Extracting the Core Structure of Social Networks using (α, β)-Community *

Liaoruo Wang[1], John Hopcroft[1], Jing He[2], Hongyu Liang[2], and Supasorn Suwajanakorn[1]

[1] Department of Computer Science, Cornell University, Ithaca, NY 14853, USA
{lw335, jeh17, ss932}@cornell.edu
[2] Institute for Theoretical Computer Science, Tsinghua University, Beijing, China
{he-j08, lianghy08}@mails.tsinghua.edu.cn

Abstract. An (α, β)-community is a connected subgraph C with each vertex in C connected to at least β vertices of C (self-loops counted) and each vertex outside of C connected to at most α vertices of C ($\alpha < \beta$). In this paper, we present a heuristic algorithm that in practice successfully finds a fundamental community structure. We also explore the structure of (α, β)-communities in various social networks. (α, β)-communities are well clustered into a small number of disjoint groups, and there are no isolated (α, β)-communities scattered between these groups. Two (α, β)-communities in the same group have significant overlap, while those in different groups have extremely small pairwise resemblance. A surprising core structure is discovered by taking the intersection of each group of massively overlapping (α, β)-communities. Further, similar experiments on random graphs demonstrate that the core structure found in many social networks is due to their underlying social structure, rather than due to high-degree vertices or a particular degree distribution.

Keywords: core structure, (α, β)-community, community discovery, graph clustering, social networks

1 Introduction

Much of the early work on finding communities in social networks was focused on partitioning the corresponding graph into disjoint communities [1–9]. Conductance was often taken as the measure of the quality of community, and algorithms were sometimes restricted to dense graphs [3, 10–12]. However, to identify well-defined communities in social networks, one needs to realize that an individual may belong to multiple communities at the same time, and is likely to have more connections to individuals outside of his/her community than inside. For example, a person in the theoretical computer science community is likely to

* This research was partially supported by the U.S. Air Force Office of Scientific Research under Grant FA9550-09-1-0675, the National Natural Science Foundation of China under Grant 60553001, the National Basic Research Program of China under Grant 2007CB807900 and 2007CB807901.

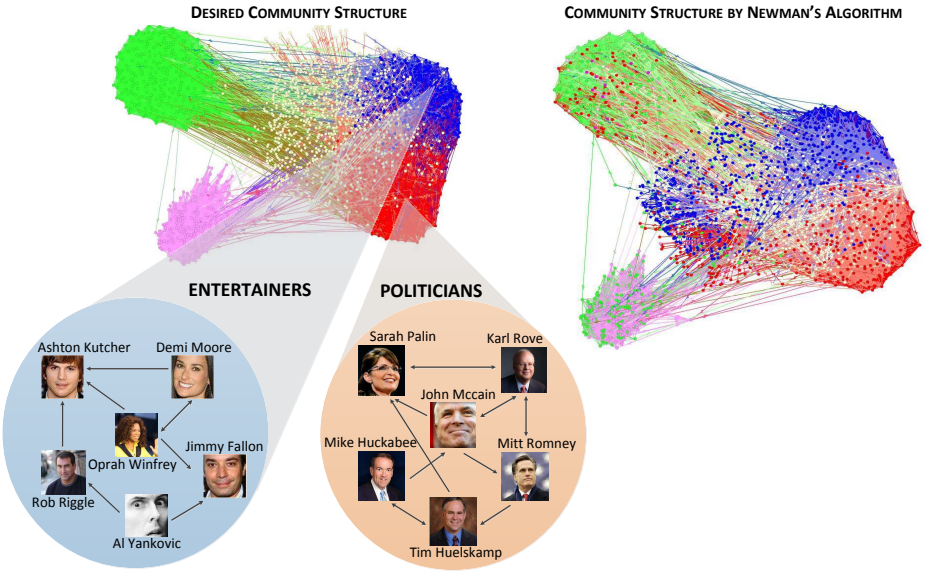


Fig. 1. Case study on the Twitter network (see PDF for colored version). Traditional community detection methods cannot extract the four meaningful communities from their numerous followers (colored yellow). The blue community consists of entertainers and the red community consists of politicians.

have many connections to individuals outside of this community, who may be his/her family and friends, or enroll in his/her institution, or attend his/her religious group. One approach to finding such overlapping communities is that of Mishra et al. [13], in which the concept of (α, β)-community was introduced and algorithms were given for finding an (α, β)-community in dense graphs, provided there exists a champion in the community. A champion of a community is an individual with a bounded number of neighbors outside of the community.

We present a case study on the Twitter network to evaluate the community structure found by many graph partitioning methods, as shown in Fig. 1. The left figure gives a fundamental structure with four meaningful communities (blue, red, green, and pink) extracted from their numerous followers (yellow nodes). Some community members are enlarged to highlight the details. Interestingly, the blue one consists of a group of well-known entertainers and the red one consists of a group of active politicians. The right figure shows the four communities obtained by Newman's modularity-based algorithm [7]. By contrast, most of the yellow nodes are grouped into one of the four communities, and the communities are heavily blended with each other. Thus, this example reveals that traditional community detection methods fail to discover the desired community structure in many cases.

In this paper, we give a definition of (α, β)-community slightly different from that of Mishra et al. [13]. Without fixing the values of α and β, our defini-

tion highlights the contrast of internal and external connectivity. We develop a heuristic algorithm based on (α, β)-community that in practice efficiently finds a fundamental community structure. Our algorithm is focused on the difference $\beta - \alpha$ and is thus robust to the specific values of α and β. Further, we thoroughly explore the structure of (α, β)-communities in various large social networks. In a Twitter following network with 112,957 vertices and 481,591 edges, there are 6,912 distinct (α, β)-communities of size 200 with 45,361 runs of the algorithm. These (α, β)-communities are neatly categorized into a small number of massively overlapping clusters. Two (α, β)-communities in the same cluster have significant overlap ($> 90\%$), while two (α, β)-communities in different clusters have extremely small ($< 5\%$) pairwise resemblance. This leads to the notion of core, which is the intersection of a group of massively overlapping (α, β)-communities. Our definition provides an intuitive criterion as to whether to classify a subgraph as a community. The edges connecting each vertex in the community to vertices of the community should be strictly more than those connecting any vertex outside the community to vertices of the community. Further, by taking the intersection of a number of massively overlapping (α, β)-communities, the set of (α, β)-communities which differ by only a few vertices is reduced to an underlying core. Thus, each (α, β)-community contains one of the few cores and some peripheral vertices, and these peripheral vertices are what gives rise to such a large number of (α, β)-communities.

We can extract the core structure by taking the intersection of a group of massively overlapping (α, β)-communities with multiple runs of the algorithm. The number of cores decreases as k increases. For large k, the (α, β)-communities are well clustered into a small number of disjoint cores, and there are no isolated (α, β)-communities scattered between these cores. The cores obtained for a small k either disappear or merge into the cores obtained for a larger k. Further, the cores correspond to dense regions of the graph, and there are no bridges of intermediate (α, β)-communities connecting one core to another. By contrast, the cores found in various random graphs usually have significant overlap among them, and the number of cores does not necessarily decrease as k increases. Extensive experiments demonstrate that the core structure found in various social networks is indeed due to their underlying social structure, rather than due to high-degree vertices or a particular degree distribution.

The number and average size of cores in the Twitter graph with respect to the community size k are given in Table 1. As k increases, some cores disappear due to their small neighborhood (Definition 2), while others merge into larger ones due to their high closeness (Definition 3). We explore some interesting questions in this paper, for example, what causes many social networks to display the core structure, why (α, β)-communities correspond to well-defined clusters, and why there are no bridges of (α, β)-communities connecting one core to another. A bridge is a sequence of intermediate (α, β)-communities that connect two cores with substantial overlap between adjacent pairs.

The rest of this paper is organized as follows. We discuss related work in Section 2, and introduce the definition of (α, β)-community in Section 3. Then, we

prove the NP-hardness of finding an (α, β)-community and present the heuristic (α, β)-COMMUNITY algorithm in Section 4. In Section 5, we apply the algorithm to various social and random graphs to demonstrate, explore, and analyze the core structure found in many social networks. Finally, we conclude in Section 6 with comments on the problems considered and future work.

2 Related Work

A closely related concept to (α, β)-community is that of degree core [32–35]. Given degree d, a degree core of a graph G is a maximal connected subgraph of G in which all vertices have degree at least d. Equivalently, it is one of the connected components of the subgraph of G formed by repeatedly deleting all vertices of degree less than d. Every d-core is a $(d-1, d+1)$-community, but there are many (α, β)-communities that are not degree cores. The concept of (α, β)-community can capture some structural properties of large social networks that other methods (such as degree core) method cannot discover. Degree cores tend to identify subsets of high-degree vertices as communities, while the concept of (α, β)-community highlights more the contrast of intra- and inter-connectivity. This is a natural type of community that we are interested in. I don't have to be a star to belong to some community, but I should belong to this community if I have (many) more connections inside this community than anybody outside this community does. We will further compare these two methods in Section 5.1.

A substantial amount of work has been devoted to the task of identifying and evaluating close-knit communities in large social networks, most of which is based on the premise that it is a matter of common experience that communities exist in these networks [3]. A community was often considered to be a subset of vertices that are densely connected internally but sparsely connected to the rest of the network [3–6, 9]. For example, Newman constructed the measure of betweenness and modularity to partition a social network into disjoint communities [5, 6]. Andersen et al. [9] proposed a local graph partitioning algorithm based on personalized PageRank vectors. An information-theoretic framework was also established to obtain an optimal partition and to find communities at multiple levels [14, 15]. However, communities can overlap and may also have dense external connections. Mishra et al. [13] proposed the concept of (α, β)-community and algorithms to efficiently find such communities. Ahn et al. [16] provided a novel perspective for finding hierarchical community structure by categorizing links instead of vertices.

A range of community detection methods have been empirically evaluated and compared in [17]. Further, community detection problem has been extended to handle query-dependent cases [18]. Many studies combined link and content information for finding meaningful communities [19, 20]. The dynamic behavior of communities was also extensively explored in previous work [21–24]. Other models have been proposed to improve the accuracy of community detection in different scenarios [25–28]. New measures have also been proposed to better evaluate the quality of community [29, 30]. For example, Zhang et al. [25] pro-

posed a novel community detection algorithm that employs a dynamic process by contradicting network topology and topology-based propinquity. Maiya and Berger-Wolf [27] utilized a novel method based on expander graphs to sample communities in networks. Yang et al. [31] explored a dynamic stochastic block model for finding communities and their evolution in dynamic social networks.

However, most existing work on community detection has not considered the existence of core structure in many social networks. It has also been ignored that many communities actually have a large number of external connections. In this paper, we demonstrate and explore the core structure, and propose a heuristic algorithm to extract cores from large networks.

3 Preliminaries

The concept of (α, β)-community was proposed by Mishra et al. [13] as a powerful tool for graph clustering and community discovery. In [13], an (α, β)-community refers to a cluster of vertices with each vertex in the cluster adjacent to at least a β-fraction of the cluster and each vertex outside of the cluster adjacent to at most an α-fraction of the cluster. Without loss of generality, we adopt a slightly different definition in this paper.

Given a subset of vertices $S \subseteq V$, for any $v \notin S$, $\alpha(v)$ is defined as the number of edges between v and vertices of S. Similarly, for any $w \in S$, $\beta(w)$ is defined as the number of edges between w and vertices of S (self-loop counted). Then, we define $\alpha(S) = \max\{\alpha(v)|v \notin S\}$ and $\beta(S) = \min\{\beta(w)|w \in S\}$.

Definition 1. *Given a graph $G = (V, E)$ with self-loops, a subset of vertices $C \subseteq V$ is called an* (α, β)-community *if each vertex in C is connected to at least β vertices of C (self-loop counted) and each vertex outside of C is connected to at most α vertices of C ($\alpha < \beta$). That is, $\alpha = \alpha(C) < \beta = \beta(C)$.*

Definition 1 is equivalent to that of [13] where C is a $(\alpha(C)/|C|, \beta(C)/|C|)$-cluster with $\alpha(C) < \beta(C)$. It acknowledges the importance of self-loops: although a maximal clique should intuitively be a community, this cannot be guaranteed without self-loops. An (α, β)-community in a graph G is called *proper* if it corresponds to a non-empty proper subgraph of G.

A maximal clique is guaranteed to be an (α, β)-community since self-loops are counted by Definition 1. Thus, every graph that is not a clique must contain an (α, β)-community (or, a maximal clique) as a proper subgraph. Starting with any vertex, either it is a proper (α, β)-community or there must be another vertex connected to it. Then, two vertices connected by an edge either form a proper (α, β)-community or there must be a third vertex connected to both. Continue this argument until a proper (α, β)-community is found or all vertices are included in a clique, contradicting the assumption that the graph is not a clique. Thus, we have the following theorem:

Theorem 1. *A non-complete graph must contain a proper (α, β)-community.*

Given an integer k and a graph G with self-loops, define k-COMMUNITY as the problem of finding an (α, β)-community of size k in G. Given an integer k and a graph G, define k-CLIQUE as the problem of determining whether there exists a clique of size k in G.

Theorem 2. *The k-COMMUNITY problem is* NP*-hard.*

Proof. We will show that if k-COMMUNITY is polynomial-time solvable, so is k-CLIQUE, which is a well-known NP-hard problem.

Let $\{k, G = (V, E)\}$ be an input to the k-CLIQUE problem, where the goal is to decide whether G contains a clique of size k. Without loss of generality, assume that G is not a clique and $k \geqslant 3$. Let $n = |V|$ and for each ℓ such that $k \leqslant \ell \leqslant n-1$, construct a graph $H_\ell = (V_\ell, E_\ell)$ as follows:

$$V_\ell = V_{\ell,1} \cup V_{\ell,2}, V_{\ell,1} = \{x_i | 1 \leqslant i \leqslant n+\ell+1\}, V_{\ell,2} = \{y_j | 1 \leqslant j \leqslant \ell+1\};$$
$$E_\ell = \{(x_{i_1}, x_{i_2}) \,|1 \leqslant i_1 < i_2 \leqslant n+\ell+1\} \cup \{(y_{j_1}, y_{j_2}) \,|1 \leqslant j_1 < j_2 \leqslant \ell+1\} \cup$$
$$\{(y_j, x_i) \,|1 \leqslant j \leqslant \ell+1, 1 \leqslant i \leqslant \ell-1\}.$$

H_ℓ contains two cliques of size $n+\ell+1$ and $\ell+1$, where each vertex of the second clique is connected to a fixed subset of $\ell-1$ vertices of the first clique. Let $G_\ell = G^* \cup H_\ell^*$, where G^* and H_ℓ^* are obtained by adding self-loops to all the vertices. Note that G^* and H_ℓ^* are disjoint.

The graph G has a clique of size k if and only if it has a maximal clique of size ℓ, $k \leqslant \ell \leqslant n-1$. Then, we proceed to prove that G has a maximal clique of size ℓ if and only if G_m contains an (α, β)-community of size $n+2\ell+1$. Assume that G contains a maximal clique on the subset $V' \subseteq V$ with $|V'| = \ell$. Let $S = V' \cup V_{\ell,1}$, and clearly, $\beta(S) = \ell$. By the maximality of V', each vertex in $V - V'$ is adjacent to at most $\ell-1$ vertices in V'. Further, by the construction of H_ℓ, each vertex in $V_{\ell,2}$ is adjacent to $\ell-1$ vertices in $V_{\ell,1}$. Thus, S is an (α, β)-community of size $n+2\ell+1$ since $\alpha(S) = \ell - 1 < \beta(S)$.

Now, assume that G_ℓ has an (α, β)-community S of size $n+2\ell+1$. Since the subset S contains at least $(n+2\ell+1) - (n+\ell+1) = \ell$ vertices in $V_{\ell,1}$, there exists at least one vertex $v \in S \cap V_{\ell,1}$ that is not connected to any vertex in $V_{\ell,2}$. Suppose that S contains k vertices in $V_{\ell,1}$, $\ell \leqslant k \leqslant n+\ell+1$, and thus $\beta(S) \leqslant \beta(v) = k$. If $k < |V_{\ell,1}|$, there exists at least one vertex outside of S adjacent to k vertices in S, leading to $\alpha(S) \geqslant k \geqslant \beta(S)$ which contradicts the definition of (α, β)-community. Hence, $V_{\ell,1} \subseteq S$.

Suppose that there exists some vertex $y_j \in S \cap V_{\ell,2}$, i.e. $|S \cap V_{\ell,2}| \geqslant 1$. Since $|S| - |V_{\ell,1}| = \ell < |V_{\ell,2}|$, at least one vertex in $V_{\ell,2}$ is outside of S. Note that $V_{\ell,2}$ is a clique and each vertex in $V_{\ell,2}$ is connected to $\ell-1$ vertices in $V_{\ell,1}$. Thus, $\beta(S) \leqslant (\ell-1) + |S \cap V_{\ell,2}|$ and $\alpha(S) \geqslant (\ell-1) + |S \cap V_{\ell,2}|$, which contradict the assumption that S is an (α, β)-community. Then, $|S \cap V_{\ell,2}| = 0$ and the remaining ℓ vertices of S are all from V. Recall that $\alpha(S) \geqslant \ell - 1$ and there are no edges between V and $V_{\ell,1}$. If $S - V_{\ell,1}$ is not a clique, then $\beta(S) \leqslant \ell - 1 \leqslant \alpha(S)$, again leading to a contradiction. Hence, $S - V_{\ell,1}$ is a clique and $\beta(S) = \ell$. $S - V_{\ell,1}$ is also a maximal clique of size ℓ, since $\alpha(S) < \beta(S) = \ell$. Therefore, we have completed

the proof by constructing a correspondence between the k-COMMUNITY problem and the k-CLIQUE problem. □

4 Algorithm

In this section, we give a heuristic algorithm for finding an (α, β)-community of size at least k in a graph $G = (V, E)$. Starting with a random subset $S \subseteq V$ of k vertices, the algorithm proceeds as follows. If $\alpha(S) > \beta(S)$, swap a vertex in S with the lowest β-value and a vertex outside of S with the highest α-value. Each such swap increases the value of $\sum_{v \in S} \beta(v)$ by $-(2\beta-1)+(2\alpha+1) = 2(\alpha-\beta)+2$ if the two vertices are not connected, or by $-(2\beta-1)+(2\alpha-1) = 2(\alpha-\beta)$ if the two vertices are connected by an edge. Note that $\sum_{v \notin S} \alpha(v)$ may also increase upon each such swap. Since $\sum_{v \in S} \beta(v)$ cannot increase infinitely, the algorithm either returns an (α, β)-community S or reaches a state in which $\alpha(S) = \beta(S)$.

Let $A = \{v \in V - S | \alpha(v) = \alpha(S)\}$ and $B = \{w \in S | \beta(w) = \beta(S)\}$ denote the two subsets of vertices with the highest α-value and the lowest β-value. If $\alpha(S) = \beta(S)$, the algorithm finds a pair of vertices $a \in A$ and $b \in B$ that are not connected, if such a pair exists, and swaps a and b. Since self-loops are counted, the sum $\sum_{v \in S} \beta(v)$ is increased by two, as illustrated in Fig. 2. Then, the condition $\alpha(S) = \beta(S)$ may no longer hold such that the algorithm continues swapping a vertex in S with the lowest β-value and a vertex outside of S with the highest α-value. Again, since $\sum_{v \in S} \beta(v)$ cannot increase infinitely, the algorithm will find either an (α, β)-community S or the case when $\alpha(S) = \beta(S)$ and the sets A and B form a bi-clique. In the latter situation, if a vertex $v \in A$ is not connected to any other vertex in A, adding v to S will increase $\beta(S)$ by one but not increase $\alpha(S)$, thus obtaining an (α, β)-community. Similarly, removing a vertex $w \in B$ that is not connected to any other vertex in B will also produce an (α, β)-community.

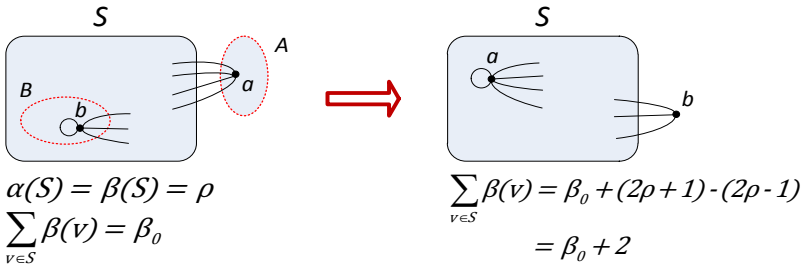


Fig. 2. The (α, β)-COMMUNITY algorithm.

Thus, upon termination, the algorithm returns either an (α, β)-community or a subset $S \subseteq V$ where $\alpha(S) = \beta(S)$ and the sets A and B form a bi-clique. Further, neither A nor B has an isolated vertex in the corresponding subgraphs

induced by the two sets. Then, we simply add all the vertices in A to S and start the algorithm over. Though we cannot guarantee to find an (α, β)-community due to this latter case, in practice when k is not too small (e.g. $\leqslant 20$), we never run into the bi-clique situation and thus always find an (α, β)-community.

A mathematical description of this (α, β)-COMMUNITY algorithm, along with a subroutine called SWAPPING, is given below. Three corollaries are also given to demonstrate the correctness and proper termination of the SWAPPING algorithm. Their proofs are straightforward and thus omitted from this paper.

Algorithm 1 (α, β)-COMMUNITY$(G = (V, E), k)$

$S \leftarrow$ a random subset of k vertices
while $\beta(S) \leqslant \alpha(S)$ **do**
  $S \leftarrow$ SWAPPING(G, S)
  $A \leftarrow \{v \notin S \mid \alpha(v) = \alpha(S)\}$
  $B \leftarrow \{v \in S \mid \beta(v) = \beta(S)\}$
  if $\{(a_i, b_j) \notin E \mid a_i \in A, b_j \in B\} \neq \emptyset$ **then**
    pick such a pair of vertices (a_i, b_j)
    $S \leftarrow (S - \{b_j\}) \cup \{a_i\}$
  else if $\{a_i \in A \mid (a_i, a_k) \notin E, \forall a_k \in A, k \neq i\} \neq \emptyset$ **then**
    pick such a vertex a_i
    $S \leftarrow S \cup \{a_i\}$
  else if $\{b_j \in B \mid (b_j, b_k) \notin E, \forall b_k \in B, k \neq j\} \neq \emptyset$ **then**
    pick such a vertex b_j
    $S \leftarrow S - \{b_j\}$
  else
    $S \leftarrow S \cup A$
  end if
end while
return S

Algorithm 2 SWAPPING$(G = (V, E), S)$

while $\beta(S) < \alpha(S)$ **do**
  $A \leftarrow \{v \notin S \mid \alpha(v) = \alpha(S)\}$
  $B \leftarrow \{v \in S \mid \beta(v) = \beta(S)\}$
  pick a vertex $a \in A$ and a vertex $b \in B$
  $S \leftarrow (S - \{b\}) \cup \{a\}$
end while
return S

Corollary 1. *Each iteration of* SWAPPING *strictly increases* $\sum_{v \in S} \beta(v)$.

Corollary 2. SWAPPING *always terminates. When it terminates, swapping any pair of vertices in* A *and* B *will not increase* $\sum_{v \in S} \beta(v)$.

Corollary 3. SWAPPING *returns a subset of vertices* S *with* $\beta(S) \geqslant \alpha(S)$.

5 Experimental Results

In this section, we conduct experiments on a number of social and random graphs to demonstrate, explore, and analyze the core structure.

5.1 Twitter

The Twitter dataset [36] was crawled in 2009 from the online social networking and microblogging service `Twitter.com` that contains friendship links among a group of Twitter users. Each vertex represents a Twitter user account, and each edge represents a following relation. For simplicity, we consider this graph as undirected, ignoring the direction of the edges and combining multiple edges with the same endpoints. Further, we remove the isolated and degree-one vertices from the graph to discard the insignificant outliers. This results in a smaller graph of 112,957 vertices and 481,591 edges with average degree 8.52.

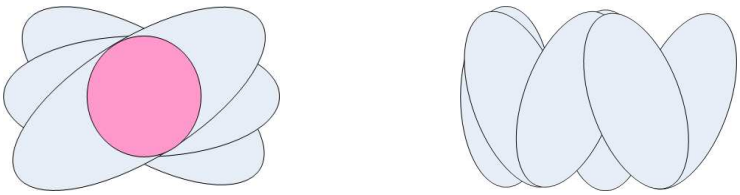
Starting with random subsets of size k, the (α, β)-COMMUNITY algorithm is applied to the Twitter graph for finding (α, β)-communities. Theoretically, this algorithm is not guaranteed to terminate within a reasonable amount of running time, thus we specify an upper bound (e.g. 1,000) on the number of iterations. However, in practice, we rarely observe the case of not finding any (α, β)-community within 1,000 iterations of the algorithm.

In most cases, 500 runs of the algorithm return 500 (α, β)-communities. However, more than 45,000 runs of the algorithm return only 6,912 distinct (α, β)-communities for $k = 200$, which gives an estimate of the number of (α, β)-communities in the Twitter graph. Surprisingly, these (α, β)-communities are all clustered into a small number of disjoint groups. Two (α, β)-communities in the same group share a resemblance higher than 0.9 and differ by only a few vertices, while two (α, β)-communities in different groups share a resemblance lower than 0.06. Here, the pairwise resemblance (a.k.a Jaccard index) $r(A, B)$ between two sets A and B is defined as:

$$r(A, B) = \frac{|A \cap B|}{|A \cup B|}.$$

Thus, the (α, β)-communities form a "core" overlapping structure rather than a "chain" overlapping structure, as shown in Fig. 3. Further, the intersection of the (α, β)-communities in each group has an over 75% resemblance with every single (α, β)-community in that group. At $k = 200$, all the 6,912 (α, β)-communities found in the Twitter graph cluster into 4 "cores". The "cores" are disjoint from each other and correspond to dense regions of the graph. In contrast to what we would have expected, there are no isolated (α, β)-communities scattered between these densely-clustered "cores".

For a group of massively overlapping (α, β)-communities, we define the *core* to be the intersection of those (α, β)-communities. The number of cores can be



(a) Core (b) Chain

Fig. 3. The overlapping structure.

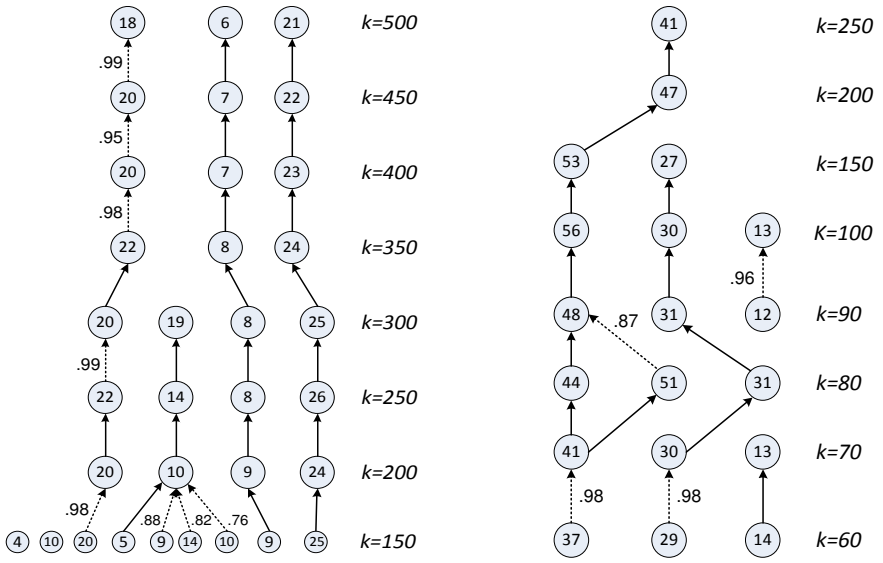
determined by computing the resemblance matrix of all the (α, β)-communities. Then, the (α, β)-communities can be categorized in a way that any two (α, β)-communities in the same category are similar to each other. A pairwise resemblance is considered sufficiently large if it is greater than 0.6, while in practice we frequently observe resemblance greater than 0.9. Thus, the cores can be obtained by taking the intersection of all the (α, β)-communities in each category. The number of cores is simply the number of blocks along the diagonal of the resemblance matrix. The number and average size of cores in the Twitter graph with respect to the community size k are given in Table 1.

Table 1. Cores in the Twitter graph.

k	25	50	100	150	200	250	300	350	400	450	500
number of cores	221	94	19	9	4	4	4	3	3	3	3
average core size	23	45	73	112	151	216	276	332	364	402	440

Observation. The number of cores decreases as the size k increases. This number becomes relatively small when k is large, and will eventually decrease to one as k further increases. Thus, (α, β)-communities are well clustered into a small number of cores before gradually merging into one large core. For example, the (α, β)-communities are clustered into 9 cores for $k = 150$ and 4 cores for $k = 200$, where the cores are disjoint in both cases. As k increases, the cores obtained for a small k either disappear or merge into the cores obtained for a larger k. A layered tree diagram is given to illustrate this phenomenon in Fig. 4(a).

Each level in the tree diagram contains the cores obtained for the corresponding size k. For a pair of cores in adjacent levels, a directed arrow is added from lower to upper level if they have significant overlap, that is, a substantial fraction (e.g. 60%) of vertices in the lower-level core is contained in the upper-level core. If this fraction of overlap is smaller than one, a dotted arrow labeled with the fraction is added to represent a partial merge. Otherwise, a solid arrow is added to represent a full merge. As shown in Fig. 4(a), the fraction of overlap is close to



(a) Twitter (b) Slashdot

Fig. 4. The tree diagram for Twitter and Slashdot. (Each circle represents a core obtained for a given k, in which the integer denotes its β value. Each dotted arrow represents a partial merge with the fraction of overlap labeled, and each solid arrow represents a full merge.)

one as we move up the tree. Thus, a lower-level core is (almost) entirely merged into an upper-level core.

The definition of (α, β)-community allows a community to have more edges connecting it to the rest of the graph than those connecting within itself. Empirically, there are many more vertices outside of an (α, β)-community, and thus the cut edges are almost always more than the internal edges. This definition provides an intuitive criterion as to whether to classify a subgraph as a community. The edges connecting each vertex in the community to vertices of the community should be strictly more than those connecting any vertex outside the community to vertices of the community. Further, by taking the intersection of a number of massively overlapping (α, β)-communities, the set of (α, β)-communities which differ by only a few vertices is reduced to an underlying core. Thus, each (α, β)-community contains one of the few cores and some peripheral vertices, and these peripheral vertices are what gives rise to such a large number of (α, β)-communities.

Analysis. One question is what causes the Twitter graph to display this core structure, and further, why the graph shows only a small number of disjoint cores for a large size k. As shown later, this is due to the fact that an underlying social structure, as opposed to randomness, exists in the Twitter network. To

take a closer look into this, we simplify the Twitter graph by removing low-degree vertices, i.e. vertices of degree lower than 19, and then obtain a smaller graph with 4,144 vertices and 99,345 edges. The minimum β value for most (α, β)-communities is 19, thus this will discard the less important low-degree vertices without destroying the fundamental structure. The (α, β)-COMMUNITY algorithm is applied to this simplified graph for $k = 200, 250, 300, 350, 400$, and we obtain exactly two disjoint cores in each case. For any two adjacent levels in the corresponding tree diagram, the two lower-level cores are completely contained in the upper-level cores. One reason for such a small number of cores could be that the vertices in the two cores are more "powerful" in pulling other vertices toward them. If we remove the two cores from the graph and repeat the experiment for $k = 200$, then (α, β)-communities are no longer clustered and form a large number of scattered communities.

Another question is why there are exactly two cores in the simplified graph. Define S_1 and S_2 as the two cores obtained for $k = 200$. Then, S_1 corresponds to a fairly dense subgraph with 156 vertices and 3,029 edges, in which the minimum degree is 23 and the average degree is 38.8. S_2 has 159 vertices and 2,577 edges, in which the minimum degree is 19 and the average degree is 32.4. Surprisingly, there are only 105 cross edges between S_1 and S_2, while 110 (70%) vertices of S_1 and 100 (63%) vertices of S_2 are not associated with any cross edge. Thus, S_1 and S_2 correspond to two subsets of vertices that are densely connected internally but sparsely connected with each other. As a result, they are returned as the cores of two groups of massively overlapping (α, β)-communities.

Disappear and Merge. We have observed that, in the Twitter graph, a core obtained for some k disappears from the tree diagram as k increases, and two cores obtained for some k merge into a larger core as k increases. By examining these interesting phenomena, we discover that the disappearance of a core is possibly due to its small effective neighborhood, and the merging of two cores is possibly due to their high closeness. Now, we give the following definitions:

Definition 2. *The* neighborhood *of a core S is defined as a subset of vertices that are more closely connected to S than any other core.*

The neighborhood of a core can be determined by an iterative process. Any vertex with more connections to one core than any other must belong to the neighborhood of that core. Thus, these vertices can be associated with some core in the first iteration, and we call them tier-1 neighbors. Then, any vertex with more connections to one core and its tier-1 neighbors should also belong to the neighborhood of that core. Thus, these vertices can be associated with some core in the second iteration, and we call them tier-2 neighbors. This process can be recursively performed until no more vertices can be categorized into any neighborhood.

Definition 3. *The* closeness *between two cores S_1 and S_2 is defined as their cross-edge density, i.e.*

$$c(S_1, S_2) = \frac{|\{(v, w) \in E \mid v \in S_1, w \in S_2\}|}{|S_1| \cdot |S_2|},$$

where $|S_1|$ and $|S_2|$ denote the number of vertices in S_1 and S_2. This is also an alternate definition of the conductance of a cut.

If a core has a small neighborhood, then there are many low-degree vertices in the neighborhood that do not contribute to the SWAPPING algorithm. Thus, vertices in an adjacent neighborhood are likely to be swapped in, since they may also have a large number of connections to the core and its neighborhood. As the adjacent neighborhood becomes dominant in the algorithm, the vertices in the starting subset are gradually replaced by the vertices in that adjacent neighborhood. Then, the algorithm converges to the corresponding core, causing the initial core to disappear. We notice that the adjacent neighborhood to which the algorithm converges is usually much larger than the small neighborhood of the initial core.

Further, we observe that two cores with comparative size of neighborhood combine to form a larger core as k increases. In such cases, these two cores are very close to each other, and the strong interconnection between them becomes dominant such that they merge rather than disappear, even if they both have small neighborhood. Thus, two cores with high closeness value will merge to form a larger core as k increases.

An example is given in Fig. 5 to illustrate the disappearance and merging of cores in the Twitter graph. We obtain 8 cores for $k = 150$. Two cores have fairly small neighborhood, and thus disappear as k increases to 250. Three cores have comparative size of neighborhood and significantly high pairwise closeness, and thus merge to form a larger core as k increases to 250. Hence, we obtain 4 cores for $k = 250$. Further, as k increases from 250 to 350, two cores merge and we obtain three cores. One core has a relatively small neighborhood compared with the others. As k continues to increase to 500, this core eventually disappears.

Bridge. A *bridge* between two cores S_1 and S_m is a sequence of intermediate (α, β)-communities $S_2, \cdots, S_{m-1}$, where the pairwise resemblance is large between adjacent subsets but small between the first and last subsets, e.g. $r(S_1, S_m) < 0.3$ and $r(S_i, S_{i+1}) > 0.6$ for all $i \in \{1, 2, \cdots, m-1\}$. The *length* of the bridge is thus given by $m-1$. Recall that for $k = 200$, (α, β)-communities are all clustered into 4 disjoint cores, and no bridge is detected between any two cores. However, the possible bias of our algorithm might prevent a bridge from being found in the Twitter graph. Then, the following experiments are designed to determine whether there exists a bridge.

Select any two cores obtained for $k = 200$ and perform the following steps repeatedly. Randomly pick r vertices from one core and $200 - r$ vertices from the other to form an initial subset of size 200, and apply the (α, β)-COMMUNITY algorithm to this subset. If every run returns an (α, β)-community substantially overlapping with one core but disjoint from the other, then it suggests that there does not exist any bridge between the two cores. With 100 runs of the algorithm, 99 return such an (α, β)-community, and only one returns an (α, β)-community C that contains 95.54% of one core A and 26.22% of the other core B. However,

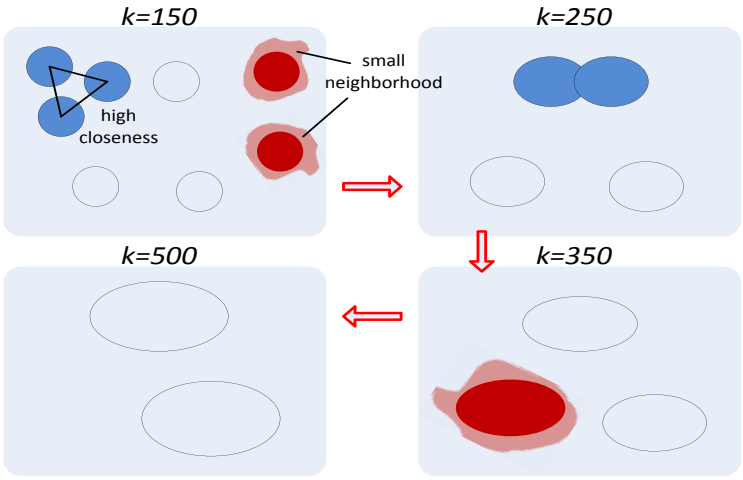


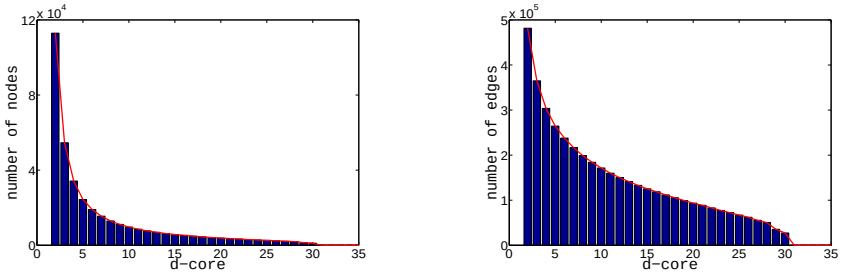
Fig. 5. The disappearing and merging of cores in the Twitter graph (see PDF for colored version). The disappearing cores are colored red and the merging cores are colored blue.

no other intermediate (α, β)-communities can be found between B and C using the same approach, which demonstrates the absence of bridge.

Another approach to finding a bridge is to search for (α, β)-communities that fall between cores. Generate random subsets of size 200 and run the (α, β)-COMMUNITY algorithm repeatedly. After 4 disjoint cores have been obtained with 500 runs of the algorithm, (α, β)-communities returned by another 45,361 runs are compared with the 4 cores to check whether there is any intermediate (α, β)-community. This approach is also useful for estimating the total number of (α, β)-communities of a given size. No intermediate (α, β)-communities are found, however, only 6,912 distinct (α, β)-communities are obtained, which indicates a relatively small number of (α, β)-communities of size 200 and/or a possible bias of our algorithm that favors some communities over others.

Overall, these experiments have suggested that there is no bridge between cores, that is, there is no sequence of intermediate (α, β)-communities that connect two cores with substantial overlap between adjacent pairs. The absence of bridge demonstrates the underlying social structure of the Twitter network with (α, β)-communities neatly clustered into a few disjoint cores.

Degree Core. We conduct experiments on the same Twitter dataset using the degree core method. When $d = 9$, it returns one connected subgraph of 11,133 nodes and 184,146 edges. When $d = 20$, it returns one connected subgraph of 3,835 nodes and 93,533 edges. When $d = 30$, it returns one connected subgraph of 1,127 nodes and 27,344 edges. The degree core method always identifies one connected subgraph of high-degree vertices as community, as shown in Fig. 6.



(a) # nodes vs. d-core (b) # edges vs. d-core

Fig. 6. The degree core method.

This means, while degree cores tend to identify subsets of high-degree vertices as communities, the concept of (α, β)-community highlights more the contrast of inter- and intra-connectivity. Our analysis is robust to the specific values of α and β, in particular, we believe the positive difference $\beta - \alpha$ gives a strong intuitive indication of community, rather than their absolute values. The greater $\beta - \alpha$ is, the better. This concept gives a natural type of community that we are interested in. I don't have to be a star to belong to some community, but I should belong to this community if I have (many) more connections inside this community than anybody outside this community does.

5.2 Slashdot

Slashdot is a technology-related news website known for its professional user community. The website features contemporary technology-oriented news submitted by users and evaluated by editors. Slashdot introduced the Slashdot Zoo feature in 2002, allowing users to tag others as friends or foes. The social network based on common interest shared by Slashdot users was obtained and released by Leskovec et al. [3] in February 2009.

The Slashdot graph has 82,168 vertices and 504,230 edges, with an average degree of 12.3. Our heuristic algorithm discovers a core structure similar to that of Twitter. As in the Twitter graph, the number of cores decreases as the community size k increases and becomes relatively small for large k. The cores found in the Slashdot graph are almost disjoint from each other, with few edges connecting in between, and they correspond to dense regions of the graph. Thus, this suggests that (α, β)-communities are well clustered into a small number of disjoint cores for large k. For example, (α, β)-communities are clustered into three nearly disjoint cores for $k = 100$, where only 171 edges connect the two cores of size 93 and 100 with 2,142 and 1,105 internal edges, respectively. As k increases, the cores obtained for a small k either disappear (due to their small neighborhood), or merge into the cores obtained for a larger k (due to their high closeness). A layered tree diagram is given to illustrate this phenomenon in the Slashdot graph, as shown in Fig. 4(b).

5.3 Coauthor

The Coauthor dataset was crawled from the e-print arXiv that contains scientific coauthorship between authors of the papers submitted to the hep-ph archive [37]. If author i coauthors a paper with author j, there is an undirected edge between vertex i and vertex j in the corresponding graph. If a paper has k authors, then there is a clique of size k in the graph. The dataset contains papers published between January 1993 and April 2003 (124 months), starting within a few months of the inception of arXiv, and thus it represents essentially the complete history of the hep-ph archive.

The arXiv hep-ph Coauthor graph contains 12,006 vertices and 118,489 edges, with an average degree of 19.7. Since there exists a clique of size 239 in this graph, the (α, β)-Community algorithm returns this clique or a substantial part of it as a core for $k \geqslant 200$. After removing this clique, we obtain a similar core structure to that of Twitter and Slashdot.

5.4 Citation

The Citation dataset was crawled from the e-print arXiv that contains 421,578 citation links among a collection of 34,546 papers in the hep-ph archive [38, 39]. If paper i cites paper j or vice versa, then there is an undirected edge between vertex i and vertex j in the corresponding graph. This dataset was originally released in the KDD Cup 2003 [38], and represents essentially the complete history of the hep-ph archive.

The Citation graph has 34,546 vertices and 420,877 edges, with an average degree of 24.4. In this graph, we again discover a core structure similar to that of Twitter, Slashdot, and Coauthor. The Citation graph contains more cores than other social graphs for the same value of k. There are 4 disjoint cores for $k = 900$, and as k continues to increase, the number of cores eventually decreases to one as in the other social graphs.

5.5 Random Graphs

A similar set of experiments can be performed on random graphs to demonstrate the existence of core structure in various social networks. The comparison of the results confirms that the structure we have found in many social graphs is more than just a random artifact.

First, we generate a random graph according to the $G(n, p)$ model with $n = 112,957$ and $p = 8.52$ (those of the Twitter graph). This graph contains 597,674 edges (self-loop counted), which are also similar to that of the Twitter graph. However, conducting the same experiment on this graph reveals a completely different structure from what we have seen in social graphs. The (α, β)-Community algorithm is employed to find 500 (α, β)-communities of size 30 to 300. For each size, the 500 obtained (α, β)-communities have little overlap (less than 5% in most cases), and are scattered all over the graph where no massively overlapping clusters can be found. We observe that $\alpha = 1$ and $\beta = 2$ for

each (α, β)-community in this random graph, as opposed to those as large as 20 in the Twitter graph. Thus, random subsets are extracted from $G(n, p)$ which are not even connected, implying the absence of an underlying social structure.

An interesting question is whether high-degree vertices lead to the massively overlapping clusters found in the Twitter graph. To answer this question, we generate random d-regular graphs with 4,144 vertices (that of the Twitter graph with low-degree vertices removed) for a wide range of values of d. Recall that the lowest β value is 19 for most (α, β)-communities in the Twitter graph, thus removing vertices of degree lower than 19 does not destroy the fundamental structure of the graph. For each value of d, the algorithm still returns scattered (α, β)-communities with little overlap among them. Thus, high-degree vertices are not the primary reason for such few number of cores in the Twitter graph.

Another question is whether a particular degree distribution of the Twitter graph leads to the massively overlapping clusters. To answer this question, we conduct similar experiments on randomly generated graphs with 4,144 vertices and a given degree distribution (e.g. power-law). There are several ways to generate random graphs with a given degree distribution, two of which give the same distribution as that of the Twitter graph while the third gives a power-law distribution.

Uniform model. Given the degree distribution, we place edges by selecting vertices uniformly at random. As a result, high-degree vertices are not as densely connected as in the Twitter graph. This uniform model displays the same behavior as the $G(n, p)$ model for small (α, β)-communities. As the size k increases, (α, β)-communities gradually overlap with each other. Cores can be extracted from the graph, but they also have significant overlap among them.

Further, most high-degree vertices are contained in the cores as expected. For example, consider the two cores obtained for $k = 450$. One core is of size 172, containing 93% of the vertices of degree higher than 200 and 63% of those of degree higher than 150. The other core is of size 351, containing 100% of the vertices of degree higher than 200 and 84% of those of degree higher than 150.

Proportional model. Given the degree distribution, we place edges by selecting vertices with probability proportional to their degree. As a result, high-degree vertices are densely connected, and for $k \geqslant 150$, there is only one core returned by the algorithm with 200 (α, β)-communities. Further, almost all high-degree vertices are contained in that core. For example, the core is of size 125 for $k = 200$, containing 94% of the vertices of degree higher than 200 and 73% of those of degree higher than 150. The core corresponds to the dense region of the graph due to the way the edges are placed, in which high-degree vertices are more likely to be selected.

Preferential attachment model. We first create a clique of small size (e.g. 5), then recursively add a new vertex and randomly pick some of the existing vertices to be its neighbors with probability proportional to their degree. Thus, the resulting graph displays a power-law degree distribution, different from that of the Twitter graph. For each size from 50 to 300, the (α, β)-Community

algorithm returns a small number of cores with substantial overlap among them. In contrast to what we have observed in the Twitter graph, the number of cores steadily increases with the size k. For example, we obtain 7 cores for $k = 90$ and 11 cores for $k = 250$.

According to these experiments, random graph models do not produce well-defined clusters as social graphs do. The cores found in random graphs usually have significant overlap among them, and correspond to dense regions due to the way the graph was generated. This demonstrates that the core structure displayed by various large social networks is indeed due to the existence of underlying social structure of those networks.

6 Conclusion and Future Work

In many social networks, (α, β)-communities of a given size k are well clustered into a small number of disjoint cores, each of which is the intersection of a group of massively overlapping (α, β)-communities. Two (α, β)-communities in the same group share a significant overlap and differ by only a few vertices, while the pairwise resemblance of two (α, β)-communities in different groups is extremely small. The number of cores decreases as k increases and becomes relatively small for large k. The cores obtained for a small k either disappear or merge into the cores obtained for a larger k. Further, the cores correspond to dense regions of the graph, and there are no isolated (α, β)-communities scattered between the cores. There are no bridges of (α, β)-communities connecting one core to another. We have explored various large social networks, all of which display the core structure rather than the chain structure.

By constructing random graphs with a power-law degree distribution or the same degree distribution as that of the social graphs, it is demonstrated that neither high-degree vertices nor a particular degree distribution can lead to the core structure displayed in many social networks. The cores found in random graphs usually have significant overlap and are increasingly scattered across the graph as the size k increases, which implies the absence of well-defined clusters in random graphs and verifies the existence of core structure in various social networks.

Our work opens several questions about the structure of large social networks. It demonstrates the successful use of the (α, β)-Community algorithm on real-world networks to discover their social structure. Further, our work inspires an effective way of finding overlapping communities and extracting the underlying core structure. We conjecture that, in many social graphs, the vertices inside an (α, β)-community but outside of the corresponding core are actually located in the overlapping regions of multiple communities. Other interesting questions include whether different types of social networks display fundamentally different social structure, how the core structure evolves over time, whether the cores represent the stable backbones of the network, and whether the vertices that belong to multiple communities constitute the unstable regions of the network.

References

1. A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70(06111), 2004.
2. M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. USA*, 99(12):7821–7826, 2002.
3. J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. Statistical properties of community structure in large social and information networks. In *Proc. 17th Int'l World Wide Web Conf. (WWW)*, 2008.
4. M. E. J. Newman. Detecting community structure in networks. *The European Physical J. B*, 38:321–330, 2004.
5. M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69(066133), 2004.
6. M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74(036104), 2006.
7. M. E. J. Newman. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA*, 103(23):8577–8582, 2006.
8. M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(026113), 2004.
9. R. Andersen, F. Chung, and K. Lang. Local graph partitioning using pagerank vectors. In *Proc. 47th IEEE Symp. Found. Comp. Sci. (FOCS)*, 2006.
10. M. Gaertler. Clustering. *Network Analysis: Methodological Foundations*, 3418:178–215, 2005.
11. K. Lang and S. Rao. A flow-based method for improving the expansion or conductance of graph cuts. In *Proc. 10th Int'l Conf. Integer Programming and Combinatorial Optimization (IPCO)*, 2004.
12. S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.
13. N. Mishra, R. Schreiber, I. Stanton, and R. E. Tarjan. Finding strongly-knit clusters in social networks. *Internet Mathematics*, 5(1–2):155–174, 2009.
14. M. Rosvall and C. T. Bergstrom. An information-theoretic framework for resolving community structure in complex networks. *Proc. Natl. Acad. Sci. USA*, 104(18):7327–7331, 2007.
15. S. Papadimitriou, J. Sun, C. Faloutsos, and P. S. Yu. Hierarchical, parameter-free community discovery. In *Proc. 19th European Conf. Mach. Learn. (ECML)*, 2008.
16. Y.-Y. Ahn, J. P. Bagrow, and S. Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466:761–764, 2010.
17. J. Leskovec, K. J. Lang, and M. W. Mahoney. Empirical comparison of algorithms for network community detection. In *Proc. 19th Int'l World Wide Web Conf. (WWW)*, 2010.
18. M. Sozio and A. Gionis. The community-search problem and how to plan a successful cocktail party. In *Proc. 16th ACM Int'l Conf. Knowl. Disc. Data Min. (KDD)*, 2010.
19. J. Gao, F. Liang, W. Fan, C. Wang, Y. Sun, and J. Han. On community outliers and their efficient detection in information networks. In *Proc. 16th ACM Int'l Conf. Knowl. Disc. Data Min. (KDD)*, 2010.
20. T. Yang, R. Jin, Y. Chi, and S. Zhu. Combining link and content for community detection: a discriminative approach. In *Proc. 15th ACM Int'l Conf. Knowl. Disc. Data Min. (KDD)*, 2009.
21. L. Tang, H. Liu, J. Zhang, and Z. Nazeri. Community evolution in dynamic multimode networks. In *Proc. 14th ACM Int'l Conf. Knowl. Disc. Data Min. (KDD)*, 2008.

22. C. Tantipathananandh and T. Y. Berger-Wolf. Constant-factor approximation algorithms for identifying dynamic communities. In *Proc. 15th ACM Int'l Conf. Knowl. Disc. Data Min. (KDD)*, 2009.
23. Y.-R. Lin, J. Sun, P. Castro, R. B. Konuru, H. Sundaram, and A. Kelliher. Metafac: community discovery via relational hypergraph factorization. In *Proc. 15th ACM Int'l Conf. Knowl. Disc. Data Min. (KDD)*, 2009.
24. T. L. Fond and J. Neville. Randomization tests for distinguishing social influence and homophily effects. In *Proc. 19th Int'l World Wide Web Conf. (WWW)*, 2010.
25. Y. Zhang, J. Wang, Y. Wang, and L. Zhou. Parallel community detection on large networks with propinquity dynamics. In *Proc. 15th ACM Int'l Conf. Knowl. Disc. Data Min. (KDD)*, 2009.
26. V. Satuluri and S. Parthasarathy. Scalable graph clustering using stochastic flows: applications to community discovery. In *Proc. 15th ACM Int'l Conf. Knowl. Disc. Data Min. (KDD)*, 2009.
27. A. S. Maiya and T. Y. Berger-Wolf. Sampling community structure. In *Proc. 19th Int'l World Wide Web Conf. (WWW)*, 2010.
28. M. D. Choudhury, W. A. Mason, J. M. Hofman, and D. J. Watts. Inferring relevant social networks from interpersonal communication. In *Proc. 19th Int'l World Wide Web Conf. (WWW)*, 2010.
29. J. Chen, O. R. Zaïane, and R. Goebel. Detecting communities in social networks using max-min modularity. In *Proc. 9th SIAM Int'l Conf. Data Min. (SDM)*, 2009.
30. W. Chen, Z. Liu, X. Sun, and Y. Wang. A game-theoretic framework to identify overlapping communities in social networks. *Data Min. Knowl. Discov.*, 21(2):224–240, 2010.
31. T. Yang, Y. Chi, S. Zhu, Y. Gong, and R. Jin. A bayesian approach toward finding communities and their evolutions in dynamic social networks. In *Proc. 9th SIAM Int'l Conf. Data Min. (SDM)*, 2009.
32. J. I. Alvarez-Hamelin, A. Barrat, L. DallAsta, and A. Vespignani. k-core decomposition: a tool for the visualization of large scale networks. *CoRR*, cs.NI/0504107, 2005.
33. J. I. Alvarez-Hamelin, A. Barrat, L. DallAsta, and A. Vespignani. k-core decomposition: a tool for the analysis of large scale internet graphs. *CoRR*, cs.NI/0511007, 2005.
34. V. Batagelj and A. Mrvar. Generalized cores. *Journal of the ACM*, 5, 2002.
35. J. Healy, J. Janssen, E. E. Milios, and W. Aiello. Characterization of graphs using degree cores. In *Proc. 3rd Worshop on Algorithms and Models for the Web Graph (WAW)*, 2006.
36. M. D. Choudhury, Y-R. Lin, H. Sundaram, K.S. Candan, L. Xie, and A. Kelliher. How does the sampling strategy impact the discovery of information diffusion in social media? In *Proc. 4th Int'l AAAI Conf. Weblogs and Social Media (ICWSM)*, 2010.
37. J. Leskovec, J. Kleinberg, and C. Faloutsos. Graph evolution: densification and shrinking diameters. *ACM Trans. Knowl. Disc. from Data (TKDD)*, 1(1), 2007.
38. J. Gehrke, P. Ginsparg, and J. Kleinberg. Overview of the 2003 kdd cup. *SIGKDD Explorations*, 5(2):149–151, 2003.
39. J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proc. 11th ACM Int'l Conf. Knowl. Disc. Data Min. (KDD)*, 2005.