pp.283-289, 2009.

[6] X. Liu and T. Murata, Advanced modularity-specialized label propagation algorithm for detecting communities in networks, *Physical A – Statistical Mechanics & Its Applications*, vol.389, no.7, pp.1493-1500, 2010.

[7] J. Ugander and L. Backstrom, Balanced label propagation for partitioning massive graphs, *Proc. of the 6th ACM International Conference on Web Search and Data Mining*, pp.507-516, 2013.

[8] Z. Lin, X. Zheng and N. Xin, CK-LPA: Efficient community detection algorithm based on label propagation with community kernel, *Physical A – Statistical Mechanics & Its Applications*, vol.416, pp.386-399, 2014.

[9] M. Girvan and M. E. J. Newman, Community structure in social and biological networks, *Proc. of the National Academy of Sciences of the United States of America*, vol.99, no.12, pp.7821-7826, 2002.

[10] F. Radicchi, C. Castellano and F. Cecconi, Defining and identifying communities in networks, *Proc. of the National Academy of Sciences of the United States of America*, vol.101, no.9, pp.2658-2663, 2004.

[11] W. Liu, M. Pellegrini and X. Wang, Detecting communities based on network topology, *Scientific Reports*, vol.4, no.4, p.5739, 2014.

[12] M. E. Newman and M. Girvan, Finding and evaluating community structure in networks, *Physical Review E Statistical Nonlinear & Soft Matter Physics*, vol.69, no.2, 2004.

[13] T. Saha, C. Domeniconi and H. Rangwala, Detection of communities and bridges in weighted networks, *International Conference on Machine Learning and Data Mining in Pattern Recognition*, pp.584-598, 2011.