Internet Mathematics Vol. 5, No. 1–2: 153–172

Finding Strongly Knit Clusters in Social Networks

Nina Mishra, Robert Schreiber, Isabelle Stanton, and Robert E. Tarjan

Abstract. Social networks are ubiquitous. The discovery of closely knit clusters in these networks is of fundamental and practical interest. Existing clustering criteria are limited in that clusters typically do not overlap, all vertices are clustered, and/or external sparsity is ignored. We introduce a new criterion that overcomes these limitations by combining internal density with external sparsity in a natural way.

This paper explores combinatorial properties of internally dense and externally sparse clusters. A simple algorithm is given for provably finding such clusters assuming a sufficiently large gap between internal density and external sparsity. Experimental results show that the algorithm is able to identify over 90% of the clusters in real graphs, assuming conditions on external sparsity.

1. Introduction

Social networks have gained in popularity recently with the advent of sites such as MySpace, Friendster, and Facebook. The number of users participating in these networks is large, e.g., hundreds of millions in MySpace, and growing. These networks are becoming a rich source of data as users populate their profiles with personal information. Of particular interest in this paper is the graph structure induced by the friendship links.

A fundamental problem related to these networks is the discovery of clusters, or communities. Intuitively, a cluster is a collection of individuals with dense friendship patterns internally and sparse friendships externally. There are many reasons to seek tightly knit communities in networks; for instance, targeted marketing schemes can be designed based on clusters.

© A K Peters, Ltd.
1542-7951/08 $0.50 per page

What defines a cluster in a social network? At first glance, the answer would seem to be identical to a traditional cluster in a graph. However, it turns out that the notions are quite different. The reason stems from some of the initial motivations for studying graph clustering: to partition a large graph into multiple processors so that interprocessor communication is minimized and load is approximately balanced. In a multiprocessing environment, each vertex of the graph is assigned to exactly one cluster, and the number of vertices assigned to each processor is approximately the same. The number of edges crossing the cut is an important component of the optimization. This criterion does not apply to a social network: a person can belong to multiple clusters, not every person needs to be clustered, and clusters can contain a varying number of members. Further, internal density of a cluster matters: the number of edges crossing between two clusters may be quite large, but any person outside a cluster should have little adjacency into the cluster.

Closer to our work is the notion of a community, which has been considered in prior work. A subset of vertices is said to form a community [Flake et al. 00] if each vertex has at least as many edges into the community as outside the community. One problem with such a definition is that individuals with a high degree of connectivity will ultimately not belong to any community. Such highly connected individuals are crucial to understanding network structure. While this definition is closer to what we seek, it is still missing many important components: external sparsity, overlapping clusters where not every vertex is clustered, and internal density.

In this paper, we formulate a new graph-clustering criterion that is ideally suited for social networks. We consider an induced subgraph to be a cluster if its internal density is sufficiently large (β) and if vertices outside the cluster have sufficiently sparse connectivity into the cluster (α). Specifically, a subset of vertices forms an (α, β)-*cluster* if every vertex in the cluster is adjacent to at least a β-fraction of the cluster and every vertex outside the cluster is adjacent to at most an α-fraction of the cluster (see Definition 3.1). Our analysis provides a rigorous understanding of the combinatorics of (α, β)-clusters, together with a provable algorithm for finding them. The (α, β)-criterion allows clusters to overlap and does not necessarily cluster every vertex.

1.1. Contributions

Clusters in social networks take on different characteristics, e.g., overlapping, internally dense, and externally sparse. We give a novel formulation, (α, β)-clustering, specifically suited to these networks.

We investigate combinatorial properties of (α, β)-clusters. We bound the extent to which two clusters can overlap. For two clusters of equal size, we show that they overlap in at most a $(1 - (\beta - \alpha))$ fraction of the vertices. For certain values of α and β, it is possible for one cluster to be contained in another. We show that if the ratio of the size of the largest cluster to the smallest cluster is at most $(1 - \alpha)/(1 - \beta)$, then one cluster cannot be contained in another. Finally, we give a loose upper bound on the number of $(\alpha, 1)$-clusters of size s, namely

$$\binom{n}{\alpha s + 1} \Big/ \binom{s}{\alpha s + 1},$$

where n is the number of vertices.

Next, we introduce the notion of a ρ-champion of a cluster, which is a vertex in the cluster with a bounded number of neighbors outside of the cluster, specifically, no more than a ρ fraction of the cluster. If ρ is less than β, then intuitively the champion has more neighbors inside the cluster than out. We assume that the goal of clustering is to find (α, β)-clusters that have at least one ρ-champion.

How can one find such clusters? We show that if there is a large gap between $\alpha/2$ and β, i.e., $\beta > \frac{1}{2} + \frac{\rho+\alpha}{2}$, then there is a deterministic algorithm for finding all clusters that runs in time roughly quadratic in the number of vertices.

To validate our ρ-champion assumption and algorithms, we conduct an experiment that evaluates the effectiveness of the clustering algorithm. We demonstrate that our clustering algorithm succeeds in finding all $(\alpha, 1)$-clusters with ρ-champions. We compare the clusters we discover with a ground-truth algorithm for finding all maximal cliques in a graph. The experiments demonstrate that our algorithm finds over 90% of the clusters in the graph, assuming conditions on external sparsity. Furthermore, (α, β)-clusters truly exist in these graphs.

2. Related Work

Our (α, β)-clustering formulation is new but has been considered in restricted settings under different guises. The problem of finding the connected $(0, \beta)$-clusters in a graph can be reduced to first finding connected components and then outputting the components that are β-connected. This problem can be solved efficiently via depth-first search in $O(|E|+|V|)$ time for a graph $G = (V, E)$. Also, the problem of finding $\left(1 - \frac{1}{n}, 1\right)$-clusters is equivalent to finding the maximal cliques in a graph. This problem has a rich history. Known algorithms find all maximal cliques in time that depends polynomially on the size of the graph and the number of maximal cliques [Tsukiyama et al. 77, Johnson et al. 88].

The problem of finding $((1-\epsilon)\beta, \beta)$-clusters, for small ϵ, has also been studied under the name of finding quasicliques. A method is presented in [Abello et al. 02] for finding subgraphs with average connectivity β. In [Hartuv and Shamir 00], the authors find densely connected subgraphs in which $\beta > \frac{1}{2}$ via a min-cut algorithm. In the bipartite case, [Mishra et al. 04] considers the problem of finding dense, well-separated bipartite subgraphs. These algorithms ignore external sparsity (α). External sparsity turns out to be quite important: an example in Figure 2 shows that there is only one $\left(\frac{1}{n}, 1 - \frac{1}{2n}\right)$-cluster, but if α is ignored, then there are 2^n $\left(\frac{n-1}{n}, 1\right)$-clusters, an undesirable consequence.

Spectral clustering is a very popular method that involves recursively splitting the graph using various criteria, e.g., the principal eigenvector of the adjacency matrix. Successful approaches have been employed in [Kannan et al. 00, Shi and Malik 00, Karypis and Kumar 98, Spielman and Teng 96, Newman 06], among many others. All of these approaches do not allow overlapping clusters, which is one of the main goals of our work.

Newman and others have advocated modularity as an optimization criterion for graph partitioning [Newman 06]. The modularity of a partition is the amount by which the number of edges between vertices in the same subset exceeds the number predicted by the degree-distribution-preserving random-graph model of [Aiello et al. 00]. Newman proposed several methods for optimizing modularity, among them a spectral approach, and others have found competitive methods as well.

In [Flake et al. 04], the authors use a recursive cut approach intended to optimize the expansion of the clustering but use Gomory–Hu trees [Gomory and Hu 62] instead of eigenvectors to find the cut. The expansion of a cut is very similar to the conductance of a cut. The minimum quality of the clustering is guaranteed by adding a sink to the graph. Again, the goal of this work is different from ours in that a partitioning is constructed, disallowing overlapping clusters.

Modeling flow through a network is another way to cluster a graph [Flake et al. 04, Van Dongen 98]. MCL models flow through two alternating Markov processes: expansion and inflation. MCL has been widely used for clustering in biological networks but requires that the graph be sparse, and it finds overlapping clusters only in restricted cases. In contrast, (α, β)-clustering has no restrictions on the general structure of the graph and allows clusters of different sizes to overlap.

There has also been considerable work in finding communities on the Web [Kumar et al. 99, Gibson et al. 98, Capocci et al. 05, Flake et al. 00, Ino et al. 05]. For instance, in [Kumar et al. 99] the problem is approached as one of finding bicliques as the cores of communities. Dourisboure et al. consider a very similar internal density community definition [Dourisboure et al. 09]. Their methods are able to find clusters in graphs with hundreds of millions of nodes. A key difference between their work and ours is the notion of external sparsity.

Finally, there has been previous work finding overlapping clusters: [Gregory 07, Palla et al. 05, Zhang et al. 07] have all developed distinct methods for uncovering overlapping community structures. For example, [Palla et al. 05] defines a community as a series of connected k-cliques, while [Zhang et al. 07] adapts the modularity definition for use with fuzzy c-means clustering. While both methods allow overlapping clusters, neither considers our external sparsity criterion.

3. Preliminaries

In this section, we give some notation that will be useful for the rest of the paper and also formally define the (α, β)-clustering problem.

3.1. Notation

We use the following notation to describe our results. For a graph $G = (V, E)$, n denotes the number of vertices and m denotes the number of edges. For a subset of vertices $A \subseteq V$, $|A|$ denotes the number of vertices in A, and $E(v, A)$ denotes the set of edges between a vertex v and a subset of vertices A. For $B \subseteq V$, $E(A, B)$ denotes the set of edges between A and B. The neighbors of a vertex v are denoted by $\Gamma(v)$. The vertices that are in a ball of radius r around v are denoted by $B_r(v)$ and include vertices that are $1, 2, \ldots, r$ hops from v. Thus, for instance, $B_2(v) = \Gamma(v) \cup \Gamma(\Gamma(v))$. The affinity that a vertex v has with a set X is exactly $|\Gamma(v) \cap X|$.

3.2. Formal Definition of (α, β)-Clustering

What is a good cluster in a social network? There are numerous existing criteria for defining good graph clusters, and a multitude of algorithms accompanies each criterion. One popular criterion is based on finding clusters of high conductance. The conductance of a cut A, B is the ratio of the number of edges crossing the cut to the minimum of the volumes of A and B, where the volume of A is the number of edges emanating from the vertices in A. Intuitively, conductance is the fraction of edges coming out of A that cross the cut. The conductance of a cluster is the minimum conductance of any cut in the cluster.

A spectral algorithm typically uses the eigenvector of a matrix related to the adjacency matrix to find a good cut of the graph into subgraphs A, B. The

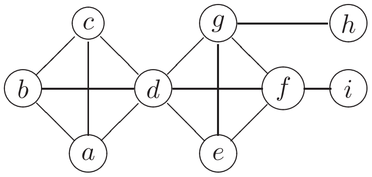

Figure 1. Overlapping clusters.

process is then recursively repeated (on A and B) until k clusters are found (where k is an input parameter) or until the conductance of the next-best cut is larger than some threshold. Formal guarantees can be proved for some variants of this basic algorithm [Kannan et al. 00].

Cut-based graph-clustering algorithms produce a strict partition of the graph, which is particularly problematic for social networks, as illustrated in Figure 1. In this graph, d belongs to two clusters $\{a, b, c, d\}$ and $\{d, e, f, g\}$. Furthermore, h and i need not be clustered. A cut-based approach will either put $\{a, b, c, d, e, f, g\}$ into one cluster, which is not desirable, since e, f, g have no edges to a, b, c, or else cut at d, putting d into one of the clusters, say $\{a, b, c, d\}$, but leaving d out of $\{e, f, g\}$, which leaves a highly connected vertex outside of the cluster.

The example in Figure 1 motivates a new formulation of the graph-clustering problem that does not stipulate that each vertex belong to exactly one cluster. Our objective is to identify clusters that are internally dense, i.e., each vertex in the cluster is adjacent to at least a β-fraction of the cluster, and externally sparse, i.e., any vertex outside of the cluster is adjacent to at most an α-fraction of the vertices in the cluster.

Definition 3.1. Given a graph $G = (V, E)$ in which every vertex has a self-loop,[1] $C \subset V$ is an (α, β)-*cluster* if it is

1. *internally dense:* $\forall v \in C$, $|E(v, C)| \geq \beta|C|$;
2. *externally sparse:* $\forall u \in V \setminus C$, $|E(u, C)| \leq \alpha|C|$.

Given $0 \leq \alpha < \beta \leq 1$, the (α, β)-*clustering problem* is to find all (α, β)-clusters.

The new clustering criterion does not seek a strict partitioning of the data. To see why clusters can overlap, return to Figure 1. Both $\{a, b, c, d\}$ and $\{d, e, f, g\}$ are $\left(\frac{1}{4}, 1\right)$-clusters. Furthermore, h and i do not fall into an (α, β)-cluster if $0 \leq \alpha < \frac{1}{2} < \beta \leq 1$, and consequently would not be clustered.

[1] This is a technical assumption needed to ensure that $\beta = 1$ clusters are possible.

Observe that when $\beta \to 1$, an (α, β)-cluster approaches a clique, and when $\alpha \to 0$, an (α, β)-cluster tends to a disconnected component. We want $\alpha < \beta$, since vertices outside of a cluster should have fewer neighbors in the cluster than vertices that belong to the cluster.

4. Combinatorics of (α, β)-Clusters

In this section, we discuss several combinatorial properties of (α, β)-clusters including cluster overlap, containment, and number of clusters.

4.1. Cluster Overlap

Given two (α, β)-clusters A, B, where $|A| \geq |B|$, we now determine the maximum size of the overlap, namely $|A \cap B|$. In the case $\beta = 1$, $|A \cap B|$ can be no larger than $\alpha|B|$ (otherwise, there would be a vertex outside of B that is adjacent to more than α of B). Alternatively, in the case $\alpha = 0$, $|A \cap B|$ must be 0. More generally, we seek a bound for arbitrary values of α and β. We express the overlap as the fraction of vertices in A, i.e., $\gamma = |A \cap B|/|A|$.

Proposition 4.1. *For two (α, β)-clusters A and B, where $|A| \geq |B|$ and $A \neq B$, an upper bound on γ is $1 - (\beta - \alpha\frac{|B|}{|A|})$.*

Proof. Let $u \in A \setminus B$. From the α criterion we know that no element of $A \setminus B$ is connected to more than α of B. Formally, $\alpha|B| \geq |E(u, A \cap B)|$. Similarly, $|E(u, A \cap B)| \geq |A \cap B| - (1 - \beta)|A|$, since u is connected to at least β of A. Combining these inequalities, we get $\alpha|B| \geq |A \cap B| - (1 - \beta)|A|$, and solving for $|A \cap B|$, we have

$$|A \cap B| \leq (1 - \beta)|A| + \alpha|B|, \quad \gamma = \frac{|A \cap B|}{|A|}, \tag{4.1}$$

so $\gamma \leq 1 - (\beta - \alpha|B|/|A|)$. □

When $\beta = 1$, the above bound implies that $\gamma \leq \alpha|B|/|A|$, which is tight. However, if we let $\alpha = 0$, the bound indicates that $\gamma \leq (1 - \beta)$, which is weak; γ should be 0, since $\alpha = 0$ implies that each cluster is disconnected from the rest of the graph. We now prove a bound that is tight in the case that $\alpha = 0$ and $\beta > \frac{1}{2}$. This bound is not useful when β is close to or less than $\frac{1}{2}$.

Corollary 4.2. *For two (α, β)-clusters A and B, where $|A| \geq |B|$ and $\beta > \frac{1}{2}$, an upper bound on the ratio of the intersection $|A \cap B|$ to the larger one, $|A|$, is*

$$\gamma \leq \frac{\alpha}{2\beta - 1} \frac{|B|}{|A|}.$$

Proof. Let $u \in A \setminus B$. From the definition of an (α, β)-cluster we know that $|E(u, A \cap B)| \leq \alpha |B|$. Therefore,

$$|E(A \setminus B, A \cap B)| \leq \alpha |B| |A \setminus B|. \tag{4.2}$$

Let $x \in A \cap B$. It follows that

$$|E(x, A \setminus B)| \geq \beta |A| - |A \cap B|$$

and

$$|E(A \cap B, A \setminus B)| \geq (\beta |A| - |A \cap B|) |A \cap B|. \tag{4.3}$$

Combining (4.2) and (4.3), we have that

$$(\beta |A| - |A \cap B|) |A \cap B| \leq \alpha |B| |A \setminus B|.$$

To simplify the equation, let $|A \cap B| = \gamma |A|$ and $|A \setminus B| = (1 - \gamma)|A|$. Also, recall from (4.1) that $|A \cap B| \leq (1 - \beta)|A| + \alpha |B|$. We have

$$\begin{aligned} (\beta |A| - [(1 - \beta)|A| + \alpha |B|]) \gamma |A| &\leq \alpha |B| (1 - \gamma) |A|, \\ (\beta |A| - |A| + \beta |A| - \alpha |B|) \gamma &\leq \alpha |B| - \alpha \gamma |B|, \\ (2\beta - 1)|A| \gamma &\leq \alpha |B|, \\ \gamma &\leq \frac{\alpha}{2\beta - 1} \frac{|B|}{|A|}, \end{aligned}$$

which completes the proof. □

If we have the situation in which $\beta > \frac{1}{2}$, then the appropriate bound on the overlap is

$$\gamma \leq \min \left(1 - \left(\beta - \alpha \frac{|B|}{|A|} \right), \ \frac{\alpha}{2\beta - 1} \frac{|B|}{|A|} \right).$$

Moreover, it can be shown that when

$$\beta - \alpha \frac{|B|}{|A|} > \frac{1}{2},$$

then

$$\frac{\alpha}{2\beta - 1}\frac{|B|}{|A|}$$

is the minimum, and otherwise,

$$1 - \left(\beta - \alpha\frac{|B|}{|A|}\right)$$

is the minimum.

4.2. Cluster Containment

Given that clusters can overlap, it is natural to ask whether one cluster can be contained in another. In some circumstances, α and β may be such that clusters are contained in each other. For example, consider two cliques, C and D, each containing $k \geq 3$ vertices. Assume that each vertex in C is adjacent to two vertices in D. When

$$\beta = \frac{1}{2} + \frac{1}{k} \quad \text{and} \quad \alpha = \frac{2}{k},$$

then C, D, and $C \cup D$ are all (α, β)-clusters and $C \cup D$ contains both C and D.

If we want to prevent our algorithm from finding clusters one of which is contained in another, we can do so by requiring that the ratio of the larger to the smaller cluster be at most $(1 - \alpha)/(1 - \beta)$.

Corollary 4.3. *Let A and B be (α, β)-clusters and assume that $|B| \leq |A|$. If*

$$\frac{|A|}{|B|} < \frac{1 - \alpha}{1 - \beta},$$

then B cannot be contained in A.

The proof follows directly from Proposition 4.1 where the assumption implies that γ is upper bounded by $\frac{|B|}{|A|}$. The larger the gap between α and β, the larger the bound. For example, if $\alpha = \frac{1}{4}$ and $\beta = \frac{3}{4}$, then the larger cluster must be at least three times the size of the smaller before the smaller can be contained in the larger. Similarly, if $\alpha = \frac{1}{8}$ and $\beta = \frac{7}{8}$, then the ratio is 7.

4.3. Bounding the Number of $(\alpha, 1)$-Clusters

We next consider the problem of bounding the number of $(\alpha, 1)$-clusters from above. We give a superpolynomial bound on the number of clusters of a fixed size $s = f(n)$. More generally, it would be interesting to bound the number of possible (α, β)-clusters, but our analysis here is focused on cliques.

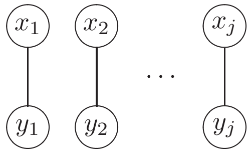

Figure 2. A graph G in which $\overline{G}$ has exponentially many clusters.

We wish to bound the number of $(\alpha, 1)$-clusters of size $s = f(n)$ in a graph $G = (V, E)$ where $|V| = n$. We know from Proposition 4.1 that no two clusters can overlap in more than αs vertices.

Proposition 4.4. *Let $G = (V, E)$, where $|V| = n$. If $\mathcal{C}$ is the set of $(\alpha, 1)$-clusters of size s in G, then*

$$|\mathcal{C}| \leq \binom{n}{\alpha s + 1} \Big/ \binom{s}{\alpha s + 1}.$$

Proof. From Proposition 4.1, two $(\alpha, 1)$-clusters of size s can share at most αs vertices. In this analysis, we upper bound the number of subsets of vertices that can be $(\alpha, 1)$-clusters. The analysis does not utilize the graph structure. Instead, consider the clusters as a collection of subsets of vertices of size s. Now we can say that every subset of size $\alpha s + 1$ must appear in at most one set in our collection. There is a total of $\binom{n}{s}$ subsets of size s, and each of these subsets contains $\binom{s}{\alpha s+1}$ subsets of size $\alpha s + 1$. By simple combinatorics we can have at most

$$\binom{n}{\alpha s + 1} \Big/ \binom{s}{\alpha s + 1}$$

clusters of size s.[2] □

We note that this bound is tight when $\alpha = 0$ and when α approaches 1. If we let $\alpha = 0$ then the bound indicates that the number of clusters is at most n/s. This is tight, because clusters cannot overlap at all. At the other extreme, consider the complement of the graph shown in Figure 2. Let $\alpha = (j - 1)/j$ and $\beta = 1$. Observe that $B = \{b_1 \cdots b_j | b_i = x_i \vee y_i\}$ are all legitimate (α, β)-clusters and further that $|B| = 2^j$. When $s = j$, Proposition 4.4 also yields an upper bound of 2^j clusters. Thus, Proposition 4.4 is tight when $\alpha = (j - 1)/j$.

We believe that the bound given in Proposition 4.4 overcounts the number of clusters when $\alpha \leq \frac{1}{2}$ because the edges are completely ignored. Consider K_4,

[2] This exactly corresponds to the construction of a Steiner system [Anderson 74].

where $s = 3$ and $\alpha = \frac{1}{3}$. The bound allows three clusters of size 3. In reality, due to α violations, there are none.

5. Gaps and Champions

In this section, we make some restrictions to the general (α, β)-clustering problem and motivate these restrictions.

5.1. Gap between Internal Density and External Sparsity

To motivate a gap between internal density and external sparsity, consider Figure 2. Observe that depending on the choice of α and β, the number of clusters may be exponential in the size of the graph. In practice, an algorithm that outputs more clusters than vertices is quite undesirable, especially given that social networks are massively large data sets. Thus, we seek a restriction that will reduce the number of clusters.

5.2. Champions

Intuitively, a vertex champions a cluster if it has more affinity into the cluster than out of it. To motivate champions, observe that for $\overline{G}$ of G given in Figure 2, each vertex in each cluster has as many neighbors outside the cluster as within it. There is no vertex that “champions” the cluster in the sense that many of its neighbors are in the cluster. For example, theoretical physicists form a community in part because there are some champions that have more friends that are theoretical physicists than not. Specifically, if *every* vertex in a subset A has as many neighbors out of A as into A, then it is arguable whether A is really even a cluster. This motivates us to define formally the notion of a ρ-champion.

Definition 5.1. A vertex $c \in C$ ρ-*champions* a cluster C if $|\Gamma(c) \cap V \setminus C| \leq \rho|C|$ for some $0 \leq \rho \leq 1$.

6. Finding Strongly Knit Clusters

In this section we prove that if

$$\beta > \frac{1}{2} + \frac{\rho + \alpha}{2},$$

Algorithm 1. (Deterministic clustering algorithm, when $\beta > \frac{1}{2} + \frac{\alpha+\rho}{2}$.)

```
1. Input: $\alpha, \beta, G, s$.
2. For each $c \in V$:
    (a) $C = \varnothing$.
3. For each $v \in \tau(c)$:
    (a) If $|\Gamma(v) \cap \Gamma(c)| \geq (2\beta - 1)s$ then add $v$ to $C$.
4. If $C$ is an $(\alpha, \beta)$-cluster then output $C$.
```

then there are at most n clusters with ρ-champions and further that there is a simple deterministic algorithm for finding the clusters.

Lemma 6.1. *If $\beta > \frac{1}{2} + \frac{\rho+\alpha}{2}$, then there are at most n (α,β)-clusters with ρ-champions of a fixed size s.*

Proof. Under the conditions of the lemma, we show that a vertex can champion at most one cluster. If c champions a cluster C, then for any other cluster C',

$$|\Gamma(c) \cap C'| = |\Gamma(c) \cap (C' \cap C)| + |\Gamma(c) \cap C' \setminus C| \leq (1 - \beta + \alpha)|C'| + \rho|C'|.$$

Thus by assumption, we have that $(1-\beta+\rho+\alpha)|C'| < \beta|C'|$, and consequently c does not have enough neighbors in C' to be β-connected into C'. Note that this proof relies on the fact that, for fixed size s, neither C nor C' can be contained in the other. □

A large gap between β and $\frac{1}{2}+\frac{\alpha+\rho}{2}$ yields a simple algorithm (Algorithm 1) for deterministically pinning down all the clusters. Let the input to the algorithm be α, β, the graph G, and the size s of the clusters to be found.

The following lemma shows that if v and c share sufficiently many neighbors, then v is necessarily part of the cluster C that c champions. When the size of the cluster is fixed, Lemma 6.2 also implies that C is unique. Additional bounds to guarantee uniqueness when the size of the cluster is allowed to vary can be easily obtained.

Lemma 6.2. *Let C be an (α,β)-cluster and c its ρ-champion. Let $\beta > \frac{1}{2} + \frac{\rho+\alpha}{2}$. A vertex v is in the cluster C if and only if $|\Gamma(v) \cap \Gamma(c)| \geq (2\beta - 1)|C|$.*

Proof. We begin by establishing two facts: (1) Any vertex in cluster C shares at least $(2\beta - 1)|C|$ neighbors with c. (2) Any vertex not in C shares at most $(\rho + \alpha)|C|$ neighbors with c.

Regarding (1), let $v \in C$. We can bound the number of neighbors that c and v share from below using the fact that v intersects at least β of C and c misses at most $(1 - \beta)|C|$. Therefore, we have that $|\Gamma(c) \cap \Gamma(v)| \geq (2\beta - 1)|C|$.

Regarding (2), let $\overline{v} \in V \setminus C$. We can bound the number of neighbors that c and v share from above by separating the neighbors that $\overline{v}$ and c could have in C and outside of C. Due to the α-disconnectedness of C, the number of neighbors that $\overline{v}$ has inside of C is at most $\alpha|C|$. Further, because c champions C, the number of neighbors that c and $\overline{v}$ can share outside of C is at most $\rho|C|$. Thus, $|\Gamma(c) \cap \Gamma(\overline{v})| \leq (\rho + \alpha)|C|$.

The assumption that $\beta > \frac{1}{2} + \frac{\rho+\alpha}{2}$ implies that $(\rho + \alpha)|C| < (2\beta - 1)|C|$. Consequently,

$$|\Gamma(\overline{v}) \cap \Gamma(c)| \leq (\rho + \alpha)|C| < (2\beta - 1)|C| \leq |\Gamma(c) \cap \Gamma(v)|.$$

We have shown if $v \in C$, then v and c share at least $(2\beta - 1)|C|$ neighbors, and if $\overline{v} \notin C$, then v and c share strictly fewer than $(2\beta - 1)|C|$ neighbors. □

Consequently, we have the following theorem.

Theorem 6.3. *Let $G = (V, E)$ be a graph and $\beta > \frac{1}{2} + \frac{\rho+\alpha}{2}$. Algorithm 1 finds exactly all the (α, β)-clusters of size s that have ρ-champions in time $O(m^{0.7}n^{1.2} + sn^{2+o(1)})$.*

To interpret the theorem, when clusters have ρ-champions with $\rho = \alpha$, a separation of $\frac{1}{2}$ is needed between β and α in order for the algorithm to find all the clusters. When ρ is larger, the gap between α and β must also be larger for the algorithm to provably succeed. For example, if $\rho = 3\alpha$, then the gap between β and α must be larger, namely $\beta > 2\alpha + \frac{1}{2}$.

The running time follows from the fact that the algorithm computes the number of neighbors that each pair of vertices share. We can precompute $|\Gamma(v_i) \cap \Gamma(v_j)|$ for all $i, j \in V$ by noting that if A is the adjacency matrix of G, then $(A^T A)_{i,j} = |\Gamma(v_i) \cap \Gamma(v_j)|$. In [Yuster and Zwick 05] it is shown that matrix multiplication can be performed in $O(m^{0.7}n^{1.2} + n^{2+o(1)})$ time. Checking the α, β conditions for a cluster of size s requires at most $O(ns)$ time, so in total, our algorithm requires time $O(m^{0.7}n^{1.2} + n^{2+o(1)} + n(n + ns)) = O(m^{0.7}n^{1.2} + sn^{2+o(1)})$.

In the case that G is a typical social network, G has small average degree and A is a sparse matrix. If we let d be the average degree of the graph, then $m = dn/2$. Thus, for small d, the algorithm runs in time $O(d^{0.7}n^{1.9} + sn^{2+o(1)})$.

6.1. Lifting the Cluster Size Assumption

We have previously assumed that the cluster size s was input to the algorithm. In order to find all clusters of any size, the deterministic algorithm would have to be run once for each value of n, requiring $O(n^4)$ time. We show how to lift this assumption by calling the previous deterministic algorithm with values of s in powers of $(1+\theta)$, where $0 < 1+\theta < \frac{1-\alpha}{1-\beta}$. In order to prove that the algorithm can still find all the clusters, we need a slightly larger gap, specifically $(1-\beta+\alpha+\rho)(1+\theta) < \beta$. Assuming such a gap, we show that each vertex can champion at most one cluster of size between $(1+\theta)^i$ and $(1+\theta)^{i+1}$, which in turn implies that there are at most $n\log_{1+\theta} n$ clusters. Furthermore, we give a small modification to the deterministic algorithm that will find all the clusters.

We begin by showing that each vertex can champion at most one cluster of size between $(1+\theta)^i$ and $(1+\theta)^{i+1}$, assuming a slightly larger gap.

Lemma 6.4. *Let $1+\theta < \frac{1-\alpha}{1-\beta}$. There are at most n clusters of size between $(1+\theta)^i$ and $(1+\theta)^{i+1}$, provided that $(1-\beta+\alpha+\rho)(1+\theta) < \beta$, for all i.*

Proof. Let C and C' be two (α,β)-clusters of size between $(1+\theta)^i$ and $(1+\theta)^{i+1}$. Further, let c be a champion of C. We show that c cannot also champion C'.

Assume that $|C| \geq |C'|$. By Corollary 4.3, C' is not a subset of C. Note that $|C \cap C'| \leq (1-\beta)|C| + \alpha|C'|$ from the proof that bounds the size of the intersection, Proposition 4.1. We now bound the number of neighbors that c has in C' from above:

$$\begin{aligned}|\Gamma(c) \cap C'| &= |\Gamma(c) \cap C' \cap C| + |\Gamma(c) \cap C' \setminus C| \\ &\leq (1-\beta)|C| + \alpha|C'| + \rho|C| \\ &\leq (1+\theta)^{i+1}(1-\beta+\alpha+\rho).\end{aligned}$$

Given the assumption that $(1-\beta+\alpha+\rho)(1+\theta)^{i+1} < \beta(1+\theta)^i$, observe that c does not have enough neighbors in C' to be a member of the cluster C'. A similar argument holds in the event that $|C'| > |C|$. □

To find the clusters, we repeatedly call the previous deterministic algorithm $O(\log n)$ times with values of s in the range $(1+\theta)^1, \ldots, (1+\theta)^{\log_{1+\theta} n}$.

To see why the algorithm works, observe that if $(1+\theta)^i \leq |C| \leq (1+\theta)^{i+1}$, then any vertex in the cluster neighbors at least $(1+\theta)^i(2\beta-1)$ vertices in C, and any vertex outside the cluster neighbors at most $(1+\theta)^{i+1}(\alpha+\rho)$ vertices in C. If there is a gap between $(1+\theta)(\alpha+\rho)$ and $(2\beta-1)$, then the modified deterministic algorithm will find all clusters. Our assumed gap of $(1-\beta+\alpha+\rho)(1+\theta) < \beta$ implies that $(\alpha+\rho)(1+\theta) < 2\beta-1$.

Theorem 6.5. *Let $\alpha, \beta, \rho, \theta$ be such that $(1-\beta+\alpha+\rho)(1+\theta) < \beta$ and $1+\theta < \frac{1-\alpha}{1-\beta}$. All (α,β)-clusters with ρ-champions can be found via $O(\log_{1+\theta}(n))$ calls to the deterministic algorithm.*

As stated, the total running time to find all clusters of any size is $O(n^3 \log_{1+\theta} n)$. However, this assumes that the maximum cluster size is n. In practice, the maximum degree in a social network is usually significantly smaller than n, and consequently, the maximum cluster size is also much smaller than n. Specifically, if Δ is the maximum degree in the graph, no cluster can be of size greater than $\frac{1}{\beta}\Delta$. Thus, when $\beta > \frac{1}{2}$, we need to call the deterministic algorithm only $\log_{1+\theta} \Delta$ times. Also, the upper bound on the cluster size improves the time it takes to check the α and β criteria from $O(n^2)$ to $O(\Delta n)$. Thus, the total running time of the algorithm is $O(n^2 \Delta \log_{1+\theta} \Delta)$.

7. Experiments

We have introduced the notion of a ρ-champion and given an algorithm for finding (α,β)-clusters with ρ-champions. A natural next question is, do (α,β)-clusters with ρ-champions even exist in real graphs? And if so, do most (α,β)-clusters have ρ-champions? To answer the first question, we study two real networks induced by coauthorship among high-energy physicists and coauthorship among theoretical computer scientists. To answer the second question, we need an algorithm that can find (α,β)-clusters independently of whether they have ρ-champions. The best previous algorithm for this problem appeared in [Tsukiyama et al. 77]. It finds all maximal cliques in a graph, i.e., all $(\alpha,1)$-clusters.

Our experiments uncovered a few surprising facts. First, our deterministic algorithm was able to find about 90% of the maximal cliques in these graphs for which $\alpha \leq \frac{1}{2}$. Next, among the cliques we missed, we found that there was no strong ρ-champion. Finally, our algorithm was orders of magnitude faster than Tsukiyama's. In short, our algorithm more quickly discovers clusters of practical interest, i.e., small α, small ρ, and large β.

7.1. Data Sets and Tsukiyama's Algorithm

As mentioned, two data sets were used: the High-Energy Physics Theory Co-Author Graph (HEP)[3] and the Theory Co-Author Graph (TA). In these graphs, authors are vertices and edges correspond to coauthorship. Some basic statistics about these graphs are given in Table 1.

[3] Available online (http://www.cs.cornell.edu/projects/kddcup/datasets.html).

Data Set	**Size of V**	**Average Degree**	$\sum_{v\in V}\lvert B_2(v)\rvert/V$
HEP	8,392	4.86	40.58
TA	31,862	5.75	172.85

Table 1. Some basic statistics about the High-Energy Physics Theory Co-Author Graph (HEP) and the Theory Co-Author Graph (TA).

Tsukiyama's algorithm finds all maximal cliques in a graph via an inductive characterization: given the maximal cliques involving the first i vertices, the algorithm shows how to extend this set to the maximal cliques involving the first $i+1$ vertices. The algorithm's running time is polynomial in the size of the graph and the number of maximal cliques. More details can be found in [Tsukiyama et al. 77].

7.2. Results

In this section we present numerical results comparing the ground truth of Tsukiyama's algorithm with our Algorithm 1. For this experiment we were interested only in cliques of size 5 or larger with α values of 0.5 or less. These are the cliques that Algorithm 1 could reasonably be expected to find. We found that the HEP graph had a total of 126 cliques satisfying this definition; our algorithm found 115, or 91%. Similarly, the theory graph had 854 cliques, and our algorithm found 797, or 93%. In Figure 3 we show the α and ρ distributions of the cliques found by Tsukiyama compared with the distribution of those found by Algorithm 1. When a bar is cut off, a number is placed next to the bar to indicate the true value. Bars have been cut off only when Algorithm 1 found all of the cliques that Tsukiyama's algorithm found.

In both theory and HEP graphs, the distribution of ρ-values among the clusters found is exactly as our theorems claim, i.e., we find all clusters for which ρ is less than $\frac{1}{2}$ and, as a bonus, a few for which ρ is larger.

Running Time. Our experiments were run on a 3-GHz Intel Xeon with 16 gigabytes of RAM. In Table 2, we report wall-clock time. The numbers for Algorithm 1 reflect the cumulative time taken with the parameter s ranging from 5 to 25.

Experiment	**HEP**	**TA**
Algorithm 1, $(\alpha,\beta)=(0.5,1)$	8 sec	2 min 4 sec
Tsukiyama	8 hours	36 hours

Table 2. Report of wall-clock times.

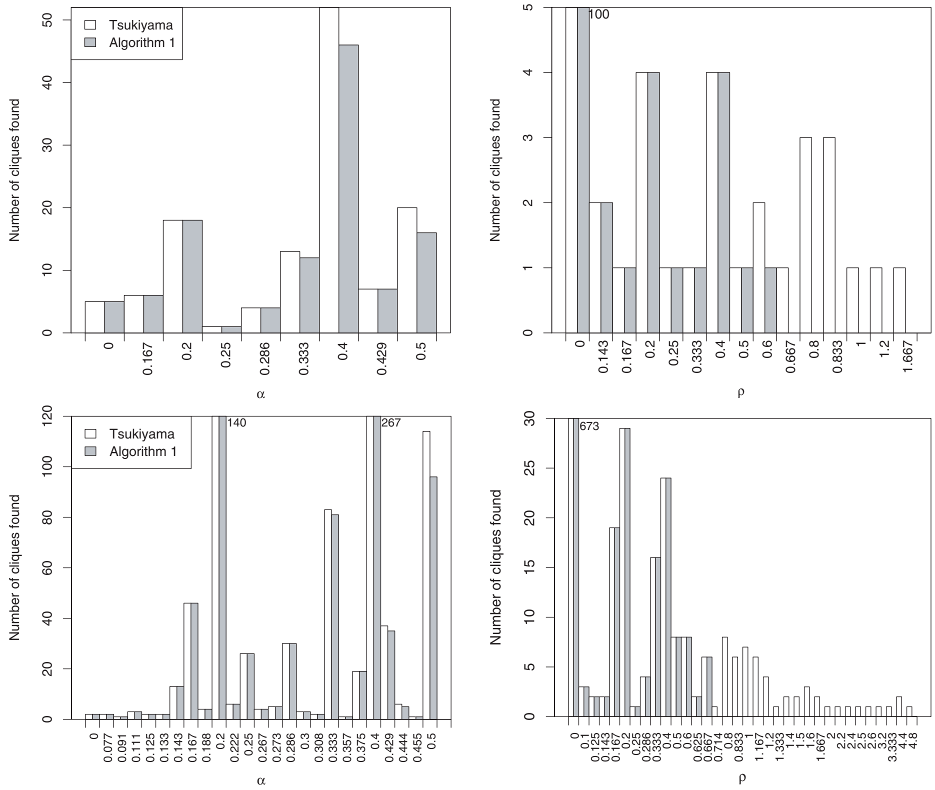

Figure 3. The figure shows α (left) and ρ (right) distributions for the cliques found by Tsukiyama's algorithm versus the cliques found by Algorithm 1. (top) HEP: Our algorithm found 115 out of 126 maximal cliques. (bottom) TA: Our algorithm found 797 out of 854 maximal cliques.

8. Summary and Future Work

In this paper, we introduced a new criterion for discovering overlapping clusters that captures intuitive notions of internal density and external sparsity. We studied several combinatorial properties of these clusters to better understand how they interact. We next introduced the idea of a ρ-champion and developed an algorithm to find (α, β)-clusters. Finally, we tested the ρ-champion assumption by comparing our algorithm with Tsukiyama's clique-finding algorithm.

With respect to future work on clustering, the most obvious direction is to develop algorithms that work when $\beta < \frac{1}{2}$. The primary difficulty in this direction is that the current definition of (α, β)-clusters allows disconnected clusters when $\beta < \frac{1}{2}$. For example, two disjoint K_5-cliques form a $(0, \frac{1}{2})$-cluster. Additional connectivity assumptions will have to be made to develop appropriate algorithms.

In addition to improving the gap between α and β, future work on generalizations of (α, β)-clustering to weighted and directed graphs is of interest. Our work assumes that edges are unweighted. But in real social networks, there is a strength of connectivity between pairs of individuals corresponding to how often they communicate. This weight could be exploited in the discovery of closely knit communities. In addition, some networks induce directed graphs; for instance, the direction of edges in email networks plays an important role in defining communities; otherwise, spam mailers would belong to every cluster. Many of our existing algorithms and theorems can be easily generalized to a directed case, but there may be other interesting results available only when directed edges are assumed.

Decentralized and streaming algorithms are essential for modern networks such as instant messaging and email graphs. In particular, it is often difficult to collect the graph in one centralized location [Kempe and McSherry 04]. Thus, algorithms that can compute clusters with only local information are needed. Further, given that social networks are dynamic data sets, i.e., users and links come and go, streaming graph-clustering algorithms are an important avenue for future research.

Acknowledgments. We thank the anonymous reviewers for their many thoughtful comments. We thank Mark Sandler and Dana Ron for valuable discussions. The third author was supported by a National Physical Science Consortium Fellowship and a Google Anita Borg Scholarship.

References

[Abello et al. 02] J. Abello, M. G. C. Resende, and S. Sudarsky. "Massive Quasi-Clique Detection." *LATIN: Latin American Symposium on Theoretical Informatics* 2286 (2002), 598–612.

[Aiello et al. 00] W. Aiello, F. Chung, and L. Lu. "A Random Graph Model for Massive Graphs." In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pp. 171–180. New York: ACM Press, 2000.

[Anderson 74] I. Anderson. *A First Course in Combinatorial Mathematics*. Oxford, UK: Clarendon Press, 1974.

[Capocci et al. 05] A. Capocci, V. Servedio, G. Caldarelli, and F. Colaiori. "Detecting Communities in Large Networks." *Physica A* 352:2–4 (2005), 669–676.

[Dourisboure et al. 09] Y. Dourisboure, F. Geraci, and M. Pellegrini. "Extraction and Classification of Dense Implicit Communities in the Web Graph." *ACM Transactions on the Web* 3:2 (2009), Article no. 7.

[Flake et al. 00] Gary William Flake, Steve Lawrence, and C. Lee Giles. "Efficient Identification of Web Communities." In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, edited by Raghu Ramakrishnan, Sal Stolfo, Roberto Bayardo, and Ismail Parsa, pp. 150–160. New York: ACM Press, 2000.

[Flake et al. 04] G. W. Flake, R. E. Tarjan, and K. Tsioutsiouliklis. "Graph Clustering and Minimum Cut Trees." *Internet Mathematics* 1:4 (2004), 385–408.

[Gibson et al. 98] "D. Gibson, J. Kleinberg, and P. Raghavan. "Inferring Web Communities from Link Topology." In *Proceedings of the 9th ACM Conference on Hypertext*, pp. 225–234. New York: ACM Press, 1998.

[Gomory and Hu 62] R. E. Gomory and T. C. Hu. "Multi Terminal Network Flows." *Journal of the Society for Industrial and Applied Mathematics* 9 (1961), 551–571.

[Gregory 07] S. Gregory. "An Algorithm to Find Overlapping Community Structure in Networks." In *Knowledge Discovery in Databases: PKDD 2007—11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, September 17–21, 2007, Proceedings*, Lecture Notes in Computer Science 4702, pp. 91–102. Berlin: Springer, 2007.

[Hartuv and Shamir 00] E. Hartuv and R. Shamir. "A Clustering Algorithm Based on Graph Connectivity." *Information Processing Letters* 76 (2000), 175–181.

[Ino et al. 05] H. Ino, M. Kudo, and A. Nakamura." "Partitioning of Web Graphs by Community Topology." In *Proceedings of the 14th International Conference on World Wide Web*, pp. 661–669. New York: ACM Press, 2005.

[Johnson et al. 88] D. S. Johnson, C. H. Papadimitriou, and M. Yannakakis. "On Generating All Maximal Independent Sets." *Information Processing Letters* 27 (1988), 119–123.

[Kannan et al. 00] R. Kannan, S. Vempala, and A. Vetta. "On Clusterings—Good, Bad and Spectral." In *Proceedings of the 41th Annual Symposium on Foundations of Computer Science*, pp. 367–377. Los Alamitos: IEEE Computer Society, 2000.

[Karypis and Kumar 98] G. Karypis and V. Kumar. "A Parallel Algorithm for Multilevel Graph Partitioning and Sparse Matrix Ordering." *J. Parallel Distrib. Comput.* 48 (1998), 71–95.

[Kempe and McSherry 04] D. Kempe and F. McSherry. "A Decentralized Algorithm for Spectral Analysis." In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, pp. 561–568. New York: ACM Press, 2004.

[Kumar et al. 99] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. "Trawling the Web for Emerging Cyber-communities." *Proceedings of WWW8, Computer Networks* 31:11–16 (1999), 1481–1493.

[Mishra et al. 04] N. Mishra, D. Ron, and R. Swaminathan. "A New Conceptual Clustering Framework." *Machine Learning* 56 (2004), 115–151.

[Newman 06] M. E. J. Newman. "Modularity and Community Structure in Networks." *Journal of the National Academy of Sciences* 103 (2006), 8577–8582.

[Palla et al. 05] G. Palla, I. Derenyi, and I. Farkas. "Uncovering the Overlapping Community Structure of Complex Networks in Nature and Society." *Nature* 435 (2005), 814–818.

[Shi and Malik 00] J. Shi and J. Malik. "Normalized Cuts and Image Segmentation." *IEEE Trans. Pattern Analysis and Machine Intelligence* 22 (2000), 888–905.

[Spielman and Teng 96] D. A. Spielman and S. Teng. "Spectral Partitioning Works: Planar Graphs and Finite Element Meshes." In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, pp. 96–105. Los Alamitos, CA: IEEE Press, 1996.

[Tsukiyama et al. 77] S. Tsukiyama, M. Ide, H. Ariyoshi, and I. Shirakawa. "A New Algorithm for Generating All the Maximal Independent Sets." *SIAM J. Comput.* 6 (1977), 505–527.

[Van Dongen 98] S. Van Dongen. "A New Cluster Algorithm for Graphs." Technical report, National Research Institute for Mathematics and Computer Science, 1998. Available at http://citeseer.ist.psu.edu/527044.html or http://www.cwi.nl/ftp/CWIreports/INS/INS-R9814.ps.gz.

[Yuster and Zwick 05] R. Yuster and U. Zwick. "Fast Sparse Matrix Multiplication." *ACM Transactions on Algorithms* 1 (2005), 2–13.

[Zhang et al. 07] S. Zhang, R.S. Wang, and X.S. Zhang. "Identification of Overlapping Community Structure in Complex Networks Using Fuzzy *c*-Means Clustering." *Physica A: Statistical Mechanics and Its Applications* 374 (2007), 483–490.

Nina Mishra, Search Labs, Microsoft Research, 1288 Pear Ave., Mountain View, CA 94043 (ninam@microsoft.com)

Robert Schreiber, HP Labs, 1501 Page Mill Road, Palo Alto, CA 94304
(rob.schreiber@hp.com)

Isabelle Stanton, Department of Computer Science, University of California, Berkeley, 387 Soda Hall, Berkeley, CA 94720 (isabelle@eecs.berkeley.edu)

Robert E. Tarjan, Department of Computer Science, Princeton University, 35 Olden Street, Princeton, NJ 08540-5233 and HP Labs, 1501 Page Mill Road, Palo Alto, CA 94304 (robert.tarjan@hp.com)

Received February 14, 2008; accepted May 6, 2008.