Pramana – J. Phys. (2016) 87: 58
DOI 10.1007/s12043-016-1270-2

Overlapping community detection using weighted consensus clustering

LINTAO YANG*, ZETAI YU, JING QIAN and SHOUYIN LIU

College of Physical Science and Technology, Central China Normal University, Wuhan 430079, China
*Corresponding author. E-mail: ltaoyang@gmail.com

MS received 8 February 2014; revised 14 December 2015; accepted 25 January 2016; published online 21 September 2016

Abstract. Many overlapping community detection algorithms have been proposed. Most of them are unstable and behave non-deterministically. In this paper, we use weighted consensus clustering for combining multiple base covers obtained by classic non-deterministic algorithms to improve the quality of the results. We first evaluate a reliability measure for each community in all base covers and assign a proportional weight to each one. Then we redefine the consensus matrix that takes into account not only the common membership of nodes, but also the reliability of the communities. Experimental results on both artificial and real-world networks show that our algorithm can find overlapping communities accurately.

Keywords. Complex networks; overlapping community; consensus clustering.

PACS Nos 89.75.−k; 89.75.Fb; 89.75.Hc

1. Introduction

Most complex systems in the real world can be described in terms of networks or graphs. A key property of many real-world networks is their community structure: the existence of groups of nodes such that nodes within the groups have higher density of edges while nodes among groups have lower density of edges [1,2]. Identifying the community structure is crucial to understand the structural, functional and dynamical properties of complex networks [3,4]. Thus, many community detection algorithms have been proposed to detect community structure in complex networks. Many of them have been limited to partitions, where each node belongs to one community.

However, communities in real-world networks are usually overlapping such that some nodes may belong to more than one community. For example, in social networks, a person may be in several social groups like family, friends and colleagues. Overlapping community structure can be represented by a cover of network [5], which is defined as a set of clusters such that each node belongs to at least one cluster and no cluster is a proper subset of any other cluster.

Many overlapping community detection algorithms have been proposed. Most of them are unstable and non-deterministic. The most typical algorithms are local expansion and optimization algorithms [5–7], which first find the seeds of communities and then expand these seeds by greedily optimizing a local community fitness function. In general, the algorithms are very sensitive to the random seeds, and modifying these may lead to different outcomes. The tie-break rules adopted by algorithms may also produce different results. For instance, in COPRA algorithm [8], it is possible that belonging coefficients of all pairs (*community identifier, belonging coefficient*) of a node are less than a threshold. In this case, if more than one pair has the same maximum belonging coefficient, the algorithm randomly selects one of the pairs. Thus, this random selection makes the algorithm non-deterministic. In order to generate more reliable and accurate results, combining outcomes of these algorithms is a promising approach.

Lancichinetti and Fortunato [9] first applied consensus clustering in community detection problems. The core idea is based on the assumption that similar nodes are very likely grouped together by the base algorithms (e.g. classic non-deterministic community detection algorithms) and, conversely, nodes that co-occur very often in the same community should be regarded as being very similar. Hence, a consensus matrix is constructed from the base partitions obtained by the base algorithms. Each matrix entry is a similarity measure

about how many times a given pair of nodes is allocated to the same community. This consensus matrix can then be used as an input for the same base algorithm, leading to a new set of partitions, which generate a new consensus matrix, etc., until a unique partition is finally reached, which cannot be altered by further iterations. Dahlin and Svenson [10] also adopted consensus matrix and developed a node-based fusion of community algorithms by agglomerative hierarchical clustering using a special linkage rule. However, one significant drawback of these algorithms is that they assume that each community in all base partitions is perfectly reliable. As the base algorithms are unstable, all communities may not be reliable and the reliability of individual communities may not be the same. Hence, a simple average of all base partitions does not have to be the best choice.

In this paper, we propose an overlapping community detection algorithm by using weighted consensus clustering. The intuition here is that, if two nodes are assigned to a community with high reliability, both the nodes are probably in this community. Based on this intuition, we first evaluate a reliability measure for each community in all base covers, which take into account both the topology of the original network and the information given by the base covers. Then we redefine the consensus matrix, in which each entry represents the common membership of two nodes and its value is proportional to the reliability of the community they belong to. We evaluate our algorithm with standard measures such as modularity and omega index on both artificial and real-world networks. Experimental results show that our algorithm can detect overlapping communities accurately.

The rest of this paper is organized as follows: Section 2 proposes a consensus-based overlapping community detection algorithm (COCDA). Experimental results on LFR benchmark and several real-world networks are given in §3. Section 4 concludes this paper.

2. Detecting overlapping communities by weighted consensus clustering

Let $G = (V, E)$ be an undirected network or graph representing a complex network, where V is a set of $|V|$ nodes and E is a set of $|E|$ edges. A community ensemble is a set of base covers, represented as $P = \{P^1, P^2, \ldots, P^H\}$, where H is the ensemble size. Each base cover is a set of communities $P^i = \{C_1^i, C_2^i, \ldots, C_{k_i}^i\}$, where k_i is the number of communities in the ith cover and C_j^i is the community of the jth cover. In our algorithm, we allow $C_u^i \bigcap C_v^i \neq \emptyset$ so that the community can overlap with each other. The goal of our algorithm is to find a consensus cover P^*, which better represents the properties of each cover in P. We use conventional non-deterministic algorithms as base algorithms to obtain the base covers. In the following, we use the default settings of parameters in the base algorithms.

2.1 *Measuring community reliability*

Inspired by [11], we propose a method to evaluate the reliability of individual communities within a base cover. We first introduce a probability p_{xy} for the edge between node x and node y of connecting two nodes in the same community. The probability p_{xy} is defined as

$$p_{xy} = \frac{1}{H}\sum_{i=1}^{H}\delta(L^i(x), L^i(y)), \tag{1}$$

where $L^i(x)$ represents the associated label of the node x in the base cover P^i and $\delta(a, b)$ is 1, if $a = b$, and 0 otherwise. The value of p_{xy} is large only for the edge (x, y) which is most frequently in the same community, whereas low value indicates that the edge probably connects two different communities.

Given a community C_j^i in the base cover P^i, an edge set $E_{\text{in}}(C_j^i) \subseteq E$ includes all such edges (x, y) with both nodes x and y included in C_j^i; another edge set $E_{\text{out}}(C_j^i) \subseteq E$ includes all such edges (x, y) with only node x or y included in C_j^i. We proposed a measure shown as eq. (2) to judge the reliability of the community C_j^i:

$$\begin{aligned}\text{rel}(C_j^i) = 1 - \frac{1}{|E_{\text{in}}(C_j^i)| + |E_{\text{out}}(C_j^i)|} \\ \times \sum_{(x,y)\in E_{\text{in}}(C_j^i)\bigcup E_{\text{out}}(C_j^i)} p_{xy}\log_2 p_{xy} \\ + (1 - p_{xy})\log_2(1 - p_{xy}),\end{aligned} \tag{2}$$

where $|E_{\text{in}}(C_j^i)|$ and $|E_{\text{out}}(C_j^i)|$ are the number of edges in $E_{\text{in}}(C_j^i)$ and $E_{\text{out}}(C_j^i)$, respectively. If all edges in $E_{\text{in}}(C_j^i)$ have probability $p_{xy} = 1$ and all edges in $E_{\text{out}}(C_j^i)$ have probability $p_{xy} = 0$, $\text{rel}(C_j^i) = 1$. We can conclude that the community is very stable. If all edges in $E_{\text{in}}(C_j^i)\bigcup E_{\text{out}}(C_j^i)$ have probability $p_{xy} = \frac{1}{2}$, $\text{rel}(C_j^i) = 0$. We can conclude that the community is totally unstable.

2.2 *Constructing consensus network*

For each base cover P^i, we first construct the membership matrix $\mathbf{U}^i \in \mathbb{R}^{|V|\times k_i}$, in which the rows correspond to nodes while the columns correspond to communities. Each element u^i_{xj} in the membership matrix represents the membership of node x to the jth community:

$$u^i_{xj} = \begin{cases} 1/l^i_x, & \text{if node } x \in C^i_j, \\ 0, & \text{if node } x \notin C^i_j, \end{cases} \tag{3}$$

where l^i_x is the number of communities the node x belongs to in the ith cover. Clearly, when node x only belongs to the jth community, i.e. $l^i_x = 1$, the membership u^i_{xj} is 1.

Given a membership matrix $\mathbf{U}^i$, a $|V|\times|V|$ similarity matrix is constructed, which is denoted as $\mathbf{S}^i$. Matrix entries represent the similarity between nodes x and y:

$$s^i_{xy} = \sum_{j=1}^{k_i} f(u^i_{xj}, u^i_{yj}) \times \text{rel}(C^i_j), \tag{4}$$

where $\text{rel}(C^i_j)$ is the reliability of the community C^i_j and $f(u^i_{xj}, u^i_{yj})$ is a suitable fuzzy t-norm (e.g. an algebraic product $f(u^i_{xj}, u^i_{yj}) = u^i_{xj} \times u^i_{yj}$), which is interpreted as the common membership of two nodes x and y to the same community.

Accordingly, $\{\mathbf{S}^1, \mathbf{S}^2, \ldots, \mathbf{S}^H\}$ are obtained from H covers. All similarity matrices $\{\mathbf{S}^1, \mathbf{S}^2, \cdots, \mathbf{S}^H\}$ are combined into a single consensus matrix as shown in eq. (5).

$$\mathbf{M} = \frac{1}{H}\sum_{i=1}^{H} \mathbf{S}^i, \tag{5}$$

where each entry of this matrix M_{xy} reflects the average similarity between the nodes x and y. A large value of M_{xy} indicates that the two nodes x and y have a high probability to be classified into the same community, whereas low value indicates that the two nodes have a small probability to be classified into the same community.

To reduce the effects of noise and improve the algorithm execution speed, we introduce a filtering procedure. The new consensus matrix $\mathbf{M}^{\text{new}}$ is filtered from $\mathbf{M}$ as shown in eq. (6).

$$M^{\text{new}}_{xy} = \begin{cases} M_{xy}, & \text{if rand(1)} < M_{xy}, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

where rand(1) is a random number in the range [0,1]. Clearly, the larger the M_{xy} is, the less probably the edge (x, y) is removed. From M^{new}_{xy}, we create a consensus network $G' = (V, E', W)$, where the weight of the edge (x, y) is M^{new}_{xy}.

2.3 *Producing the consensus cover*

We apply H times a non-deterministic overlapping community detection algorithm to the consensus network G', where the algorithm used should be able to detect overlapping communities in weighted networks, because the consensus network is weighted. Then we use the new set of covers produced to construct a new community ensemble. The procedure is iterated until all covers $\{P^1, P^2, \ldots, P^H\}$ are equal. Here, we compute the Omega index [12] between a given cover and the rest of the covers. If all Omega indices are 1, we think that all covers are equal. As the filtering procedure introduces the stochasticity, our algorithm needs more iteration to converge. In practice, we have observed that running our algorithm for ten iterations gives good results.

3. Experiments

3.1 *Experiments with LFR benchmark networks*

We use the LFR benchmark proposed by Lancichinetti *et al* [13]. The LFR benchmark introduces heterogeneity into degree and community size distributions of a network and thus it is closer to the features observed in real world than standard benchmarks. To evaluate the accuracy and stability of the proposed COCDA algorithm, we create 100 benchmark networks with the same set of parameters. We set $n = 5000$, $\bar{k} = 10$, $k_{\max} = 50$, $\mu = 0.3$, $\tau_1 = 2$, $\tau_2 = 1$, $c_{\min} = 20$, $c_{\max} = 100$ and $O_n = \{500, 1000\}$. For every network we have produced $H = 10$ covers with the chosen base algorithm and average the results. Using H covers obtained as input, we also perform the COCDA algorithm H times and average the results. In the experiments, we use Omega index to calculate how similar the known cover is to the covers found by the algorithms. The Omega index is between 0 and 1, with 1 corresponding to a perfect matching.

Figure 1 shows the Omega index between the known covers and the covers found by the algorithm as a function of O_m. The curve OLSOM shows the average of the Omega index between each cover found by the OLSOM algorithm and the known cover. The curve COCDA reports the average of Omega index between the consensus cover found by COCDA and the known

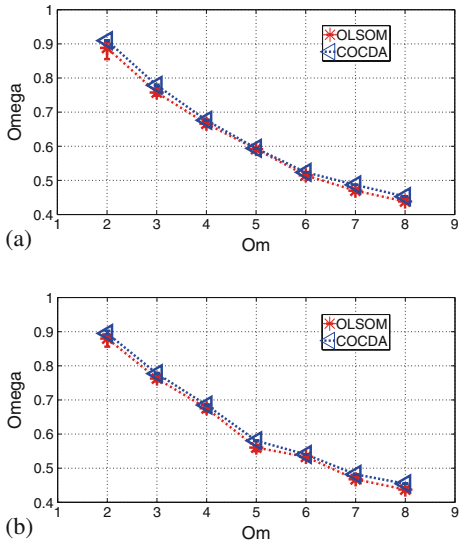

Figure 1. The Omega index of the base algorithm and the COCDA algorithm on the LFR benchmark with different O_m for (**a**) $O_n = 500$ and (**b**) $O_n = 1000$.

cover. The data points shown in the figure are the result of averaging 100 benchmark networks (each network with the same of parameters). Error bars show the minimum and maximum Omega indices. As expected, the performance of both algorithms consistently and significantly drop as the diversity of overlapping increases (i.e., O_m getting larger). We can see that the COCDA algorithm slightly improves the Omega index, because the performance of OLSOM algorithms on the LFR benchmark networks is very good already.

3.2 *Experiments with real-world networks*

As real networks may have some topological properties different from synthetic ones, we consider 17 representative real-world networks drawn from disparate fields. Table 1 lists the real-world networks for our tests and their statistics. n and m are the total numbers of nodes and edges, respectively. $\langle k \rangle$ is the average degree of the network. $\langle d \rangle$ denotes the average distance, $\langle C \rangle$ indicates the clustering coefficient [14]. For the networks of Roget, polblogs, ODLIS and CA-GrQc, we reduced the sizes of these networks by only keeping the largest connected component and by iteratively removing all the one-degree vertices. In the following, we use an overlapping modularity (Q_{ov}) measure [15] to evaluate the performance of the algorithms. The values of Q_{ov} vary between 0 and 1. The larger the value is, the better the performance is.

In table 2, we present the mean and the standard deviation of the modularity Q_{ov} computed from OLSOM and COCDA for 17 real-world networks. The results of both algorithms are computed by 10 executions using Dell PowerEdge R820 (Xeon E5-4607 v2*4, 128GB main memory). For each network, the mean modularity of COCDA is higher than that of OLSOMs which obviously indicates that COCDA is more accurate than OLSOM. While the standard deviation of the modularity computed from COCDA is smaller than that computed from OLSOM, we can clearly figure out that COCDA is more stable than

Table 1. The basic topological features of the 17 real-world networks.

Networks	Ref.	n	m	$\langle k \rangle$	$k_{\max}$	$\langle d \rangle$	$\langle C \rangle$
Karate	[16]	34	78	4.59	17	2.41	0.5706
High school	[17]	69	219	6.35	14	2.96	0.4660
Lesmis	[18]	77	254	6.60	36	2.64	0.5731
Jazz	[19]	198	2742	27.70	100	2.24	0.6175
USAir	[20]	332	2126	12.81	139	2.74	0.6252
C. elegans	[14]	453	2025	8.94	237	2.66	0.6465
Roget	[18]	994	3640	7.32	28	4.07	0.1541
Email	[21]	1133	5451	9.62	71	3.61	0.2202
SmaGri	[22]	1024	4916	9.60	232	2.98	0.3071
Polblogs	[23]	1222	16714	27.36	351	2.74	0.3203
Yeast	[24]	2375	11693	9.85	118	5.10	0.3057
ODLIS	[25]	2898	16376	11.30	592	3.17	0.2967
Facebook	[26]	4039	88234	43.69	1045	3.69	0.6055
Power	[14]	4941	6584	2.67	19	18.99	0.0801
CA-GrQc	[27]	4158	13422	6.46	81	6.05	0.5569
Router	[28]	5022	6258	2.49	106	6.45	0.0116
PGP	[29]	10680	24316	4.55	205	7.49	0.2659

Table 2. The modularities of the base algorithm and the COCDA algorithm on 17 real-world networks.

	Q_{ov}		Running time (s)	
Nets	OLSOM	COCDA	OLSOM	COCDA
Karate	0.7194(0.0108)	0.7386(0.0053)	0.7	1
High school	0.7846(0.0234)	0.8097(0.0169)	1	2
Lesmis	0.6920(0.0164)	0.7426(0.0128)	0.5	0.8
Jazz	0.5395(0.0464)	0.6407(0.0151)	1	4
USAir	0.5937(0.0413)	0.6757(0.0034)	2	11
C. elegans	0.5098(0.0184)	0.6475(0.0186)	2	12
Roget	0.4039(0.0359)	0.5284(0.0142)	6	37
Email	0.4682(0.0384)	0.5722(0.0081)	4	31
SmaGri	0.6244(0.0036)	0.6369(0.0003)	3	22
Polblogs	0.4050(0.0359)	0.5257(0.0325)	28	113
Yeast	0.6202(0.0038)	0.6326(0.0002)	3	28
ODLIS	0.2711(0.0166)	0.4012(0.0086)	40	195
Facebook	0.5621(0.0154)	0.6411(0.0080)	55	275
Power	0.4040(0.0075)	0.4752(0.0034)	5	101
CA-GrQc	0.6475(0.0116)	0.7337(0.0033)	7	153
Router	0.5577(0.1960)	0.6316(0.0005)	3	26
PGP	0.5561(0.0090)	0.6596(0.0027)	42	956

OLSOM. We also show the comparison of the average running time of both the algorithms in table 2. COCDA has a similar time complexity when the experiment is conducted in a small network like karate or lemis. However, it has about 20 times higher time complexity when the experiment is conducted in a rather large network. For instance, when we execute our program to CA-GrQc network, the OLSOM algorithm takes 7 s while the consensus algorithms takes 153 s on an average.

A high modularity might not necessarily result in the true cover. We used the karate network with known attributes to verify the output of the detection algorithms. This network characterizes the social interactions between individuals in a karate club in an American university, which has 34 members and 78 pairwise links. As a conflict had arisen between the club's administrator and the main teacher, the club eventually splits into two smaller clubs. They are {1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13, 14, 17, 18, 20, 22} and {9, 10, 15, 16, 19, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34}. By multiple executions applying OLSOM algorithm to this network, different covers are obtained. For example, at one time, the cover contains two large communities (e.g. {1, 2, 3, 4, 5, 6, 7, 8, 11, 13, 14, 17, 18, 22}, {9, 15, 16, 19, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34}), and three smaller communities which contains only one node (e.g. {12}, {20} and {10}). At another time, the cover contains some different communities (e.g. {3, 9, 10, 15, 16, 19, 21, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34}, {1, 2, 3, 4, 5, 6, 7, 8, 11, 13, 14, 17, 18, 20, 22}, {12}). Our algorithm combines the base covers obtained by each OLSOM execution, and results in two communities, which is in accordance with the real split.

4. Conclusion

In this paper, we presented an algorithm, COCDA, to detect overlapping communities in complex networks using weighted consensus clustering. We proposed a method to evaluate the reliability of individual communities and redefined the consensus matrix. Experimental results showed that the COCDA algorithm can combine the covers obtained from the base algorithms, considerably enhancing both the stability and the accuracy. However, COCDA has some drawbacks. For example, it has high time complexity. A possible approach for speeding up COCDA is the use of parallel computing. The community ensemble is simultaneously generated on modern multicore processors and reduces the running times.

Acknowledgements

The authors would like to acknowledge the reviewers for their useful comments and advice. This work

was financially supported by self-determined research funds of CCNU from the colleges basic research and operation of MOE (CCNUI5A05044) and the National Natural Science Foundation of Hubei province under Grant No. 2013CFB210.

References

[1] S Gregory, *J. Stat. Mech.* **2011(02)**, P02017 (2011)
[2] M Seifi and J L Guillaume, in: *WWW: Proc. of the 21st International Conference Companion on World Wide Web* (2012), pp. 1173–1180
[3] P Sah, L O Singh, A Clauset and S Bansal, bioRxiv (2013)
[4] X Wang, L Jiao and J Wu, *Physica A* **388(24)**, 5045 (2009)
[5] A Lancichinetti, S Fortunato and J Kertész, *New J. Phys.* **11(3)**, 033015 (2009)
[6] J Xie, B K Szymanski and X Liu, in: *ICDMW: Proc. Data Mining Technologies for Computational Collective Intelligence Workshop at ICDM* (2011), pp. 344–349
[7] C Lee, F Reid, A McDaid and N Hurley, arXiv:1002.1827 (2010)
[8] S Gregory, *New J. Phys.* **12(10)**, 103018 (2010)
[9] A Lancichinetti and S Fortunato, *Sci. Rep.* **2** (2012)
[10] J Dahlin and P Svenson, arXiv:1309.0242 (2013)
[11] D Gfeller, J C Chappelier and P De Los Rios, *Phys. Rev. E* **72(5)**, 056135 (2005)
[12] L M Collins and C W Dent, *Multivar. Behav. Res.* **23(2)**, 231 (1988)
[13] A Lancichinetti, S Fortunato and F Radicchi, *Phys. Rev. E* **78(4)**, 046110 (2008)
[14] Duncan J Watts and Steven H Strogatz, *Nature* **393**, 440 (1998)
[15] V Nicosia, G Mangioni, V Carchiolo and M Malgeri, *J. Stat. Mech.* **2009(03)**, P03024 (2009)
[16] Wayne W Zachary, *J. Anthropol. Res.* **33(4)**, 473 (1977)
[17] Jierui Xie and Boleslaw K Szymanski, in: *Advances in knowledge discovery and data mining* (Springer, 2012) pp. 25–36
[18] Donald E Knuth, *The Stanford GraphBase: A platform for combinatorial computing* (ACM, 1993)
[19] Pablo M Gleiser and Leon Danon, *Adv. Complex Syst.* **06(4)**, 565 (2003)
[20] V Batagelj and A Mrvar, *Graph Drawing* pp. 77–103 (2003)
[21] Roger Guimera, Leon Danon, Albert Diaz-Guilera, Francesc Giralt and Alex Arenas, *Phys. Rev. E* **68(6)**, 065103 (2003)
[22] Norman P Hummon and Patrick Dereian, *Soc. Networks* **11(1)**, 39 (1989)
[23] Lada A Adamic and Natalie Glance, in: *Proceedings of the 3rd International Workshop on Link Discovery* (2005), pp. 36–43
[24] Mering C Von, R Krause, B Snel, M Cornell, S G Oliver, S Fields and P Bork, *Nature* **417(6887)**, 399 (2002)
[25] J Culpepper, *Electron. Res. Rev.* **4(10)**, 124 (2000)
[26] Julian Mcauley and Jure Leskovec, *Adv. Neural Inform. Proces. Systems*, pp. 539–547 (2012)
[27] Jure Leskovec, Jon Kleinberg and Christos Faloutsos, *ACM Trans. Knowledge Disc. Data* **1**, 2 (2007)
[28] Neil Spring, Ratul Mahajan and David Wetherall, in: *ACM SIGCOMM computer communication review* (ACM, 2002) Vol. 32, pp. 133–145
[29] Marián Boguñá, Romualdo Pastor-Satorras, Albert Díaz-Guilera and Alex Arenas, *Phys. Rev. E* **70(5)**, 056122 (2004)