SHRINK: A Structural Clustering Algorithm for Detecting Hierarchical Communities in Networks

Jianbin Huang
School of Software
Xidian University
Xi'an, China
jbhuang@xidian.edu.cn

Heli Sun
Dept. of Computer Science
and Technology
Xi'an Jiaotong University
Xi'an, China
helisun@stu.xjtu.edu.cn

Jiawei Han
Dept. of Computer Science
University of Illinois at
Urbana-Champaign
Urbana, IL, USA
hanj@cs.uiuc.edu

Hongbo Deng
Dept. of Computer Science
University of Illinois at
Urbana-Champaign
Urbana, IL, USA
hbdeng@uiuc.edu

Yizhou Sun
Dept. of Computer Science
University of Illinois at
Urbana-Champaign
Urbana, IL, USA
sun22@uiuc.edu

Yaguang Liu
School of Computer Science
and Technology
Xidian University
Xi'an, China
ygliu521@gmail.com

ABSTRACT

Community detection is an important task for mining the structure and function of complex networks. Generally, there are several different kinds of nodes in a network which are cluster nodes densely connected within communities, as well as some special nodes like hubs bridging multiple communities and outliers marginally connected with a community. In addition, it has been shown that there is a hierarchical structure in complex networks with communities embedded within other communities. Therefore, a good algorithm is desirable to be able to not only detect hierarchical communities, but also identify hubs and outliers. In this paper, we propose a parameter-free hierarchical network clustering algorithm SHRINK by combining the advantages of density-based clustering and modularity optimization methods. Based on the structural connectivity information, the proposed algorithm can effectively reveal the embedded hierarchical community structure with multiresolution in large-scale weighted undirected networks, and identify hubs and outliers as well. Moreover, it overcomes the sensitive threshold problem of density-based clustering algorithms and the resolution limit possessed by other modularity-based methods. To illustrate our methodology, we conduct experiments with both real-world and synthetic datasets for community detection, and compare with many other baseline methods. Experimental results demonstrate that SHRINK achieves the best performance with consistent improvements.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining; G.2.2 [**Graph Theory**]: Graph Algorithms; I.5.3 [**Clustering**]: Algorithms

General Terms

Algorithms, Design, Performance

Keywords

Hierarchical Community Discovery, Graph Clustering, Hubs and Outliers

1. INTRODUCTION

Nowadays, many real-world networks possess intrinsic community structure, such as large social networks, Web graphs, and biological networks. A community (also referred to as a module or cluster) is typically thought of a group of nodes with dense connections within groups and sparse connections between groups as well. Detecting communities in a network can provide insight into how network function and topology affect each other and has received a great deal of attention in recent years. For example, communities in a co-authorship network might imply researchers working together with the same interests, and communities in a citation network might indicate related papers on a single topic, meanwhile communities on the Web graph might represent pages of related topics.

Finding communities in complex networks is a nontrivial task, since the number of communities in the network is typically unknown and the communities are often of unequal size or density. Moreover, it has been shown that there is a hierarchical structure of complex networks with communities embedded within other communities. Essentially, small communities group together to form larger ones, which in turn group together to form even larger ones [16]. Taking the co-authorship network extracted from DBLP in Figure 1 as an example, a research field can be composed of many research groups with the same academic interests. For example, there are many groups in DM research field, while a group may consist of several subgroups like "data stream mining", "graph mining","mining moving object" and so on.

Besides the general nodes that are densely connected with communities, there are some special nodes like hubs (denoted as red diamonds) and outliers (denoted as white triangles) in Figure 1. For example, some researchers, like "Ji-
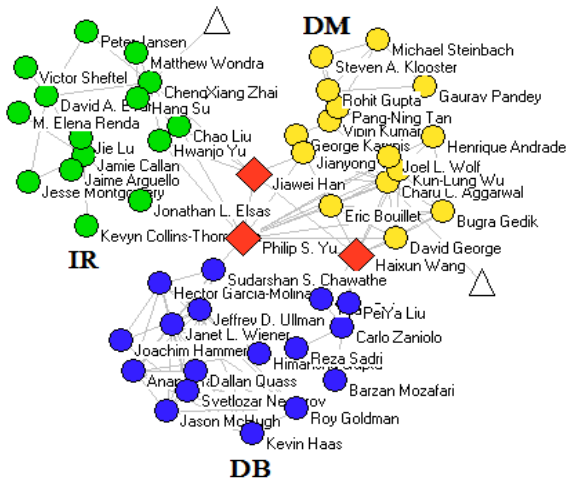
Figure 1: Community structure and node roles for an example of co-authorship network extracted from DBLP.

awei Han" and "Philip S. Yu", have published a large amount of papers in collaboration with people from various research communities. These nodes should be considered as hubs that are closely related to different communities, forming overlapping communities. As we know, hubs play special and important roles in many real-world networks. For example, hubs in the WWW could be utilized to improve the search engine rankings for relevant authoritative Web pages [14], and hubs in viral marketing [7] and epidemiology [5] could be central nodes for spreading ideas or diseases. Furthermore, there are some nodes that are marginally connected with the community members, such as the white triangles in Figure 1. In reality, a visiting scholar who only publishes one paper with researchers in the hosted group should not be considered as a member of the group, and meanwhile it is better to be regarded as an outlier. Since outliers have little or no influence in a community, they may be isolated as noise in the network. Therefore, how to detect hierarchical communities as well as hubs and outliers in a network becomes an interesting and challenging problem. However, most existing approaches only study the community detection without considering hubs and outliers. In this paper, we propose a parameter-free hierarchical network clustering algorithm SHRINK by combining the advantages of density-based clustering and modularity-based methods. The main contributions are summarized in the following:

1. We propose a novel parameter-free network clustering algorithm. Through shrinkage of the local micro-communities into super-node iteratively, our algorithm does not only reveal the meaningful hierarchical community structure in networks, but also identify the hubs and outliers.

2. Our algorithm can find the communities with various densities. Moreover, the clustering result does not depend on the order of processed nodes. Experimental results show that our algorithm is effective and efficient.

3. By combining the advantages of density-based clustering and modularity optimization, our algorithm over-

comes not only sensitive threshold problem of density-based clustering algorithm, but also the resolution limit that other modularity-based algorithms suffer from.

The rest of the paper is organized as follows. First we briefly review some related work in Section 2. In section 3, we formulize the notion of hierarchical structural-connected clusters. In section 4, we describe the algorithms in detail. In section 5, we report the experimental results. Finally, we summarize our conclusions and suggest future work in section 6.

2. RELATED WORK

Community discovery in complex networks has been studied for years in multiple fields, particularly computer science and physics. Traditional graph partitioning methods, such as Kernighan-Lin algorithm [13], Girvan-Newman algorithm [11], normalized cut [24], and spectral bisection methods [25] have been widely applied to find network communities. Recently, significant progress has been archived in this research field and many approaches have been presented for detecting communities in networks.

Modularity-based methods: For evaluating the quality of network partitions, Newman and Girvan proposed the modularity measure Q [20] which has been widely used in community discovery. Modularity-based methods assume that high values of modularity indicate good partitions. But it has been proven that modularity optimization is an NP-complete problem. Most of the modularity-based algorithms find good approximation of the modularity maximum with high computational complexity such as SA (Simulated Annealing) [12], FN [19], and CNM [6]. Recently, Blondel *et al.* proposed a greedy modularity-based algorithm, called BGLL[3], for finding communities in weighted networks. This algorithm has a low computational complexity and can discover hierarchical communities. However, the results of the algorithm depend on the order in which the nodes are visited. Actually, the methods of greedy optimization of modularity often tend to form large communities through combination of small ones. Recent research shows that modularity is not a scale-invariant measure, and hence, by relying on its maximization, detection of communities smaller than a certain size is impossible. This serious problem is famously known as the resolution limit of modularity-based algorithms [10]. Compared with the traditional modularity-based methods, our work use the modularity as a quality function to guide the selection of optimal hierarchical communities.

Hierarchical and Overlapping methods: In the presence of hierarchy, the concept of community structure becomes richer. Agglomerative or divisive hierarchical clustering are well-known techniques to solve this problem [19, 11]. Starting from a partition in which each node is its own community, or all nodes are in the same community, one merges or splits clusters according to a topological measure of similarity between nodes. In this way, one builds a hierarchical tree of partitions. Though this type of methods naturally produces a hierarchy of partitions, it needs a metric to stop the algorithm. Recently, some work focused on the problem of identifying meaningful community hierarchies [23] and detecting multiresolution levels [2, 16, 22].

The issue of finding overlapping communities has become a hot topic. Palla *et al.* proposed a clique percolation

method (CPM) [21]. A complete sub-graph of k nodes, called k-clique, is rolled over the network through other cliques with $k-1$ common nodes. In this way, a set of nodes can be reached, which is regarded as a community. One node can belong to more than one community; therefore, overlaps naturally occur. The CPM algorithm is limited by its assumption that the graph has a large number of cliques. Furthermore, the method is not suitable to detect hierarchical structure. Recently, Nepusz *et al.* considered the problem of fuzzy community detection in networks, which expands the concept of overlapping community structure [18]. Every node is allowed to belong to multiple communities with different degrees of membership. A measure was introduced to identify regular nodes in a community, hubs that have significant membership in more than one single community, and outliers that do not belong to any of the communities.

In real networks, communities are usually both hierarchical and overlapping. Most existing methods investigate these two phenomena separately. Our work is one of the few methods that try to discover both hierarchical communities and overlapping nodes in a given network.

Density-based methods: Density-based clustering approaches (e.g., DBSCAN [8] and OPTICS [1]) have been widely used in data mining owing to their ability of finding clusters of arbitrary shape even in the presence of noise. Recently, Xu *et al.* proposed an efficient structural network clustering algorithm SCAN [26] through extension of the DBSCAN [8]. This algorithm can find communities as well as hubs and outliers in a network. However, it requires a minimum similarity parameter ε and a minimum cluster size μ to define clusters, and is sensitive to the parameter ε which is difficult to determine automatically. To deal with this problem, Bortner *et al.* proposed a new algorithm, called SCOT+HintClus [4], to detect the hierarchical cluster boundaries of network by extending the algorithm OPTICS [1]. However, it does not find the global clustering result and needs an additional pruning process to expose the reasonable hierarchical structure of the networks. Our work tries to develop a parameter-free method to explore the hierarchy of structural-connected communities with multiresolution levels in networks.

3. DENSELY CONNECTED HIERARCHICAL COMMUNITIES

The goals of our algorithm are not only to cluster networks hierarchically but also to identify two kinds of special nodes: hubs and outliers. Therefore, local connectivity structure of the network is used in our optimal clustering. In this section, we formalize some notions and properties of the hierarchical structure-connected clusters.

Definition 1. (Structural Similarity) Let $G = (V, E, w)$ be a weighted undirected network and $w(e)$ be the weight of the edge e. For a node $u \in V$, we define $w(\{u, u\}) = 1$. The structure neighborhood of a node u is the set $\Gamma(u)$ containing u and its adjacent nodes which are incident with a common edge with u : $\Gamma(u) = \{v \in V | \{u, v\} \in E\} \cup \{u\}$. The structural similarity between two adjacent nodes u and v is then

$$\sigma(u, v) = \frac{\sum\limits_{x \in \Gamma(u) \cap \Gamma(v)} w(u, x) \cdot w(v, x)}{\sqrt{\sum\limits_{x \in \Gamma(u)} w^2(u, x)} \cdot \sqrt{\sum\limits_{x \in \Gamma(v)} w^2(v, x)}}. \quad (1)$$
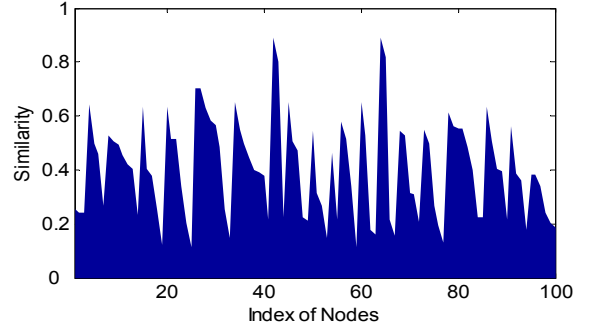

Figure 2: A segment of similarity-plot for the DBLP co-authorship network.

The above structural similarity is extended from a cosine similarity used in [26] which effectively denotes the local connectivity density of any two adjacent nodes in a weighted network. It can be replaced by other similarity definitions such as Jaccard similarity, and our experimental results show that the cosine similarity is better.

The density-based clustering algorithm OPTICS [1] shows that the hierarchical cluster structure of a dataset can be obtained from the reachability-similarity values plotted for each object in the cluster-ordering. Here we intend to design a parameter-free algorithm, and we do not use the minimum similarity threshold ε and the minimum cluster size μ any more. Actually, the reachability-similarity of any adjacent nodes u and v are equal to their structural similarity when $\mu = 2$ and the clustering results are not sensitive to the parameter μ. In Figure 2, we give a segment of ordered similarity-plot extracted from the DBLP co-authorship network. It is able to observe that the similarity distribution describes the intrinsic clustering structure accurately with high similarity regions surrounded by low similarity regions. The clusters are clearly discernible as "mountains" in the plot, and the hubs and outliers are located in the low regions between the mountains. Thus, if we explore the clusters from the top of each mountain to the plain, we would find not only the nested cluster structure, but also clusters with a variety of densities. Each local maximum of the similarity in the plot corresponds to a densely connected node pair.

Definition 2. (Dense Pair) Given a network $G = (V, E)$, $\sigma(u, v)$ is the structural similarity of nodes u and v. If $\sigma(u, v)$ is the largest similarity between nodes u, v and their adjacent neighbor nodes: $\sigma(u, v) = \max\{\sigma(x, y) | (x = u, y \in \Gamma(u) - \{u\}) \vee (x = v, y \in \Gamma(v) - \{v\})\}$, then $\{u, v\}$ is called a dense pair in G, denoted by $u \leftrightarrow_\varepsilon v$, where $\varepsilon = \sigma(u, v)$ is the density of pair $\{u, v\}$.

A dense pair is a pair of nodes with the largest similarity from each other. That is to say, the connectivity density of the two nodes is not less than their surrounding links. As shown in Figure 3, $\{9, 13\}$ is a dense pair with density 0.8165 in the example network.

Definition 3. (Micro-community) Given a network $G = (V, E)$, $C(a) = (V', E', \varepsilon)$ is a connected sub-graph of G represented by a node a. $C(a)$ is a local micro-community iff 1)$a \in V'$; 2)for all $u \in V'$, $\exists v \in V'(u \leftrightarrow_\varepsilon v)$; 3)$\nexists u \in$

Figure 3: The micro-communities in an example network weighted by structural similarity.

$V(u \leftrightarrow_\varepsilon v \wedge u \in V' \wedge v \notin V')$. ε is the density of the micro-community $C(a)$.

The micro-community is an isolated node or a sub-graph that consists of one or more connected dense pairs with certain density ε. As shown in Figure 3, the single node set $\{7\}$ forms a micro-community with density 1 and the nodes set $\{8, 11, 12\}$ forms a micro-community with density 0.8 in the toy network. For any node $v \in V$, v must be in the same micro-community with itself. Obviously, it is symmetric and transitive for the relation of nodes being in the same micro-community. Thus, a network will be partitioned into one or more local micro-communities by this equivalence relation. The involved properties are introduced in the following theorems.

THEOREM 1. *Given a network $G = (V, E)$, $C(a) = (V', E', \varepsilon)$ is a micro-community in G. For all $u \in V$, $C(u) = C(a)$ iff $u \in V'$.*

THEOREM 2. *Given a network $G = (V, E)$, $C = (V', E', \varepsilon)$ is a micro-community with density ε in G. If $u \in V'$ and $v \in \Gamma(u) - \{u\}$, then $\sigma(u, v) \leq \varepsilon$.*

According to the partitioning by micro-communities, the original network can be reduced into a smaller super-graph by shrinking the micro-communities into super-nodes. Then a dense micro-community can be regarded as a single node in the following process.

Definition 4. (Super-network) Given a network $G = (V, E, \sigma)$, $\tilde{V} = \{V_1, V_2, \cdots, V_k\}$ is a partition of the node set V and $\forall V_i \in \tilde{V}$, the sub-network $G_i = (V_i, E_i)$ induced by the node set V_i is a local micro-community in G. Define $\tilde{E} = \{\{V_i, V_j\} | \exists u \in V_i, \exists v \in V_j, \{u, v\} \in E\}$ and $\tilde{\sigma}(V_i, V_j) = \max\{\sigma(u, v) | u \in V_i, v \in V_j\}$; then $\tilde{G} = (\tilde{V}, \tilde{E}, \tilde{\sigma})$ is called a super-network of G.

When a hierarchical tree of local micro-communities has been built, the following Theorem 3 shows that the density of a local micro-community is not less than the density of the bigger micro-community in which it is embedded.

THEOREM 3. *Given a network $G = (V, E, \sigma)$ and its super-network $\tilde{G} = (\tilde{V}, \tilde{E}, \tilde{\sigma})$, if $C = (V', E', \varepsilon)$ is a local micro-community in G, $\tilde{C} = (\tilde{V}', \tilde{E}', \tilde{\varepsilon})$ is a local micro-community in $\tilde{G}$, and $V' \in \tilde{V}'$, then $\tilde{\varepsilon} \leq \varepsilon$.*

4. THE ALGORITHMS

In this section, we describe the hierarchical clustering algorithm SHRINK-H which reveals the densely connected

clusters, hubs and outliers in networks. A similarity-based modularity gain is adopted to evaluate the quality of micro-communities and to stop the algorithm. In order to reduce the running time, we also introduce a greedy algorithm SHRINK-G which is more efficient with almost the same clustering results.

4.1 Measurement of Modularity Gain

A metric is necessary for our algorithm to measure the goodness of the discovered hierarchical communities. Many quality functions have been proposed, such as modularity, fitness, *etc*. Here we select the modularity measure Q as the quality function because of its effectiveness in practice and efficiency for calculation. We use the similarity-based modularity function Q_s proposed by Feng *et al.* in [9]. It is extended from the connection-based modularity Q and has a better ability to deal with hubs and outliers. Given a cluster $CR = \{C_1, C_2, \cdots, C_k\}$ of the network $G = (V, E)$, the function Q_s is defined as follows:

$$Q_s = \sum_{i=1}^{k} \left[\frac{IS_i}{TS} - \left(\frac{DS_i}{TS} \right)^2 \right], \qquad (2)$$

where k is the number of clusters, $IS_i = \sum_{u,v \in C_i} \sigma(u, v)$ is the total similarity of nodes within cluster C_i, $DS_i = \sum_{u \in C_i, v \in V} \sigma(u, v)$ is the total similarity between nodes in cluster C_i and any node in the network, and $TS = \sum_{u,v \in V} \sigma(u, v)$ is the total similarity between any two nodes in the network.

To enhance the efficiency of the algorithm, we calculate the modularity Q_s incrementally. Given two adjacent module C_i and C_j, the modularity gain ΔQ_s can be computed by

$$\Delta Q_s = Q_s^{C_i \cup C_j} - Q_s^{C_i} - Q_s^{C_j} = \frac{2US_{ij}}{TS} - \frac{2DS_i \cdot DS_j}{(TS)^2}, \quad (3)$$

where $US_{ij} = \sum_{u \in C_i, v \in C_j} \sigma(u, v)$ is the total similarity of the links between two modules C_i and C_j.

Based on the equation (3), the gain of modularity Q_s for merging a micro-community $C = \{c_1, c_2, \cdots, c_k\}$ into a super-node can be easily computed as

$$\Delta Q_s(C) = \frac{\sum_{i,j \in \{1,2,\cdots,k\}, i \neq j} 2US_{ij}}{TS} - \frac{\sum_{i,j \in \{1,2,\cdots,k\}, i \neq j} 2DS_i \cdot DS_j}{(TS)^2}. \quad (4)$$

The similarity-based modularity described above is a metric to evaluate the quality of a partition. Here we use the gain of modularity to control the shrinkage of the micro-communities. If the modularity gain ΔQ_s of a micro-community C is positive, we argue that the nodes of C should be clustered in the same community. Thus, given a network G, the task of our community discovery algorithms is to find a higher modularity solution under the principle of density-based clustering, rather than to search a partition greedily maximizing the modularity Q_s.

4.2 Clustering via Hierarchical Shrinkage

The pseudo-code of our hierarchical clustering algorithm, called SHRINK-H, is given in Algorithm 1. The main process can be divided into two phases that are repeated iteratively, as shown in Figure 4(a). Given a network with n nodes, first we initialize each node with a different community label. In this initial partition, the number of communities is the same as the number of nodes. Then, for each node
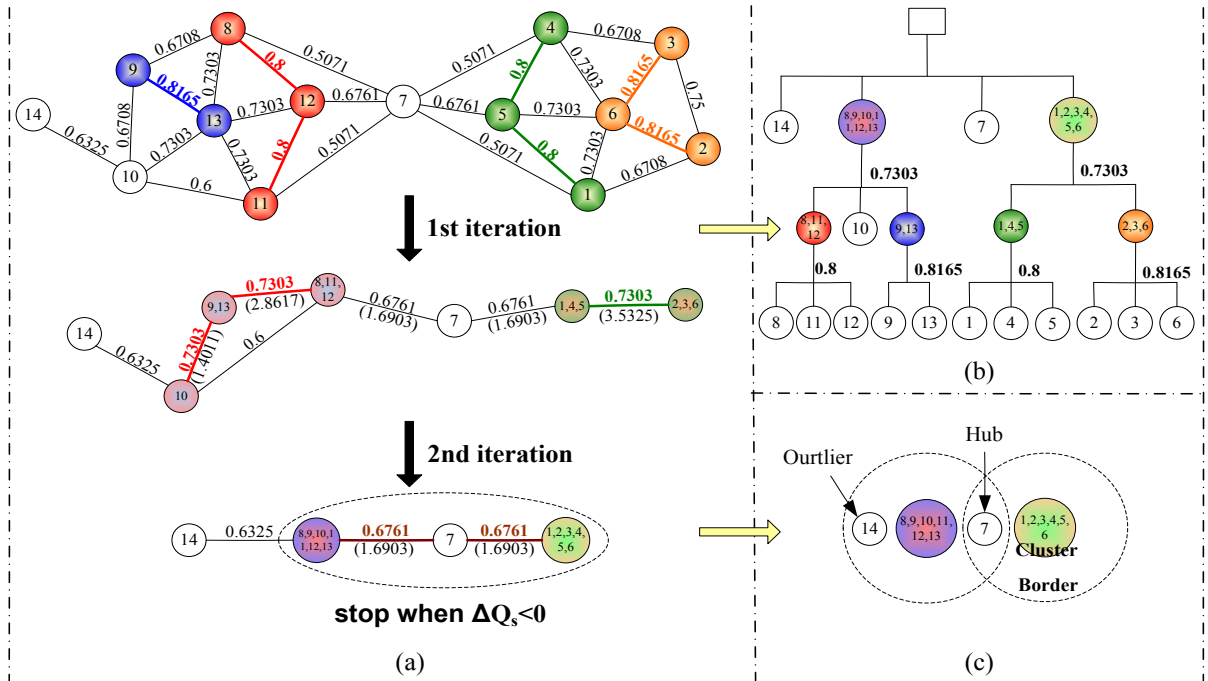
Figure 4: Illustration of the procedure and result of the hierarchical network clustering algorithm SHRINK-H: (a) the process of the hierarchical shrinkage of the micro-communities, (b) the hierarchical structure of the nested communities with different densities, and (c) the final two-layers overlapping communities.

i we find its local micro-community. This process is applied sequentially for all nodes. We record all the different micro-communities which represent a partition of the network and then the first phase is completed. The second phase of the algorithm is to build a super-network. We evaluate the gain of Q_s for the shrinkage of the micro-communities found during the first phase. If the gain is positive, the corresponding local community is replaced by a super-node. The above two phases are executed in turns until there is no micro-community with positive modularity gain. Then the hierarchy of communities naturally occurs, as shown in Figure 4(b). This algorithm is efficient because the size of the network is reduced rapidly in the process. In each iteration, a node is visited only once and the corresponding local micro-communities do not depend on the order in which the nodes are visited.

If one wants to get traditional non-overlapping partitions or overlapping communities without hubs and outliers, a post-process can be employed to deal with the "homeless" nodes: hubs and outliers. The homeless nodes whose neighbors are within at most one cluster are outliers. Each outlier can be assigned to its adjacent cluster as a border node. Other homeless nodes are regarded as hubs. If overlapping communities are considered, the hubs can be assigned to their adjacent communities as border nodes shown in Figure 4(c). Otherwise, they can be assigned to the adjacent community that harvests the largest positive gain of Q_s.

4.3 Clustering via Greedy Shrinkage

To enhance the efficiency, we propose a modified greedy algorithm SHRINK-G. Compared with micro-community, the dense pair is a smaller unit which has the largest density among its surrounding links. If we do not consider the hierarchical structure of communities, we can cluster the net-

work via greedy shrinkage of the dense pairs. Thus, each dense pair in a micro-community is considered separately. Starting with an arbitrary node u in a network G, we find the dense pair containing u. If there is a node v adjacent to u that forms a dense pair $\{u, v\}$ and its modularity gain is positive, we merge node v and u to form a super-node u'. Then we check whether there exists a dense pair containing u' and try to shrink it. The above process is repeated until there does not exist a shrinkable dense pair containing current node. Then the algorithm continues with next unvisited node. The clustering is accomplished when all the nodes in the network G are visited. The pseudo-code of this clustering algorithm is given in Algorithm 2, called SHRINK-G. Since this algorithm needs to visit all the nodes in a network only once, it is much faster than the SHRINK-H. However, the clustering result may rely on the visiting sequence of the nodes. Nevertheless, our experimental results on a large amount of networks show that the clustering results of the above two algorithms are the same in most cases.

5. EXPERIMENTAL RESULTS

In this section, we evaluate the proposed algorithm SHRINK using some real-world datasets and synthetic benchmark datasets. We compare our algorithms with the density-based network clustering algorithm SCAN and two representative modularity-based methods: CNM [6] and BGLL [3]. Our algorithms are implemented in ANSI C++. All the experiments were conducted on a PC with a 2.4 GHz Pentium IV processor and 2GB of RAM.

5.1 Evaluation Criteria

In our experiments, we adopt Normalized Mutual Information (NMI), an information-theoretic based measurement,

Algorithm 1: SHRINK-H

Input: Network $G = (V, E)$
Output: Set of clusters $CR = \{C_1, C_2, \cdots, C_k\}$; Set of hubs and outliers N

begin
    $CR \leftarrow \{\{v_i\} | v_i \in V\}$;
    while *true* **do**
        // Phase 1:Detect local micro-communities
        $MC \leftarrow \emptyset$;
        for *each* $v \in V$ **do**
            $C(v) \leftarrow \emptyset$;
            Queue q;
            $q.\text{insert}(v)$;
            $\varepsilon \leftarrow max\{\sigma(v, x) | x \in \Gamma(v) - \{v\}\}$;
            while $q.empty() \neq true$ **do**
                $u \leftarrow q.\text{pop}()$;
                if $u = v \vee max\{\sigma(u, x) | x \in \Gamma(u) - \{u\}\} = \varepsilon$
                then
                    $C(v) \leftarrow C(v) \cup \{u\}$;
                    for *each* $w \in \Gamma(u) - \{u\}$ **do**
                        if $\sigma(w, u) = \varepsilon$ **then**
                            $q.\text{insert}(u)$;
                        end
                    end
                end
            end
            $MC \leftarrow MC \cup C(v)$;
        end
        // Phase 2: Shrink micro-communities
        $\Delta Q_s \leftarrow 0$;
        for *each* $C \in MC$ **do**
            if $|C| > 1 \wedge \Delta Q_s(C) > 0$ **then**
                $\tilde{v} \leftarrow \{v | v \in C\}$;
                $CR \leftarrow (CR - \underset{v_i \in C}{\cup}\{\{v_i\}\}) \cup \{\tilde{v}\}$;
                $\Delta Q_s \leftarrow \Delta Q_s + \Delta Q_s(C)$;
            end
        end
        if $\Delta Q_s = 0$ **then**
            break;
        end
    end
    $N = \emptyset$;
    for *each* $C \in CR$ **do**
        if $|C| = 1$ **then**
            $CR = CR - C$;
            $N = N \cup C$;
        end
    end
    return *CR, N;*
end

Algorithm 2: SHRINK-G

Input: Network $G = (V, E)$
Output: Set of clusters $CR = \{C_1, C_2, \cdots, C_k\}$; Set of hubs and outliers N

begin
    $CR \leftarrow \{\{v_i\} | v_i \in V\}$;
    for *each* $v \in V$ **do**
        $u \leftarrow v$;
        $L = \emptyset$;
        $L \leftarrow \Gamma(u) - \{u\}$;
        for *each* $l \in L$ **do**
            if $u \leftrightarrow l \wedge \Delta Q_s(\{u, l\}) > 0$ **then**
                $CR \leftarrow (CR - \{\{u\}, \{l\}\}) \cup \{\{u, l\}\}$;
                $L \leftarrow L \cup (\Gamma(l) - \{l\})$;
                $u \leftarrow \{u, l\}$;
            end
        end
    end
    $N = \emptyset$;
    for *each* $C \in CR$ **do**
        if $|C| = 1$ **then**
            $CR = CR - C$;
            $N = N \cup C$;
        end
    end
    return *CR, N;*
end

---

to evaluate the quality of clusters generated by different methods. It is currently widely used in measuring the performance of network clustering algorithms [15]. Formally, the measurement metric NMI can be defined as

$$NMI = \frac{-2 \sum_{i,j} N_{ij} log(\frac{N_{ij}N}{N_{i.}N_{.j}})}{\sum_i N_{i.} log(\frac{N_{i.}}{N}) + \sum_j N_{.j} log(\frac{N_{.j}}{N})}, \quad (5)$$

where N is the confusion matrix, N_{ij} is the number of nodes in both cluster X_i and Y_j, $N_{i.}$ is the sum over row i of N and $N_{.j}$ is the sum over column j of N. Note that the value of NMI ranges between 0.0 (total disagreement) and 1.0 (total agreement).

5.2 Evaluation on Real-world Networks

To assess the performance of the proposed method in terms of accuracy, we conduct experiments on the DBLP Co-authorship network and two popular real-world networks from Newman[1].

5.2.1 DBLP Co-authorship Network

The DBLP Co-authorship network in four research fields (i.e., DB, IR, DM and ML) was extracted from the DBLP computer science bibliographical dataset. We only consider the authors who have published more than twenty papers. Then we obtain a weighted undirected network with 1,547 nodes and 7,789 edges, in which each node corresponds to a distinct author and the edge between two nodes represents their co-author relationship. The integral weight of an edge denotes the number of papers co-authored by these two authors.

Our algorithms SHRINK-H and SHRINK-G get the same clustering result on this network, where 172 communities as well as 162 hubs and 47 outliers are found. Due to the limited space, we can not present all the extracted communities. We then select six representative communities and list no more than ten cluster members along with two representative hubs and outliers in Table 1. Each community represents a group of scientists with the same research interests, such as machine learning community (36) and information retrieval community (147) in Table 1. Here we are able to observe that SHRINK can discover meaningful co-authorship communities from a large amount of real academic associations. The identified hubs indicate some famous researchers who have published a large number of papers in collaboration with a variety of research groups. On the contrary, the identified outliers always correspond to those researchers who may only publish one or few papers coauthored with other scholars. Based on the results, we can see that SHRINK is effective to find the meaningful hubs and outliers from the research communities.

5.2.2 Zarchary's Karate Network

The Zachary's karate network [27] consists of 34 nodes and 78 edges as shown in Figure 5. This network can be separated into two distinct groups by the dashed line since there is a conflict between one of the administrator (represented by node 1) and the instructor (represented by node 33) of the club.

As shown in Figure 5, our algorithms SHRINK-H and SHRINK-G can find four communities in this network represented by different colors. The roles of nodes are repre-

Table 1: Six communities discovered by SHRINK on DBLP Co-authorship network. The last two rows are hubs and outliers associated with the corresponding communities which are labeled by ◇ and △ respectively.

Community [17]	Community[36]	Community[64]	Community[93]	Community[116]	Community[147]
Jon M. Kleinberg	Michael I. Jordan	Jeffrey D. Ullman	Charu C. Aggarwal	Soumen Chakrabarti	James P. Callan
Ravi Kumar	Dan Klein	Michael Stonebraker	Guy M. Lohman	Shashank Pandit	Jaime G. Carbonell
Deepayan Chakrabarti	Zoubin Ghahramani	Yannis Papakonstantinou	Sheng Ma	Sunita Sarawagi	Russell Greiner
Jure Leskovec	Thomas Hofmann	Jim Gray	Vijayshankar Raman	Gaurav Bhalotia	Yiming Yang
David Liben-Nowell	Tao Li	Jinren Zhou	Daniel Barbarĺć	Rushi Desai	Nick Cercone
Ronald Fagin	Chris H. Q. Ding	Sharma Chakravarthy	Joel L. Wolf	B. Aditya	Stephen E. Robertson
Ziv Bar-Yossef	Zhongfei Zhang	Per-Ake Larson	Kun-Lung Wu	Rahul Gupta	Jamie Callan
	Tobias Scheffer	Wolfgang Lehner	Calisto Zuzarte	Byron Dom	Nick Craswell
	John Shawe-Taylor	Cĺesar A. Galindo-Legaria	Chang-Shing Perng		Vibhu O. Mittal
	Eric P. Xing	Janet L. Wiener	Sam Lightstone		Yasushi Ogawa
... ...		... ...	... ...		... ...
◇Christos Faloutsos	◇Nick Koudas	◇Hector Garcia-Molina	◇Jiawei Han	◇Rakesh Agrawal	◇John D. Lafferty
		◇Rajeev Motwani	◇Philip S. Yu	◇S. Sudarshan	
	△Robert A. Jacobs	△John McPherson		△Arpit Mathur	
	△Roded Sharan				

Figure 5: The clustering result of SHRINK on the Zachary's karate network.

sented by different shapes: two hubs denoted by diamonds, six outliers denoted by triangles in the network, and others are general cluster members. In our algorithms, nodes 10 and 20 are identified as hubs. The reason is that these two nodes connected with two adjacent communities in the same way. Hence, it is better for them to be considered as shared nodes (i.e., hubs). Due to the sparse links of this network, nodes 12, 15, 16, 19, 21 and 23 are identified as outliers which are loosely connected with the communities. In short, the SHRINK algorithm can successfully detect the community and identify the hubs and outliers.

Although this partition of four communities in Figure 5 does not match the ground truth of the dataset, many other methods obtain the same result which indicates that it is topologically meaningful. The SCAN algorithm get the same clustering result as our algorithms by using manually detected parameters ($\varepsilon = 0.527$, $\mu = 3$). The BGLL algorithm also find four communities in this network, but it can not find the hubs and outliers and it assigns the nodes 10 and 20 to the community of administrator. We also cluster this network using the CNM algorithm, but it only detects three communities in this network, among which the group of administrator is divided into two unreasonable sub-groups: {1, 5, 6, 7, 11, 12, 17, 20} and {2, 3, 4, 8, 10, 14, 18, 22}. The result of CNM indicates that the agglomerative hierarchical performs badly in greedy modularity maximization.

5.2.3 NCAA College-football Network

The National Collegiate Athletic Association (NCAA) College-football is a social network with communities (or conferences) of American college football teams. In total, there are 115 college football teams, which are divided into eleven conferences and five independent teams (Utah State, Navy, Notre Dame, Connecticut and Central Florida) that do not belong to any conference. The network, representing the schedule of Division I-A games for the 2000 season, contains 115 nodes and 613 edges. Now the question is to find out the communities from the graph. Figure 6(a) illustrates the football network with each vertex represents a school team. The teams belonging to a conference and the independent teams are denoted by circles and diamonds respectively, and teams in the same conference are identified by the same color. There is a link between two teams if they played a game together. The number of teams in a conference ranges from seven to thirteen. Each team plays about ten games in the season. Consequently, the inner link density of each conference is different.

The clustering result of our algorithms SHRINK-H and SHRINK-G is presented in Figure 6(b). We obtain eleven clusters in this network which demonstrates a good match with the original conference system. Four independent teams are correctly identified as hubs. Although there is an independent team that is falsely merged into a conference, and three misclassified teams (i.e., Louisiana Monroe, Louisiana Lafayette, and Louisiana Tech), our algorithm still performs much better than other methods including the SCAN, CNM and BGLL algorithms, which will be described as follows.

The SCAN algorithm finds thirteen communities as its best result in this dataset with parameters ($\varepsilon = 0.53$, $\mu = 2$). The teams in the conference denoted by black circles in Figure 6(a) are divided into two clusters. Meanwhile, five hubs are identified including four correct independent teams: CentralFlorida, Connecticut, Navy, and NotreDame. Another independent team UtahState is misclassified into a conference. The accuracy of SCAN is worse than our algorithm, because it is hard for the SCAN algorithm to detect communities with various densities by using a global density threshold ε. The modularity-based algorithm CNM and BGLL discover seven and ten communities in this network respectively. The algorithm CNM only finds four clusters matching with the conferences. For the five independent teams, they are assigned to three different clusters.

In summary, SHRINK generates promising clustering results along with hubs and outliers in community detection, consistently outperforming baseline methods including the SCAN, CNM and BGLL algorithms.
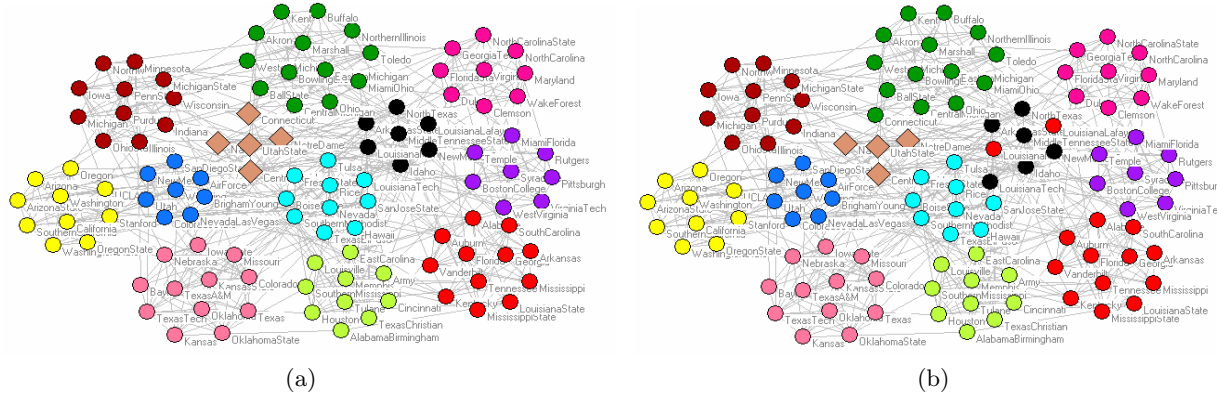
Figure 6: NCAA College-football Network: (a) ground truth, (b) the clustering result of SHRINK.

Table 2: The parameters of the computer-generated datasets for performance evaluation.

Dataset	n	m	k	$maxk$	$minc$	$maxc$
5000S	5,000	48,811	20	50	10	50
5000B	5,000	49,009	20	50	20	100
50000S	50,000	989,737	40	100	50	100
50000B	50,000	990,687	40	100	100	200

5.3 Evaluation on Synthetic Networks

So far, we have presented the experimental results of our algorithms using several real-world networks. Now we also use the Lancichinetti-Fortunato-Radicchi (LFR) benchmark graphs [17, 15] to evaluate the performance of our algorithms. By varying the parameters of the networks, we can analyze the behavior of the algorithms in detail. Some important parameters of the benchmark networks are given in Table 2. We generate several weighted undirected benchmark networks with the number of nodes $n = 5,000$ and 50,000. For each n, two individual networks are generated with different ranges of the community sizes, where S means that the sizes of the communities in the dataset are relatively small and B means that the sizes of communities are relatively big. For each type of dataset, we range the mixing parameter mu from 0.1 to 0.8 with a span of 0.05 and get fifteen networks. Generally, the higher the mixture parameter of a network is, the more difficult it is to reveal the community structure. Some important parameters of the benchmark networks are:

- n: number of nodes
- m: average number of edges
- k: average degree of the nodes
- $maxk$: maximum degree
- mu: mixing parameter, each node shares a fraction mu of its edges with nodes in other communities
- $minc$: minimum for the community sizes
- $maxc$: maximum for the community sizes

Due to the difficulty of detecting the parameter ε in the benchmark networks for the algorithm SCAN, we only compare our algorithm with two baseline methods of modularity optimization: CNM and BGLL. Because these two algorithms both assign each node to just one community, a post-process is used in our algorithms to assign the homeless nodes into the community with largest positive modularity gain. The clustering results of our algorithms SHRINK-H and SHRINK-G are almost the same or only slightly differ-
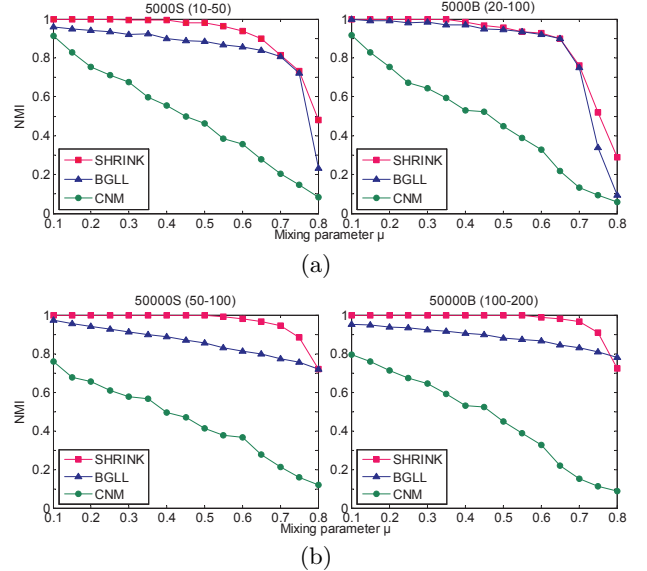

Figure 7: Test of the accuracy of SHRINK, BGLL, and CNM algorithms on the computer-generated benchmark networks.

ent in all generated networks. Thus, we report the average values of these two algorithms. The NMI scores of the three methods are plotted in Figure 7. On most of the benchmark datasets, our algorithm gets NMI = 1 when $mu < 0.5$, which means a perfect match with the original network structure. We can see that the performances of SHRINK are better than that of BGLL on the generated networks in most cases, because the BGLL algorithm tends to produce small number of big communities on the large-scale networks, due to the well known resolution limit of modularity [10]. For the pure modularity optimization algorithm CNM, it performs worse than both BGLL and SHRINK algorithms. However, the performance of our algorithm is decreased when $mu > 0.5$, especially in the small-scale network with big communities (e.g. 5000B). This is because our algorithms have to deal with more and more isolated hubs and outliers with the increasing of parameter mu.

5.4 Analysis of the Resolution Limit Problem

Despite the good performance of the modularity measure on many practical networks, it may lead to apparently un-
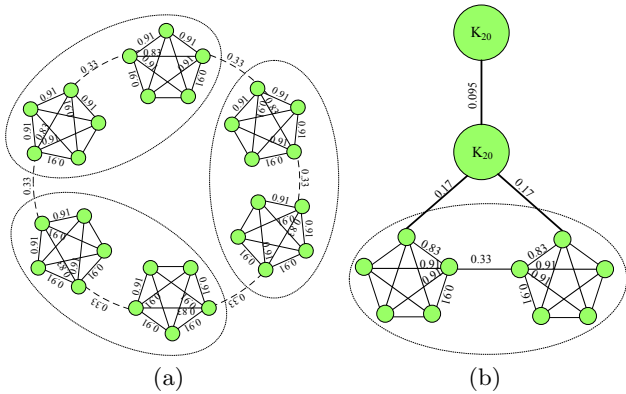
Figure 8: Two schematic networks (the numbers on the edge represent the structural similarity): (a) the Ring network made out of identical cliques connected by single links, and (b) the Pairwise network with four identical cliques.

Table 3: The number of communities on Ring and Pairwise datasets found by SA, CNM, BGLL, and SHRINK.

Dataset				SA	CNM	BGLL	SHRINK
Name	n	m	c				
Ring	150	330	30	15	16	15	**30**
Pairwise	50	404	4	3	3	3	**4**

reasonable partitions in some cases. It has been shown that modularity contains an intrinsic scale that depends on the total number of links in the network. Communities that are smaller than this intrinsic scale may not be resolved, even in the extreme case where they are complete graphs connected by some single bridges. The resolution limit of modularity actually depends on the degree of interconnectedness between pairs of communities and can reach values of the order of the size for the whole network [10].

In Figure 8(a), we show a network consisting of a ring of several cliques, connected through single links. Each clique is a complete graph with n nodes and $n(n-1)/2$ links. Suppose there are c cliques (with c even), the network has a total of $N = nc$ nodes and $M = cn(n-1)/2 + c$ edges. According to [10], modularity optimization would lead to a partition where the cliques are combined into groups of two or more (represented by dotted lines). Here, we use a synthetic dataset with $n = 5$ and $c = 30$, called Ring. Another synthetic network is shown in Figure 8 (b). In this network, the larger circles represent cliques with n nodes, denoted as K_n, and the small cliques with p nodes. According to [10], we set $n = 20$, $p = 5$ and get the network called Pairwise. Modularity optimization merges the two smallest communities into one (shown with a dotted line).

We present the clustering results on the above two datasets in Table 3, where n is the number of node, m is the number of edges, and c is the correct number of communities. Our algorithms SHRINK-H and SHRINK-G find the exact communities. For the Ring and Pairwise datasets, the modularity-based algorithms SA (optimized by simulated annealing), CNM, and BGLL all possess the resolution limit problem which result in merging two small cliques into one cluster.

Following [10], we also conduct experiments on five exam-

Table 4: The real-world datasets for analyzing resolution limit of the modularity-based algorithms and the clustering results by SA, CNM, BGLL, and SHRINK.

Dataset				SA	CNM	BGLL	SHRINK
Name	n	m	$c(Q)$				
Yeast	688	1079	57 (0.677)	9	27	26	**49**
E. coli	423	519	76 (0.661)	27	40	41	**61**
Elect. circuit	512	819	70 (0.640)	11	32	29	**64**
Social	67	182	21 (0.532)	**10**	7	7	**10**
C. elegans	306	2345	20 (0.319)	4	4	6	35

ples of real-world networks: Yeast[2], E. coli[2], Elect. circuit[2], Social[2], and C. elegans[3]. We consider the above five networks as undirected. The datasets and clustering results are listed in Table 4. In most cases, the numbers of communities obtained by our algorithms are the most accurate results, which are very close to the ground truth.

The reason that our algorithms can overcome the resolution limit is that it combines the density-based clustering principle and the modularity measure. The connected nodes with higher similarity will be considered preferentially as in the same community than the lower ones. Moreover, all of the adjacent nodes with equal similarities will be merged in one community or be staying alone.

5.5 Running Time Complexity

Finally, we analyze the computational complexity of our algorithm SHRINK. The running time of SHRINK-H is mainly consumed by finding micro-communities and merging the nodes in them in each iteration. The time complexity is $O(m)$ for the network with m edges. If there are h steps for the algorithm to terminate, the time complexity of is $O(m \cdot h)$. Our tests show that h is always linear in logarithm of the number of nodes n (i.e., $\log n$), which results in an overall time complexity of $O(m \log n)$.

To illustrate the running time of the proposed algorithms SHRINK-H and SHRINK-G, we generate seven networks with the number of nodes n ranging from 1,000 to 300,000. For each network, the number of edges m is ten times of the number of nodes. The running time for SHRINK-H and SHRINK-G are plotted as a function of the number of nodes in Figure 9, respectively. It shows that our algorithm SHRINK-H can process the network of 300,000 nodes within an hour. The greedy clustering algorithm SHRINK-G is faster than the hierarchical one. Actually, we are able to reduce more than half running time of SHRINK-H with the similar performance.

6. CONCLUSIONS

In this paper we present a novel parameter-free network clustering algorithm SHRINK by combining the advantages of density-based clustering and modularity optimization methods. Based on the structural connectivity information, the proposed algorithm can effectively reveal the embedded hierarchical community structure in large-scale weighted undirected networks, and identify hubs and outliers as well. Moreover, it overcomes the sensitive threshold problem of density-based clustering algorithms and the resolution limit possessed by other modularity-based methods. Experimental
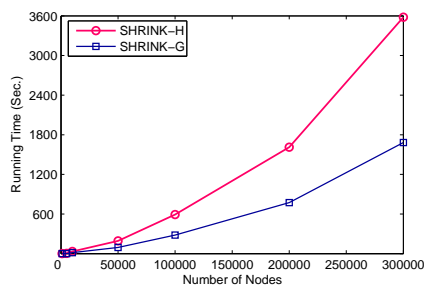
Figure 9: Running time for SHRINK with varying network sizes.

results on the real-world and synthetic datasets show that our algorithm achieves the best performance when compared with the baseline methods. It is efficient with time complexity $O(m \log n)$. In the future, it is interesting to investigate the local communities in large-scale online networks, and to use our method to analyze complex networks in various applications.

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure. In *SIGMOD*, pages 49–60, 1999.

[2] A. Arenas, A. Fernandez, and S. Gomez. Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics*, 10(5):053039, 2008.

[3] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008+, October 2008.

[4] D. Bortner and J. Han. Progressive clustering of networks using structure-connected order of traversal. In *Proc. of ICDE'10*, pages 653–656, 2010.

[5] D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec, and C. Faloutsos. Epidemic thresholds in real networks. *ACM Transactions on Information and System Security*, 10(4):1–26, 2008.

[6] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70(6):066111, Dec 2004.

[7] P. Domingos and M. Richardson. Mining the network value of customers. In *Proc. of KDD'01*, pages 57–66, New York, NY, USA, 2001. ACM.

[8] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD'99*.

[9] Z. Feng, X. Xu, N. Yuruk, and T. A. J. Schweiger. A novel similarity-based modularity function for graph partitioning. In *DaWak'07*, pages 385–396, 2007.

[10] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *PNAS*, 104(1):36, 2007.

[11] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *PNAS*, 99(12):7821–7826, 2002.

[12] R. Guimerà and L. A. Nunes Amaral. Functional cartography of complex metabolic networks. *Nature*, 433(7028):895–900, February 2005.

[13] B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *The Bell System Technical Journal*, 49(1):291–307, 1970.

[14] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *SODA*, pages 668–677, 1998.

[15] A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *Phys. Rev. E*, 80(5):056117, Nov 2009.

[16] A. Lancichinetti, S. Fortunato, and J. Kertesz. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics*, 11(3):033015+, Mar 2009.

[17] A. Lancichinetti, S. Fortunato, and F. Radicchi. Benchmark graphs for testing community detection algorithms. *Phys. Rev. E*, 78(4):046110, Apr 2008.

[18] T. Nepusz, A. Petróczi, L. Négyessy, and F. Bazsó. Fuzzy communities and the concept of bridgeness in complex networks. *Phys. Rev. E*, 77(1):016107, 2008.

[19] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69(6):066133, Jun 2004.

[20] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113, Feb 2004.

[21] G. Palla, I. Derenyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, June 2005.

[22] P. Ronhovde and Z. Nussinov. Multiresolution community detection for megascale networks by information-based replica correlations. *Phys. Rev. E*, 80(1):016109, Jul 2009.

[23] M. Sales-Pardo, R. Guimera, A. A. Moreira, and L. A. N. Amaral. Extracting the hierarchical organization of complex systems. *PNAS*, 104(39):15224–15229, 2007.

[24] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8):888–905, 2000.

[25] S. White and P. Smyth. A spectral clustering approach to finding communities in graph. In *Proc. of SDM'05*, 2005.

[26] X. Xu, N. Yuruk, Z. Feng, and T. Schweiger. SCAN: a structural clustering algorithm for networks. In *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.