

A Density-Based Approach for Mining Overlapping Communities from Social Network Interactions

Sajid Yousuf Bhat
Department of Computer Science
Jamia Millia Islamia (A Central University)
New Delhi-110025, India
s.yousuf.jmi@gmail.com

Muhammad Abulaish^{*}
Center of Excellence in Information Assurance
King Saud University
Riyadh-11653, Kingdom of Saudi Arabia
mAbulaish@ksu.edu.sa

ABSTRACT

In this paper, we propose a density-based community detection method, **CMiner**, which exploits the interaction graph of online social networks to identify overlapping community structures. Based on the average reciprocated interactions of a node in the network, a new distance function is defined to find the similarity between a pair of nodes. The proposed method also provides a basic solution for automatic determination of the neighborhood threshold, which is a non-trivial problem for applying density-based clustering methods. Considering the local neighborhood of a node p , the distance function is used to determine the distance between the node p and its neighbors in the interaction graph to identify core nodes, which are then used to define overlapping communities. On comparing the experimental results with clique percolation and other related methods, we found that **CMiner** is comparable to the state-of-the-art methods and is also computationally faster.

Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications—*Data mining*; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Clustering, Information filtering*

General Terms

Algorithms, Design

Keywords

Social network analysis; Density-based clustering; Community finding; Overlapping community detection

1. INTRODUCTION

Due to increasing popularity of the Online Social Networks (OSNs) and its wide application areas, community

^{*}To whom correspondence should be made

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WIMS '12, June 13-15, 2012 Craiova, Romania

Copyright © 2012 ACM 978-1-4503-0915-8/12/06 ...\$10.00.

mining research has received a lot of attention in recent past and the field is still rapidly evolving. Numerous methods based on spectral clustering [6, 22, 24], partitional clustering [17], modularity optimization [5, 19], likelihood [4], mathematical programming [1], and latent space clustering [13] have been developed for community detection in social networks. Detecting communities in a network depends on various factors like, whether the definition of community relies on global or local network properties, whether nodes can simultaneously belong to several communities, whether link weights are utilized and whether the community definition allows for hierarchical community structure. The fact that nodes in a network can belong to more than one community and a solution based on k -clique percolation given by Palla et al. [20] have resulted in an increased attention towards detecting overlapping communities in social networks. Although, most of the methods consider overlap of communities at boundaries, some methods allow central vertices of communities to overlap, making it unclear as how to characterize overlapping vertices [9]. The proposed method allows any central or boundary node to belong to multiple communities.

Considering the case of OSNs like Facebook, Twitter and so on, community structures have mostly been analyzed using traditional community detection techniques over social networks representing explicit relations (friends, colleagues, etc.) of users. However, we argue that using only explicit social graph can be reductive. The interaction links represent relationships that social actors use to diffuse information through a social network and thus provide important weighted structural information. This information can be used to determine functional, meaningful and useful community structures in social networks. For example, the observations made by Wilson et al. [25] and Chun et al. [2] on Facebook friendship and interaction data reveals that for most users, majority of their interactions occur only across a small subset of their social links proving that only a subset of social links actually represents interactive relationships. Their findings suggest that social network-based systems should be based on the activity network, rather than on the social link network. Furthermore, considering the interaction degree of nodes in online social networks, the likelihood of nodes to link to other nodes of similar degree is more than friend network. This means that nodes in an interaction graph show more assortativity than friend network and places it close to known social networks.

In this paper, we propose a density-based method, **CMiner**, for detecting overlapping community structures from the in-

teraction graphs of online social networks. The proposed method is in line with SCAN [26] and DENGRAPH [8] that are based on DBSCAN [7]. Both of them find dense communities and detect outliers in networks. In addition to these, **CMiner** has the following novel contributions.

- Based on the average reciprocated interactions of a node in the network, **CMiner** defines a new distance function to find the similarity between a pair of nodes.
- **CMiner** does not need a neighborhood threshold (often difficult to determine) to be specified by the users manually. Rather, it uses a simple approach to automatically determine the neighborhood threshold value for each node locally from the underlying network. Determining a way to optimally calculate the neighborhood threshold for density-based community detection methods is a long-standing and challenging task.
- **CMiner** finds *overlapping community structures* from the interaction graph of online social networks using a *density-based* approach. To the best of our knowledge, this approach is not used in existing state-of-the-art methods for community detection.

Rest of the paper is organized as follows. Section 2 presents a brief review of the related works on detecting communities from social networks. In section 3, we define the distance function and present a density-based overlapping community detection method. Section 4 presents our experimental setup and evaluation results. Section 5 presents a discussion on the proposed approach and also gives its time complexity and the possible directions for future work. Finally, section 6 concludes the paper.

2. RELATED WORK

Traditional community detection techniques include graph partitioning methods, which divide the vertices of a network into a predefined number of groups in such a way that the number of edges lying between groups is minimal. Kernighan-Lin algorithm [14] is one of the earliest known partitioning methods. Besides graph partitioning, partition-based clustering methods have also been used for detecting communities in networks. Given a set of data points and a predefined value k (number of clusters to be found) the problem is to divide the nodes into k clusters that optimizes a cost function based on distances between nodes and/or from points to centroids. However, a main drawback of these methods is to determine the number of clusters a priori.

Hierarchical clustering is another well-known technique used in social network analysis [21, 23]. Starting from a partition in which each node is in its own community or all nodes are in the same community, one splits or merges clusters according to a topological measure of similarity between nodes. Based on the concept of sociological notion of betweenness centrality, Girvan and Newman [11] have proposed a divisive hierarchical clustering algorithm for community detection, which calculates betweenness of all edges in a network and removes the one with highest betweenness value. The process continues until there is no edge remaining or a stopping criterion is met. But, the method does not provide a measure to determine the best split of communities in a network. Later on, Newman and Girvan [19] proposed *modularity* to measure the quality of a division of a network

into groups or communities. The idea of modularity is to compare the number of links inside communities to the expected number of links in a random reference network, which contains no community structure. High values of modularity indicate network partitions in which more edges fall within groups than expected by chance. In many methods proposed later, modularity became an objective function to be maximized leading to modularity optimization-based methods for community detection. Although, modularity optimization methods have been proved highly effective in practice for community evaluation, there are some major problems with the modularity measure. Firstly, modularity requires information about the entire structure of a network, which is unrealistic in case of large networks like the World Wide Web. As a solution to this problem, Clauset [3] proposed a measure of local community structure, called local modularity, for graphs which lack global knowledge. Secondly, modularity-based methods have a resolution limit and may fail to identify smaller (possibly important) communities [10].

Extending the DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm [7] to undirected and un-weighted graph structures, Xu et al. [26] propose SCAN (Structural Clustering Algorithm for Networks) to find clusters, hubs, and outliers in large networks. SCAN uses structural similarity which involves the neighborhood of vertices as clustering criteria.

$$dist(p, q) = \begin{cases} 0 & \text{if } p = q \\ \min(I_{\overline{pq}}, I_{\overline{qp}})^{-1} & \text{if } (I_{\overline{pq}} > 0) \wedge (I_{\overline{qp}} > 0) \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

Similarly, considering the weighted interaction graph of online social networks, Falkowski et al. [8] extend DBSCAN algorithm [7] to weighted interaction graph structures of online social networks. They define a distance measure based on the interaction between two actors p and q in a network as given in equation 1. In equation 1, $I_{\overline{pq}}$ and $I_{\overline{qp}}$ are the number of interactions between actors p and q initiated by p and q , respectively. The basic idea is that for each point in a cluster the neighborhood of a given radius (ϵ) has to contain at least a minimum number of points (η) such that the density of the points in the cluster exceeds some threshold.

Motivated by the fact that entities in networks can simultaneously belong to multiple communities, the issue of detecting overlapping communities has received a lot of attention recently. The most popular method for identifying overlapping communities is the Clique Percolation Method (CPM) proposed by Palla et al. [20] which is based on the concept of a k -clique, i.e., a complete subgraph of k nodes. The method relies on the observation that communities seem to consist of several small cliques that share many of their nodes with other cliques in the same community. A k -clique community is the largest connected subgraph obtained by taking the union of a k -clique and of all other k -cliques which are adjacent to it. Gregory [12] handles overlapping communities by adding one more action (node splitting) to the Newman-Girvan method [19]. The algorithm recursively splits nodes that are likely to reside in multiple communities, or removes edges that seem to bridge two different communities. This process is repeated until the network is disconnected into desired number of communities. In [16], the authors have proposed a method, called LFM, for un-

Table 1: Notations and their descriptions

Notation	Description
V	The set of nodes in the social network
I_p	Total count of outgoing interactions of a node p
$I_{\overrightarrow{pq}}$	Total count of outgoing interactions from node p to node q
$I_{\overleftarrow{p}}$	Total count of reciprocated interactions of a node p : $\sum_{\forall q \in V_p} \min(I_{\overrightarrow{pq}}, I_{\overleftarrow{qp}})$
$I_{\overleftrightarrow{pq}}$	Total count of reciprocated interactions of p and q : $\min(I_{\overrightarrow{pq}}, I_{\overleftarrow{qp}})$
V_p	Set of nodes in the networks with which node p interacts
V_{pq}	Set of nodes with which both nodes p and q interact: $V_p \cap V_q$

covering overlapping community structures based on local optimization of a fitness function. The method performs a local exploration of the network, searching for the natural community of each node (community structure is revealed by peaks in the fitness histogram). The procedure enables each node to be included in more than one module, leading to a natural description of overlapping communities.

3. PROPOSED METHOD

In this section, we discuss the proposed method, **CMiner** for detecting overlapping community structures in social networks. In order to ease the discussion, we have used a set of notations that are described in table 1. In line with the community detection method **SCAN** proposed by Xu et al. [26] and **DENGRAPH** proposed by Falkowski et al. [8], our approach is based on **DBSCAN** [7] method, where a cluster is searched by checking the neighborhood of each point in the underlying database. If the neighborhood of a given radius ε of a point p contains more than η points, a new cluster with p as a core object is created. The process then iterates to find directly density-reachable objects from these core objects and defines a density-connected cluster using density reachability and density connectivity properties [7]. The overall concept of our proposed method however significantly varies from them as discussed in the following paragraphs. We start with defining the distance function which measures the distance between a pair of nodes in the social network. In order to find the distance from a node p to a node q , the function computes average interaction weight between p and commonly interacted nodes of p and q , including q . This average interaction weight is normalized by the total outgoing interaction weight of p as shown in equation 2.

$$dist(p, q) = \begin{cases} 0 & \text{if } p = q \\ \left\{ \frac{\sum_{s \in V_{pq}} (I_{\overleftrightarrow{ps}}) + I_{\overleftrightarrow{pq}}}{|V_{pq}| + 1} \right\}^{-1} & \text{if } I_{\overrightarrow{pq}} \neq 0 \wedge |V_{pq}| \neq 0 \\ 1 & \text{otherwise} \end{cases} \quad (2)$$

Unlike [7, 8, 26] where the global neighborhood threshold parameter ε is manually required to be set at the beginning of the process and is mostly difficult to determine, we

propose a function that determines the local neighborhood threshold ε_p for a node p by taking into consideration the node's interaction data. The local neighborhood threshold aims to give a measure of the average interaction behavior of a node with all its neighbors in the interaction graph. The local neighborhood threshold ε_p of a node p is obtained using equation 3, where the symbols have the same interpretations as given in table 1.

$$\varepsilon_p = \begin{cases} \left\{ \frac{I_{\overleftarrow{p}}}{|V_p|} \right\}^{-1} & \text{if } I_{\overleftarrow{p}} \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

The local ε_p -neighborhood of a node p ($N_{local\varepsilon_p}$) is then defined as follows:

$$N_{local\varepsilon_p} = \{q : q \in V_p \wedge dist(p, q) \leq \varepsilon_p\} \quad (4)$$

In simple terms, the local ε_p -neighborhood of a node p is defined as the set of nodes in its interaction network whose distance from node p is less than or equal to the local neighborhood threshold ε_p of node p . To proceed with defining a community in our context, we need to define the concept of a *core node*. For this, we introduce the concept of *local minimum-number-of-points threshold* (η_p) as shown in equation 5, where η forms the only parameter to be manually set for **CMiner**. Now for a node p to qualify as a core node with respect to its local neighborhood threshold ε_p and local minimum-number-of-points threshold η_p , it is required to have at least η_p of its recipient nodes (nodes to which it sends interactions) within its local ε_p -neighborhood. The core nodes act as seeds from where communities are grown outwards based on the following definitions.

$$\eta_p = \frac{\eta \times |V_p|}{100} \quad (5)$$

Definition 1. A node q is *direct density-reachable* from a node p with respect to η if p is a core node and q belongs to the local ε_p -neighborhood of p . Direct density-reachability in this case is however asymmetric, as in the proposed context the distance from a node p to q may not be same as the distance from q to p .

Definition 2. Two nodes p and q are called *mutual cores* if both p and q are core nodes, and p belongs to the local ε_q -neighborhood of q , and q belongs to the local ε_p -neighborhood of p . In other words two core nodes p and q are called mutual cores if p and q are direct density-reachable from each other. Mutual core relation is symmetric and it is the symmetric relation which is used transitively to grow a community as discussed in the following paragraphs.

Definition 3. A node q is *density-reachable* from a node p with respect to η , if there is a chain of nodes v_1, v_2, \dots, v_n , where $v_1 = p$ and $v_n = q$, such that v_{i+1} and v_i are mutual cores for i ranging from $1, 2, \dots, n-2$, and v_n is direct density-reachable from v_{n-1} . Density reachability function is asymmetric and transitive, and it is not necessary that two nodes belonging to the same community be density reachable. They may belong to the same community because they are density reachable from some third node belonging to that community. This condition is justified in the following definition of density connectivity.

Definition 4. A node q is *density-connected* to a node p with respect to η , if there exists a node v such that both p

and q are density reachable from v . Density connectivity is a symmetric relation and for the density reachable vertices, it is also reflexive.

Definition 5. A non-empty subset $C \subseteq V$ is called as a *density-connected community* with respect to η , if all the vertices in C are density-connected and C is maximal with respect to density reachability.

The basic process of the proposed community detection method is as follows. Initially all nodes being un-labeled and un-visited, the process iteratively finds a density-connected community by randomly selecting an un-visited node to grow a community using density-reachable property. For each un-visited node p , it checks whether p is a core node and if p qualifies the test, it finds all the density-reachable nodes of p to identify its community. To do so, it first computes the local ε_p threshold for p using equation 3. If the ε_p threshold for p is greater than zero, then it uses the distance function of equation 2 to determine the local ε_p -neighborhood of p , i.e., $N_{local\varepsilon_p}$. Now if node p qualifies as a core node, its community list is appended with the current community label and the community list of each node in $N_{local\varepsilon_p}$ is also appended with the same. We use the term appended as the nodes belonging to $N_{local\varepsilon_p}$ including p can already be labeled by some other community labels during some previous iterations. A node is assigned to a new community irrespective of its previous community allotments, thus allowing a node to belong to multiple communities. Once a node p is identified as a core-node, the following key steps are used to identify a density-connected community for p .

1. All un-visited mutual-core nodes of the node p in $N_{local\varepsilon_p}$ are appended with the current community label. They are marked as visited and pushed to the stack in order to identify the density-reachable nodes of p . This step is later repeated for each node in the stack for the current community in order to find the connected sequences of mutual-core nodes p . These nodes are called the primary-core nodes of that community.
2. If a core-node q in $N_{local\varepsilon_p}$ is not a mutual-core of p , it is appended with the current community label, however, it is not pushed into the stack to grow the current community and its visited/un-visited status is kept un-altered.
3. Non-core nodes in $N_{local\varepsilon_p}$ are marked as visited and they are appended with the current community label. Such nodes form boundary nodes for the community of p and are thus not pushed into the stack as they cannot be used to grow a community.

The steps through 1-3 are repeated for each node in the stack thus identifying a density-connected community for each randomly selected un-visited node p in the social network. It is worthwhile to note that if a core-node q , assigned to a community C , does not show a mutual-core relation with any primary-core node of C , then q is called a secondary-core node of community C and C is called a secondary-community of q . Similarly, if a core-node r is a primary-core node of a community C then community C is called the primary-community of r .

The whole process is repeated for each un-visited node to find the overlapping community structure in the social

network. At the end of the process, un-labeled nodes (if any) represent outlier nodes, i.e., they do not belong to any community as they do not show an interaction behavior that is similar to any node or enough number of nodes in the social network.

4. EXPERIMENTAL RESULTS

In this section, we present the experimental results conducted on various synthetic and real world social network datasets, and compare them with results from some of the known community detection methods. Results are generated for overlapping community structures and the concept of Normalized Mutual Information (NMI) is used to compare the community structures found by various algorithms. We have specifically used the definition of NMI proposed and implemented by Lancichinetti et al. [16]. Unlike other comparison methods that work only with non-overlapping communities, NMI implemented by Lancichinetti et al. [16] is most commonly used to compare both overlapping and non-overlapping community structures.

4.1 Experiments on Synthetic Datasets

Lancichinetti and Fortunato [15] have proposed a synthetic network generation method that can be used to generate a class of artificial networks usually referred as LFR-benchmarks. As claimed by the authors, the networks generated so reflect important aspects of real networks and they can be used as benchmarks for testing community detection algorithms. We have used their method to generate various undirected-weighted networks for our experiments by varying different parameters required for the generation of networks. A description of the available parameters can be seen in their original paper [15]. Here, we only mention the parameter name and the value used to generate the network. For all generated synthetic networks, the average degree $\langle k \rangle$ and the maximum degree $maxk$ have been set to 20 and 50, respectively. Similarly, the minus exponent for the degree sequence t_1 and minus exponent for the community size distribution t_2 have been set to -3 and -1 , respectively. Rest of the parameter settings used to generate different networks is mentioned in the following paragraphs. It should further be noted that each point of every curve in figure 1 corresponds to an average over 25 realizations of the benchmark.

Figure 1 compares the result of **CMiner** with **LFM** [16] and **CFinder** which implements the weighted clique percolation method [20] for detecting overlapping communities in synthetic undirected and weighted networks. For N1000-S- β 1.5 networks, number of nodes (N) is set to 1000, community size is set to relatively small in the range of 20-50 (represented using S), and the exponent for weight distribution β is set to 1.5. Similarly, for N1000-B- β 1.5 networks, the parameters are same as the previous networks except the community size which is relatively bigger and set in the range of 20-100 (represented using B). For N5000-S- β 1.5 networks, number of nodes (N) is set to 5000, community size is relatively small in the range of 20-100, and exponent for the weight distribution β is 1.5. Similarly, for N5000-B- β 1.5 networks, the parameters are same as N5000-S- β 1.5 networks except the community size which is relatively bigger and set in the range of 20-200. For each of these settings, we have set the mixing parameter for weight values as $w=1$; number of memberships of the overlapping nodes $om=3$ and

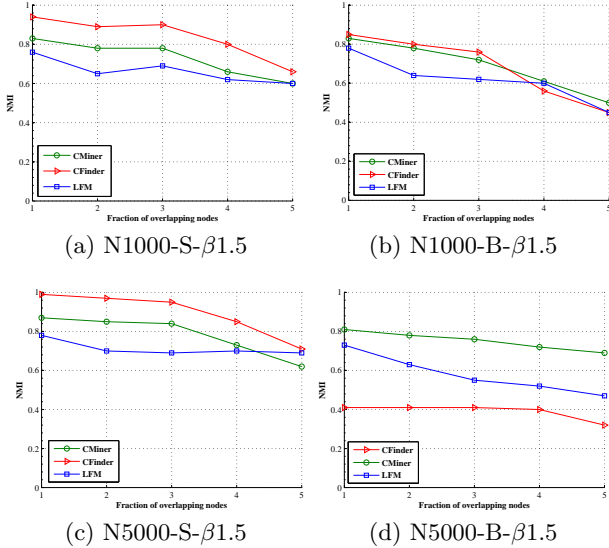


Figure 1: Experimental results on undirected-weighted LFR-Benchmarks with overlapping communities.

varied the fraction of overlapping nodes to generate benchmark networks. For **CFinder** [20], we have used its weighted clique percolation method with clique size $k=4$ and varied the intensity threshold parameter accordingly to get the best results. For **CMiner**, we have varied the minimum-number-of-points parameter η between 50% – 85% to get the best results. For **LFM**, we have used the default parameter setting with $\alpha = 1$. As can be seen from figure 1, the results obtained by **CMiner** for synthetic networks with overlapping nodes, when the size of communities is smaller in both small and large networks, are comparable to the results obtained by **CFinder** [20] and better than **LFM** [16]. However, when the size of communities is larger, **CMiner** performs similar to **CFinder** for small networks and marginally better for larger networks. It also performs better than **LFM** in this case as well.

4.2 Experiments on Zachary’s Network

Zachary’s network of karate club members [27] is a well-known graph regularly used as a benchmark to test community detection algorithms. The dataset consists of an undirected graph representing weighted interactions between 34 members of a club at a U.S. university, as recorded over a two-year period. During the course of the study, the club split into two groups as a result of a dispute within the organization, and the members of one group left to start their own club. The actual division of the club into two groups is shown in figure 2a. We have compared the results of our method on the Zachary’s network with the result obtained from the fast modularity optimization method for weighted networks proposed by Newman [18] shown in figure 2c and with the best result obtained by using **DENGRAPH** proposed by Falkowski et al. [8], by setting the neighborhood threshold to $1/3$ and the minimum number of points threshold to 3, shown in figure 2d. The result of **LFM** [16] on Zachary’s network is shown in figure 2b. The result obtained after applying our method is shown in figure 2f for which $\eta = 50\%$ of friends. In figure 2, nodes belonging to the same community

are enclosed within the same community boundary. Overlapping nodes are represented by nodes lying within multiple community boundaries. Nodes that are not enclosed within any boundary represent un-clustered/outlier nodes. As is

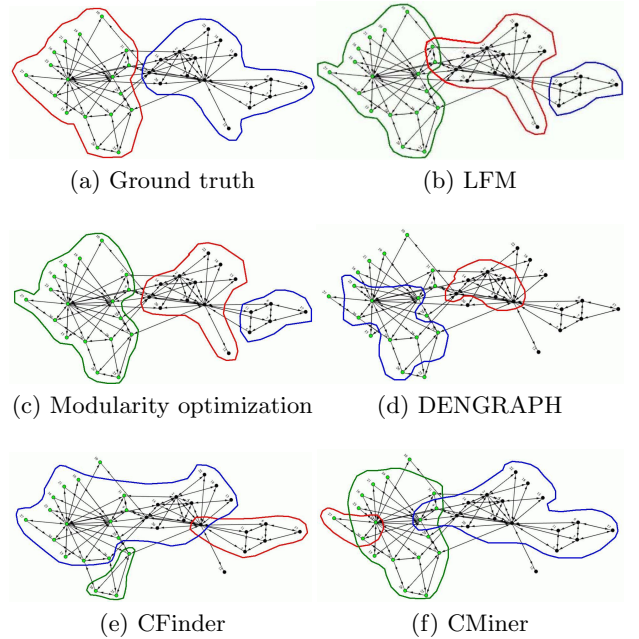


Figure 2: Experimental results on Zachary’s karate club network (best viewed in color).

obvious from figure 2, only **CMiner**, **LFM** and the weighted fast modularity optimization method [18] give the most appropriate results. **CMiner** detects 3 overlapping communities and 2 outliers from the Zachary’s network. Among the three communities the smallest community (enclosed by red boundary in the figure 2a) consists of three nodes and two of its nodes overlap with a much larger community (enclosed by green boundary in the figure 2a). This implies that the smaller community can be considered as an integral part of the larger community thus finding exactly two main communities which are most close to the ground truth. Similarly, **LFM** finds 3 communities among which 2 communities overlap. We argue that **CMiner** gives a more descriptive representation of the community structure in Zachary’s network as it identifies a split closer to the actual one and also finds the nodes where the communities overlap which could possibly represent the nodes, that held the club together or played a role in the split of the club. This is because, between the two main communities identified by **CMiner** (which are most close to the ground truth) it marks nodes labeled 9 and 33 as overlapping nodes, i.e., belonging to both the main communities. It means that the whole group could be thought of being held together by nodes labeled as 9 and 33. Analogously, a dispute between nodes labeled as 1 and 33 had resulted in the actual split of the club. Moreover, node 9 is a common neighbor of nodes 1 and 33. Thus we can say that the communities identified by **CMiner** in the zachary network are more realistic than the others. However, the result of **LFM** on the Zachary’s network is also comparable with **CMiner**.

5. DISCUSSION AND FUTURE WORK

On the basis of the experimental results discussed in the previous section, it is quite obvious that **CMiner** is comparable with the Clique Percolation Method (CPM) [20] and better than LFM [16] considering the overlapping nature of communities. The results on the Zachary dataset also show that **CMiner** performs better than some of the well known non-overlapping community detection methods like [8] and [18]. Considering the complexity analysis, **CMiner** involves analyzing the local neighborhood of each node in the network, and for each node this cost is proportional to its interaction degree. Thus, the total cost for this step is $O(\text{deg}(p_1) + \text{deg}(p_2) + \dots + \text{deg}(p_n))$, where $\text{deg}(p_i), i = 1, 2, \dots, n$ is the degree of node p_i . For a complete graph of n nodes, the degree of each node is $n - 1$, leading to a worst case complexity for this step as $O(n^2)$. So, the total cost of **CMiner** in worst case is $O(n^2)$. However, in general, real-world networks show sparser degree distributions, resulting in an $O(n)$ average case complexity. Thus, on an average, the total cost of the whole method on sparse real-world networks is $O(n)$, which is significantly better than $O(n^2)$ for [18] and $O(e^n)$ for CPM [20].

Although the proposed approach seems promising, the major concern related to it is regarding the asymmetry of the distance function. The proposed method however compensates this asymmetry using the mutual-cores relation between core-nodes. However, the notion of density-based methods usually requires that the distance function be symmetric. In our future work, we aim to give a symmetric representation of the proposed distance function. Moreover, besides using only interaction data from social networks, we also aim to use the social (friend) relation data for defining the distance between two nodes in a social network as it also represents important explicit relations between them.

6. CONCLUSION

In this paper, we have proposed a density-based method, **CMiner**, to identify overlapping community structures from social network interactions. **CMiner** is based on an average interaction behavior of nodes in the social network. Unlike related density-based methods, **CMiner** does not need the global neighborhood threshold to be specified by the users which is mostly difficult to determine. Instead, the proposed method automatically computes a local version of the neighborhood threshold from the underlying interaction graph. Moreover, it requires a single tunable parameter to be set by the user which determines the density of the communities to be identified from the network. This property also enables **CMiner** to extract community structures of varying densities in a hierarchical manner. We have presented some preliminary results on synthetic and real-world networks. The results show that **CMiner** is at par with state-of-the-art methods and performs better than some other related methods. **CMiner** is also faster and naturally scalable to large social networks.

7. ACKNOWLEDGEMENTS

The authors acknowledge the support provided by King Abdulaziz City for Science and Technology (KACST) and King Saud University (KSU), Kingdom of Saudi Arabia. This work has been funded by the KACST under the NPST project number 11-INF1594-02.

8. REFERENCES

- [1] G. Agarwal and D. Kempe. Modularity maximizing network communities using mathematical programming. *The European Physical Journal B*, 66:409–418, 2008.
- [2] H. Chun, H. Kwak, Y. Eom, Y. Ahn, S. Moon, and H. Jeong. Comparison of online social relations in volume vs interaction: a case study of cyworld. In *Proceedings of the 8th ACM SIGCOMM Conference on Internet Measurement*, volume 5247, pages 57–70, 2008.
- [3] A. Clauset. Finding local community structure in networks. *Physical Review E*, 72:026132, 2005.
- [4] A. Clauset, C. Moore, and M. E. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008.
- [5] A. Clauset, M. E. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70:066111, 2004.
- [6] C. H. Ding, X. He, H. Zha, M. Gu, and H. D. Simon. A min-max cut algorithm for graph partitioning. In *Proceedings of the International Conference on Data Mining*, pages 107–114, 2001.
- [7] M. Ester, H. Kriegel, S. Jörg, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the International Conference on Knowledge Discovery from Data*, pages 226–231, 1996.
- [8] T. Falkowski, A. Barth, and M. Spiliopoulou. DENGGRAPH: a density-based community detection algorithm. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 112–115. IEEE Computer Society, Washington, DC, USA, 2007.
- [9] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.
- [10] S. Fortunato and M. Barthélemy. Resolution limit in community detection. In *Proceedings of the National Academy of Science*, volume 104, pages 36–41, 2007.
- [11] M. Girvan and M. E. Newman. Community structure in social and biological networks. In *Proceedings of the National Academy of Sciences*, volume 99, pages 7821–7826, 2002.
- [12] S. Gregory. An algorithm to find overlapping community structure in networks. In *Proceedings of the 11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 91–102, 2007.
- [13] M. S. Handcock, A. E. Rafter, and J. M. Tantrum. Model-based clustering for social networks. *Journal of the Royal Statistical Society A*, 170:301–354, 2007.
- [14] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal*, 49:291–307, 1970.
- [15] A. Lancichinetti and S. Fortunato. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Physical Review E*, 80:016118, 2009.
- [16] A. Lancichinetti, S. Fortunato, and J. Kertész. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11:033015, 2009.

- [17] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.
- [18] M. E. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69:066133, 2004.
- [19] M. E. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69, 2004.
- [20] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.
- [21] J. P. Scott. *Social Network Analysis: A Handbook*. Sage Publications Ltd., 2 edition, 2000.
- [22] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.
- [23] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.
- [24] S. White and P. Smyth. A spectral clustering approach to finding communities in graphs. In *Proceedings of the 5th SIAM International Conference on Data Mining*, pages 76–84, 2005.
- [25] C. Wilson, B. Boe, A. Sala, K. P. Puttaswami, and B. Y. Zhao. User interactions in social networks and their implications. In *Proceedings of the 4th ACM European Conference on Computer Systems*, pages 205–218. ACM, New York, 2009.
- [26] X. Xu, N. Yuruk, Z. Feng, and T. A. J. Schweiger. SCAN: a structural clustering algorithm for networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '07)*, pages 824–833. ACM, 2007.
- [27] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.