International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.5 (2014), pp.131-142
http://dx.doi.org/10.14257/ijmue.2014.9.5.12

An Improved Random Walk Based Community Detection Algorithm

Daxiang Ji, Yuqing Sun* and Demin Li

School of Computer Science and Technology
Shandong University, Jinan 250100
jidaxiang1990@163.com, sun_yuqing@sdu.edu.cn , lidemin1014@gmail.com

Abstract

Community detection is an important issue in social network analysis, which aims at finding potential community structures such that the internal nodes of a community have higher closeness than external nodes. Taking into account node attribute information, this paper presents an improved community detection algorithm based on random walk. Based on the basic understanding that people getting together often relies on their common interests, node similarities are initially calculated with node attributes and iteratively updated based on the random walk model. Meanwhile, node importance is computed to represent how much it can influence other nodes, based on which some important nodes are selected as seeds for community clustering. As for overlapping community detection, some construction is made on a given social network. Experimental results on several real datasets show our approach has better effects than previous methods on both overlapping and non- overlapping communities.

Key words: *social network; community detection; random walk*

1. Introduction

Social network is a collection of individuals or organizations as well as the links between them, in which each node represents an individual and each link between two nodes denotes their relationship. Social network analysis has emerged as a key technique in many areas, such as biology, economics and etc. A key task of social network analysis is to find community structure, which is quite common in real networks, and being able to identify communities within a network can provide insight into how network function and topology affect each other.

Modularity is a common criterion to evaluate community structure by the comparison between the actual density of edges within a community and the density one would expect to have in the community if nodes were attached without community structure [1]. Typical community detection algorithms based on optimization of modularity iteratively divide nodes into communities until modularity convergence and achieve community structure with a high modularity [1, 2]. These methods allow nodes to belong to only one community, namely non-overlapping communities, which cannot reflect the reality in practical social networks.

Overlapping community reveals the characteristics of multiple memberships of nodes in communities, and thus can better reflect the real community structure. Some overlapping community detection algorithms have been proposed [3-5] and the extended modularity based on the spirit of general modularity is usually used to evaluate overlapping communities by dividing the contribution of node to modularity by the number of communities it belongs to [6]. However, most of these algorithms ignore node attributes which is highly related to which community a node belongs to.

Some algorithms considering node attributes are proposed to detect communities in social network, such as the SA-Cluster algorithm (Clustering Based on Structural/Attribute)[7].The SA-Cluster have better results on community structure than previous methods but it is not suitable for a complex social network with multiple attributes.

ISSN: 1975-0080 IJMUE

In this paper, we take into account both node attributes and node links in social networks. By expressing node attributes into standardized vectors, the node similarity is calculated to represents how many common characteristics these nodes have. Adopting the random walk model, we calculate the node closeness matrix and node importance which respectively reflect the closeness of nodes and how much one node can influence others. To solve the overlapping community problem, we make some construction on a given social network. Finally, some important nodes are selected as initial community seeds for clustering to find communities.

Rest of this paper is organized as follows. Chapter 2 surveys related work. In Chapter 3, we present the details of the proposed community detection algorithm. Experimental evaluation are discussed in chapter 4 and conclusions are given in Chapter 5.

2. Related Work

2.1. Non-overlapping Community Detection Algorithms

The typical non-overlapping community detection algorithm is the Girvan and Newman algorithm (GN algorithm), which detects communities by literately deleting the edges with the largest number of intermediaries [1]. Although its results achieve high value of modularity, it is difficult to apply in a large-sized network due to its high computational complexity.

Another representative method is aggregation-based community detection. Newman fast algorithm treats each node as a community and merges communities iteratively on the condition that two communities are capable of producing maximum modularity [2]. Lai *et al.* pretreat the network using random walk model and uses polymerization method to detect communities, which achieves high modularity [8].Yang *et al.* propose SA-Cluster (Clustering Based on Structural/Attribute) which regards each possible value of attributes as a node in the graph. Edges between the node and the corresponding attribute nodes are added to the graph. Then, a clustering process is performed using random-walk-based node similarity [7]. The Inc-Cluster (incremental SA-Cluster) improve SA-Cluster by computing node similarity gradually according to the change of attribute weight [9]. Such approaches take into account node attributes in a novel way but the augmented graph will become much more complex when the number of attribute values is large, so they are not suitable for complex social networks.

Algorithms based on the Extreme Optimization regards the influence of each node on modularity as a local variable. On the basis of random division, it adjusts local variables to improve global modularity using greedy strategy [10].The Kernighan_Lin algorithm initially defines the difference between the edge number inside communities and that between communities as a gain function and randomly divides nodes into two communities [11]. Then nodes are exchanged based on whether the change will increase the value of gain function until traverse all node pairs. However, the number of nodes contained in a community should be acknowledged in advance and they are not suitable for overlapping community detection.

2.2. Overlapping Community Detection Algorithms

Typical overlapping community detection algorithms are the clique penetration algorithm and edge clustering [3, 5].The clique penetration algorithm regards communities as a series of k-group (complete sub graph of size k) in which nodes are reachable from each other. The community structure is found by merging adjacent k-groups. However, when networks are sparse, these methods can only find a portion of overlapping communities. Edge clustering algorithm constructs a dual graph by mapping each edge in the original graph to a node in the dual graph, and communities is disclosed by existing non-overlapping community algorithms on the dual graph. Anh clusters tree diagram using hierarchical methods which regards community as a closely-linked edge set [12].Edges connecting nodes in different communities are overlapping, but they will be assigned just to one cluster, which may not well reflect the real community structure in practice.

Based on seed expansion, LMF algorithm use several nodes as seeds, and cover the entire network through expansion [4].It defines two objective functions: community fitness and node fitness to community. Initially, each node is regarded as a community. Then, it add an adjacent node with largest fitness to the community, recalculate current fitness of each node in each community and remove nodes with negative fitness. Repeat the process until fitness of all community neighbors become negative. However, it is difficult to apply if the size of target communities is of the same order of magnitude with the size of the given network.

Some methods reduce overlapping community detection problem to traditional methods, such as non-negative matrix factorization and fuzzy clustering [13, 14]. However, it is difficult to select a feature matrix to represent the inherent topology or determine how to construct the distance matrix in practice. In addition, there's selection method classifies networks into four categories and selects different detection algorithms based on the characteristics of the given network [15]. Ruan's algorithms delete (add) edges from (to) the network according to the semaphore obtained by mixing attribute similarity and link similarity and use existing community detection algorithms to find community structure [16]. By preprocessing the network, these algorithms can achieve more realistic results than previous methods, but this will lead to a negative impact on the performance of the algorithm.

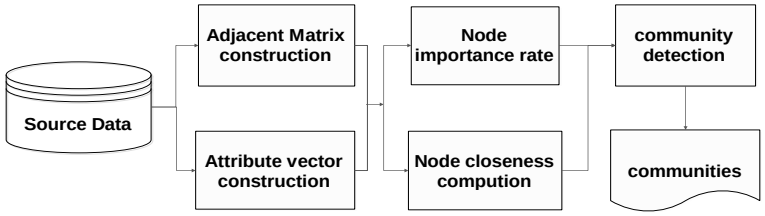


Figure 1. Framework of ARCD

3. Community Detection based on Random Walk and Node Attributes

In this section, we present the Attributes-Based Random Walk Community Detection framework, shown as Figure 1. At first, we denote the network as a graph *G (V, E, A)*, where *V* is the collection of nodes in the network, *E* is the set of edges represented as the adjacent matrix, and $A= (a_1, a_2, \ldots, a_n)$ is the set of *n* attribute vectors associated with *n* nodes in the network. Secondly, we iteratively calculate similarity between nodes based on the attribute vectors. Combined with random walk model, the importance of each node is computed to represent how much it can influence other nodes. In the third stage, a clustering process using some important nodes as initial centers is performed to find communities.

3.1. Attributes Processing and Node Rating

As mentioned before, node attributes are important factors when clustering nodes in a social network. Since node attributes in practical social networks are described in different ways and their semantics depend on context, standard attribute vectors are required to be established for node similarity computation. In this paper, we discuss how to generate and standardize the attribute vector for two representative classes of data: unstructured data and structured data.

For unstructured data, such as web-based social network data, the content includes HTML tags, text, images, and client scripts etc. We adopt the well-known model, Bag-of-Words (BoW), to establish node attribute vectors using the information on the pages of social network users. BoW models a web document as a collection of words contained in the document and the appearance of each word is independent. For two web documents, we build a dictionary according to the words appeared in the documents, and each word in the

dictionary has a unique index. Then, two documents can be express into two vectors with a same dimension.

Attribute vector of structured data is associated with data type, such as integer, text, etc. For the standardization purpose, different data types need different operations. For example, if the data is ordinal, we map the different discrete values to a discrete set of values, such as {good, better, best} corresponding to {0, 0.5, 1}.As for numerical data, we reduce it to a decimal in [0, 1] using $a_{ik} = (a_{ik} - \min)/(max -$, where a_{ik} is the numerical data and *min/max* is the possible minimum/maximum value in the value range of a_{ik}. After each column of node attribute vector is standardized, it can be used to calculate node attribute similarity.

For the node attribute vectors, we use the well-known cosine similarity to denote nodes attribute similarity as defined in equation (1).

$$\mathrm{sim}(v_1, v_2) = \frac{a_1 * a_2}{||a_1|| * ||a_2||} = \frac{\sum_{i=1}^{n} a_{1i} * a_{2i}}{\sqrt{\sum_{i=1}^{n} a_{1i}^2} * \sqrt{\sum_{i=}^{n}}} \quad (1)$$

wherev_1 and $v_2$are two nodes in a social network, a_1 and $a_2$represent their attribute vectors respectively, a_{1i} anda_{2i} are respectively the i^{th} value in a_1 and a_2.

Based on the random walk model, node importance is calculated as follow:

$$\mathrm{VI}(v) = c * \frac{1}{N} + (1 - c) * \sum_{p_i \in p} \mathrm{VI}(p_i) * \left(\frac{\mathrm{sim}(p_j, v}{\sum_{p_i \to p_j} \mathrm{sim}(} \right. \quad (2)$$

Where *N* denotes the number of nodes in the network, *c* is the restart coefficient for random walk, *P* denotes the neighborhood set of *v* and *sim* is the aforementioned similarity function. Calculating node importance is an iterative process until the change of each node importance is lower than a predefined parameter *d*. Important users are usually active than other users in social networks so they can be used as the initial clustering centers.

3.2. Community Detection Algorithm

We propose Attributes-Based Random Walk Community Detection Algorithm (ARCD), including three stages: the preprocessing stage, the graph reconstruction phase, and the clustering stage, as shown in Algorithm 1.

Algorithm1. Attributes-Based Random Walk Community Detection Algorithm

Input: Graph *G=(V,E,A)*, number of communities: ,active node threshold: ,largest number of communities a node belongs to: $\boldsymbol{M}$, restart coefficient: $\boldsymbol{c}$, the maximum steps: $\boldsymbol{L}$

Output: A community set $\boldsymbol{C} = \{\boldsymbol{C_1}, \boldsymbol{C_2}, \ldots, \boldsymbol{C_k}\}$

PART Ⅰ Pre_Process

1. Initialize() //initialize parameters
2. Attributes_Process() // standardize attribute vectors
3. For each node $\boldsymbol{v_i}$ //calculate attribute similarity
4. For each node $\boldsymbol{v_j}$
5. If($\boldsymbol{v_i}$!) $\boldsymbol{M}(\boldsymbol{v_i}, \boldsymbol{v_j}) = \mathrm{sim}(\boldsymbol{v}$
6. Calculate Node Impact Vector R={ $\boldsymbol{r_1}$,$\boldsymbol{r_2}$,…,$\boldsymbol{r_{|V|}}$ } and Closeness Matrix $\boldsymbol{D}$

PART Ⅱ Graph_ReBuild

7. Select $\boldsymbol{k}$ nodes with highest as Center Set
8. For each node $\boldsymbol{v}$
9. For each $\boldsymbol{s_i}$
10. If($\boldsymbol{D}(\boldsymbol{v},\boldsymbol{s_i})$>) and (Count($\boldsymbol{v}$)<$\boldsymbol{maxG}$)))

11. Add a fake node***v'*** of active node ***v*** to ***V*** and
12. Add edges between ***v'*** and neighbors of ***v***
13. For each node $\check{v} \in$ Neighbor(***v***) Set ***D(v', $\check{v}$)***= ***D(v, $\check{v}$)***and ***D(v,v')***=0

PART Ⅲ Community Detection

14. $C = \text{Cluster\_Algorithm}(S,$
15. Merge fake nodes with corresponding active nodesand remove corresponding edges
16. Return $C=\{C_1,C_2,\ldots,C_k\}$

In the preprocessing phase, we establish and standardize attribute vectors as described in Section 3.1.The similarity between nodes is calculated based on the standardized attribute vectors. Adopting the random walk model, the closeness matrix between nodes as well as the importance of nodes is calculated. The closeness between node v_i and v_j is denoted as the probability that a walker start from v_i and arrived at v_j after L jumps where L is the maximum step of random walk. During the process of random walk, a walker randomly selects a neighbor node of the current node as next station. The probability of selecting v_x depends on the similarity between current node and v_x, as shown in equation (3).

$$p(v_i \to v_x) = \alpha * \frac{sim(vi,vx)}{\sum_{v' \epsilon N(vi)} sim(vi,v')} + \beta * \frac{}{|} \quad (3)$$

Where $N(vi)$ is the neighbor collection of v_i; α and β are ratios of attribute similarity and link similarity, and $\alpha+\beta=1$. The closeness of $v_i$and v_j is defined as:

$$c(v_i, v_j) = \sum_{\substack{\tau:v_i \to v_j \\ length(\tau)<L}} p(\tau)c(1-c)^{|\tau|} \quad (4)$$

where is a path from to ;c is the restart coefficient, and L is the maximum step of random walk path.

Based on node closeness, the clustering process can generate non-overlapping communities. For the overlapping community case, we import the concept of active node which is allowed to belong to multiple communities corresponding fake nodes are added if the similarity to cluster center nodes are greater than a certain threshold θ and the similarities between the active node and corresponding fake nodes equal to 0. Parameter *maxG* is the max number of communities that a node is allowed to attend. The complexity of the graph reconstruction phase is related to the number of nodes n and the number of target communities k, that is $O(nk)$.

After the above construction, existing clustering algorithm can be used to get community structure with some important nodes selected as cluster seeds, which may contain the active nodes and their duplicates. After having the clustered results, we could merge those fake nodes with their corresponding active nodes, which maps to the overlapping community structure. The complexity of the clustering stage depends on the selected clustering algorithm.

The choice of parameters will influence the quality and efficiency of the algorithm. From a theoretical analysis, the larger active threshold θ is, the less copies of active node there will be, and that will lead to a community structure near non-overlapping. The larger *maxG* is, the more copies of active nodes there will be, which will lead to a better overlapping community structure. However, larger *maxG* will lead to more nodes in the graph, which will affect the efficiency of the algorithm. Likewise, the larger the number of communities is, the more complex the algorithm will be, while it will also lead to a better community structure.

3.2. Community Metric

Community detection is to find potential community structure between social network users, and the community metrics should objectively reflect the quality of community structure. Modularity is used to measure the non-overlapping community structures, which is able to quantitatively compare the dense degree of edges inside communities with that of edges outside communities [1]. Higher modularity represents a better the community

structure. Let element e_{ij} of matrix e be the fraction of edges in the network that connect nodes in community i and j. Let $a_i =$ and $Tre(e) =$. Modularity, denoted as Q, is calculated in equation (5):

$$Q = \sum_i (e_{ii} - a_i^2) = Tre - \quad (5)$$

As for overlapping community structure, an extended modularity (EQ) is proposed in reference [6], which divides the contribution of an edge to modularity by the number of communities containing its end-node, as shown in equation (6).

$$EQ = \frac{1}{2m} \sum_i \sum_{v \in C_i, w \in C_i} \frac{1}{O_v O_w} [A_{vw} - \frac{k_v k_w}{2m}] \quad (6)$$

Here, O_v is the number of communities containing node v; $k_v$is the degree of$v$; A is the adjacent matrix of the network, and m is the number of edges in the network. Higher value of EQ indicates a more significant overlapping community structure.

The above modularity is actually regarded as the Maximum Likelihood Estimator (MLE) of the random variables whether a node is the start or end of an edge in the given graph comparing with the random graph. In this paper, based on the same spirit of community evaluation, we proposed the extended weighted modularity (EWQ for short), as shown in equation (7).

$$EWQ = \frac{1}{2h} \sum_i \sum_{v \in C_i, w \in C_i} \frac{1}{O_v O_w} [M_{vw} - \quad (7)$$

Here, $s_v = \sum_{v_i \in N_v} sim($ is the sum of attribute similarity between v and its neighbors; M is the attribute similarity matrix, and $h = \sum_{v \in V} \sum_{v_i \in N_v} sim($ is the sum of M. EWQ is able to better evaluate overlapping community structure from the perspective of node attributes.

In addition, we adopt density and entropy to measure the community structure on structured datasets. Density evaluates community structure from the perspective of edge clustering.

$$density(\{V_i\}_{i=1}^k) = \sum_{i=1}^k \frac{|\{(v_p, v_q) | v_p, v_q \in V_i, (v_p, v_i}{|E|} \quad (8)$$

Here, V_i is a community, v_p and v_q are nodes in the network, and (v_p,v_q) denotes an edge between v_p and v_q. Entropy of attributes in one community V_i is defined as equation (9):

$$entropy(a_i, V_j) = -\sum_{k=1}^{n_i} p_{ijk} log_2 p_{ijk} \quad (9)$$

Where a_i is an attribute of nodes in V_j, n_i is the total number of possible values of a_i and p_{ijk} is the probability that the attribute value a_i of nodes in V_i equals to a_{ik} (k^{th} value of a_i).

Entropy of attributes in the whole community structure is defined as equation (10):

$$entropy(\{V_i\}_{i=1}^k) = \sum_{i=1}^m \frac{\omega_i}{\sum_{p=1}^m \omega_p} \sum_{j=1}^k \frac{|V_j|}{V} entropy(a \quad (10)$$

Where m is the number of attributes and ω_i denotes the weight of the ith attribute. Entropy reflects the distribution of node attribute values in different communities and lower entropy represents a more concentrated distribution of node attribute values.

4. Experiments and Analysis

In this section, we firstly introduce three real datasets used in our experiments. Then, we analyze the effectiveness and efficiency of our algorithm from two aspects. One is to evaluate how parameters affect the community results. Another is to make comparison with other representative methods.

4.1. Data Sets

We choose three representative real social network data sets in our experiments, namely *New movies*, *Citation* and *Polbolg*, which are widely used in the related works on social network analysis [17-19]. The corresponding parameters of these datasets are given in Table 1.

Table 1. Data Sets used for Community Detection

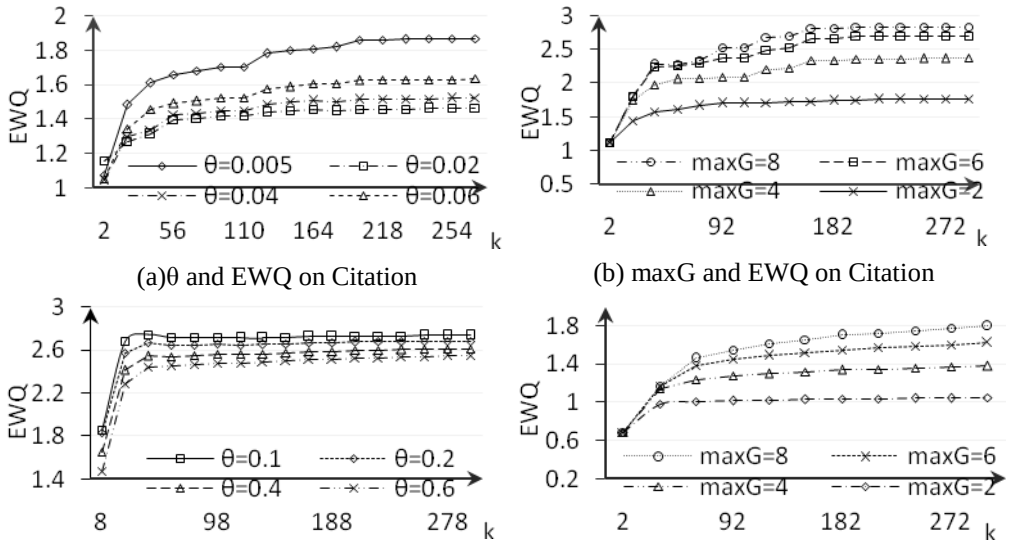
Data Set	Nodes	Edges	Attribute Description
New movies	34282	142427	name, profile, etc
Citation	2555	6101	title, author name, etc
Pol blogs	1490	19090	political preferences

The dataset *New movies* is obtained from Wikipedia and it is a textual dataset, including movies, actors, directors, authors and so on. *New movies* has 34282 nodes and 142427 edges in total, among which 16255 nodes represent people such as actors, directors and others famous movies. We choose 16255 nodes and 86336 edges concerned with people in the data set. Attribute description include name, status, and profile of each person. *Citation* is a structured dataset which contains attribute description of 2555 papers and 6101 correlations between them. Paper attribute description includes title, publication date, published journal, and author name, etc. The data set *Pol blogs* is downloaded from a data website of University of Michigan. It includes 1490 nodes represented for 1490 politicians and their blog relationships. Moreover, each node has a radical or conservative attribute. We modeled the above datasets as a graph *G(V,E,A)*.For *Citation*, paper set correspond to nodes set *V* of the graph *G*, correlations among papers correspond to edges set *E*, and attribute description of papers correspond to attribute information set *A*. For *New movies*, nodes set *V* of the graph *G* corresponds to the collection of actors, directors or authors, correlations among them correspond to the edge set *E*, and the collection of attribute description correspond to attribute information set *A*.

4.2. Discussion on Parameter Setting

In this subsection, we analyze how the performance of the proposed algorithm scales with the selection of parameters. There are three main parameters in our method, the threshold θ to determine whether a node is active, the maximum number *max G* of communities that each active node is allowed to attend and the number *k* of communities in the community structure. We refer to the experience in related work [7] when setting parameters *c* and *L*.

Figure 2 shows how parameters influence community quality. The X axis is the number of communities, the Y axis is community quality (in the form of EWQ). Setting c=0.15, L=4, *max G*=3, the results on *Citation* is shown in Figure 2(a).We find EWQ increases to a relatively stable state with *k* increasing and smaller threshold θ leads to a higher EWQ curve. When setting c=0.15, L=4, *max G*=3, we find the same tendency on *New movies* as shown in Figure 2(c). The parameter *max* G is also an important parameter in our algorithm. Setting threshold θ =0.005, c=0.15on data set *Citation*, we can see EWQ scales with the parameter *max G* as shown in figure 2(b). And on data set *New movies*, setting threshold θ=0.1, c=0.15, the relation between *max G* and EWQ is shown in figure 2(d). The results conform to our analysis that larger *max G* will lead to more fake nodes which achieves more overlaps in community structure.

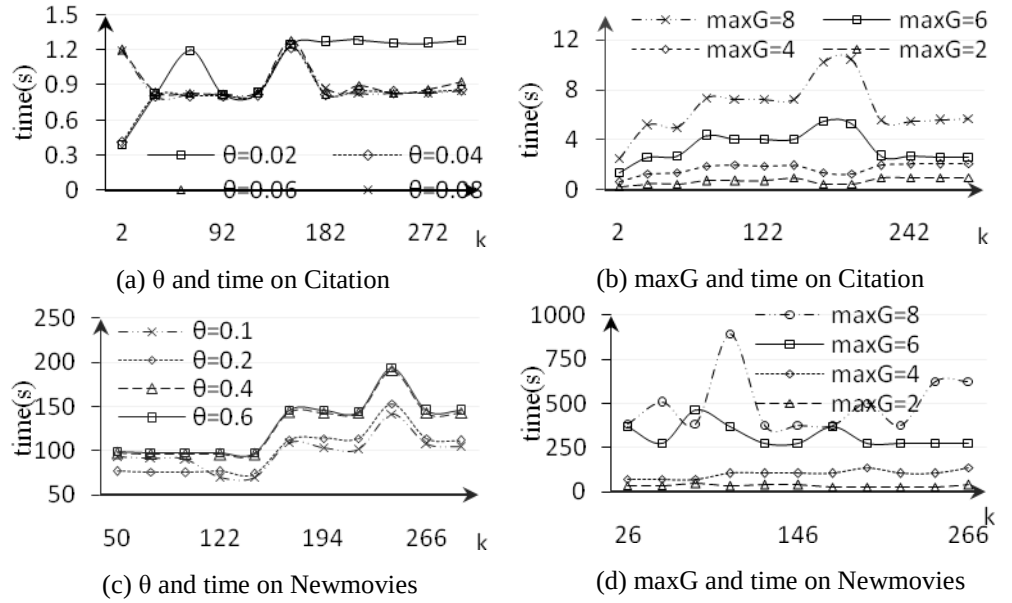


(a)θ and EWQ on Citation (b) maxG and EWQ on Citation

(c) θ and EWQ on NewMovies (d) maxG and EWQ on Newmovies

Figure 2. Community Quality Evaluation

Then, we analyze how parameters influence the performance of the ARCD algorithm. In Figure 3, the X axis is the number of communities, the Y axis is performance (in the form of running time) and each curve denotes different settings. Figure 3(a) and (c) respectively show how the running time scales with θ on *Citation* and *Newmovies*. We see that the running time increases small with the increasing of θ. From Figure 3(b) and (d) we can see that running time almost remains the same with the increase of k, and larger *max G* will lead to higher curve, which conforms to our analysis that larger *max G* will lead to more fake nodes in the reconstructed graph, so longer running time is needed.



(a) θ and time on Citation (b) maxG and time on Citation

(c) θ and time on Newmovies (d) maxG and time on Newmovies

Figure 3. Performance Evaluation

4. 3.Comparison with Related Methods

In this subsection, we firstly compare our algorithm with the state-of-the-art overlapping community detection algorithm LMF [4].When we using LFM to find communities on *New movies*, the running time exceeds the acceptable time. On *Citation*, let parameters c=0.15, L=4, *max G*=6, θ=0.005 in ARCD, and we get the comparison with the result of LFM as

shown in Table 2. From Table 2, we can see both performance and community quality of ARC Dare better than LMF.

Table 2. Performance of LMF and ARCD on Citation

Algorithm	EWQ	time
LMF	0.435	208.8s
ARCD	2.179	2.752s

Then, we compare our algorithm with the method which randomly selects cluster centers on *New movies*. The parameters in our experiments are chosen as follows: *max G*=4, θ=0.1, *c*=0.15 and *L*=4. The performance are shown in Figure 4.The results show that choosing important nodes as the seeds of clusters can obviously reduce the running time of the clustering process.

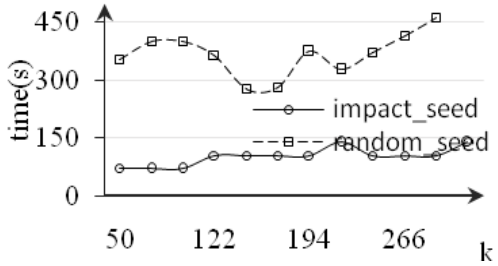
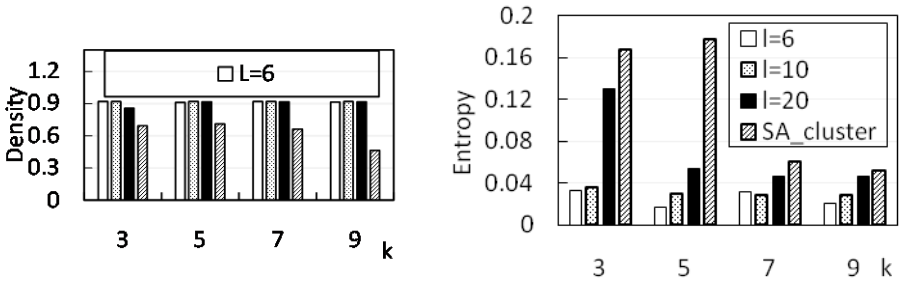


Figure 4. Performance of Different Seeds on New Movies

As for structured data, we compare our method with SA-Cluster [7] on *Pol bolgs*. In this experiment, we choose parameters as follow: *L*=6, 10 and 20, *c*=0.15, the attribute similarity weight α=0.5, the link similarity weight β=0.5, the number of communities *k* is 3, 5, 7 and 9, and each node belongs to one community. The result is shown in Figure 5, in which the X axis denotes the number of communities. According to Figure 5(a), we can see that ARCD finds community structure with a higher density than that of SA-Cluster algorithm. Meanwhile, we get statistical value of the entropy, as shown in Figure 5(b). Comparing to SA-Cluster, algorithm ARCD achieve lower entropies in condition of different walk step than that of SA_Cluster.



(a) comparison of density (b) comparison of entropy

Figure 5. Results on Pol Blogs

5. Conclusion and Future Work

The analysis of the community structure in social network has important practical significance on the economic, social security and other areas. In this paper, we consider both node attributes and node links to detect communities in social networks. We establish and standardize the node attribute vectors described in different ways. Using the standardized node attributes vectors, we use cosine similarity to denote similarity between nodes. Combined with random walk model, the closeness matrix between nodes and the importance

of nodes is calculated. For the problems of overlapping community detection, we reconstruct the given social network by adding some fake nodes to the network. Experimental results on several real data sets show the proposed approach have better effects than previous methods.

For the future work, we would analyze other information in social networks, such as how to improve the accuracy and efficiency of community detection by using users' action information. Another research direction is to integrate node attributes into the analysis on community structure evolution in social network.

Acknowledgements

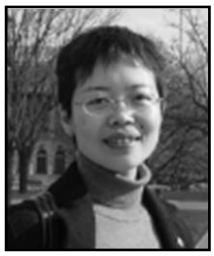
Part of this work is supported by National Natural Science Foundation of China (61173140), the National Science and Technology Pillar Program (2012BAF10B03-3).

References

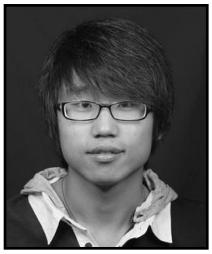
[1] M. E. J. Newman, "The structure and function of complex networks", SIAM Review, vol. 45, issue 2, (**2003**), pp. 167–256.
[2] M. E. J. Newman, "Fast Algorithm for Detecting Community Structure in Networks", Physical Review E, vol. 6, no. 69, (**2004**).
[3] G. Palla, I. Derenyi, I. Farkas and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society", Nature, vol. 435, no. 7043, (**2005**) June, pp. 814-818.
[4] A. Lancichinetti, S. Fortunato and J. Kertész, "Detecting the overlapping and hierarchical community structure in complex networks", New Journal of Physics, vol. 11, no. 3, (**2009**).
[5] T. Evans and R. Lambiotte, "Line Graphs, Link partitions, and Overlapping Community", Physics Review E, vol. 80, no. 1, (**2009**).
[6] H. Shen, X. Cheng, K. Cai and M. Hu, "Detect overlapping and hierarchical community structure in networks", Physica A: Statistical Mechanics and its Applications, vol. 388, no. 8, (**2008**).
[7] Y. Zhou, H. Cheng and J. Yu, "Graph Clustering Based on Structural/Attribute Similarities", VLDB, Lyon, France, (**2009**) August 24-28, pp.718-729.
[8] D. Lai, H. Lu and C. Nardini, "Enhanced Modularity-based Community Detection by Random Walk Network Preprocessing", Physical Review E, vol. 6, no. 81, (**2010**).
[9] H. Cheng, Y. Zhou, X. Huang and J. Yu, "Clustering large attributed information networks: an efficient incremental computing approach", Data Mining and Knowledge Discovery, (**2012**), pp. 219-242.
[10] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization", Physical Review E, vol. 72, 027104, (**2005**).
[11] Y. Y. Ahn, J. P. Bagro and S. Lehmann, "Link communities reveal multiscale complexity in networks", Nature, vol. 466, (**2010**) 05 August, pp. 761–764.
[12] T. A. Dang and E. Viennet, "Community Detection based on Structural and Attribute Similarities", The Sixth International Conference on Digital Society, Valencia, Spain, (**2012**), January 30-February 4, Article No. 10.
[13] S. Zhang, R. S. Wang and X. S. Zhang, "Uncovering Fuzzy Community Structure in Complex Networks", Phys. Rev. E, vol. 76, 046103.
[14] S. Zhang, R. Wang and X. Zhang, "Identification of Overlapping Community Structure in Complex Networks Using Fuzzy C-means Clustering", vol. 374, issue 1, (**2007**) 15 January, pp. 483–490.
[15] H. Elhadi and G. Agam, "Structure and Attributes Community Detection: Comparative Analysis of Composite", Ensemble and Selection Methods. SNA-KDD'13, August 11, Chicago, U.S. Article No. 10.
[16] Y. Ruan, F.David and S. Parthasarathy, "Efficient Community Detection in Large Networks using Content and Links", WWW, Rio de Janeiro, Brazil, (**2013**), 13-17 May, pp. 1089-1098.
[17] J. Tang, J. Sun, C. Wang and Z. Yang, "Social Influence Analysis in Large-scale Networks", KDD'09, (**2009**) June 28 - July 1, Paris, France. pp. 807-816.
[18] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang and Z. Su, "ArnetMiner: Extraction and Mining of Academic Social Networks", KDD'08, August 24 – 27, Las Vegas, Nevada, USA, pp. 990-998.
[19] L. Adamic and N. Glance, "The political blogosphere and the 2004 US Election", In Proceedings of the WWW-2005 Workshop on the Weblogging Ecosystem, Chiba, Japan, (**2005**) May 10-14, pp. 36 – 43.

Authors

DaxiangJi, he was born in 1990. He received his Master Degree Candidate of the School of Computer Science and Technology, Shandong University，China. Jidaxiang1990@163.com. His main research interests include social network analysis and data mining.



Yuqing Sun, she was born in 1967. She is a Professor of the School of Computer Science and Technology, Shandong University，China. Sun_yuqing@sdu.edu.cn. Her research interests include access control model and technology, security policy, privacy protection, data and application security, location based application and workflow management.



Demin Li, he was born in 1988. Master Degree Candidate of the School of Computer Science and Technology, Shandong University，China. lidemin1014@gmail.com. His main research interests include social network analysis and differential privacy.