A

Discovering Social Circles in Directed Graphs

SCOTT H. BURTON, Brigham Young University
CHRISTOPHE G. GIRAUD-CARRIER, Brigham Young University

We examine the problem of identifying social circles, or sets of cohesive and mutually-aware nodes surrounding an initial query set, in directed graphs where the complete graph is not known beforehand. This problem differs from local community mining, in that the query set defines the circle of interest. We explicitly handle edge direction, as in many cases relationships are not symmetric, and focus on the local context because many real-world graphs cannot be feasibly known. We outline several issues that are unique to this context, introduce a quality function to measure the value of including a particular node in an emerging social circle, and describe a greedy social circle discovery algorithm. We demonstrate the effectiveness of this approach on artificial benchmarks, large networks with topical community labels, and several real-world case studies.

Categories and Subject Descriptors: H.2.8 [**Database Management**]: Database Applications; I.2.m. [**Artificial Intelligence**]: Miscellaneous

General Terms: Algorithms, Performance

Additional Key Words and Phrases: Social Circles; Directed Graphs; Local Community Search

ACM Reference Format:
Scott H. Burton and Christophe G. Giraud-Carrier, 2013. Discovering Social Circles in Directed Graphs. *ACM Trans. Knowl. Discov. Data.* V, N, Article A (January YYYY), 28 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Humans are inherently social beings that tend to associate with one another through homophily [McPherson et al. 2001]. It follows that human society, both online and offline, is characterized by a complex network of interconnections within which somewhat homogeneous groups, or communities, emerge naturally. In turn, these communities tend to have a powerful influence on the attitudes and behaviors of their members [de Klepper et al. 2010], so that an individual's social environment can often be leveraged to infer important information about that individual's attitudes, behaviors and decisions [Ajzen and Fishbein 1980; Rosenberg 1989; Yen and Syme 1999; Ståhl et al. 2001; Wei 2004; Goldberg et al. 2010; Mislove et al. 2010]. For example, a health professional may be able to improve the efficacy of his/her intervention by considering the social circle of an at-risk individual, or a workshop organizer may better target potential participants by issuing invitations around a core group of known experts.

The problem of discovering such communities around one or more individuals has recently been referred to as the *community search* problem [Sozio and Gionis 2010], to differentiate it from the well-known *community detection* problem [Fortunato 2010; Newman 2011]. Unlike community detection, which is concerned with finding arbitrary highly-interconnected subgraphs within larger networks, the goal of community

Author's addresses: S.H. Burton and C.G. Giraud-Carrier, Department of Computer Science, Brigham Young University, Provo, UT 84062

DOI: http://dx.doi.org/10.1145/0000000.0000000

search is to identify a single subgraph that includes an initial set of query individuals. The role of the query set is to provide some context to the search. Indeed, most individuals belong to multiple overlapping communities, such as work organizations, clubs, and neighborhood associations. While the issue of overlapping communities has received some attention in the context of community detection [Nicosia et al. 2009; Wang et al. 2010; Goldberg et al. 2010; Stanoev et al. 2011; Yang and Leskovec 2012c], it is clearly intrinsic to the local community search problem where a single node cannot uniquely identify the community of interest. Instead, by adding other nodes to the query set it is possible to extract different overlapping communities for the same individual depending on the content of the query set. Overlapping communities are thus handled naturally, because a node's community membership is established for each query set separately. The specification of the additional seed nodes is what determines the desired community, and ideally, these nodes are selected such that their only element of commonality is the characteristic that defines the desired community, e.g., coworkers, teammates, fellow hobbyists. For example, the social circle of an individual, that begins with two of his/her sisters, is likely to center around family relationships, while the social circle of that same individual, that begins with two of his/her professional colleagues, is likely to include mostly business relationships. In that sense, such local communities resemble what sociologists refer to as social circles [Kadushin 1966; 1968], and we refer to them as such in the following.

Here, we focus our attention on the *local* social circle discovery problem. Analogous to the difference between community detection and local community detection, the local variant of the social circle discovery problem operates under the constraint that the entire graph is not known *a priori*, and that new edges and nodes are discovered only through their adjacencies to the currently-known portion of the network. The local constraint is intrinsic to many contexts wherein knowing the entire graph is either impossible or infeasible (e.g., Web pages, Twitter users, YouTube videos).

We further focus on *directed* graphs, since many relations are naturally directed and opposite-directional links are not synonymous (e.g., publication citations, links on Web pages, followers on Twitter). Most extant community mining algorithms are designed for undirected graphs, with the assumption that they can be applied to directed graphs simply by ignoring direction and treating the graph as if it were undirected. However, if edge direction is ignored, valuable information is lost [Leicht and Newman 2008]. Furthermore, incoming links to a node may not be known without an exhaustive search of the graph rendering this approach clearly inadequate.

In this paper, we propose an effective local social circle discovery algorithm for directed graphs. Ideally, seed nodes are selected such that the only element of commonality among them is the underlying characteristic, or shared interest, that defines the desired social circle. We adopt a greedy expansion approach where nodes adjacent to the social circle are iteratively added, or those in the social circle are periodically removed, by maximizing a particular heuristic function, until a specified size is reached. We demonstrate the effectiveness of the proposed algorithm using standard benchmarks as well as case studies in large real-world social networks.

2. RELATED WORK

Most of the research regarding communities has focused on detection, where a graph is partitioned into distinct communities, based on such ideas as random walks [Pons and Latapy 2005; Rosvall and Bergstrom 2008; Kim et al. 2010], label propagation [Raghavan et al. 2007; Gregory 2010], spectral methods [Capocci et al. 2005; Smyth and White 2005], modularity [Clauset et al. 2004; Newman 2004; Reichardt and Bornholdt 2006; Fortunato and Barthélemy 2007; Berry et al. 2011], and generative models of affilia-

tion [Yang and Leskovec 2012a]. A recent, and excellent, survey of the field is in [Fortunato 2010].

In the past decade, however, efforts have begun to be expanded on community search. Significant examples of global approaches include k-cliques [Palla et al. 2005], normalized conductance [Mislove et al. 2010], and variants of density [Sozio and Gionis 2010]. As far as local approaches are concerned, there are two major groups, one that emphasizes the existence of a community boundary and one that focuses instead on density-related measures. The first group is characterized by approaches based on such ideas as local modularity [Clauset 2005], bridges to other communities [Papadopoulos et al. 2009], triangles to outside nodes [Friggeri et al. 2011], link addition rate [Bagrow and Bollt 2005], and iterative local expansion [Chen et al. 2009]. The second group includes approaches such as Iterative Scan, which alternates through phases of adding new nodes and removing community members to maximize a density metric [Baumes et al. 2005], Greedy Clique Expansion, which builds upon earlier work from [Lancichinetti et al. 2009] and adds/removes nodes in a greedy fashion to maximize a ratio of internal to total edges [Lee et al. 2010], Max-flow [Flake et al. 2002], internal density maximization [Nguyen et al. 2011], and spectral clustering [Andersen and Lang 2006; Yang and Leskovec 2012b]. Interestingly, all of these local community mining approaches assume undirected edges, with the stated (and sometimes only implicit) assumption that the algorithm can be applied to directed graphs by ignoring edge direction. However, because edge direction may limit the knowledge of links *into* an emerging community, applying undirected local approaches to directed graphs is not trivial and may embed assumptions that adversely affect the algorithm or metric. By contrast, we propose a local social circle discovery algorithm for directed graphs.

Recently, McAuley and Leskovec have presented a generative, unsupervised approach to discover an individual's social circles among their friends, which combines link and profile information [McAuley and Leskovec 2012], while Qin et al. do something similar as they cluster blogs around a given vertex of the blogosphere [Qin et al. 2012]. To the best of our knowledge, these authors are also the first, within the Computer Science research community, to use the term *social circle*. Their definition is similar to ours since they "expect that circles are formed by densely-connected sets of alters...[and] each circle is not only densely connected but its members also share common properties or traits" [McAuley and Leskovec 2012], but their motivation is different. They focus exclusively on ego networks, and essentially cluster ego's alters, building a number of circles around ego. By contrast, we take two (or more) individuals (think of ego and a small set of its alters only) and build a single social circle around them. One significant distinction is that we may get in our circle someone who is not directly connected to ego (i.e., not one of the current alters) but who is strongly connected with others in the social circle. One may think of this as a case where ego may not have yet established an explicit connection to that individual but probably should. A simple example would be a situation where an individual, say John, is connected to a number of people in his family but has no direct link to aunt Sally, whereas most other in his social circle do. McAuley and Leskovec's algorithm would not be able to put aunt Sally in any of John's circles since she is not one of his alters. Our algorithm, on the other hand, would add aunt Sally to John's family circle, on the strength of her associations with John's other family alters. Hence, while their work focuses on *organizing* the neighbors of a node into different groups, we seek to *discover* nodes that belong with the initial query set, including those that are not directly adjacent. Hence, we extend the concept of social circles to include nodes within the same community, not just those connected to a particular ego.

Finally, we note two problems that bear similarity to community search, or social circle discovery, but also differ in significant ways. First, the team formation problem,

whose goal is to identify a compatible team of experts possessing required skills, may involve an initial set of query nodes, yet the problem itself is quite different in that the defining requirements are the skills and personalities of the potential members, not their connections [Lappas et al. 2009]. Second, the graph theory problem of finding a minimum set of nodes connecting an initial query set shares some similarities with the local community search problem, but is also very different in that local community search seeks to build a cohesive set of nodes around the query set, as opposed to simply finding paths among them, and it is also very likely that the initial query nodes are already connected [Faloutsos et al. 2004; Tong and Faloutsos 2006].

3. SOCIAL CIRCLE DISCOVERY

The local social circle discovery problem for directed graphs consists of identifying a set of cohesive and mutually-aware nodes surrounding an initial query set, using only information from known nodes and the directed edges among them. This definition raises a number of important issues that must be addressed in order to formulate a node selection function that captures the underlying intuition of what "good" social circles should be like. We examine these issues in turn, and show how they affect the design of our node selection function.

Before we proceed, however, we first consider one of the fundamental tenets of our work, namely that we work explicitly with directed graphs. Many relations are naturally directed yet not inherently reciprocal (e.g., publication citations, links on Web pages, followers on Twitter), resulting in graphs of directed edges. Interestingly, most existing community mining algorithms are designed for undirected graphs with an assumption that they can be applied as is to directed graphs by simply ignoring direction. Leicht and Newman note that ignoring edge direction works reasonably well in some cases, but not in others, and in all cases it discards potentially valuable information that could enable more accurate community discovery [Leicht and Newman 2008]. Consider the four graphs shown in Figure 1, which are all isomorphic to graph 1 if edge direction is simply ignored.

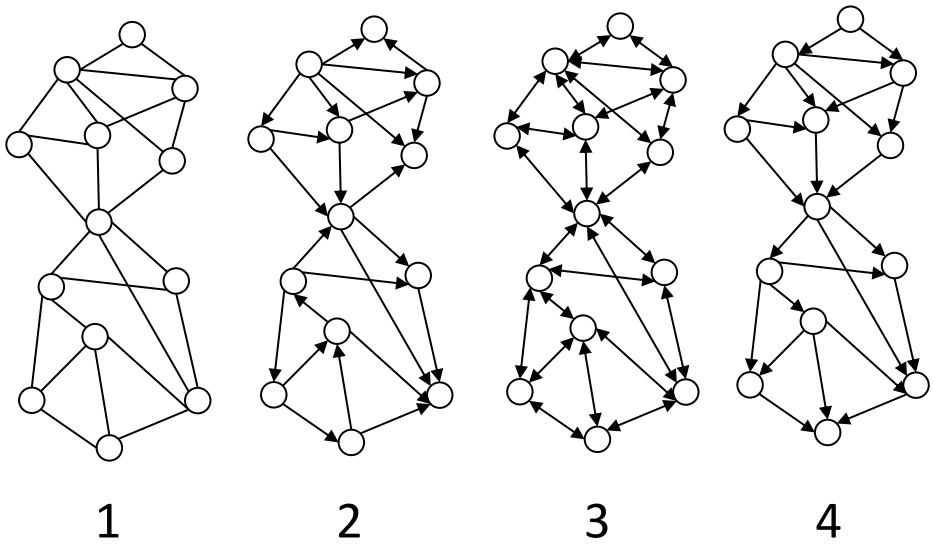

Fig. 1. Differences among graphs when edge direction is taken into account.

When edge direction is taken into account, obvious differences emerge among these graphs. For example, since Graph 3 is composed exclusively of bidirectional links, it has twice as many edges as Graph 2, which is composed of only unidirectional links. Even more relevant to social circle discovery, since all of the edges in Graph 4 point downward, the nodes at the bottom of the graph may be completely unaware of those above them that link to them. Applying undirected algorithms to directed graphs requires two important assumptions to be made. First, an assumption must be made about how to count edges. For example, if a metric requires counting the number of edges between two nodes, should a bidirectional edge count as 1 or 2? Treating directed graphs as undirected, implicitly causes bidirectional edges to be counted as 1, like any other edges in the graph. Second, an assumption is needed about whether both incoming and outgoing edges should be considered. While the natural answer may be that all edges should be used, in many instances incoming links cannot be directly discovered (e.g., links to a website) [Mislove et al. 2007]. This is exactly the situation in the local discovery context, where nodes can only be found through their links from the known portion of the graph. Because some inward links may be known through exploration of other nodes, and yet, there may exist any number of additional unknown inward links, using any of these links in calculations could lead to unexpected behavior. Furthermore, even if all edges were known, there seems to be a significant semantic difference in terms of social circle membership between a node with high in-degree (e.g., a news site that many readers link to) and a node with high out-degree (e.g., a directory-like service providing pointers to a large number of resources). Yet, treating edges as undirected would view both cases as identical.

For all of these reasons, we contend that it is important to design algorithms that handle directedness explicitly. We now return to the specific issues raised by the local discovery of social circles in that context.

3.1. The Lab Advisor Problem

Since a social circle is defined as a cohesive group of nodes around an initial query set, one would expect that the decision to include a new node in a given social circle should be independent from the existence of other collateral social circles to which that node may also belong [Friggeri et al. 2011].

As an example, consider the task of discovering the social circle around a few students who work in the same research lab. One would expect that social circle to encompass all students in the lab, as well as the lab advisor. Now, for the most part, the students are likely to have limited professional contacts outside the lab. The advisor, on the other hand, is likely to be well connected within the broader research community to many individuals outside the lab. This scenario is depicted in Figure 2, where there is a link between two nodes if the corresponding individuals have a professional relationship, Node A is the advisor, and the shaded nodes represent the students that make up the current lab social circle.

While it is true that A is part of a select group of people with whom she interacts in her research community, it is equally true that A is part of her research lab. Provided that the focus is originally on a few of A's students (query set), the lab should here be the discovered social circle. Note that many community mining algorithms, that view a community simply as a subgraph of nodes that are more densely interrelated among themselves than with the rest of the graph, put emphasis on the community's boundary to outside nodes, and would thus miss Node A and instead prefer an outside colleague of a single lab member, such as Node B, because of its fewer total number of connections.

In the case of naturally overlapping social circles, the fundamental assumption of "more in than out" does not seem to hold as "highly overlapping communities can have

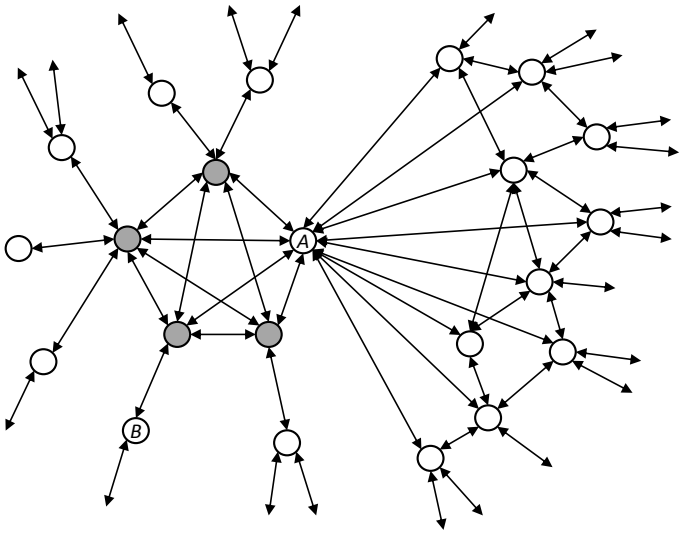

Fig. 2. The problem of selecting an advisor (A) as a member of her research lab (shaded nodes).

many more external than internal connections" [Ahn et al. 2010]. A node's membership to a specific social circle should depend solely upon the strength of its ties to that circle, and not on the presence of links (or lack thereof) to others outside of it. Hence, as [Nguyen et al. 2011], we turn our attention to the idea of maximizing internal density. Given a directed graph with N nodes and E edges, density is defined as [Wasserman and Faust 1994]:

$$Density_d = \frac{E}{(N)(N-1)} \tag{1}$$

When selecting the next node to add to an existing social circle, the denominator is the same for all candidates, since in all cases the social circle's size increases by 1, regardless of the number of links of the candidate node to the social circle. Hence, to maximize $Density_d$, one only needs to maximize its numerator, E. Now, E counts the number of edges in the social circle so that for all candidate nodes, E starts at the same value, and the differentiating factor among candidate nodes is the number of links that exist between these nodes and the social circle.

Let $e(x, y)$ be an edge indicator function defined by $e(x, y) = 1$ if there is an edge from x to y, and $e(x, y) = 0$ otherwise. Let SC be a social circle and n a node that may be added to SC. Then, the number of links between n and SC, denoted by $d_d(n, SC)$, is the sum of the number of links from nodes in SC to n and the links from n to nodes in SC, namely:

$$\begin{aligned} d_d(n, SC) &= \sum_{c \in SC} [e(c, n) + e(n, c)] \\ &= \sum_{c \in SC} e(c, n) + \sum_{c \in SC} e(n, c) \\ &= InDeg(n, SC) + OutDeg(n, SC) \end{aligned} \tag{2}$$

where $InDeg(n, SC)$ is the in-degree of n with respect to SC and $OutDeg(n, SC)$ is the out-degree of n with respect to SC. It follows that maximizing $Density_d$ (Equation 1) is the same as maximizing $d_d(n, SC)$ (Equation 2) across candidate nodes.

It is clear that, starting with the shaded nodes of Figure 2, maximizing $d_d(n, SC)$ would allow A to be added to the lab social circle. Similarly, as expected, if the set of query nodes were to include a few of A's colleagues from her broader research community, rather than a few of her students, the resulting social circle would include A and her colleagues, but none of her students. Hence, maximizing $d_d(n, SC)$ provides a principled solution to the Advisor Problem based on maximizing internal density in the context of directed graphs. Further refinements are needed, however, in response to other important issues.

3.2. The Fringe Problem

A social circle is defined as a cohesive group of nodes that surround a set of query nodes. Recall that the role of the query set is to provide context, such that, ideally, its nodes capture the characteristic that defines the desired social circle. As a result, one would expect the query set to remain somewhat prominent in, or central to, the discovered social circle, and not to be pushed out to the fringe by dense but remote groups of nodes.

One such scenario is depicted in Figure 3. Assume that the query set consists of the shaded nodes (area labeled 1). As some of the nodes in the area labeled 2 begin to be added to the growing social circle, they will have a tendency to cause the highly-connected nodes in the area labeled 3 to be added, thus leaving the initial query set on the fringe of the social circle.

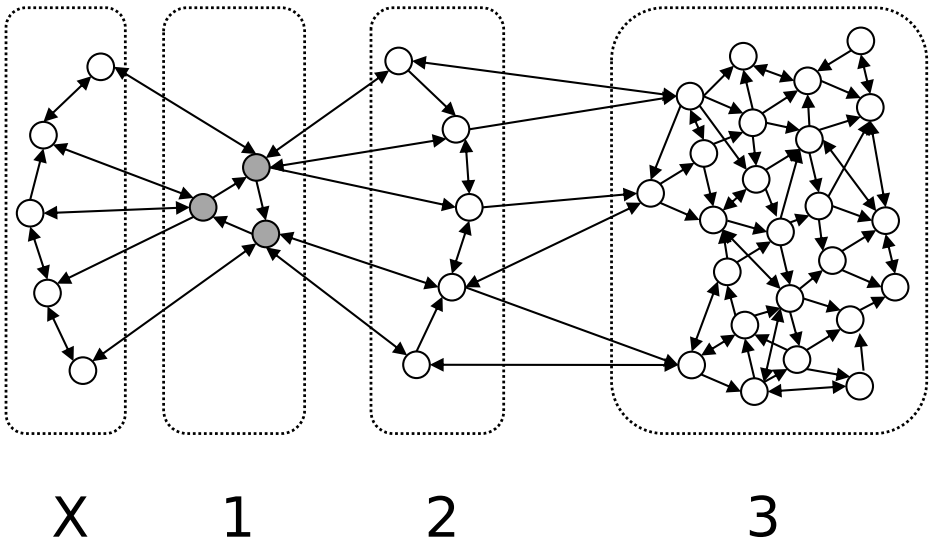

Fig. 3. The problem of adding nodes away from rather than around the query set, leaving it on the fringe of the final social circle.

Given the position of the query set in the graph, it would seem natural to expect that the nodes in the area labeled X be part of the final social circle, rather than the nodes in the dense area marked 3. Interestingly, Sozio and Gionis noticed the same

problem in the context of community search [Sozio and Gionis 2010]. In their case, in undirected graphs and having a complete knowledge of the graph, the solution was to use the minimum degree rather than the average degree of the nodes of a community as a measure of density for that community.

We are, of course, operating at the local level only, where nodes are added one at a time, based only on information from nodes in the growing social circle. If newer members of the social circle are treated the same as older ones, then they exert the same influence on new ones and can thus easily cause the social circle to divert from the initial query set, as illustrated above. Hence, our solution is based on the idea of discounted importance through length of membership to the social circle, as follows.

To maintain the relative importance of early members of the social circle, especially the query set, edges between nodes are discounted according to the time when the nodes were included in the social circle. Let $s(n)$ denote the step in the social circle-building process at which node n was added to the social circle. We modify Equation 2 to obtain the step-discounted value $\delta_d(n, SC)$ of $d_d(n, SC)$ as:

$$\begin{aligned}\delta_d(n, C) &= \sum_{c \in SC} e(c, n) s(c)^{-\alpha} + \sum_{c \in SC} e(n, c) s(c)^{-\alpha} \\ &= WgtInDeg(n, SC) + WgtOutDeg(n, SC) \end{aligned} \tag{3}$$

where $WgtInDeg(n, SC)$ is the weighted in-degree of n with respect to SC and $WgtOutDeg(n, SC)$ is the weighted out-degree of n with respect to SC. The parameter $\alpha \geq 0$ is the discount factor.

The discount factor α can be varied to specify the relative centrality of the query set. Larger values of α give prominence to the query set, and make the nodes added later in the process less and less important to the growing social circle. Smaller values of α reduce the impact of when nodes are added in the process, and thus decrease the prominence of the query set. Extremes on either end of the spectrum are undesirable, however. Indeed, as $\alpha \to \infty$, $\delta_d(n, C) \to 0$, and the addition of nodes to the growing social circle becomes a random process. On the other hand, as $\alpha \to 0$, $\delta_d(n, C) \to \sum_{c \in SC}[e(c, n) + e(n, c)] = d_d(n, SC)$, so that the algorithm focuses exclusively on density and is thus prone to the fringe problem. A balance must therefore be found. Note that there is a clear connection between the density of the underlying graph and the impact of the value of α on $\delta_d(n, C)$. Indeed, if the graph is sparsely connected, even small values of α may act as if α was very large, whereas more densely connected graphs may tolerate larger values of α. As a result, we recommend setting $\alpha = 1$, which weighs the value of connections to a node in the social circle inversely to the step in which it was added, thereby giving prominence to the query set and the first nodes added, while still allowing those with numerous connections to a number of later nodes to be added.

Maximizing $\delta_d(n, SC)$ allows us to avoid the Fringe Problem while retaining the advantages of maximizing $d_d(n, SC)$. Yet, one more problem remains, which we alluded to above when introducing directed graphs, and the distinction between in-degree and out-degree and its impact on social circle membership.

3.3. The Famous Person Problem

We have already addressed the issues of cohesiveness and overlap, and of query set centrality. There remains as part of the definition of a social circle the fact that it should be composed of nodes that are mutually aware, in line with Shaw's view that a group is "two or more persons who are interacting with one another in such a manner that each person influences and is influenced by each other person" [Shaw 1976]. While we do not require a social circle to be a k-clique, it is reasonable to expect that each

member of the social circle influences and is influenced by at least some other members of the circle. It is clearly not sufficient for a potential node to have links from every member of the social circle if there are no links back, and vice versa.

As an example, consider two cases. In the first, the graph is made up of research scientists and there is a link from one research to another if the former has cited the work of the latter. There likely exist in such a graph dense groups, or social circles, of respected research scientists who have cited each other's work extensively. For a new researcher to cite the work of these scientists (i.e., link to them) does not make her part of their social circle in any meaningful way. In the second, and somewhat reciprocal, case, the graph is made up of individuals with varying levels of popularity and there is a link from one individual to another if the former is interested in the latter's activities and life events. While such a graph will contain a number of what may be viewed as genuine friendship networks, it will also contain celebrities whose social status makes them more visible to the graph at large. Then, one may likely find a celebrity who garners the interest of (i.e., is linked from) members of the same social circle. Surely again, this does not make the celebrity a part of the social circle in any meaningful way (it is unlikely that any celebrity is keenly interested in the life of any of her fans). Both of these scenarios are captured abstractly in Figure 4 where the shaded nodes mark the current social circle, Node A represents the new researcher in the first instance and Node B represents the celebrity in the other instance.

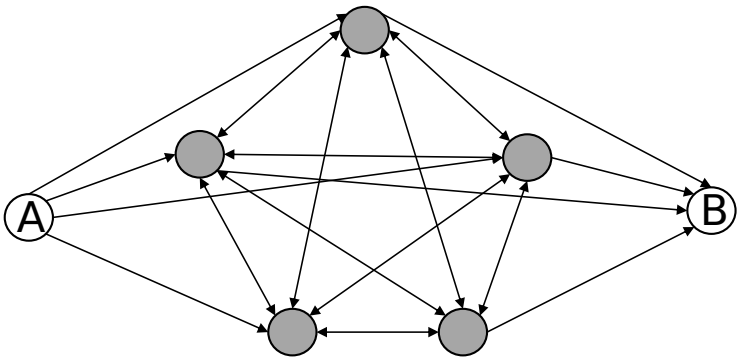

Fig. 4. Two types of nodes that should not be included in the community because they do not have mutual influence.

Note that what makes Node A and Node B unusual is that they possess only one type of directed edges. Node A links to several nodes in the social circle, but there are no links from members of the circle back to it. Conversely, Node B has links from several members of the social circle, but does not link back to any of them. Neither one of these nodes should be part of the social circle. To help exclude such nodes and enforce some level of mutual influence as per the definition of social circles, we make one final change to the node selection function, wherein we modify Equation 3, so that rather than summing over the step-discounted in-degrees and out-degrees, we select their minimum, as follows:

$$\phi(n, SC) = \min\left(WgtInDeg(n, SC), WgtOutDeg(n, SC)\right) \tag{4}$$

By maximizing $\phi(n, SC)$ over all candidate nodes, we ensure that the social circle is dense, centered around the initial query set, and its members have a significant

level of mutual awareness. Furthermore, the use of $\min$ in $\phi(n, SC)$ naturally handles the problem caused by nodes that link to the social circle, but of which the algorithm is currently unaware (due to its local nature). In this case, the $\min$ function will result in a 0, because such nodes have no links from the social circle, thus they can be consistently excluded. Only those nodes that have links from the social circle can have a score greater than 0 (the $\min$ term would result in 0 otherwise), so the set of candidate nodes can be safely reduced to only those that are known.

3.4. Social Circle Discovery Algorithm

Sozio and Gionis have proven that a greedy algorithm is guaranteed to solve the community search problem for any node-monotone function to be optimized, where node-monotonicity is defined by [Sozio and Gionis 2010]:

DEFINITION 3.1. *Let V be an underlying set of nodes, and let G_V be the collection of all possible graphs defined over subsets of V. Let f be a function that assigns a score value to any graph in G_V and node $n \in G_V$, that is, $f : V \times G_V \rightarrow \mathbb{R}$. A function f is monotone non-increasing if for every graph G, for every induced subgraph H of G, and every node v in H, $f(H, v) \leq f(G, v)$. Node-monotone non-decreasing functions are defined similarly.*

THEOREM 3.2. *$\phi(.)$ is node-monotonic non-increasing.*

PROOF. Let G be a graph and H be any induced subgraph of G. Let n be a node in H. Then, it is clear that

$$\begin{aligned} WgtInDeg(n, H) &= \sum_{c \in H} e(c, n) s(c)^{-\alpha} \\ &\leq \sum_{c \in G} e(c, n) s(c)^{-\alpha} \\ &= WgtInDeg(n, G) \end{aligned}$$

Similarly,

$$\begin{aligned} WgtOutDeg(n, H) &= \sum_{c \in H} e(n, c) s(c)^{-\alpha} \\ &\leq \sum_{c \in G} e(n, c) s(c)^{-\alpha} \\ &= WgtOutDeg(n, G) \end{aligned}$$

It follows immediately that $\phi(n, H) \leq \phi(n, G)$, which establishes the result. □

Other than the function to optimize, which captures the specific group properties one is interested in, the formal definition of the community search problem and that of the social circle discovery problem are identical. Hence, it follows from Theorem 3.2 that Sozio and Gionis' greedy algorithm, equipped with the function ϕ, is guaranteed to solve the social circle discovery problem. However, as one may expect, that algorithm, and the subsequent guarantee of optimality, require complete knowledge of the graph. Here, we are concerned specifically with the local version of the problem, where only those nodes that members of the growing social circle link to are available. While we cannot guarantee global optimality in this context, if the algorithm adopts an alternative greedy approach where at each step it selects the node that maximizes ϕ among all candidate nodes, then we retain at least some local optimality. Details are shown as Algorithm 1.

ALGORITHM 1: Social Circle Discovery Algorithm

Input: Set Q of initial query nodes, maximum size max of the social circle, and frequency of node removal f
Output: A social circle SC of size at most max

1: $AddStep \leftarrow 1$
2: **for all** q in Q **do**
3: $s(q) \leftarrow AddStep$
4: **end for**
5: $SC \leftarrow Q$

6: $NumIter \leftarrow 1$
7: **while** $|SC| < max$ **do**
8: $AddStep \leftarrow AddStep + 1$
9: $N \leftarrow \{n \mid \exists c \in SC \wedge e(c, n) = 1\}$
10: **if** $N = \emptyset$ **then**
11: **break**
12: **end if**
13: $w \leftarrow \text{argmax}_{n \in N}(\phi(n, SC))$
14: $s(w) \leftarrow AddStep$
15: $SC \leftarrow SC \cup \{w\}$
16: **if** $NumIter \mod f = 0$ **then**
17: $c \leftarrow \text{argmin}_{c \in \{SC \setminus Q\}}(\phi(c, SC))$
18: $SC \leftarrow SC \setminus c$
19: **for all** $x \in SC : s(x) > s(c)$ **do**
20: $s(x) \leftarrow s(x) - 1$
21: **end for**
22: $AddStep \leftarrow AddStep - 1$
23: **end if**
24: $NumIter \leftarrow NumIter + 1$
25: **end while**

26: Return SC

Algorithm 1 takes as input the query nodes, the maximum size of the desired social circle and the frequency of removal, and produces as output a social circle of at most the specified size. Lines 1-5 set the add-step counter to 1, assign that value to all of the query nodes, and initialize the social circle to the query set. The number of iterations is initialized to 1 on line 6. Its purpose is to assist in the node removal process. As we wish to consider node removal with frequency f, i.e., after every f iterations through the main loop, we can use the number of iterations so far and check for node removal every time it is divisible by f, as shown on line 16, where $\mod$ is the modulo operator. Lines 7-25 contain the main loop, which runs until the social circle reaches the user-specified size. On line 8, the add-step counter is incremented by 1. On line 9, the set of all neighbors of the current social circle is computed as the set of all nodes that any member of the social circle links to. Lines 10-12 handle the possibility that the algorithm runs out of candidate nodes to add to the social circle before reaching the maximum size limit set by the user. If there are no neighbors to consider, the algorithm simply breaks out of the loop. Otherwise, on line 13, the neighbor node that maximizes ϕ is selected. In the event of a tie, the tie is broken by the δ function from Equation 3 (i.e., using the sum of the terms rather than the $\min$). If a tie still remains, it is broken arbitrarily. On lines 14-15, the winning node's add-step is set and the node is added to the social circle. Upon successfully passing the test of line 16, every f iterations,

lines 17-22 effect node removal. On lines 17-18, the node in the current social circle with the smallest ϕ value is selected and removed from the social circle. Note that we explicitly exclude the nodes of the query set from this selection as it makes little sense to remove them. In order to avoid skipping add-step values, lines 19-22 decrement by 1 the add-step values of all of the nodes that were added to the social circle after the node being removed, and then decrement by 1 the add-step counter. Finally, the number of iterations is incremented by 1 on line 24. Once a social circle of size at most max has been found, it is returned (line 26).

3.4.1. Stopping Criterion. One important aspect of the local social circle discovery problem is the decision of when to stop growing the circle. Determining an adequate stopping criterion is indeed a difficult problem. Some algorithms have a built-in stopping criterion (e.g., see [Luo et al. 2008]), while others (e.g., see [Clauset 2005; Bagrow and Bollt 2005]), like ours, do not, so that external criteria may be applied. In this latter case, the mechanisms for building a social circle and the decision to stop it are decoupled. Using a pre-set maximum size is a simple, user-driven approach that may leverage human domain knowledge. Other approaches rely on properties deemed to characterize desirable communities (e.g., density, p-strength), but even these are not without problems. For example, in the case of p-strength, one generally has to try several values of p, and in the case of least-square fit the procedure is involved and has "several semi-arbitrary factors" [Bagrow 2008]. Furthermore, empirical results suggest that different combinations of algorithms and stopping criteria produce different results depending on, for example, the size of the community or the degree of the starting node. Choosing a stopping criterion is even more challenging when dealing with local methods, and when overlapping and hierarchical communities are sought. In one instance a smaller set may be preferred while in another, with identical starting conditions, a larger set could be desired. In some situations, algorithms with built-in stopping criteria stop short of the whole community, while in other situations, algorithms with a poorly chosen pre-set maximum size may extend beyond the "true" community. There probably is not a universal stopping criterion for local community search. Like others, our social circle discovery algorithm does not have a built-in stopping criterion. Our current implementation uses a pre-set maximum size (parameter max in Algorithm 1). Perhaps an additional parameter could be used in our approach to specify a relative size or a density desired for stopping. This, of course, would require proper treatment in its own right, with various settings examined on different types of graphs, which we leave for future work.

3.4.2. Query Set Selection. Another important aspect of the local social circle discovery problem is the selection of the query set (parameter Q in Algorithm 1). In theory, one may argue that the problem of selecting the query set is independent of the mechanics of discovering a social circle. Indeed, given any query set, Q, Algorithm 1 will produce a social circle. However, recall that the specification of Q is what determines the desired social circle. Thus, the quality of a social circle depends heavily on the nodes in Q exhibiting and sharing the characteristics that define the desired social circle, so that, in practice, the choice of Q is critical. Finding an adequate query set can be a significant challenge in its own right, since identifying similar nodes, such as at-risk individuals in a health-related application or genes with related functions in a genomic application, is often not trivial and may require domain expertise. We note that some work has recently been done in the more general case of seeding strategies [Lee et al. 2011], but the question remains largely open. While we touch upon query selection strategies in our experiments, our focus here is directed primarily at identifying social circles once a query set has been defined.

3.4.3. Computational Complexity. There are two main contributors to the computational complexity of Algorithm 1: the discovery of neighbor nodes (line 9) and the evaluation of the ϕ value with regard to these nodes (lines 13 at each iteration, and line 17 every f iterations). For simplicity, let $c = |SC|$ and let d be the average out-degree of any node in the overall graph. For each node in SC, the algorithm checks all of its out-links and adds the corresponding nodes to the set of neighbors. Hence, the complexity of computing the set N of neighbors (line 9) is $O(cd)$. Now, let n be one of the neighbors in N. In order to compute $\phi(n, SC)$, the algorithm needs the in-degree and out-degree of n with respect to SC. The in-degree, $InDeg(n, SC)$, can be obtained by iterating over the elements of SC and checking whether n is one of the nodes they link to. Hence, the complexity of computing $InDeg(n, SC)$ is $O(cd)$, if we assume a linear search through the out-nodes. The out-degree, $OutDeg(n, SC)$, requires finding all of the nodes that n links to, and for each, check whether it belongs to SC. Hence, the complexity of computing $OutDeg(n, SC)$ is also $O(cd)$, again assuming linear search through SC. Computing the weighted versions of these quantities and finding the minimum is O(1), so that the complexity of computing $\phi(n, SC)$ is $O(cd)$. Since the size of N is $O(cd)$, the complexity of finding the node that maximizes phi (line 13) is $O(c^2d^2)$. All other steps of the algorithm are trivially $O(1)$. Now, the main loop (lines 7-25) is executed a finite number of times bounded by max, hence the algorithm's overall computational complexity is $O(c^2d^2)$. Furthermore, note that $c \leq max$ ad max is a finite value selected by the user. Hence, the complexity of Algorithm 1 is $O(d^2)$.

Notice that if incoming links can be observed directly, so that any node may have access to all of the nodes it links to as well as all of the nodes that link to it (e.g., Twitter users that follow an account), then with the use of hash tables to store these lists, it is possible to reduce the complexity of computing both $InDeg(n, SC)$ and $OutDeg(n, SC)$ to $O(c)$. And in this case, the complexity of Algorithm 1 is only $O(d)$. This savings can be dramatic in some situations, such as those shown in Section 6.1 where the degree of the nodes is large (e.g., $d > 10^6$).

We now turn to an empirical analysis of Algorithm 1 through synthetic benchmark datasets, networks for which topical communities have been identified a priori (and thus may serve as ground-truth for testing purposes), and several real case studies that exercise the unique features of our approach.

4. BENCHMARK RESULTS

Using the established LFR benchmark [Lancichinetti et al. 2008; Lancichinetti and Fortunato 2009] for directed graphs we can objectively evaluate the quality of our algorithm. It is important to note that these benchmarks were designed for the more traditional community mining problem, in which the community boundaries are clearly defined. Yet, our method is not hurt by this added property. We first consider the case of disjoint communities, wherein every node belongs to exactly one community. Next, we use benchmarks that include nodes with overlapping community memberships, wherein a certain number of nodes belong to multiple communities. For simplicity, we restrict our attention to unweighted graphs, where edges have a value of 1 when a connection exists and 0 otherwise.

For comparison, we consider three common local community mining algorithms. Even though these algorithms were designed for community mining, as opposed to finding social circles around a query set of nodes, the comparison provides a quantifiable way to evaluate our approach. We consider 1) Clauset's local modularity, which seeks to find a steep boundary [Clauset 2005]; 2) Greedy Clique Expansion, which maximizes the number of internal to total links in a density-like fashion [Lee et al. 2010; Lancichinetti et al. 2009]; and 3) Iterative Scan, which alternates between phases of adding and removing nodes to maximize a density metric [Baumes et al. 2005]. For pa-

rameters, for the Greedy Clique Expansion, a value of $\alpha = 1$ in the recommend range is used, and for our algorithm we use default values of $\alpha = 1$ and $f = 3$.

4.1. Disjoint Communities

First, we consider the case of graphs where every node belongs to a distinct community with no overlapping memberships. We generate a set of directed LFR benchmark graphs, each with 1,000 nodes, varying the community size range to be 20-50 nodes and also 40-100 nodes. In addition, we use two different values, 0.2 and 0.4, for the mixing parameter μ, which defines the amount of linking between nodes in different communities. The other parameters were held constant at standard default values, as follows: average in-degree $k = 15$, maximum in-degree $maxk = 50$, minus exponent for degree sequence $t_1 = 2$, minus exponent for community size distribution $t_2 = 1$, and total number of nodes $N = 1,000$.

For each configuration setting, a separate social circle is discovered around each of the 1,000 nodes as the initial query node. It should be noted that the Greedy Clique Expansion and Iterative Scan methods are designed to find all communities in a network and in so doing, they prescribe processes for determining pockets of nodes from which to begin, and then expand around them. However, in this case we are interested in finding a separate social circle around every node in the graph. Thus, we compare only the expansion phases of these algorithms, not their seeding strategies. Similarly, it should be noted, that our algorithm (and likely the others as well) would perform better if the initial query set included additional nodes from the desired community, but for comparison, only the single starting node is used.

Because local modularity and our approach do not contain a hard stopping criterion, all of the algorithms are stopped when the size of the social circle matches the size of the correct community defined in the benchmark (e.g., if the correct community has 25 members, the algorithms run until the social circle contains at most 25 nodes). In the case of the Greedy Clique Expansion and Iterative Scan methods, if their terminating conditions are reached prior to this point, then the discovery is halted at that point. These benchmarks are directed and are treated as if only outgoing connections can be determined, as is the case with many real-world networks (e.g. blog links, citations, etc.). Thus, if an algorithm seeks to discover the neighbors of a node, only the outgoing neighbors are returned. Once a social circle is discovered, it is compared against the correct community by evaluating the F-Measure. A separate F-Measure value is determined for the social circle around each start node, and then averaged across the 1,000 circles. The results are shown in Figure 5, where the error bars represent one standard deviation.

As shown, in each case our method performed significantly better than the other three methods on these benchmarks (t-test, $p < 0.01$). The large variance in the values is a result of the fact that if algorithms added nodes outside the community early in the process, they would likely continue adding nodes outside the community. This further demonstrates the value of starting with a set of query nodes, as opposed to just a single one. The relatively smaller variance of our method is the result of enforcing that the start node remains prominent which helps avoid discovering a completely different dense set with the start node on the fringe. The Iterative Scan method had excellent precision (> 0.98 on average for each network), but often terminated before identifying the complete set.

4.2. Overlapping Communities

Next, we evaluate the ability of each algorithm to discover social circles in graphs where nodes may belong to multiple overlapping communities. For this comparison, we generate a set of directed LFR benchmark graphs with overlapping communities.

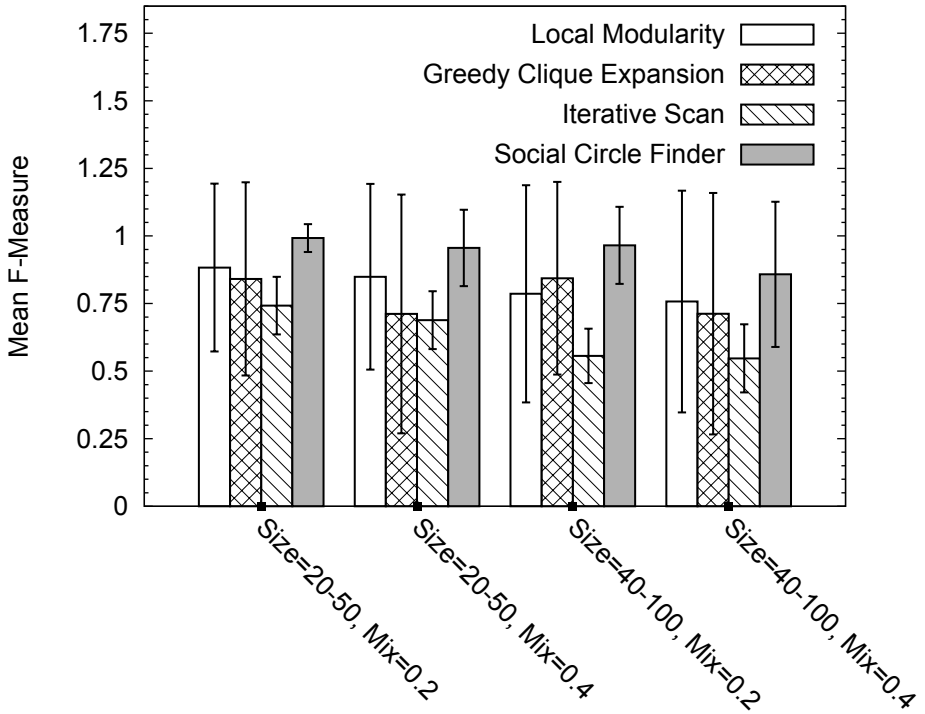

Fig. 5. Comparison on non-overlapping communities.

We use the same parameters as before, this time holding constant the community size range at 20-50, and the mixing parameter $\mu = 0.2$. We vary the number of nodes that have overlapping memberships to be either 100 or 300 (10% or 30% of the nodes), and also vary the number of memberships for those overlapping nodes to be either 2 or 4 communities.

As above, separate social circles are discovered around each of the 1,000 nodes in the graph. However, in the case of the nodes that belong to multiple communities, it is ambiguous which of the overlapping communities is desired, so in this case we attempt to discover each of the overlapping communities, by starting the algorithm separately with the node and an arbitrary neighbor in each desired community. The results were again averaged across all social circles discovered in the graph with the mean and standard deviations of the F-Measure shown in Figure 6.

In this case, our approach outperformed each of the others significantly on the first two benchmarks (t-test, $p < 0.01$). It was also the highest in the third case, but the results were not statistically significant, and in the fourth case, Local Modularly was slightly higher, but again the results were not significant. As before, the large amounts of variation arise from the fact that when algorithms "missed" the correct community, they tended to completely miss it. On the other hand, our algorithm was less susceptible to adding dense sets away from the start node, resulting in lower variance and suggesting increased robustness.

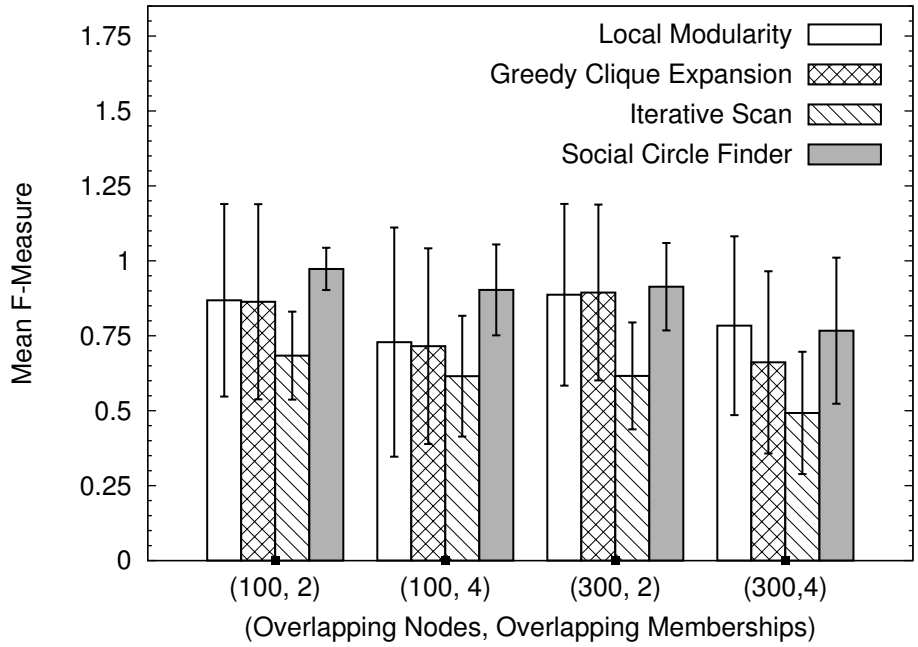

Fig. 6. Comparison on overlapping communities.

5. TOPICAL COMMUNITIES

In addition to the artificial benchmarks described above, we also evaluate our approach on real-world networks with user-specified labels. For our evaluation, we use two large directed datasets, one containing data from Flickr and the other containing data from YouTube [Mislove et al. 2007]. The Flickr dataset consists of 1.8M users crawled from the site, containing 22M directed links, and 104K user groups. The YouTube dataset consists of 1.2M users crawled from the site, along with 4.9M directed links, and 30K user-groups.

The user-groups of these datasets express common interests of the users, in other words, topical communities. There is no reason a priori to assume that there is some congruence between topical and structural communities. Indeed, establishing the correspondence (or lack thereof) between topical and structural communities in YouTube and Flickr would be very valuable (and warranted to support any related claim). However, we feel that such a study is beyond the scope of the current paper. Since all of the algorithms compared here build structural communities, the risk of assuming that the group labeling also captures structural properties is that the absolute results, i.e., the matching of the extracted structure with the topic groups, may be suboptimal (indeed in some cases here members of topical communities do not make up a structural community). Yet, for the same reason, none of algorithms are at an advantage or a disadvantage in this context. For purposes of comparison (i.e., relative results only), there is no harm in assuming correspondence, and the labeling offers a nice way to quantify performance across algorithms. Hence, we treat the topic labeling as a type of ground-truth community assignment.

5.1. Query Node Selection

As discussed, an important element of the local social circle discovery problem is the selection of query nodes. While in many cases the query set is determined beforehand, and is the reason for discovering the social circle, in other cases, it may be that an initial member and a characteristic of interest are known, but neighbors of the individual need to be added to the query set to discover the desired social circle. For example, consider the case of identifying a social circle of business contacts around an individual. The question arises, which co-workers should be included in the query set to best define the social circle? Using the labeled datasets, we evaluate different selection mechanisms for selecting a second member of the query set, given a start node and a community of interest. We consider the following possible selection criteria:

(1) Arbitrary. Select an arbitrary member of the community.
(2) Least Other Groups. Select the node that belongs to the least number of other communities.
(3) Least Overlapping Groups. Select the node that has the fewest number of communities in common with the first.
(4) Least Outside Friends. Select the node that has the fewest connections outside the desired community.
(5) Highest In-group Ratio. Select the node that has the highest ratio of friends inside the community to those outside.
(6) Most Inside Friends. Select the node that has the greatest number of friends in the desired community.

To evaluate these different criteria, we select arbitrary nodes from the network, and discover the various overlapping communities it belongs to. For each of these communities, we select a second node based on each of the different criteria and use the two nodes as a query set to discover a social circle. We treat the network as if only outgoing connections are known. For consistent comparison, in each case, we add 20 nodes to the query set and count the number of them that are part of the desired labeled set. We exclude communities where none of the neighbors are in the desired set and those that had fewer than 20 additional connected members in the labeled community. Figure 7 shows the average number of correct nodes added to the set for 8,158 evaluations each on the YouTube dataset and 1,949 evaluations each on the Flickr dataset. As shown, for each data set, selecting the node with the highest ratio of internal friends to external friends is the most effective way of selecting a second query node.

5.2. Importance of Additional Query Nodes

In addition to the manner of selecting additional query nodes, the *number* of these initial nodes can also potentially impact the effectiveness of discovering other nodes in the desired local social circle. Using the best method of query node selection above (highest in-group ratio) we discover social circles around query sets that range in size from 2 to 10 members. Again, for consistent comparison, we evaluate the number of correctly identified nodes in the first 20 added after the initial query set. Similar to the previous experiment, we select arbitrary nodes from the network and for each of their labeled community memberships, we select a query set of different sizes. As the largest query set requires starting with 10 nodes and discovering 20 more, we exclude communities where the initial node has fewer than 9 direct neighbors in the community, and where the connected component of the labeled community contains fewer than 30 members. Figure 8 shows the number of additional nodes correctly identified in the first 20 added after the query set for 2,132 evaluations each on the YouTube data and 1,331 evaluations each on the Flickr data.

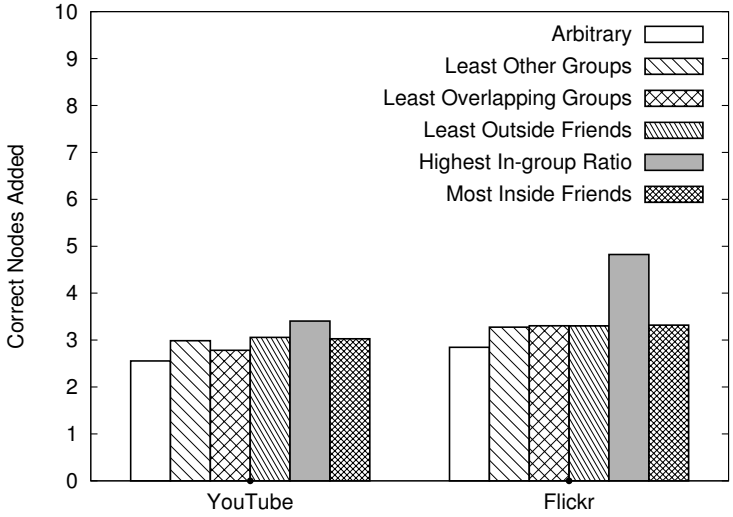

Fig. 7. Number of correct nodes found in the first 20 for different query selection mechanisms.

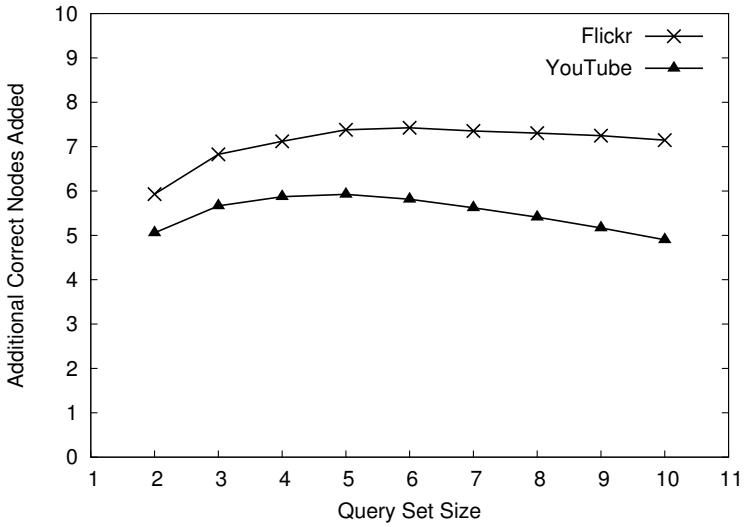

Fig. 8. Number of correct nodes found in the first 20 (after the query set) for query sets of different sizes.

As shown, using more than two nodes helps to better identify the social circle of interest with the largest increase in value coming from adding a third and fourth member to the query. Interestingly, for these datasets, having more than 5 or 6 query nodes does not increase effectiveness. The fact that the number of correct additional nodes declines may be because the additional query members were some of the "easier" nodes to identify, so by starting with them already in the social circle, the task is to find other, potentially more "difficult," members.

5.3. Algorithm Comparison

Using these same data sets, we can also compare the performance of the different algorithms in discovering the labeled communities. As before, we compare the number of correct nodes of the first 20 added after the query set. Because the Iterative Scan algorithm alternates through phases of addition and deletion, it cannot be cleanly stopped at a specific number of members, and therefore is not included in this comparison. As with the previous experiments, we arbitrarily select nodes from the networks and for each community to which they belong, we select a second node for the query set and discover a social circle around them. For the selection of the second node, we use the best approach from before (highest in-group ratio), and exclude communities where the initial node has no direct neighbors in the community and where the number of additional connected members of the labeled community is less than 20.

As the algorithms run at different levels of efficiency, some were able to complete more evaluations than others. The relative efficiency of our approach is noteworthy. Thus, we show the rolling average over the number of trials completed. Here, each trial represents the discovery of a separate community/social circle from different query sets. For example, 100 trials means that 100 different communities have been discovered around 100 different query sets in the graph. Hence, the rolling average after, for example 500 trials, represents the average number of correct nodes added to 500 separate social circles. Note also that, as stated above, we focus strictly on the number of correct nodes within the first 20 added to the community after the query set, so that the reported rolling averages should be viewed as number of correct nodes added out of 20.

Figure 9 shows the rolling average per iteration for the YouTube dataset and Figure 10 shows the averages for the Flickr dataset. As can be seen, our method clearly outperforms and is more efficient than the other methods.

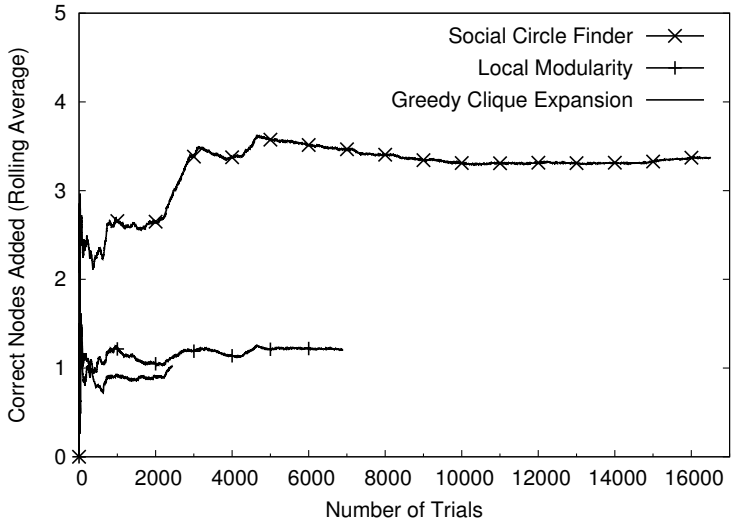

Fig. 9. Number of correct nodes added out of the first 20 on the YouTube dataset.

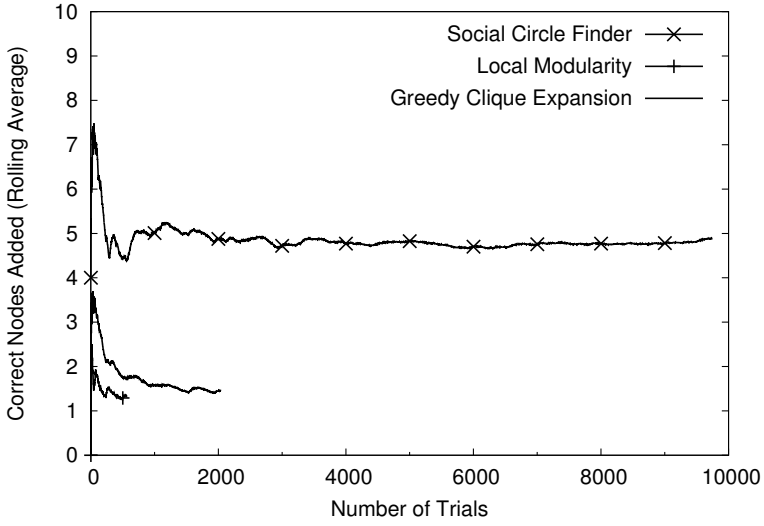

Fig. 10. Number of correct nodes added out of the first 20 on the Flickr dataset.

6. CASE STUDIES

Finally, in addition to analyzing results on benchmark graphs and on anonymous networks with topical community assignments, we also qualitatively validate our approach in two different real-world social networks: Twitter and the blogosphere. These networks complement each other as case studies because they have significantly different graph properties. On the other hand, each of these graphs is directed, incredibly large and complex, and requires a local solution.

6.1. Twitter User Social Circles

We first apply our approach to building social circles of users on the social network platform Twitter. For demonstration purposes, we have chosen to consider social circles around well-known individuals (at least in the United States). For all query individuals, we show the number of individuals who follow them (followers) and the number of individuals they follow (following) as of 25 January 2013. Clearly, an individual's lists of followers and following vary over time. We include the numbers here only to give a sense of the relative sizes of these lists and the "social status" of the corresponding individuals. Also, the teams of professional athletes and the positions of politicians are not constant over time, and we report them as they were at the time of the discovery in January 2013.

We first turn to professional basketball players, and discover a social circle around three prominent players: LeBron James, NBA player for the Miami Heat (Twitter account: @KingJames; followers: $\sim$ 7M; following: 286), Derek Fisher, NBA player for the Dallas Mavericks and president of the NBA Players Association (Twitter account: @DerekFisher; followers: $\sim$ 930K; following: 189), and Rajon Rondo, NBA Player for the Boston Celtics (Twitter account: @RajonRondo; followers: $\sim$ 885K; following: 62). By choosing players from different cities, we avoid discovering a social circle focused on a certain market such as radio or TV personalities from that city. As discussed earlier, the goal is to choose an initial set such that the only common characteristic is the desired trait (in this case, NBA players). Using these three accounts as the query set, we apply our Social Circle Discovery algorithm to build a social circle of $max = 75$

members, with $\alpha = 1$ and $f = 3$. The first 20 members of the resulting social circle are shown in Table I. Of the 75 members of the discovered social circle, 62 were NBA players or groups, 4 were affiliated with the NBA (such as former players, trainers, and agents), 2 were other professional athletes, 5 were other popular figures (such as musicians and actors), and 2 were athletic news organizations.

Table I. NBA Social Circle Members

Step	Twitter Account	Name	NBA Team
1.	KingJames	LeBron James	Miami
1.	derekfisher	Derek Fisher	Dallas
1.	RajonRondo	Rajon Rondo	Boston
2.	KDTrey5	Kevin Durant	Oklahoma City
3.	rudygay22	Rudy Gay	Memphis
4.	John_Wall	John Wall	Washington
5.	russwest44	Russell Westbrook	Oklahoma City
6.	DWRIGHTWAY1	Dorell Wright	Philadelphia
7.	Baron_Davis	Baron Davis	New York
8.	JCrossover	Jamal Crawford	Portland
9.	CP3	Chris Paul	LA Clippers
10.	NBA		NBA Account
11.	nate_robinson	Nate Robinson	Golden State
12.	MikeVick	Mike Vick	
13.	KyrieIrving	Kyrie Irving	Cleveland
14.	BooBysWorld1	Daniel Gibson	Cleveland
15.	RealTristan13	Tristan Thompson	Cleveland
16.	SteveNash	Steve Nash	LA Lakers
17.	Avery_Bradley	Avery Bradley	Boston
18.	unclejeffgreen	Jeff Green	Boston

Using LeBron James as a starting point, it is actually possible to be interested in, and discover, other social circles or overlapping communities. Indeed, in addition to being a professional basketball player, LeBron James is also a figure of popular culture, so that another social circle may be obtained if we include popular figures rather than professional basketball players in the query set with him. To verify this hypothesis and further validate our Social Circle Discovery algorithm, we re-run the algorithm with a query set comprising Lebron James and two pop culture individuals: Ciara, a musician (Twitter account: @Ciara; followers: ~ 3M; following: 67), and Charlie Sheen, and actor (Twitter account: @CharlieSheen; followers: ~ 9M; following: 106). As before, $\alpha = 1$ and $f = 3$. However, we set $max = 20$ as most of these individuals have very large lists of followers, which greatly affects computation time due to the request rate restrictions enforced by Twitter. The members of the resulting social circle are listed in Table II.

All of the members of this social circle are entertainers of some kind, and each of their Twitter accounts has been "verified" by Twitter as the correct account of a popular figure. Of the 20 members of this set, 11 are musicians, 5 are entertainers (actors, musicians/actors, etc.), and 4 are professional athletes. This group clearly represents a rather different social circle to which Lebron James also belongs. Incidentally, his friendship with the famous rapper, and part-owner of an NBA team, Jay-Z, was made newsworthy over whether the friendship could help lure him to that team.

We note that it would be difficult for boundary-focused community detection algorithms to discover a community of popular figures because of their numerous links with outsiders (the Lab Advisor Problem). Properly handling the links from outsiders also requires a directed approach, and illustrates the importance of accounting for mutual connection to the growing set (the Famous Person Problem). In addition, algorithms that require iteratively trying each outside member as a member of the community,

Table II. Popular Culture Social Circle Members

Step	Twitter Account	Name / Stage Name	Status
1.	KingJames	LeBron James	Athlete
1.	ciara	Ciara	Musician
1.	charliesheen	Charlie Sheen	Actor
2.	Ludacris	Ludacris	Musician
3.	SnoopDogg	Snoop Dogg	Musician
4.	lala	La La	Entertainer
5.	iamdiddy	P. Diddy	Musician
6.	NeYoCompound	Ne-Yo	Musician/Actor
7.	chrisbrown	Chris Brown	Musician
8.	KevinHart4real	Kevin Hart	Actor
9.	carmeloanthony	Carmelo Anthony	Athlete
10.	myfabolouslife	Fabolous	Musician
11.	Wale	Wale Folarin	Musician
12.	Tyrese	Tyrese Gibson	Musician/Actor
13.	djkhaled	DJ Khaled	Music Producer
14.	MeekMill	Meek Mill	Musician
15.	CP3	Chris Paul	Athlete
16.	Nas	Nasir Jones (Nas)	Musician
17.	DwyaneWade	Dwyane Wade	Athlete
18.	DJCLUE	DJ Clue?	Musician

such as those in the benchmark comparison, cannot be effectively run on Twitter with these highly-popular users because it would require millions of calls to the Twitter API (which limits request rates). For this reason, we have not included comparison with the other algorithms used on the benchmark graphs. By contrast, our approach can be run, albeit still slowly in some cases due to the rate limitations, because we are required only to know the follower/following lists of the members of the growing social circle.

In addition to professional athletes, members of the United States Congress have become prominent users of Twitter, and have strong ties to one another, particularly other members of the same political party. Using our approach and a query set of members of each party, we can discover other representatives from that party. Choosing five Democrats and five Republicans, we build two separate social circles of 100 members (i.e., $max = 100$), with $\alpha = 1$ and $f = 5$. For the initial query set, we selected the party leaders in the House of Representatives, as well as two additional members of the House, and two members of the Senate. The initial query sets for the two social circles are as follows.

— Democratic Congress Query Set
  — Nancy Pelosi, House Minority Leader (Twitter account: @NancyPelosi; followers: ~ 300K; following: 248)
  — Steve Israel, House of Representatives (Twitter account: @RepSteveIsrael; followers: ~ 10K; following: 226)
  — John Conyers, House of Representatives (Twitter account: @RepJohnConyers; followers: ~ 6K; following: 438)
  — John Kerry, Senate (Twitter account: @JohnKerry; followers: ~ 60K; following: 223)
  — Charles Schumer, Senate (Twitter account: @ChuckSchumer; followers: ~ 47K; following: ~ 28K)
— Republican Congress Query Set
  — John Boehner, Speaker of the House (Twitter account: SpeakerBoehner; followers: ~ 437K; following: ~ 14K)
  — Jason Chaffetz, House of Representatives (Twitter account: @JasonInTheHouse; followers: ~ 35K; following: ~ 22K)

— Darrell Issa, House of Representatives (Twitter account: @DarrellIssa; followers: $\sim$ 72K; following: $\sim$ 23K)
— John Boozman, Senate (Twitter account: @JohnBoozman; followers: $\sim$ 12K; following: 259)
— Roy Blunt, Senate, (Twitter account: @RoyBlunt; followers: $\sim$ 22K; following: $\sim$ 8K)

Each of the members of the discovered social circles were involved in politics, even though not all of them were actually representatives. Table III shows statistics of the resulting social circles.

Table III. United States Congress Twitter Social Circles

	Democratic	Republican
Total Members	100	100
Congress (same party)	94	71
Congress (other party)	0	0
News and Reporters	6	14
Foundations and Activists	0	15

Of the 100 members of the Democratic set 94 were accounts for Democratic representatives (either individual accounts, or groups such as the official account for a Democratic congressional committee). In the Republican set, 71 of the 100 members were accounts for Republican representatives or their groups. In each of these social circles there were many accounts of other politically involved users (news organizations, foundations, etc.) that were included due to their large number of mutual connections with the representatives. In the case of the Republican set, there were more foundations and political activists than the Democratic set, possibly suggesting that the Republican representatives are more likely to have mutual links to these users. It is also interesting that no representatives of the opposite political party were discovered in the social circles, suggesting little direct overlap among members.

6.2. Blog Social Circles

One of the characteristics that has contributed to the success of the blogosphere is the fact that authors link to each other's posts. Dense connections between common blogs can define social circles within the blogosphere and because the entire set of blogs cannot be feasibly known, a local discovery method is required to discover these sets. To discover a social circle of blogs, we downloaded the latest 50 blog entries for each blog, and crawled the content for links. The links were then examined and if the resulting page contained a FeedURL in its metadata, it was considered a blog. A case study of blog social circles complements that of Twitter social circles nicely, because whereas on Twitter many graphs are densely connected and it is common for many users to follow those that follow them, a blog social circle defined by links to other blogs is much more sparse.

A prominent interest that exists within the blogosphere is that of "mommy-blogs," where mothers post about their experiences raising children and homemaking, and link to one another. To discover a social circle around a set of mommy-blogs, we selected the query set by choosing an arbitrary blog from the "Top Rated Mommy Blogs" at TopMommyBlogs.com [1], and crawled its neighbors to identify four more that had connections between them, and that by manual inspection appeared to be mothers talking about events, as opposed to an automated feed or coupon service.

[1] http://www.topmommyblogs.com/pages /top_rated_mommy_blogs.html

Using this initial set, and our algorithm with $\alpha = 1$ and $f = 5$, we identified a social circle of $max = 100$ blogs. A visual representation of this set is shown in Figure 11, and the first 25 blogs are listed in Table IV.

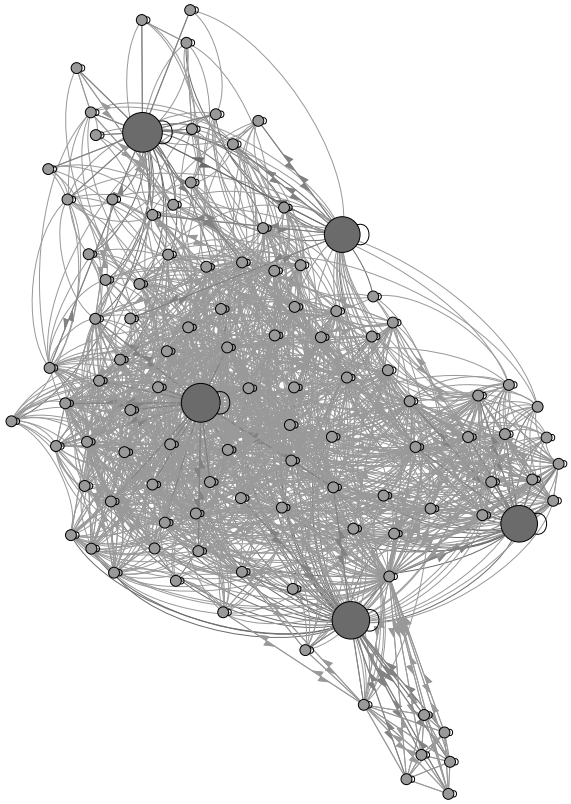

Fig. 11. The social circle of mommy-blogs. The larger nodes are the initial query set.

Each of the 100 blogs were considered mommy-blogs to some degree, in that they dealt with issues related to homemaking, children, and thriftiness. In addition, Figure 11 shows that the initial query set (shown as larger nodes) remain highly-connected and prominent in the resulting social circle, as opposed to being left on the fringe while a dense adjacent group is discovered (the Fringe Problem).

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have defined the local social circle discovery problem in directed graphs, and proposed a novel algorithm to discover such social circles around an initial query set, based on a degree-inspired quality function that quantifies the value of adding a node to the growing social circle. Our approach does not focus on boundaries and can therefore include appropriate nodes in a social circle regardless of their membership in other circles. In addition it stays focused around the original query set, as opposed to drifting into other parts of the graph, leaving the initial query nodes on the fringe of the final social circle. Further, our approach explicitly accounts for edge direction and avoids including celebrities or unknown nodes that do not have mutual

Table IV. The First 25 Members of the Mommy-blog Social Circle

Step	Blog URL
1.	momtobedby8.com
1.	stuckathomemom.com
1.	guideformoms.blogspot.com
1.	autumnandkids.com
1.	mamaluvsbooks.com
2.	thegiveawaygals.com
3.	blog.stay-a-stay-at-home-mom.com
4.	lifesabargain.net
5.	mewreview.com
6.	confessionsofamessymama.blogspot.com
7.	tinklemonkey.com
8.	swanksavings.com
9.	amedicsworld.com
10.	prmomambassador.com
11.	countingtoten.com
12.	earndollarspinoy.info
13.	to-sew-with-love.com
14.	funnypregnantlady.blogspot.com
15.	nikkicole22654.blogspot.com
16.	budgetearth.com
17.	justjennifer.net
18.	carolscrittercorner.com
19.	mommies-in-orbit.com
20.	alittlesimplicity.com
21.	momat40.com

interaction with the social circle. We show that our Social Circle Discovery algorithm performs well on artificial benchmark problems, large networks with pre-labeled topical communities, and through case studies in real-world networks. Our method is able to efficiently discover meaningful social circles even when the degree of the included nodes is extremely high.

There are several interesting extensions to our algorithm that could be pursued. While we have explicitly accounted for directed edges, we are still only handling unweighted graphs. It would be interesting to consider ways to incorporate weighted edges in the algorithm. One simple solution would be to replace the current indicator function $e(x, y)$ by a number-valued function corresponding to the weight of the edge. If the semantic associated with edge weights is that of a notion of strength of the relationship between the connected nodes, then the interaction between this extension and the existing discounted importance mechanism of our algorithm may result in the expected behavior. If not, further extensions may be needed to properly account for the intended meaning of the weights. Another area of interest, also related to the directed nature of the graphs, has to do with the relative value of incoming and outgoing edges. In the current implementation, both $WgtInDeg$ and $WgtOutDeg$ are treated equally in $\phi(n, C)$. There may be value, depending on the application, in weighing these quantity differently, perhaps using a parameter β to transform $\phi(n, C)$ into $\min(\beta WgtInDeg(n, C), (1 - \beta)WgtOutDeg(n, C))$. Further experiments are needed. As also noted, in some cases, determining the right query set may not be trivial. While we have examined some query selection strategies, further work remains to study the most effective query sets to uniquely identify the social circle of interest. While this may be domain-specific in some cases, we suspect that there are common properties that hold across domains that could be valuable. Finally, it would also be useful to examine automatic stopping criteria. In its present form, the $\phi(n, C)$ value could be monitored for significant decline to signify a good stopping point, but perhaps an ad-

ditional parameter could be added to specify the relative size or density desired for termination. These ideas require additional testing on different types of graphs with varied structures and properties.

REFERENCES

Y-Y. Ahn, J.P. Bagrow, and S. Lehman. 2010. Link communities reveal multiscale complexity in networks. *Nature* 466 (2010), 761–764.

I. Ajzen and M. Fishbein. 1980. *Understanding Attitudes and Predicting Social Behavior*. Prentice-Hall, Inc.

R. Andersen and K.J. Lang. 2006. Communities from Seed Sets. (2006).

J.P. Bagrow. 2008. Evaluating Local Community Methods in Networks. *Journal of Statistical Mechanics: Theory and Experiment* (2008), P05001.

J.P. Bagrow and E.M. Bollt. 2005. A local method for detecting communities. *Phys. Rev. E* 72, 4 (2005), 046108.

J. Baumes, M. Goldberg, and M. Magdon-Ismail. 2005. Efficient Identification of Overlapping Communities. In *Proceedings of the International IEEE International Conference on Intelligence and Security Informatics*. 27–36.

Jonathan W. Berry, Bruce Hendrickson, Randall A. LaViolette, and Cynthia A. Phillips. 2011. Tolerating the community detection resolution limit with edge weighting. *Phys. Rev. E* 83 (May 2011), 056119. Issue 5.

A. Capocci, V.D.P. Servedio, G. Caldarelli, and F. Colaiori. 2005. Detecting communities in large networks. *Physica A: Statistical Mechanics and its Applications* 352, 2-4 (2005), 669 – 676.

J. Chen, O. R. Zaïane, and R. Goebel. 2009. Detecting communities in large networks by iterative local expansion. In *Proceedings of the International Conference on Computational Aspects of Social Networks*. 105–112.

A. Clauset. 2005. Finding local community structure in networks. *Phys. Rev. E* 72, 2 (2005), 026132.

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

M. de Klepper, E. Sleebos, G. van de Bunt, and F. Agneessens. 2010. Similarity in friendship networks: Selection or influence? The effect of constraining contexts and non-visible individual attributes. *Social Networks* 32, 1 (2010), 82–90.

C. Faloutsos, K.S. McCurley, and A. Tomkins. 2004. Fast Discovery of Connection Subgraphs. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 118–127.

G.W. Flake, S. Lawrence, C.L. Giles, and F.M. Coetzee. 2002. Self-organization and identification of Web communities. *Computer* 35, 3 (mar 2002), 66 –70.

S. Fortunato. 2010. Community detection in graphs. *Physics Reports* 486, 3-5 (2010), 75–174.

S. Fortunato and M. Barthélemy. 2007. Resolution limit in community detection. *Proceedings of the National Academy of Sciences* 104, 1 (2007), 36–41.

A. Friggeri, G. Chelius, and E. Fleury. 2011. Egomunities, Exploring Socially Cohesive Person-based Communities. *CoRR (http://arxiv.org/abs/1102.2623)* abs/1102.2623 (2011).

M. Goldberg, S. Kelley, M. Magdon-Ismail, K. Mertsalov, and A. Wallace. 2010. Finding Overlapping Communities in Social Networks. In *Proceedings of the 2nd IEEE International Conference on Social Computing*. 104 –113.

S. Gregory. 2010. Finding Overlapping Communities in Networks by Label Propagation. *New Journal of Physics* 12, 10 (2010), 103018.

C. Kadushin. 1966. The Friends and Supporters of Psychotherapy: On Social Circles in Urban Life. *American Sociological Review* 31, 6 (1966), pp. 786–802. http://www.jstor.org/stable/2091658

C. Kadushin. 1968. Power, Influence and Social Circles: A New Methodology for Studying Opinion Makers. *American Sociological Review* 33, 5 (1968), pp. 685–699. http://www.jstor.org/stable/2092880

Y. Kim, S-W. Son, and H. Jeong. 2010. Finding communities in directed networks. *Phys. Rev. E* 81 (Jan 2010), 016103. Issue 1.

A. Lancichinetti and S. Fortunato. 2009. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E* 80, 1 (Jul 2009), 016118.

A. Lancichinetti, S. Fortunato, and J. Kertész. 2009. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics* 11, 3 (2009), 033015. http://stacks.iop.org/1367-2630/11/i=3/a=033015

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

T. Lappas, K. Liu, and E. Terzi. 2009. Finding a team of experts in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '09)*. ACM, 467–476.

C. Lee, F. Reid, A. McDaid, and N. Hurley. 2010. Detecting highly overlapping community structure by greedy clique expansion. In *The 4th SNA-KDD Workshop (SNA-KDD 2010)*. 33–42.

C. Lee, F. Reid, A. McDaid, and N. Hurley. 2011. Seeding for pervasively overlapping communities. *Phys. Rev. E* 83 (Jun 2011), 066107. Issue 6. DOI:http://dx.doi.org/10.1103/PhysRevE.83.066107

E. A. Leicht and M. E. J. Newman. 2008. Community Structure in Directed Networks. *Phys. Rev. Lett.* 100, 11 (Mar 2008), 118703.

F. Luo, J.Z. Wang, and E. Promislow. 2008. Exploring local community structures in large networks. *Web Intelligence and Agent Systems* 6, 4 (2008), 387–400.

J. McAuley and J. Leskovec. 2012. Learning to Discover Social Circles in Ego Networks. In *Advances in Neural Information Processing Systems 25*, P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger (Eds.). 548–556. http://books.nips.cc/papers/files/nips25/NIPS2012_0272.pdf

M. McPherson, L. Smith-Lovin, and J.M. Cook. 2001. Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology* 27 (2001), pp. 415–444. http://www.jstor.org/stable/2678628

Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. 2007. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement (IMC '07)*. ACM, New York, NY, USA, 29–42. DOI:http://dx.doi.org/10.1145/1298306.1298311

A. Mislove, B. Viswanath, K.P. Gummadi, and P. Druschel. 2010. You are who you know: inferring user profiles in online social networks. In *Proceedings of the 3rd ACM International Conference on Web Search and Data Mining (WSDM '10)*. ACM, New York, NY, USA, 251–260. DOI:http://dx.doi.org/10.1145/1718487.1718519

M.E.J. Newman. 2011. Communities, modules and large-scale structure in networks. *Nature Physics* 8, 1 (2011), 25–31.

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

N.P. Nguyen, T.N. Dinh, D.T. Nguyen, and M.T. Thai. 2011. Overlapping Community Structures and Their Detection on Social Networks. In *Proceedings of 3rd IEEE International Conference on Social Computing*. 35–40.

V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri. 2009. Extending the definition of modularity to directed graphs with overlapping communities. *Journal of Statistical Mechanics: Theory and Experiment* 2009, 03 (2009), P03024.

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

S. Papadopoulos, A. Skusa, A. Vakali, Y. Kompatsiaris, and N. Wagner. 2009. Bridge bounding: A local approach for efficient community discovery in complex networks. *Arxiv preprint arXiv:0902.0871* (2009).

P. Pons and M. Latapy. 2005. Computing Communities in Large Networks Using Random Walks. In *Proceedings of the 20th International Symposium on Computer and Information Sciences (LNCS 3733)*. 284–293.

H. Qin, T. Liu, and Y. Ma. 2012. Mining User's Real Social Circle in Microblog. In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. 348–352.

U.N. Raghavan, R. Albert, and S. Kumara. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* 76 (Sep 2007), 036106. Issue 3.

J. Reichardt and S. Bornholdt. 2006. Statistical mechanics of community detection. *Phys. Rev. E* 74 (Jul 2006), 016110. Issue 1.

M. Rosenberg. 1989. *Society and the adolescent self-image (rev)*. Wesleyan University Press.

M. Rosvall and C.T. Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences* 105, 4 (2008), 1118–1123.

M.E. Shaw. 1976. *Group dynamics: the psychology of small group behavior*. McGraw-Hill, New York.

S. Smyth and S. White. 2005. A spectral clustering approach to finding communities in graphs. In *Proceedings of the 5th SIAM International Conference on Data Mining*. 76–84.

M. Sozio and A. Gionis. 2010. The community-search problem and how to plan a successful cocktail party. In *Proceedings of 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 939–948.

A. Stanoev, D. Smilkov, and L. Kocarev. 2011. Identifying communities by influence dynamics in social networks. *Phys. Rev. E* 84 (Oct 2011), 046102. Issue 4.

T Ståhl, A Rütten, D Nutbeam, A Bauman, L Kannas, T Abel, G Lüschen, Diaz J.A Rodriquez, J Vinck, and J van der Zee. 2001. The importance of the social environment for physically active lifestyle — results from an international study. *Social Science & Medicine* 52, 1 (2001), 1–10.

H. Tong and C. Faloutsos. 2006. Center-Piece Subgraphs: Problem Definition and Fast Solutions. In *Proceedings of 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 404–413.

X. Wang, L. Tang, H. Gao, and H. Liu. 2010. Discovering Overlapping Groups in Social Media. In *Proceedings of the 10th IEEE International Conference on Data Mining*. 569–578.

S. Wasserman and K. Faust. 1994. *Social Network Analysis: Methods and Applications*. Cambridge University Press.

C. Wei. 2004. Formation of norms in a blog community. In *Into the Blogosphere: Rhetoric, Community, and Culture in Weblogs*, L. Gurak, S. Antonijevic, L. Johnson, and C. Ratliff (Eds.). University of Minnesota.

J. Yang and J. Leskovec. 2012a. Community-Affiliation Graph Model for Overlapping Network Community Detection. In *Proceedings of the IEEE International Conference on Data Mining*. 1170–1175.

J. Yang and J. Leskovec. 2012b. Defining and Evaluating Network Communities based on Ground-truth. In *Proceedings of the IEEE International Conference on Data Mining*. 745–754.

J. Yang and J. Leskovec. 2012c. Structure and Overlaps of Communities in Networks. In *Proceedings of the 6th SNA-KDD Workshop on Social Network Mining and Analysis*.

I. H. Yen and S. L. Syme. 1999. The social environment and health: a discussion of the epidemiologic literature. *Annual review of public health* 20, 1 (1999), 287–308.