

# Micro-blog User Community Detection by Focusing on Micro-blog Content and Community Structure

Jia Liu<sup>1</sup>(✉), Ya-jun Du<sup>1</sup>, and Ji-zhong Ren<sup>2</sup>

<sup>1</sup> School of Computer and Software Engineering,  
Xihua University, Chengdu 610039, China  
xiaoke92@foxmail.com

<sup>2</sup> School of Political and Public Administration,  
Electronic Science and Technology University, Chengdu 611731, China

**Abstract.** Micro-blog community detection is one of the hot problems of Micro-blog platform. There are many existing community detection methods that are dedicated to detect community by only considering the topological structure. To detect Micro-blog community better, we considering the Micro-blog content as well as the topological structure. In Micro-blog community, the essence of a concept is semantic objects in the real world. The concept is composed of the object's attribute set, and the attribute set is a set of nouns that essentially can represent the object. In this article, we let user be object and calculate the interest similarity by the object's attribute set. First, we establish a micro-blog social network by analyzing the object's attribute set. Second, we find the clustering directions for each object by the Random Walk method. Then, we detect micro-blog user community following the clustering directions. Finally, experiments performed to verify the efficiency of our method from the two aspects of community structure and interest cohesion.

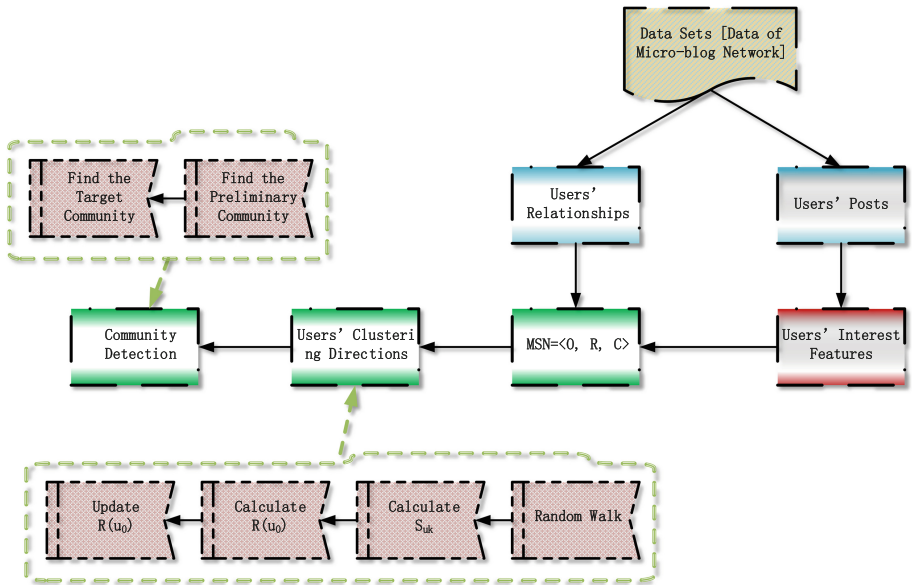
**Keywords:** Community detection · Micro-blog content · Random walk

## 1 Introduction

In recent years, the researches that focus on the Micro-blog platform mainly include: Micro-blog user behavior analysis [1], Micro-blog language analysis and identification, standard word processing [2], information dissemination and public opinion model [3], and Micro-blog community division and detection. Micro-blog community detection is the main research work. The objective of a community detection is to identify groups of nodes that have common interests, habits and hobbies in a specific network. Many algorithms have been proposed to find communities, such as the local network community detection method [4], the cavity method [5], and the parallel community detection algorithm [6]. The vast majority of those methods concentrate on community structure to detect a community.

A micro-blog is the epitome of social reality because it provides people a huge amount of valuable data concerning micro-blog contents. To detect a community in the micro-blog network, we let user be object and calculate the interest similarity by the object’s attribute set. Different than the existing methods, we combine community structure and micro-blog contents to detect a community. Detecting a community in a social network raises new challenges to traditional community detection algorithms and is the basis for studying the knowledge map that based on Micro-blog community. To analyze a social network community, higher rationality and adaptability force us to mine user preferences. The entire process of our method is shown in Fig. 1. Our main contributions in this paper can be summarized as follows:

- (1) The micro-blog social network is described as a tuple  $(MSN = \langle O, R, C \rangle)$ . Then, we let user be object and calculate the interest similarity by the object’s attribute set.
- (2) Detect the clustering directions based on the Random Walk method.
- (3) We detect communities in the micro-blog network by following the clustering directions.
- (4) We analyze the performance of our method via experiments from two aspects: community interest cohesion and community structure.



**Fig. 1.** The entire process of detecting micro-blog user community.

The structure of the paper is as follows: In the next section, we present a review of the literature. In Sect. 3, we provide a detailed discussion of building a

micro-blog network graph. In Sect. 4, we will find clustering directions for each object. Community detection will be shown in Sect. 5. Finally, we present the results of our experiments in Sect. 6.

## 2 Related Work

According to the characteristics of network structure [7], community detection methods can be roughly categorized into positive network community detection and signed network community detection. Moreover, among efficient heuristics for community detection we can distinguish between those based on community agglomeration and those based on local node moves.

The Spectral Bisection algorithm [8] and the Kernighan-Lin algorithm [9] are two classical algorithms, which have been recommended for detecting positive network communities. Brandes et al. [10] showed the NP-complete problem of the Spectral Bisection algorithm. Additionally, the variations in the order of the partitions may significantly alter the results because of the sensitivity of the Kernighan-Lin algorithm to the initial partition. Several algorithms based on optimizing the modularity are developed to detect the community structures of complex networks, especially for weighted networks and directed networks [11] to address the issues of these two classical methods [12, 13]. Furthermore, to obtain the suboptimal solutions, a few of the optimization algorithms [14] have been introduced to reduce the two major limitations of the methods based on modularity maximization. The two major limitations are that the maximization of modularity is an NP-hard problem and the resolution limit. The resolution limit means that the community detection methods cannot detect the communities whose node numbers are smaller than a predefined threshold [7]. Mu et al. [15] proposed a memetic algorithm based on genetic algorithms to maximize the modularity density and resolve the resolution limit. Community detection in social network analysis is usually considered as a single-objective optimization problem, in which different heuristics or approximate algorithms are employed to optimize an objective function that captures the notion of community. Due to the inadequacy of those single-objective solutions, several algorithms [16] based on a multi-objective framework are proposed.

A globally greedy agglomerative method known as CNM [17] is proposed. Community detection by label propagation belongs to the class of local move heuristics. It has originally been described by Raghavan et al. [18]. Several variants of the algorithm exist, one of them is due to Gilbert et al. [19]. The latter use the algorithm as a prototype application within a parallel toolbox that uses numerical algorithms for combinatorial problems.

## 3 Establish the Micro-blog Network

We describe a Micro-blog Social Network (MSN) by a tuple  $(MSN = \langle O, R, C \rangle)$ , where  $O$  represents a set of users or objects,  $R$  is a set of objects' relationships

and  $C$  denotes a set of groups of objects. Each group  $c_i \in C(i=1,2,3,\dots)$  is a subset of  $O$ .

Based on the majority of traditional community detection methods that focus on the social network topology, we consider the content analysis of the MSN in our approach. In our method,  $O = \{u_1, u_2, \dots, u_i, \dots\}$  represents a set of objects,  $u_i = (\mathbf{p}_i)$  is an object, and  $\mathbf{p}_i$  is an interest feature vector. If object  $u_i$  follows object  $u_j$ , there is an edge  $e_{ij} \in R$  between  $u_i$  and  $u_j$ . We define  $w_{ij}$  as the interest similarity between two objects,  $u_i$  and  $u_j$ . Simultaneously, we allow  $w_{ij}$  be the weight of the edge  $e_{ij}$ . To compute the  $w_{ij}$ , we improve the Tanimoto coefficient. The improved Tanimoto coefficient formula is as follows:

$$w_{ij} = \frac{\mathbf{p}_i' \bullet \mathbf{p}_j'}{|\mathbf{p}_i'|^2 + |\mathbf{p}_j'|^2 - \mathbf{p}_i' \bullet \mathbf{p}_j'}. \quad (1)$$

Where  $\mathbf{p}_i'$  is a dimension-oriented Boolean expression of  $\mathbf{p}_i$ , which can be obtained by incorporating features and changing the type of features to Boolean. Formula (2) shows each element  $p_{ik}'$  of  $\mathbf{p}_i'$ . In formula (2),  $\mathbf{p}_i \bar{U} \mathbf{p}_j$  represents a vector that contains all features belonging to the vector  $\mathbf{p}_i$  or  $\mathbf{p}_j$ . In the  $\mathbf{p}_i \bar{U} \mathbf{p}_j$ , elements are not equal to each other. The formula for each element  $p_{ik}'$  of  $\mathbf{p}_i'$  is as follows:

$$p_{ik}' = \begin{cases} 1, & \text{if } r_k \in \mathbf{p}_i, \\ 0, & \text{others.} \end{cases} \quad r_k \in \mathbf{p}_i \bar{U} \mathbf{p}_j \text{ and } k = 1 \dots \text{Count}(\mathbf{p}_i \bar{U} \mathbf{p}_j). \quad (2)$$

where  $r_k$  is an element of the  $\mathbf{p}_i \bar{U} \mathbf{p}_j$ . The function  $\text{Count}(\text{Vector } e)$  is used to calculate the count of the element in vector  $e$ . Then,  $\mathbf{p}_i'$  can be represented as  $\mathbf{p}_i' = (p_{i1}', p_{i2}', \dots, p_{ik}')$ .

## 4 Detect the Objects' Clustering Directions

We adopt characteristics of a social network to determine the clustering directions for each object. We use game theory to quantify the three characteristics of social networks. We take four steps to detect the clustering directions.

To calculate the decision for random walking steps, we utilize a possibility function proposed by Xin et al. [20]. They designed the random walk process as follows:  $a \in (0,1)$ ,  $f(x)$ , where  $x$  is the current step of the walker and  $f(x)$  is the possibility function for deciding whether to continue walking or not. At the current walking step, if  $a < f(x)$ , the walker performs the next step, otherwise, terminates walking. The random walk possibility function is the following:

$$f(x) = \frac{1}{\exp((x - \frac{1}{\eta d^\omega + 1})^\delta - \beta) + 1} \quad (3)$$

Where  $\delta$  controls the distribution of the walking steps, and  $\beta$  controls the furthest walking steps. Because of the 'Six Degrees of Separation', the diameter

of the community is less than 6. Thus, the value of  $\beta$  should be 6. The formula  $x - \frac{1}{\eta d^\omega + 1}$  is a topology gain function based on the logistic function. In formula (3),  $d$  is the degree of the object.  $\omega$  and  $\eta$  are the adjustable parameters of the logistic function.

In the second step, we adopt the game theory to quantify the importance of objects to the original object. Additionally, we can obtain more useful objects for the origin object by this method. The random walking earning and penalty function is as follows:

$$S_{u_k} = \varphi \left( \sum w_{ij} - \beta' * S_{loss} \right) + (1 - \varphi) \left( 1 - \frac{Path_{min}(u_0, u_{ter})}{\beta} \right) \quad (4)$$

$$S_{loss} = 1 - \frac{\sum \sum w_{kl}}{2n} \quad u_k, u_l \in O \wedge k \neq l \quad (5)$$

Where  $S_{u_k}$  is the actual score of  $u_k$  that is one of the neighbors of the origin object.  $\beta'$  represents a practical walking step. The  $u_0$  and  $u_{ter}$  represent the origin object and the terminal object for the walker, respectively. To prevent the walker from having repellency back to the origin object  $u_0$ , we design the formula,  $1 - \frac{Path_{min}(u_0, u_{ter})}{\beta}$ , to solve the problem. It means that if the shortest distance between the origin object and the terminal object is large, the walker has a low probability of returning to the origin object. That is the terminal object has little significance to the origin object. The function  $Path_{min}(u_0, u_{ter})$  is designed to calculate the shortest distance between  $u_0$  and  $u_{ter}$ . The parameter  $\varphi$  measures the importance of the similarity between objects and the walking trend of the walker. The formula (5) is a punishment value of each step.

In the third step, we utilize the results of  $N$  times random walking to find the clustering directions for each object. First, we determine one of the clustering directions  $u_k$  for the origin object  $u_0$  by the following formula:

$$R(u_0) = \{u_k | u_k = \arg \max(\overline{S_{u_k}}) \wedge u_k \in Ne(u_0)\} \quad (6)$$

Where  $R(u_0)$  represents the set of the clustering directions for object  $u_0$ , and  $\overline{S_{u_k}}$  represents the average score of  $n$  ( $n \leq N$ ) times random walking for the walker through  $u_k$ .  $Ne(u_0)$  denotes the set of neighbors of  $u_0$ . Formula (6) means that the object  $u_k$ , which has a maximal score, is one of the clustering directions for object  $u_0$ . Specially, objects that have a maximal average score of more than one are all the clustering directions of the object  $u_0$ .

In the fourth step, we will find additional clustering directions for object  $u_0$ . If object  $u_{k'}$  has a score which is close to the maximal average score, we would determine that  $u_{k'}$  is a clustering direction of object  $u_0$ . The clustering directions of object  $u_0$  can be updated by the following formula:

$$R(u_0) = \{u_k, u_{k'}, u_{k''}, \dots | u_k = \arg \max \overline{S_{u_k}}, (u_k - u_{k'}) < \varepsilon, (u_k - u_{k''}) < \varepsilon, \dots\} \quad (7)$$

Where the parameter  $\varepsilon$  is a small value. This shows that if the objects have a score close to object  $u_k$ , these objects would be the clustering directions of the object  $u_0$ .

## 5 Detect Community

After detecting the clustering directions for each object, we will detect community by two steps.

In the first step of detecting community, we will find some small communities by analyzing the similarity and clustering directions of objects. First, we define two sets ( $m\_Set$  and  $s\_Set$ ) to show the clustering directions of two objects. If two objects have a mutual clustering direction, they belong to the  $m\_Set$ . However, the two objects will belong to the  $s\_Set$  if they have a single-track clustering direction. Then, we sort the two sets according to the similarity of the two objects. Finally, if two objects in the  $m\_Set$  or  $s\_Set$  do not belong to any community, we will create a new community for them. In this process, we first consider the clustering directions in  $m\_Set$ , and those in  $s\_Set$  are reviewed later. Because the two objects, who have mutual clustering directions and a higher similarity, are more interested in a common community.

In the second step of detecting community, the target communities will be detected by considering the topological structure. First, if an element of the  $m\_Set$  and  $s\_Set$  has been formed into a small community, we delete it from the corresponding set. Each element in the two updated sets must contain an object  $u_i$  who already belongs to a community and another object  $u_j$  who is not member of any community. Then we will detect community for object  $u_j$  by analyzing the change of the topological structure. The modularity [21] has the unique privilege of also being a global criterion to assess the compactness of the community structure. The formula of modularity is shown in formula (8). Arab et al. [22] studied the modularity [21] in more depth. They [22] analyzed the change of the modularity when two communities unite to one. Therefore, we use the change of modularity as an evaluation criteria for the change of the topological structure when two communities unite into one. The formula for the change of the modularity is as follows:

$$Q = \sum_i (e_{ii} - a_i^2) \quad (8)$$

$$\Delta Q = \frac{E_{ij}}{m} - 2a_i a_j \quad (9)$$

Where  $Q$  is the modularity of a network,  $i$  is the number of a community,  $e_{ii}$  is the fraction of edges in community  $i$ ,  $a_i$  is the fraction of edges that connect two vertices in community  $c_i$ , and  $\Delta Q$  represents the change in the modularity value when two communities unite to a new community. The number of edges between the two communities,  $c_i$  and  $c_j$ , are denoted as  $E_{ij}$ . The edges of the entire social network are denoted as  $m$ . Assuming  $d_v$  is the degree of node  $v$ ,  $a_i$  can be presented as  $a_i = \frac{\sum_{v \in c_i} d_v}{\sum_{v \in MSN} d_v}$ . Based on the  $m\_Set$  and the  $s\_Set$ , we will merge communities if  $\Delta Q > \kappa (\kappa > 0)$ . The parameter  $\kappa$  is a threshold for  $\Delta Q$ . We merge communities following the decrease in  $\Delta Q$  until the merging condition cannot be met.

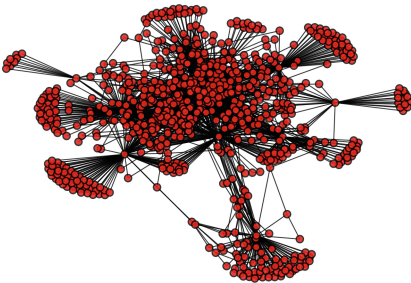
## 6 Experiments

Two datasets which are labeled *Dataset1* and *Dataset2* are subject to experimentation in this study. We obtain the *Dataset1* from Sina micro-blog open platform and the *Dataset2* from the Micro-blog User Portrait Contest in 2016. Figures 2 and 3 show the network structure distribution of two datasets. The detailed information of the two datasets is shown in Table 1.

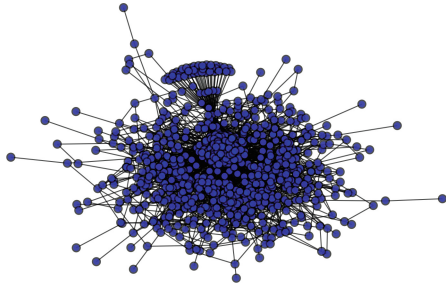
Our method is implemented in Python. We conduct our experiments on two datasets and contrast the results with other proposed algorithms, such as the CNM approach [17], Infomap [23], COPRA [24] and NRW [20]. The CNM approach [17] is a popular community detection approach and is currently widely used. Hence, we utilize CNM as one of the contrast approaches. The Infomap [23] and NRW [20] methods are both random-walks methods based on the community structure. Therefore, we chose them be our additional contrast approaches. Furthermore, our method not only considers community structure, but detects communities by analyzing micro-blog content. Therefore, we will analyze the performance of our method from two aspects: the community structure and community interest cohesion.

**Table 1.** The two datasets for the experiments.

Description	<i>Dataset1</i> (Size)	<i>Dataset2</i> (Size)
Followers and followees	311361	321712
User posts	43721	49921
Micro-blog users	781	980



**Fig. 2.** The network structure distribution of *Dataset1*.



**Fig. 3.** The network structure distribution of *Dataset2*.

### 6.1 Evaluate by the Interest Cohesion

The objects in a common community have a highly similar interest in general. Thus, we contrast our method with other methods using an evaluating indicator,

e.g., interest cohesion. The interest cohesion  $coi(c_i)$  of a community  $c_i$  can be calculated by formula (10).

$$coi(c_i) = \frac{\sum_{u_j \in c_i, u_k \in c_i} w_{jk}}{\sum_{u_j \in MSN, u_k \in MSN} w_{jk}} \quad (u_j \in Ne(u_k)) \quad (10)$$

Table 2 shows the comparison of the interest cohesion for the methods mentioned above. In Table 2,  $Num(Cs)$  represents the number of communities. Our method produces a  $Num(Cs)$  value close to the NRW and Infomap methods. Because these three methods are all based on the random walking method. The  $Num(Os')$  denotes the number of objects in a community that has a maximal number of objects, and  $Os' \in \max Num(Cs)$ . From the values of  $Num(Os')$ , we can see that the CNM method has maximal number of objects in a community than other methods. The  $\sqrt{s(coi(c_i))}$  represents the statistical dispersion of the interest cohesion. The maximal values of  $Num(Os')$  and  $\sqrt{s(coi(c_i))}$  show that the communities detected by the CNM method have an unbalanced distribution, i.e., there is a community  $c_i$  far greater than another community  $c_j$  from the number of objects. In contrast, our method has a lower value in  $\sqrt{s(coi(c_i))}$ . Thus, our communities have a balanced distribution. The  $\overline{coi(c_i)}$  is an average interest cohesion and measures the goodness of communities from their interest similarity. Our method has the highest similarity value, because our method considers interest similarity when it detects a community. The interest cohesion of communities in *Dataset2* is higher than in *Dataset1*, because *Dataset2* is pre-processed artificially.

**Table 2.** The comparison of the community interest cohesion for the methods mentioned above. In the  $Num(Os')$ ,  $Os' \in \max Num(Cs)$ , the  $Num(Os')$  denotes the number of objects in a community that has a maximal number of objects.

	<i>Dataset1</i>				
	CNM [17]	COPRA [24]	Infomap [23]	NRW [20]	Our method
$Num(Cs)$	37	89	51	58	64
$Num(Os')$	213	80	97	105	81
$\overline{coi(c_i)} (\times 10^{-2})$	0.5557	0.1217	0.6863	0.5388	0.7227
$\sqrt{s(coi(c_i))} (\times 10^{-2})$	001.1024	0.0351	0.5545	0.5387	0.0295
	<i>Dataset2</i>				
	CNM [17]	COPRA [24]	Infomap [23]	NRW [20]	Our method
$Num(Cs)$	28	65	49	39	46
$Num(Os')$	271	117	178	137	142
$\overline{coi(c_i)} (\times 10^{-2})$	0.9212	0.5667	0.8333	0.8921	1.1721
$\sqrt{s(coi(c_i))} (\times 10^{-2})$	0.7291	0.0313	0.4923	0.5127	0.0301

We sort the communities following the number of objects to better compare the methods mentioned above. Figure 4(a) and (b) show the distribution the



objects' number in the top-ten ranked communities. In the CNM method, the number of objects in communities has an unbalanced distribution. The number of objects in the first-ranked community is roughly twice as high as that in the second-ranked community. The other four methods have a similar distribution of the number of objects. Figure 5(a) and (b) show the distribution of the interest cohesion in the top-ten ranked communities. We can find that the community interest cohesion in our method is greater than the other methods. The communities with a large number of objects have high interest cohesion. The first-ranked community in our method has a smaller number of users than that in the CNM method. However, the first-ranked community in our method has higher interest cohesion than that in the CNM method. This means that the objects in communities in our method have higher interest cohesion.

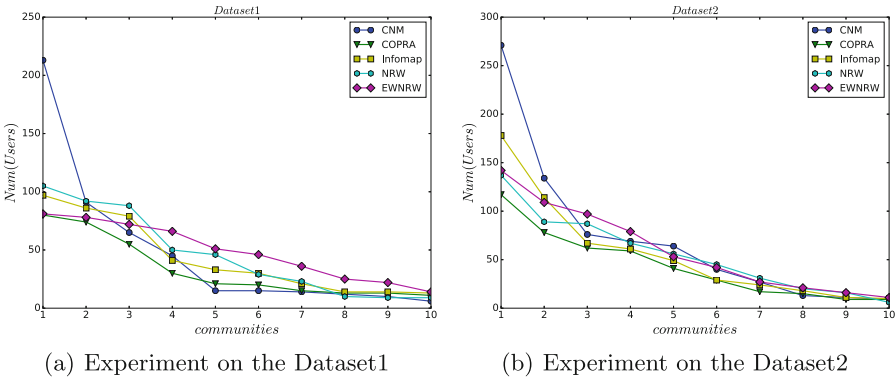
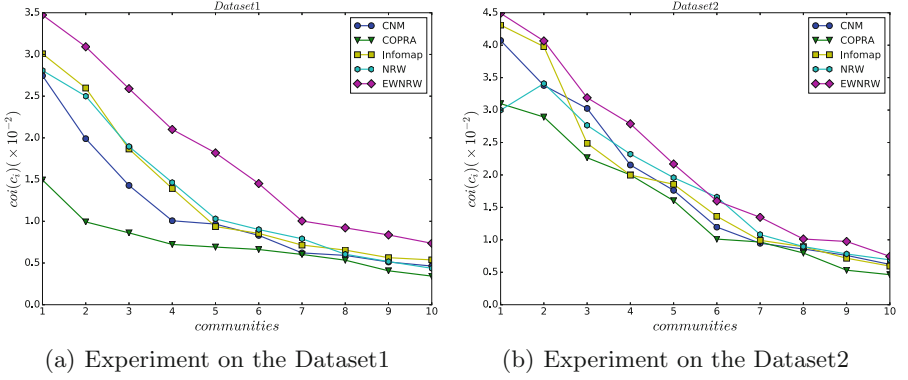


Fig. 4. Distribution of the number of objects in the top-ten ranked communities.

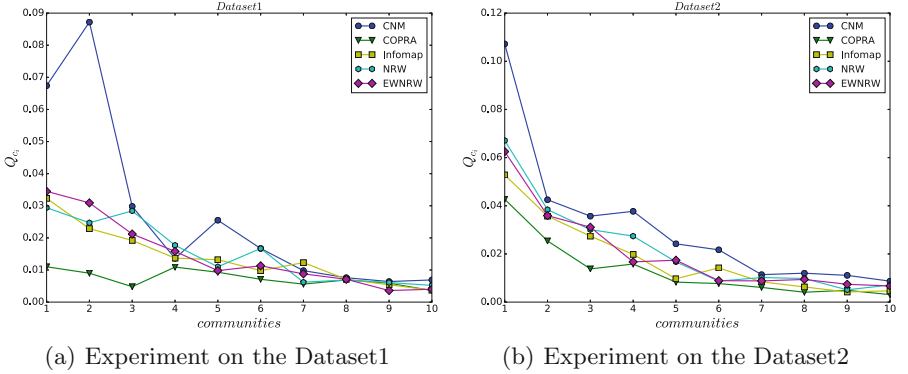
## 6.2 Evaluate by the Community Structure

Newman et al. [12] proposed a quality measure called modularity  $Q$  to quantify the community quality of a network or graph structure. The higher the value of modularity  $Q$ , the stronger the community structure is. Therefore, we adopt the modularity  $Q$  to analyze the performance of our method. The formula of modularity is shown in formula (8).

Table 3 shows the comparison of the community modularity for the methods mentioned above. In Table 3, the CNM method has a maximal modularity for the entire network and has a maximal single community modularity. This demonstrates that the objects are densely connected in an intra-community from the community structure. Moreover, our method has an approximate modularity with other methods, and thus, our method has good performance in the community structure. Figure 6(a) and (b) show the distribution of the modularity in the top-ten ranked communities. We find that the CNM method has the largest modularity and our method has high modularity. Therefore, our method is effective in detecting communities from the community structure.



**Fig. 5.** Distribution of the community interest cohesion in the top-ten ranked communities.



**Fig. 6.** Distribution of the modularity in the top-ten ranked communities.

**Table 3.** The comparison of community modularity for the methods mentioned above.

	<i>Dataset1</i>				
	CNM [17]	COPRA [24]	Infomap [23]	NRW [20]	Our method
$\max Q_{c_i}$	0.0872	0.0110	0.0323	0.0294	0.0345
$\sum_i Q_{c_i}$	0.3680	0.1667	0.1912	0.2459	0.2572
	<i>Dataset2</i>				
	CNM [17]	COPRA [24]	Infomap [23]	NRW [20]	Our method
$\max Q_{c_i}$	0.1072	0.0467	0.0528	0.0671	0.0625
$\sum_i Q_{c_i}$	0.5124	0.3023	0.3741	0.3908	0.4099

## 7 Conclusion and Future Work

In this paper, we deal with Micro-blog community detection problem from the topological structure and the Micro-blog content. We let user be object and calculate the interest similarity by the object's attribute set. Moreover, community structure is an important aspect in a network. Thus, we detect community by considering the community structure as well. First, we build the MSN by analyzing the objects' attributes and relationships. Then, we detect clustering directions. Finally, we detect community by the clustering directions and community structure. The experimental results show that our method has an advantage in detecting social network communities. In future work, we will use our method on additional data sets and verify that our method is superior for detecting social network communities.

**Acknowledgments.** This work is supported by the National Nature Science Foundation (Grant No. 61271413, 61472329 and 61532009) and the Innovation Fund of Postgraduate, Xihua University.

## References

1. Yan, Q., Wu, L.R., Zheng, L.: Social network based microblog user behavior analysis. *Physica A* **392**, 1712–1723 (2013)
2. Carter, S., Weerkamp, W., Tsagkias, M.: Microblog language identification: overcoming the limitations of short, unedited and idiomatic text. *Lang. Res. Eval.* **47**, 195–215 (2013)
3. Yan, Q., Wu, L.R., Liu, C., Li, X.Y.: Information propagation in online social network based on human dynamics. *Abstr. Appl. Anal.* **2013**, 173–186 (2013)
4. Van Laarhoven, T., Marchiori, E.: Local network community detection with continuous optimization of conductance and weighted kernel K-means. *J. Mach. Learn. Res.* **17**, 1–28 (2016)
5. Zhang, P., Moore, C., Newman, M.E.J.: Community detection in networks with unequal groups. *Phys. Rev. E* **93**, 012303 (2016)
6. Palsetia, D., Hendrix, W., Lee, S., Agrawal, A., Liao, W., Choudhary, A.: Parallel community detection algorithm using a data partitioning strategy with pairwise subdomain duplication. In: Kunkel, J.M., Balaji, P., Dongarra, J. (eds.) *ISC High Performance 2016*. LNCS, vol. 9697, pp. 98–115. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-41321-1\\_6](https://doi.org/10.1007/978-3-319-41321-1_6)
7. Guo, W.F., Zhang, S.W.: A general method of community detection by identifying community centers with affinity propagation. *Physica A* **447**, 508–519 (2016)
8. Fiedler, M.: Algebraic connectivity of graphs. *Czech. Math. J.* **23**, 298–305 (1973)
9. Kernighan, B.W., Lin, S.: An efficient heuristic procedure for partitioning graphs. *Bell Syst. Tech. J.* **49**, 291–307 (1970)
10. Brandes, U., Delling, D., Gaertler, M., Görke, R., Hofer, M., Nikoloski, Z., Wagner, D.: On finding graph clusterings with maximum modularity. In: Brandstädt, A., Kratsch, D., Müller, H. (eds.) *WG 2007*. LNCS, vol. 4769, pp. 121–132. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-74839-7\\_12](https://doi.org/10.1007/978-3-540-74839-7_12)

11. Li, Y.D., Liu, J., Liu, C.L.: A comparative analysis of evolutionary and memetic algorithms for community detection from signed social networks. *Soft. Comput.* **18**, 329–348 (2014)
12. Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. *Phys. Rev. E* **69**, 026113 (2004)
13. Wu, J.S., Wang, F., Xiang, P.: Automatic network clustering via density-constrained optimization with grouping operator. *Appl. Soft Comput.* **38**, 606–616 (2016)
14. Atay, Y., Koc, I.: Community detection from biological and social networks: a comparative analysis of metaheuristic algorithms. *Appl. Soft Comput.* **50**, 194–211 (2016)
15. Mu, Ca.H., Xie, J., Liu, Y.: Memetic algorithm with simulated annealing strategy and tightness greedy optimization for community detection in networks. *Appl. Soft Comput.* **34**, 485–501 (2015)
16. Zou, F., Chen, D.B.: Community detection in complex networks: multi-objective discrete backtracking search optimization algorithm with decomposition. *Appl. Soft Comput.* **53**, 285–295 (2017)
17. Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. *Phys. Rev. E* **70**, 066111/1–066111/6 (2004)
18. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E Stat. Nonl. Soft Matter Phys.* **76**, 036106 (2007)
19. Gilbert, J.R., Reinhardt, S., Shah, V.B.: High-performance graph algorithms from parallel sparse matrices. In: Kågström, B., Elmroth, E., Dongarra, J., Waśniewski, J. (eds.) *PARA 2006*. LNCS, vol. 4699, pp. 260–269. Springer, Heidelberg (2007). [https://doi.org/10.1007/978-3-540-75755-9\\_32](https://doi.org/10.1007/978-3-540-75755-9_32)
20. Xin, Y., Xie, Z.Q., Yang, J.: The adaptive dynamic community detection algorithm based on the non-homogeneous random walking. *Physica A* **450**, 241–252 (2016)
21. Newman, M.E.J.: Modularity and community structure in networks. *Proc. Natl. Acad. Sci.* **103**, 8577–8582 (2006)
22. Arab, M., Afsharchi, M.: Community detection in social networks using hybrid merging of sub-communities. *J. Netw. Comput. Appl.* **40**, 73–84 (2014)
23. Rosvall, M.: Maps of random walks on complex networks reveal community structure. *Proc. Natl. Acad. Sci.* **105**, 1118–1123 (2008)
24. Gregory, S.: Finding overlapping communities in networks by label propagation. *New J. Phys.* **12**, 2011–2024 (2010)