Community Detection in Scale-free Networks: Approximation Algorithms for Maximizing Modularity

Thang N. Dinh and My T. Thai

Abstract—Many networks, indifferent of their function and scope, converge to a scale-free architecture in which the degree distribution approximately follows a power law. Meanwhile, many of those scale-free networks are found to be naturally divided into communities of densely connected nodes, known as community structure. Finding this community structure is a fundamental but challenging topic in network science. Since Newman's suggestion of using *modularity* as a measure to qualify the strength of community structure, many efficien methods that fin community structure based on maximizing modularity have been proposed. However, there is a lack of *approximation algorithms* that provide provable quality bounds for the problem. In this paper, we propose polynomial-time approximation algorithms for the modularity maximization problem together with their theoretical justification in the context of scale-free networks. We prove that the solutions of the proposed algorithms, even in the worst-case, are optimal up to a constant factor for scale-free networks with either bidirectional or unidirectional links. Even though our focus in this work is not on designing another empirically good algorithms to detect community structure, experiments on real-world networks suggest that the proposed algorithm is competitive with the state-of-the-art modularity maximization algorithm.

Index Terms—Network science, approximation algorithm, community structure, modularity, social networks;

1 Introduction

Many complex systems of interest such as the Internet, social, and biological networks are found to have the degree distributions approximately follow the power laws [8], [9], [15]. That is the fraction of nodes in the network having k connections to other nodes is proportional to $k^{-\gamma}$, where γ is a parameter whose value is typically in the range $2 < \gamma < 3$. Meanwhile, many of those scale-free networks are known to exhibit notable structural properties including the existence of community structure where nodes in the network are naturally clustered into tightly connected communities with only sparser connections between them [24]. Finding this community structure is a fundamental but challenging problem in the study of network systems and not yet satisfactorily solved, despite the huge effort of a large interdisciplinary community of scientists working on it over the past few years [23].

The ability to detect such communities can be of significant practical importance, providing insight into how network function and topology affect each other. For instance, communities within the World Wide Web may correspond to sets of web pages on related topics; communities within mobile networks may correspond to sets of friends or colleagues; communities in computer networks may correspond to users that are sharing files with peer-to-peer traffic, or collections of compromised computers controlled by remote hackers, e.g. botnets [42]. In the social network visualization perspective, the detection of community structure is extremely helpful since it only displays core groups of users and their mutual interactions, hence presents a more compact and understandable description of the network as a whole [34]. Detecting this special sub-structure also finds itself extremely useful in deriving social-based solutions for many network problems, such as forwarding and routing strategies in communication networks [17], [27], [33], Sybil defense [40], [41], worm containment on cellular networks [33], [43], and sensor programming [35].

Newman-Girvan's modularity that measures the "strength" of division of a network into modules (also called communities or clusters) [24] has rapidly become an essential element of many communities detection methods. Despite of some drawbacks [22] [25], modularity is by far the most used and best known quality function, particularly because of its success in many applications in social and biological networks [26]. Modularity is defined as the fraction of the edges that fall within the given communities minus the expected such fraction if edges were distributed at random. It can be either positive or negative with positive values indicating the possible presence of community structure. Thus, one can search for community structure precisely by looking for the divisions of a network that have positive, and preferably large, values of the modularity. This is the main motivation for numerous optimization methods that find communities in the network via maximizing modularity as surveyed in [23].

Unfortunately, maximizing modularity is an NP-hard problem [12] i.e. unless P = NP, there is no efficient algorithm to find optimal solutions to the problem, where we

• *Manuscript received 15 August 2012; revised 31 January 2013. A part of this paper was presented at the 3rd IEEE SocialCom conference, MIT, Boston, USA, 2011. This work is partially supported by NSF Career Award 0953284 and DTRA, Young Investigator Award, Basic Research Program HDTRA1-09-1-0061.*
T. Dinh and M. Thai are with the Department of Computer & Information Science & Engineering, University of Florida, Gainesville, FL, 32611. E-mail: {tdinh, mythai}@cise.ufl.edu.

follow the convention that an efficient algorithm is one that run in time bounded by a polynomial in its input size. Although existing methods are efficient enough to find sub-optimal solutions in a reasonably fast time, they do not theoretically guarantee the output quality, thus, they might perform very badly for a particular set of input instances.

There is a class of efficient algorithms for NP-hard problems that find provably near-optimal solutions for all input instances, called *approximation algorithms* [39]. A *ρ-approximation algorithm* for an optimization problem is a polynomial-time algorithm that for all instances of the problem produces a solution whose value is within a factor of ρ of the value of an optimal solution. Designing approximation algorithms often results in a deeper understanding of the problem's structure and provides a mathematically rigorous basis on which to study new algorithmic approaches. None of such algorithms is known for the modularity maximization problem on scale-free networks.

In this paper, we provide approximation algorithms to the modularity maximization problem in scale-free networks. The algorithms are optimal up to a constant factor when the network's power exponent $\gamma > 2$ and are optimal up to an $O(1/\log n)$ factor when $1 < \gamma \leq 2$. The algorithms are first presented in undirected networks and then generalized for directed networks to take into the account the edge directions. The algorithms encourage further exploration to design both theoretically and empirically justified methods for finding community structure in complex networks.

Related work. In contrary to the vast amount of work on maximizing modularity, the only known polynomial-time approach to find a good community structure with error bounds is due to G. Agarwal and D. Kempe [1] in which they rounded the fractional solution of a linear programming (LP). However, no approximation algorithms were provided.

In [38], the authors provided constant-factor approximation algorithms for identifying communities in dynamic social networks. The communities are identified in order to minimize switching group costs of individuals in the network. It is worth noting that the concept of communities considered in the paper is rather different with the usual notion of communities and clusters. Thus, the proposed approaches are inapplicable to most biological, social and communication networks.

Some clustering problems also yield algorithms with good performance guarantees [7], [29], where the performance guarantees are often related to the eigenvalues of the graph. It might be worth to differentiate between community detection and graph clustering problems. They all share the same objective of partitioning network nodes into groups; however, the number of clusters is predefined or given as part of the input in the graph clustering problems whereas the number of communities is typically unknown in community detection.

A comprehensive survey on community detection algorithms can be found in [23]. The survey includes both modularity-based algorithms and non-modularity approaches, for example, see [2], [32], [36], [37].

Organization. We present the preliminaries of the modularity maximization problem in Section 2. For simplicity, we first consider the problem in undirected scale-free networks with power-law exponents $\gamma > 2$, and a simple $O(1/\log n)$ approximation algorithm for networks with low power-law exponents, $\gamma \leq 2$. We generalize our results for directed networks in Section 4 and present experimental results on both small complex networks and large online social networks in Section 5. We conclude our paper in Section 6.

2 Maximizing Modularity

We consider a network represented as an undirected graph $G = (V, E)$ consisting of $n = |V|$ vertices and $m = |E|$ edges. The *adjacency matrix* of G is denoted by $\boldsymbol{A} = (A_{ij})$, where $A_{ij} = 1$ if i and j share an edge and $A_{ij} = 0$ otherwise. We also denote the degree of vertex i, the number of edges incident at i, by k_i.

We study the division of the vertices in V into a collection of disjoint subset of vertices $\mathcal{C} = \{C_1, C_2, \ldots, C_l\}$ that the union gives back V. Each subset $C_i \subseteq V$ is called a community and we wish to have more edges connecting vertices in the same communities than edges that connect vertices in different communities. The *modularity* [31] of $\mathcal{C}$, denoted by $Q(\mathcal{C})$, is defined as

$$Q(\mathcal{C}) = \frac{1}{2m} \sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) \delta_{ij}, \tag{1}$$

where $\delta_{ij} = 1$ if i and j are in the same community, and $\delta_{ij} = 0$ otherwise. The *modularity maximization problem* asks to find a division which maximizes the modularity.

The modularity can be equivalently defined as

$$Q(\mathcal{C}) = \sum_{t=1}^{l} (\frac{E_t}{m} - \frac{K_t^2}{4m^2}), \tag{2}$$

where E_t is the number of edges that both two ends are inside community C_t and K_t is the *volume* of C_t i.e. the total degree of vertices in C_t. It can be derived from the formula 2 that the modularity is upper bounded by one.

3 Constant Factor Approximation Algorithms for Scale-free Networks

We present a constant factor approximation algorithm for the modularity maximization problem in scale-free networks. Let Q_{opt} be the maximum of modularity values of all possible divisions of the network. Our algorithm finds in a polynomial-time a division with the modularity value at least ρQ_{opt} for some constant $0 < \rho < 1$. Here by convention, the approximation factor ρ is less than one for maximization problems and $\rho > 1$ for minimization problems and the closer the approximation factor to one, the better performance guarantee.

The class of networks considered in this section are scale-free networks with the power exponents $\gamma > 2$.

This class covers a wide range of scale-free networks of interest, since typically $2 < \gamma < 3$. For example, scientific collaboration networks has γ in the range $2.1 < \gamma < 2.45$ [9], Word Wide Web with γ for in-degree and out-degree of 2.1 and 2.45, respectively [5]; Internet at router and intra-domain level with $\gamma = 2.48$ [20] and so on.

We continue with the description of the algorithm in subsection 3.1. The approximation factor is later derived for different power-law network models, starting with a proof for networks with prescribed degree sequence in subsection 3.2 to a more general proof for various model in subsection 3.4. For networks with given degree sequence in subsection 3.4, we are able to obtain an explicit approximation factor in term of the power exponent γ.

3.1 Low-Degree Following (LDF) Algorithm

Basically, LDF decides for each vertex u, which neighbor to follow, and if u follows a neighbor v, the algorithm eventually assigns u and v to the same community. The algorithm follows three rules to assign each vertex one of the three labels *leader*, *member*, or *orbiter* as follows:

1. All *members* and *orbiters* have degree at most d_0, for some predefined parameter d_0.
2. There are only two types of following: a *member* follows a *leader* and an *orbiter* follows a *member*. This implies that *members* cannot follow each other and *orbiters* cannot directly follow *leaders*.
3. All neighbors of an *orbiter* must be *members*.

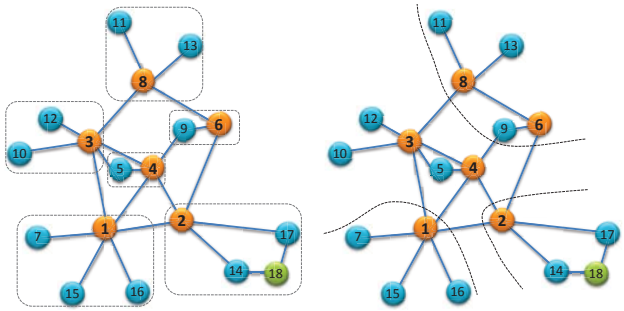
Algorithm 1. Low-degree Following Algorithm (Parameter $d_0 \in \mathbb{N}^+$)

1. $L := \emptyset, M := \emptyset, O := \emptyset, p_i = 0 \ \forall i = 1..n$
2. **for each** vertex $i \in V$ **do**
3. **if** $(k_i \leq d_0)$ & $(i \notin L \cup M)$ **then**
4. **if** $N(i) \setminus M \neq \emptyset$ **then**
5. Select a vertex $j \in N(i) \setminus M$
6. Let $M = M \cup \{i\}, L = L \cup \{j\}, p_i = j$
7. **else**
8. Select a vertex $t \in N(i)$
9. $O = O \cup \{i\}, p_i = t$
10. $\mathcal{L} = \emptyset$
11. **for each** vertex $i \in V \setminus (M \cup O)$ **do**
12. $C_i = \{i\} \cup \{j \in M \mid p_j = i\} \cup \{t \in O \mid p_{p_t} = i\}$
13. $\mathcal{L} = \mathcal{L} \cup \{C_i\}$
14. Perform post-optimization on $\mathcal{L}$ (section 3.1.2)
15. Return $\mathcal{L}$

The algorithm to construct a community structure of the network, called *Low-degree Following* (LDF) (Algorithm 1.), works as the follows. The algorithm uses three sets L, M, and O to store *leaders*, *members*, and *orbiters*, respectively, and an array p_i to store which neighbor vertex i follows. A vertex that does not belong to any of the sets L, M, or O is said to be unlabeled. Initially, L, M, and O are empty i.e. all vertices are unlabeled and $p_i = 0$ for all i.

At each step, the algorithm considers an unlabeled vertex i of degree at most d_0. If there is a neighbor $j \in N(i) \setminus M$ of i that is a *leader* or an unlabeled vertex, we add i to M and j to L, if necessary, and set i to follow j, $p_i = j$. Otherwise, all neighbors of i must be labeled with *member*, thus, we can add i to the set of orbiters O and set i to follow an arbitrary neighbor t.

Finally, two types of communities are formed. First, all the *members* that follow the same leader and all the *orbiters* that follow those *members* are assigned into the same community. Second, each unlabeled vertex forms a singleton community of size one. The union of all the communities is returned as the community structure $\mathcal{L}$.



(a) LDF algorithm (b) Community structure with the maximum modularity value

Fig. 1. (a) A community structure found by the LDF algorithm when $d_0 = 2$. Each rounded square represents a community in which the *leader* vertex is in orange; the *members* are in blue; and the *orbiters* are in light green (b) The optimal community structure found by solving the Integer Programming formulation in [1] with the mathematical software package CPLEX.

An illustration example for the algorithm is shown in Fig. 1a. We consider all vertices of degree at most $d_0 = 2$ in a non-decreasing order of degree. All vertices of degree one and vertices $5, 9, 14$, and 17 are labeled with *member*; vertices $1, 2, 3, 4, 6, 8$ are labeled with *leader* and there is only one *orbiter*, vertex 18 as all its neighbors are labeled with *member*. In comparison with the division with the maximum modularity, shown in the right, the LDF algorithm finds more (smaller) communities, which is due to the small value of d_0.

Note that the solution produced by LDF can be further optimized by local search procedures. For example, we can merge adjacent communities to form larger communities or refine the community structure by the vertex moving method [11] to reach the highest possible modularity value. However, designing another empirically good heuristic for the modularity optimization problem is not the focus of this paper.

The selection of d_0 is important to derive the approximation factor. First, d_0 is the upper bound on the degree of *members* and *orbiters*, hence, it positively correlates to the formed communities' volumes. Second, larger d_0 lessens the number of unlabeled vertices, thus, possibly increases the fraction of edges that both endpoints are in the same communities. Therefore, we shall select d_0 to be a sufficient large constant that is still relatively small to the network size, n when n tends to infinity.

Regardless of the choice of the constant d_0, the algorithm takes linear time which certainly satisfies the

polynomial-time requirement of an "efficient" algorithm. First, the labeling part (lines 1 to 9) is linear in complexity. Second, the community allocation part can be done in linear time provided that all *members* and *orbiters* keep track of their (only) leaders.

3.1.1 Automatic selection of d_0

Selecting parameter d_0 is an important part of LDF. For the analysis of the adaptive approximation ratio in next Section, it is sufficient to select d_0 as a large constant that relies only on γ. In an actual implementation of the algorithm, d_0 should be selected automatically to maximize modularity Q. This can be done by trying all possible values of d_0 from 1 to $n = |V|$, and selecting d_0 that maximizes Q. In addition, this can be done without increasing time complexity of LDF. First we sort nodes in a non-decreasing order of their degrees (for example by using counting sort which takes an $O(n)$ time). Then, if we set $d_0 = n$, the loop from lines 2 to 9 in Algo. 1 will eventually iterate through all possible values of d_0. Then d_0 is set to the degree value which gives maximum modularity.

Lemma 1: Automatic selection of the best d_0 can be done in $O(|V| + |E|)$.

3.1.2 Post-optimization

We can further optimize the LDF algorithm without changing its performance guarantee presented in the sequential subsections. First, we can derandomize the selection of neighbor inside LDF at lines 5 and 8 by selecting the neighbor that maximizes the local modularity gain. Second, each community can be abstracted into a single meta-node whose degree equals the total degree of nodes inside that community to obtain an abstract network [11], [17]. We then apply LDF recursively on top of the abstract network. Finally, local search [31] method is performed to increase the overall modularity.

3.2 Networks with Power-law Degree Sequences

We first analyze the performance of the LDF algorithm in networks with power-law degree sequences. Those networks have been used in studying different aspects of the scale-free networks [3], [4], [21].

In our network, the number of vertices of degree k is $\lfloor \frac{e^\alpha}{k^\gamma} \rfloor$ where e^α is the normalization factor as in the $P(\alpha, \gamma)$ model [3]. For convenience, we shall refer to such a network as a $P(\alpha, \gamma)$ network. While previous works [3], [4], [21] focus on power-law networks in which vertices are connected at *random*, we make no such assumption in our network model.

We can deduce that the maximum degree in a $P(\alpha, \gamma)$ network is $e^{\frac{\alpha}{\gamma}}$ (since for $k > e^{\frac{\alpha}{\gamma}}$, the number of edges will be less than 1). The number of vertices / edges are

$$n = \sum_{k=1}^{e^{\frac{\alpha}{\gamma}}} \frac{e^\alpha}{k^\gamma} \approx \begin{cases} \zeta(\gamma)e^\alpha & \text{if } \gamma > 1 \\ \alpha e^\alpha & \text{if } \gamma = 1 \\ \frac{e^{\frac{\alpha}{\gamma}}}{1-\gamma} & \text{if } \gamma < 1 \end{cases},$$

$$m = \frac{1}{2}\sum_{k=1}^{e^{\frac{\alpha}{\gamma}}} k\frac{e^\alpha}{k^\gamma} \approx \begin{cases} \frac{1}{2}\zeta(\gamma-1)e^\alpha & \text{if } \gamma > 2 \\ \frac{1}{4}\alpha e^\alpha & \text{if } \gamma = 2 \\ \frac{1}{2}\frac{e^{\frac{2\alpha}{\gamma}}}{2-\gamma} & \text{if } \gamma < 2 \end{cases} \quad (3)$$

where $\zeta(\gamma) = \sum_{i=1}^{\infty} \frac{1}{i^\gamma}$ is the Riemann Zeta function [3] which converge absolutely for $\gamma > 1$ and diverges for all $\gamma \leq 1$. Without affecting the conclusion, we will simply use real number instead of rounding down to integers. The error terms can be easily bounded and are sufficiently small in our proofs.

While the scale of the network depends on α, the parameter γ decides the connection pattern and many other important characterizations of the network. For instance, the larger γ, the sparser and the more "scale-free" the network is. Hence, the parameter γ is often regarded as the characteristic constant for scale-free networks. In term of modularity, the following theorem states that the higher power-law exponent γ implies the existing of community structure with higher modularity value and the better approximation factor.

Theorem 1: For scale-free networks with $\gamma > 2$, the modularity of the community structure $\mathcal{L}$, found by the *Low-degree Following* (LDF) algorithm will be at least $\frac{\zeta(\gamma)}{\zeta(\gamma-1)} - \epsilon$; that is the approximation factor of LDF will be at least $\frac{\zeta(\gamma)}{\zeta(\gamma-1)} - \epsilon$, where $\epsilon > 0$ is an arbitrary small constant.

Proof: We shall bound the modularity of $\mathcal{L}$ using the alternative definition of modularity in Eq. 2. Specifically, we give a lower-bound for the number of edges with both ends inside the same community, denoted by $E(\mathcal{L})$ and an upper bound for the volume of each community in $\mathcal{L}$.

We first bound $E(\mathcal{L})$. Since, the edges between *orbiters* and *members* and the vertices that they follow have both ends inside the same community, we have $E(\mathcal{L}) \geq |M \cup O|$. Furthermore, all vertices with degree at most d_0 are labeled vertices and at least half of them are labeled with *member* or *orbiter*, from Eq. 3 we have

$$E(\mathcal{L}) \geq |M \cup O| \geq \frac{1}{2}e^\alpha \sum_{i=1}^{d_0} i^{-\gamma}.$$

For an arbitrary small constant $\epsilon > 0$, the convergence of $\sum_{i=1}^{\infty} i^{-\gamma}$ when $\gamma > 1$, ensures that there exists a constant d_0 that depends only on ϵ such that

$$E(\mathcal{L}) \geq \frac{1}{2}e^\alpha \sum_{i=1}^{d_0} i^{-\gamma} \geq \frac{1}{2}e^\alpha(\zeta(\gamma) - \epsilon). \quad (4)$$

We now give an upper-bound for the volume of a community based on the degree of its *leader* vertex. Let vertex i be a *leader* of some community. Vertex i is

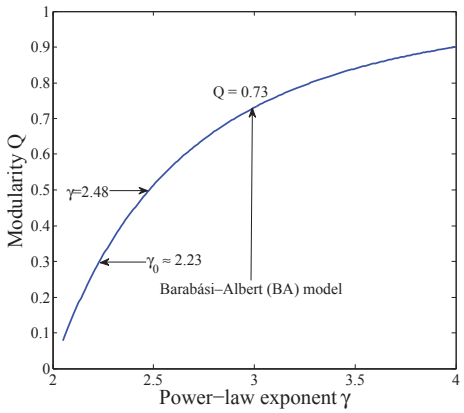


Fig. 2. Modularity as a function of power-law exponent γ

followed by at most k_i *members* of degree at most d_0, and each *member* is followed by at most d_0 *orbiters* of degree d_0 or less. The volume, the total degree of vertices in the community, is, hence, bounded by

$$k_i + k_i d_0 + k_i d_0^2 = k_i(d_0^2 + d_0 + 1) < 2k_i d_0^2.$$

Therefore, we have:

$$\begin{aligned} Q(\mathcal{C}) &\geq \frac{e^{\alpha}\left(\zeta(\gamma) - \epsilon\right)}{2m} - \sum_{i \in L} \frac{\left(2k_i d_0^2\right)^2}{4m^2} \\ &\geq \frac{e^{\alpha}\left(\zeta(\gamma) - \epsilon\right)}{2m} - \sum_{i=1}^{n} \frac{4d_0^4 k_i^2}{4m^2} \\ &= \frac{n - \epsilon e^{\alpha}}{2m} - 8d_0^4 D, \end{aligned} \tag{5}$$

where

$$D = \sum_{i=1}^{n} \frac{k_i^2}{8m^2} = \sum_{k=1}^{e^{\frac{\alpha}{\gamma}}} \frac{e^{\alpha}}{k^{\gamma}} \frac{k^2}{8m^2} = \frac{e^{\alpha}}{8m^2} \sum_{k=1}^{e^{\frac{\alpha}{\gamma}}} k^{2-\gamma}. \tag{6}$$

Since $\gamma > 2$, we have $k^{2-\gamma} < 1$, from equation (6) for sufficient large n we have

$$\begin{aligned} Q(\mathcal{C}) &\geq \frac{\zeta(\gamma) - \epsilon}{\zeta(\gamma - 1)} - 8d_0^4 \frac{e^{\alpha}}{8m^2} e^{\frac{\alpha}{\gamma}} \\ &\geq \frac{\zeta(\gamma) - \epsilon}{\zeta(\gamma - 1)} - \frac{4d_0^4}{\zeta(\gamma-1)^2 e^{\alpha(1-1/\gamma)}} \\ &\geq \frac{\zeta(\gamma) - \epsilon}{\zeta(\gamma - 1)} - \frac{4d_0^4}{n^{(1-1/\gamma)}} \geq \frac{\zeta(\gamma)}{\zeta(\gamma - 1)} - \epsilon. \end{aligned} \tag{7}$$

From 2, we have $Q_{\text{opt}} < 1$. Thus LDF is an $\left(\frac{\zeta(\gamma)}{\zeta(\gamma-1)} - \epsilon\right)$-approximation algorithm for the modularity maximization problem. □

For scale-free networks with $\gamma > \gamma_0 \approx 2.23$, by Theorem 1, the modularity value is at least 0.3, see Fig. 2, even when the nodes are randomly connected. Here, γ_0 is found by solving the equation $\zeta(\gamma) - 0.3\zeta(\gamma - 1) = 0$. Thus even when the network has no natural community structure, it still has a partition with a high modularity. This provides one more evidence against the statement of Newman and Girvan [24] that modularity values between 0.3 and 0.7 indicate strong community structure and higher values are rare.

In addition, higher λ implies higher modularity values, for example, large scale-free networks with $\gamma = 2.48$, e.g. the Internet at router and intra-domain level, will have community structure with the modularity at least $\frac{\zeta(2.48)}{\zeta(1.48)} \approx 0.5$ that means LDF is an $\frac{1}{2}$-approximation algorithm in that case. Another special case of scale-free networks is the BA model [8] in which the degree distribution can be approximated with a power-law of the form $P(k) \approx k^{-3}$. This corresponds to a theoretical modularity value of 0.73. However, due to the degree fluctuation the actual modularity values in BA's networks might be lightly smaller than the theoretical value.

Our result gives a theoretical explanation to why community structure with high modularity were found in scale-free networks [23], [24], [31] and prove with rigorous arguments that modularity should not be used alone as a quantifier for the goodness of the detected community structure.

3.3 Networks with Low Power-law Exponent

For power-law networks with $\gamma < 2$, the network has much fewer low degree vertices. As a result, LDF does not produce enough edges with both endpoints inside a same community to provide any performance guarantee. However, the modularity maximization problem can be approximated within a factor $O(1/\log n)$ using the following bisection algorithm.

Algorithm 2. Modularity Bisection

1. Compute the modularity-matrix B with $B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$.
2. Obtain the matrix $\boldsymbol{B}_0$ by replacing the main diagonal of $\boldsymbol{B}$ with zeros.
3. Find vector $x \in \{-1, 1\}^n$ that approximates the quadratic programming $\max_{x \in \{-1,1\}^n} x^T \boldsymbol{B_0} x$ using the $O(\log n)$ approximation algorithm in [14].
4. Return the division of the networks implied by x as the community structure.

The approximation algorithm for low power-law exponent networks, called *Modularity Bisection*, is presented in Algorithm 2. The algorithm first computes the modularity matrix $\boldsymbol{B}$, and obtains a matrix $\boldsymbol{B}_0$ from $\boldsymbol{B}$ by nullifying all main diagonal entries to zeros. The problem is then transformed into a quadratic programming problem and the approximation algorithm in [14] is applied to find a vector $x \in \{-1, 1\}^n$ that approximates the solution of the maximizing problem $x^T B_0 x$. The final division of the network into two communities is then derived from x by putting all vertices i with $x_i = -1$ to one part, and all vertices j with $x_j = 1$ to the other.

Alg. 2 is proposed independently in [19] and [16] for two different graph classes. While Bhaskar et. al. [16] analyze the algorithm on d-regular graph with maximum degree at most $\frac{n}{2 \ln n}$, Dinh et. al. [19] provide the following performance guarantee for power-law networks: *For scale-free networks with $1 < \gamma \leq 2$, the modularity maximization problem can be approximated within a factor $O(1/\log n)$.*

We present a detailed proof for the result, extending the proof sketch in [19]. The two key steps in our proof are as follows

- The division of the network into two communities can yield modularity values at least half of the maximum modularity value, as shown in Lemma 3
- The difference $D = x^{\mathrm{T}}(\boldsymbol{B} - \boldsymbol{B}_0)\,x$ is sufficiently small in comparison to the maximum modularity, as shown in Lemma 4.

The approximation ratio for the Modularity Bisection algorithm is finally derived in the Theorem 2.

In the first step, for a division of the network into two communities, define a column vector x having element $x_i = 1$ if vertex i belongs to the first community and $x_i = -1$, otherwise. We can write the modularity for the division into two communities as

$$Q = \frac{1}{4m}\sum_{i,j} B_{ij}(x_i x_j + 1) = \frac{1}{4m}\sum_{ij} B_{ij}x_i x_j = \frac{1}{4m}x^{\mathrm{T}}\boldsymbol{B}x$$

Hence, the division into two communities is a special case of the maximizing quadratic program problem that finds a vector $x \in \{-1, 1\}^n$ to maximize $x^{\mathrm{T}}\boldsymbol{B}x$. The following results was due to M. Charikar et al. [14].

Lemma 2: [14] Given an arbitrary matrix $\boldsymbol{A}$, whose diagonal elements are nonnegative, the problem of finding $x \in \{-1, 1\}^n$ such that $x^{\mathrm{T}}\boldsymbol{A}x$ is maximized can be approximated within $O(1/\log n)$.

Note that we have to indirectly maximize $\boldsymbol{x}^T B_0 x$, since $\boldsymbol{B}$ does not meet the requirement to apply the result in Lemma 2 as the ith entry of the modularity matrix $\boldsymbol{B}$ is $-\frac{k_i^2}{4m^2} < 0$. Thus, we can only derive the approximation factor when the difference $x^{\mathrm{T}}(\boldsymbol{B} - \boldsymbol{B}_0)x$ is relatively small to the optimal value of $x^{\mathrm{T}}\boldsymbol{B}x$.

Lemma 3: [19] Let Q_k be the maximum modularity obtained by a division of the network into *at most* k communities and $Q_{\mathrm{opt}} = Q_n$, the maximum modularity over all possible divisions. We have

$$Q_k \geq (1 - \frac{1}{k})Q_{\mathrm{opt}}$$

Lemma 3 implies that an approximation algorithm with a factor ρ for maximizing Q_2 will also be an approximation with a factor 2ρ to the modularity maximization problem.

Lemma 4: For scale-free networks with $1 < \gamma \leq 2$, we have $D = o\left(\frac{Q_{\mathrm{opt}}}{\log n}\right)$, where $D = \sum_{i=1}^{n} \frac{k_i^2}{8m^2}$.

Proof: Let $\mathcal{L}$ be the community structure obtained by the LDF algorithm. Since $Q(\mathcal{L}) \leq Q_{\mathrm{opt}}$, it is sufficient to prove that $D = o\left(\frac{Q(\mathcal{L})}{\log n}\right)$.

By Eq. (5) and (6), we have

$$Q(\mathcal{L}) \geq \frac{e^\alpha}{2m} - 8d_0^4 D$$

and

$$D = \sum_{i=1}^{n}\frac{k_i^2}{8m^2} = \sum_{k=1}^{e^{\frac{\alpha}{\gamma}}}\frac{e^\alpha}{k^\gamma}\frac{k^2}{8m^2} = \frac{e^\alpha}{8m^2}\sum_{k=1}^{e^{\frac{\alpha}{\gamma}}} k^{2-\gamma} \qquad (8)$$

For the cases $\gamma = 2$ and $\gamma < 2$, we prove that $\lim_{\alpha\to\infty}\frac{e^\alpha/2m}{D\log n} = \infty$. It follows that, $D = o\left(\frac{Q_{\mathrm{opt}}}{\log n}\right)$.

Case $\gamma = 2$: We have $\log n < 2\alpha$. Hence,

$$D\log n \leq \frac{2e^\alpha}{\alpha^2 e^{2\alpha}}\left(\sum_{k=1}^{e^{\frac{\alpha}{\gamma}}} 1\right)2\alpha = \frac{4e^{\alpha/\gamma}}{\alpha e^\alpha}$$

Thus,

$$\lim_{\alpha\to\infty}\frac{e^\alpha/2m}{D\log n} \geq \lim_{\alpha\to\infty}\frac{e^\alpha}{2e^{\alpha/\gamma}} = \infty$$

Case $2 > \gamma > 1$:

$$\begin{aligned} D\log n \leq & \frac{e^\alpha}{8m^2}e^{\frac{\alpha}{\gamma}(3-\gamma)}\sum_{k=1}^{e^{\frac{\alpha}{\gamma}}}\left(\frac{k}{e^{\frac{\alpha}{\gamma}}}\right)^{2-\gamma}\frac{1}{e^{\frac{\alpha}{\gamma}}}2\alpha \\ \leq & \frac{2\alpha e^\alpha}{\frac{2}{(2-\gamma)^2}e^{\frac{4\alpha}{\gamma}}}e^{\frac{\alpha}{\gamma}(3-\gamma)}\int_0^1 k^{2-\gamma}\mathrm{d}k \leq \frac{(2-\gamma)^2}{e^{\frac{\alpha}{\gamma}}}\frac{\alpha}{3-\gamma} \end{aligned}$$

Therefore,

$$\begin{aligned} \lim_{\alpha\to\infty}\frac{e^\alpha/2m}{D\log n} &\geq \lim_{\alpha\to\infty}\frac{e^\alpha}{2\frac{e^{2\alpha/\gamma}}{2-\gamma}}\frac{(3-\gamma)e^{\alpha/\gamma}}{\alpha(2-\gamma)^2} \\ &\geq \lim_{\alpha\to\infty}\frac{3-\gamma}{\alpha(2-\gamma)}e^{\alpha(1-\gamma^{-1})} = \infty. \end{aligned}$$

□

We are ready to show the approximation ratio for the Modularity Bisection algorithm.

Theorem 2: For scale-free networks with $1 < \gamma \leq 2$, the modularity maximization problem can be approximated within a factor $O(1/\log n)$.

Proof: Consider the case $1 < \gamma \leq 2$. From Lemma 3 with $k = 2$, we have $\frac{1}{2}Q_{\mathrm{opt}} \leq Q_2 \leq Q_{\mathrm{opt}}$. Hence, it is sufficient to approximate Q_2 within a factor of $O(1/\log n)$.

We have

$$\begin{aligned} Q_2 &= \frac{1}{4m}\max_{x\in\{-1,1\}^n} x^T\boldsymbol{B}x \\ &= \frac{1}{4m}\max_{x\in\{-1,1\}^n} x^T\boldsymbol{B_0}x - \sum_{i=1}^{n}\frac{k_i^2}{8m^2}, \end{aligned} \qquad (9)$$

where $\boldsymbol{B_0}$ is obtained by replacing the diagonal of $\boldsymbol{B}$ with zeros.

The second term in Equation (9) is $D = \sum_{i=1}^{n}\frac{k_i^2}{8m^2}$ as in the proof of Lemma 4. Denote

$$\mathrm{OPT}_0 = \max_{x\in\{-1,1\}^n} x^T\boldsymbol{B_0}x = Q_2 + D$$

We can approximate OPT_0 within a factor of $O(\log n)$ by the algorithm in [14]. That is we can find a division of the network into two communities with the modularity at least

$$\begin{aligned} &\frac{c}{\log n}\mathrm{OPT}_0 - D = \frac{c}{\log n}(Q_2 + D) - D \\ \geq &\frac{c}{\log n}Q_2 - D \geq \frac{c}{2\log n}Q_{\mathrm{opt}} - D \\ \geq &\frac{Q_{\mathrm{opt}}}{\log n}(c - o(1)) = O\left(1/\log n\right)Q_{\mathrm{opt}} \end{aligned}$$

The last step holds due to Lemma 4. □

We leave the approximability of the maximizing modularity when $\gamma < 1$ as an open problem. For $\gamma \leq 1$, Eq. 3 gives us $m = \Omega(n^2)$ i.e. the network has such a high edge density that even deciding whether there is a division of the network into two communities with significant modularity is hard [12]. No scale-free networks with such a low exponent has been encountered in the literature.

3.4 General Power-law Networks

We extend our analysis to a wider class of scale-free networks in which the degree sequences might slightly deviates from the power-law distribution. Specifically, we consider the networks that satisfy the following two common properties of real-world complex systems and theoretical scale-free models [3], [10], [13], [21]: 1) the network is sparse i.e. $m \leq cn$ for some constant $c > 0$, and 2) the network does not have "super" giant hubs, vertices of degree $\Omega(m)$. For this class, we prove in the following theorem that the network will still have a community structure with a significant modularity value, hence, the modularity maximization problem can still be approximated within a constant factor.

Theorem 3: For a sufficient large network satisfying $m \leq cn$ for some constant c and the maximum degree $\Delta = o(m)$, the LDF algorithm returns a community structure with a modularity value at least $\frac{2c-1}{4c^2}$. Thus, LDF is an $\frac{1}{2(c+1)}$-approximation algorithm for the modularity maximization problem.

The key idea is to select $d_0 = \lfloor 2c \rfloor$. The rest of the proof follows that of Theorem 1. Note that although we obtain the constant ratio approximation, the ratio is not explicitly represented in terms of the power-law exponent γ as in the case for the networks with fixed degree sequences.

4 Maximizing Modularity in Directed Networks

Many complex networks such as the Internet, the World Wide Web, Cellular networks and social networks such as Twitter are directed networks. The directions edges provide more insightful information about relationship among entities the network, but as well, exhibits multiple difficulties for analyzing the network structure. Simply ignoring edge directions and apply methods developed for community detection in undirected networks might discard potentially useful information in the edge directions and leads to unnatural divisions of the network into communities [30]. To benefit from the edge direction, the modularity measure has been generalized to take into account edge directions [30].

This section generalize our algorithms in the previous sections to work with directed networks. We show that with little modifications the proposed approximation algorithms can provide the same level of performance guarantees for this extension of the modularity maximization problem. For a given directed network, denote the in-degree of a vertex i (the number of incoming edges) and the out-degree of i (the number of outgoing edges) by k_i^{in} and k_i^{out}, respectively. Also, we define by $N^-(i) = \{j \mid (j,i) \in E\}$ the set of *incoming neighbors* and $N^+(i) = \{j \mid (i,j) \in E\}$, the set of *outgoing neighbors*. By definition, $|N^-(i)| = k_i^{\text{in}}$ and $|N^+(i)| = k_i^{\text{out}}$.

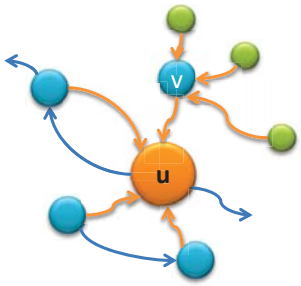


Fig. 3. A community formed by the LDF algorithm for directed networks. The *leader* is colored in orange, the *members* are in blue, and the orbiters are in light green

The probability of an edge from vertex i to vertex j will be $\frac{k_i^{\text{out}} k_j^{\text{in}}}{m}$, where k_i^{out} and k_j^{in} are the out- and in-degrees of the vertices. Thus, the directed equivalent formulation of Eq. 1 is

$$Q(\mathcal{C}) = \frac{1}{m} \sum_{i,j} \left(A_{ij} - \frac{k_i^{\text{out}} k_j^{\text{in}}}{m} \right) \delta_{ij} \qquad (10)$$

where A_{ij} is defined to be 1 if there is an directed edge from i to j and zero otherwise.

The approximation algorithms in Section 3.3 for undirected networks takes advantages of the dominance of the low degree vertices. This approach, however, cannot be applied for the directed networks, unless there are many vertices with both small in-degrees and out-degrees, which, in general, is not true due to the asymmetry of the in-degree and the out-degree. To make the matter worse, the in-degrees and the out-degrees often have different power-law distributions, for example the Web network [5], or only the out-degrees but not the in-degrees, follows a power-law distribution [28].

Instead of requiring the strong assumption that both in-degrees and out-degrees follow power laws, we provide approximation ratio for our algorithm even when only out-degrees (or only in-degree) follow a power-law. Without loss of generality, we assume that the out-degrees follow a power-law distribution $P(\alpha^{\text{out}}, \gamma^{\text{out}})$ with the power-law exponent $\gamma^{\text{out}} > 2$ while the in-degree can follow any distribution as long as there is no major giant hubs with out-degree of order $\Omega(n)$. For example if the in-degrees also follow a power law in which $\gamma^{\text{in}} > 1$, then the maximum degree is automatically bounded by $n^{1/\gamma^{\text{in}}} = o(n)$.

Algorithm 3. Directed LDF Algorithm
(Parameter $d_0, d_c \in \mathbb{N}^+$)
1. $L := \emptyset, M := \emptyset, O := \emptyset, p_i = 0\ \forall i = 1..n$
2. **for each** vertex $i \in V$ **do**
3. **if** $(1 \le k_i^{\text{out}} \le d_0)\ \&\ (i \notin L \cup M)$ **then**
4. **if** $N^+(i) \setminus M \neq \emptyset$ **then**
5. Select a vertex $j \in N^+(i) \setminus M$
6. Let $M = M \cup \{i\}, L = L \cup \{j\}, p_i = j$
7. **else**
8. Select a vertex $t \in N^+(i)$
9. $O = O \cup \{i\}, p_i = t$
10. **for each** $j \in M$ that $|t : p_t = j| > d_c$ **do**
11. $p_j = 0, L = L \cup \{j\}$, $M = M \setminus \{j\}$
12. **for each** t that $p_t = j$ **do**
13. $M = M \cup \{t\}$, $O = O \setminus \{t\}$
14. $\mathcal{L} = \emptyset$
15. **for each** vertex $i \in V \setminus (M \cup O)$ **do**
16. $C_i = \{i\} \cup \{j \in M \mid p_j = i\} \cup \{t \in O \mid p_{p_t} = i\}$
17. $\mathcal{L} = \mathcal{L} \cup \{C_i\}$
18. Return $\mathcal{L}$

The algorithm to approximate the maximum modularity, called Directed LDF or DLDF, is, fundamentally, similar to the LDF algorithm (see Algorithm 3). The major difference is that the DLDF algorithm has a pruning phase, lines 10 to 13, in which we cut-off oversize communities into smaller ones. In the pruning phase, we first identify all *members* that were followed by more than d_c *orbiters*, where d_c is a constant. Then we cut-off those oversize *members* from their *leaders*, and 'promote' those members to *leader* (line 11), and we also, 'promote' the *orbiters* that follow those *members* to *members* (lines 12 and 13).

The final communities are constructed in the same way as in the LDF algorithm. Let $\mathcal{F} = V \setminus (M \cup O)$. Each vertex $u \in \mathcal{F}$ associates with a community, denoted by $C_u \in \mathcal{L}$. Let K_u^{in} and K_u^{out} denote the total in-degree and the total out-degrees of vertices in C_u. The modularity of the division can be calculated, similarly to Eq. 2, as

$$Q_{\mathcal{F}} = \sum_{u \in \mathcal{F}} \left(\frac{E_u}{2m} - \frac{K_u^{\text{in}} K_u^{\text{out}}}{4m^2} \right), \tag{11}$$

where E_u is the number of directed edges with both ends inside C_u.

We first give a lower bound for $\sum_{u \in \mathcal{F}} E_u$ by the total number of the orbiters and the members $|M \cup O|$, noting that if a vertex i (*orbiter* or *member*) follows a vertex j, then the edge (i, j) has both ends inside the same community. Right **before** the pruning phase, all vertices of out-degree at most d_0 have been labeled. Thus,

$$|M| + |O| + |L| \ge e^{\alpha} \sum_{i=1}^{d_0} i^{-\gamma^{\text{out}}}$$

In addition, we have $|M| \ge |L|$, therefore,

$$|M| + |O| \ge 1/2(|M| + |O| + |L|) = 1/2 e^{\alpha} \sum_{i=1}^{d_0} i^{-\gamma^{\text{out}}}.$$

During the pruning phase, each vertex $i \in M$ is 'promoted' to *leader*, if and only if it has at least d_c following *orbiters*. Hence, the number of promoted *members* is at most $\frac{1}{d_c+1}(|O| + |M|)$. Therefore, **after** the pruning phase, the total number of *orbiters* and *members* decreases at most a fraction of $\frac{1}{d_c+1}$. Thus,

$$\sum_{u \in \mathcal{F}} E_u \ge \frac{d_c}{2(d_c+1)} e^{\alpha} \sum_{i=1}^{d_0} i^{-\gamma^{\text{out}}}.$$

Since, $\sum_{i=1}^{\infty} i^{-\gamma^{\text{out}}}$ converges to $\zeta(\gamma^{\text{out}})$ for $\gamma^{\text{out}} > 2$, for a given $\epsilon > 0$ there exist positive constants d_0 and d_c so that

$$\sum_{u \in \mathcal{F}} E_u \ge 1/2 e^{\alpha} \left(\zeta(\gamma^{\text{out}}) - \epsilon/2 \right). \tag{12}$$

The rest is to prove that $\sum_{u \in \mathcal{F}} \frac{K_u^{\text{in}} K_u^{\text{out}}}{4m^2}$, the second term in Eq. 11, is only of $o(1)$. Recall that all *members* and *orbiters* have out-degree at most d_0, and each member is followed by less than d_c *orbiters* due to the pruning phase. Hence,

$$\begin{aligned} K_u^{\text{out}} &\le \underbrace{k_u^{\text{out}}}_{\text{leader}} + \underbrace{k_u^{\text{in}} d_0}_{\text{members}} + \underbrace{k_u^{\text{in}} d_0 d_c}_{\text{followers}} = k_u^{\text{out}} + k_u^{\text{in}} d_0 (1 + d_c) \\ &\le \tau_0 \Delta \end{aligned} \tag{13}$$

where $\tau_0 = 1 + d_0(1 + d_c)$ and $\Delta = \max_{i=1..n}\{k_i^{\text{in}}, k_i^{\text{out}}\}$, the maximum degree.

Apply inequalities (12) and (13) to Eq. 11, we have

$$\begin{aligned} Q_{\mathcal{F}} &\ge \frac{e^{\alpha}(\zeta(\gamma^{\text{out}}) - \epsilon/2)}{4m} - \sum_{u \in \mathcal{F}} \frac{\tau_0 \Delta K_u^{\text{in}}}{4m^2} \\ &\ge \frac{\zeta(\gamma^{\text{out}}) - \epsilon/2}{2\zeta(\gamma^{\text{out}})} - \frac{\tau_0 \Delta}{2m} \frac{\sum_{u \in \mathcal{F}} K_u^{\text{in}}}{2m} \\ &\ge \frac{\zeta(\gamma^{\text{out}}) - \epsilon/2}{2\zeta(\gamma^{\text{out}})} - o(1) \ge \frac{\zeta(\gamma^{\text{out}})}{2\zeta(\gamma^{\text{out}})} - \epsilon \end{aligned} \tag{14}$$

The above inequalities hold because of $\sum_{u \in \mathcal{F}} K_u^{\text{out}} \le \sum_{i=1..n} k_i^{\text{out}} = m$ and the fact that $\Delta = o(m)$ (since the maximum out-degree is $e^{\alpha^{\text{out}}/\gamma^{\text{out}}} = o(m)$, and the maximum in-degree is $o(m)$ by our assumption.)

$$\frac{\sum_{u \in \mathcal{F}} e_u}{2m} \ge \frac{e^{\alpha^{\text{out}}} \sum_{i=1}^{d_0} i^{-\gamma^{\text{out}}}}{2\zeta(\gamma^{\text{out}} - 1) e^{\alpha^{\text{out}}}} \ge \frac{d_c}{d_c + 1} \frac{\zeta(\gamma^{\text{out}}) - \epsilon}{2\zeta(\gamma^{\text{out}} - 1)}$$

Hence, the communities formed by the DLDF algorithm has a non-trivial constant modularity. This also implies that the modularity maximization problem can be approximated within a factor $\frac{\zeta(\gamma^{\text{out}})}{2\zeta(\gamma^{\text{out}}-1)} - \epsilon$.

Apparently, the directed LDF algorithm and the analysis is also applicable for the case that only the in-degrees follows a power-law distribution. This can be done simply by replacing 'out-degree' with 'in-degree' and vice versa.

5 Computational Experiments

In this section, we evaluate the performance of LDF on both small and large real-world complex networks.

TABLE 1
Network sizes

Problem ID	Name	Nodes n	Edges m
1	Zachary's karate club	34	78
2	Dolphin's social network	62	159
3	Les Miserables	77	254
4	Books about US politics	105	441
5	American College Football	115	613
6	Electronic Circuit (s838)	512	819

TABLE 2
The modularity obtained by published methods GN [24], Blondel [11], LDF, and the optimal modularity values OPT [6].

ID	n	GN	Blondel	LDF	OPT
1	34	0.401	**0.4198**	**0.4198**	0.4198
2	62	0.520	**0.5277**	0.5179	0.5285
3	77	0.540	**0.5600**	**0.5600**	0.5600
4	105	-	**0.5270**	0.5257	0.5272
5	115	0.601	**0.6046**	0.6028	0.6046
6	512	-	0.7969	**0.8159**	0.8194

5.1 Small Complex Networks

We first compare LDF with other modularity maximization algorithms on several standard test cases for community structure identification, consisting of real-world networks. The datasets names together with their sizes are are listed in Table 1. The compared algorithms include the Girvan-Newman algorithm [24]; Blondel's algorithm (aka Louvain method), the state-of-the-art algorithm to maximize modularity [11], which can quickly detect high quality community structure in large networks; and the optimal modularity found by the column-generation method in [6]. All experiments are performed on a PC with AMD 2.00 Ghz processor and 32 GB of RAM.

The modularity values found by different algorithms are shown in Table 2. LDF produces higher quality than the Girvan-Newman algorithm (GN). In addition, LDF find the maximum modularity community structure for test cases 1 and 3, and produces better result than Blondel's and GN's for test case 6. Overall, LDF approaches closely Blondel and the optimal results.

TABLE 3
The modularity obtained by the state-of-the-art method Blondel [11], LDF, and the optimal modularity values OPT [6].

Network	n	m	Blondel	LDF	γ	Q_L
Foursq	45k	1,664k	0.4498 / 3.9s	**0.4502** / 2.0s	1.64	-
Facebook	64k	906k	0.6367 / 1.6s	**0.6414** / 2.2s	2.27	0.337
Twitter	88k	2,364k	**0.5491** / 2.3s	0.5470 / 2.8s	1.69	-
Flickr	81k	5,900k	0.5214 / 3.9s	**0.5215** / 5.4s	2.21	0.277

5.2 Online social networks

We further perform experiments on the snapshots of four popular online social networks: Facebook, Twitter, Foursquare, and Flickr. The size of each snapshots, the modularity, and the running time of LDF and Blondel's algorithm are shown in Table 3. The detail description on those network snapshots can be found in [18]. In addition, we apply the method in [15] to approximate the power-law exponent for each network and report the theoretical lower bound $Q_L = \frac{\zeta(\gamma)}{\zeta(\gamma-1)}$ on the modularity whenever the exponent is greater than 2. Neither Girvan-Newman nor the column-generation method in [6] are scalable enough for those network snapshots.

In most cases (three out of four cases), LDF has higher modularity than Blondel. This suggests that while LDF is less efficient than Blondel for small networks, it (LDF) is more competitive for larger networks. In the cases of Facebook and Flickr snapshots, the theoretical lower bounds is about half the modularity values found by LDF and Blondel's. We do not observe clear power-law distributions for the cases of Twitter and Foursquare snapshots. The method in [15] finds approximate power-law exponents 1.64 and 1.69 (with low likelihoods) for Foursquare and Twitter, respectively.

Both algorithms perform quickly, taking only few seconds on the networks of millions of edges. The running time of the two are nearly linear-time in term of the network size. Comparing closely, Blondel runs about 25% faster than LDF. One of the reason is that the implementation of Blondel's algorithm is mature and better optimized than our implementation for LDF.

6 Conclusions and Discussions

In this paper, we provide approximation algorithms for finding community structure via maximizing the modularity together with their performance guarantees. Scale-free networks with $\gamma > 2$ can be approximated within a factor of $\left(\frac{\zeta(\gamma)}{\zeta(\gamma-1)} - \epsilon\right)$; scale-free networks with lower exponent $\gamma > 1$ can be approximated within a factor $O(1/\log n)$, while the approximability of the modularity maximization problem when $\gamma < 1$ remains an open question. Our approaches hint the possibility of designing both theoretically and empirically efficient algorithms for finding community structure in complex networks.

In addition, our algorithms also theoretically explain why high modularity values were found in the scale-free networks [9], [24], [31]. The high value of modularity found in scale-free networks suggests that modularity should not be used as a quantitative for the goodness of community structure. High modularity values do not imply the existence of community structure in scale-free networks (and other classes of networks). Instead, modularity should be used only as an optimization objective to detect the community structure, if such a structure exists.

References

[1] G. Agarwal and D. Kempe, "Modularity-maximizing graph communities via mathematical programming," *Eur. Phys. J. B*, vol. 66, 2008.

[2] Y. Ahn, J. P. Bagrow, and S. Lehmann, "Link communities reveal multi-scale complexity in networks," *Nature*, 2010.

[3] W. Aiello, F. Chung, and L. Lu, "A random graph model for massive graphs," in *STOC '00*. New York, NY, USA: ACM, 2000.

[4] ——, "Random evolution in massive graphs," in *In Handbook of Massive Data Sets*. Kluwer Academic Publishers, 2001.

[5] R. Albert, H. Jeong, and A. Barabasi, "Error and attack tolerance of complex networks," *Nature*, vol. 406, 2000.

[6] D. Aloise, S. Cafieri, G. Caporossi, P. Hansen, S. Perron, and L. Liberti, "Column generation algorithms for exact modularity maximization in networks." *Phys. Rev. E*, vol. 82, 2010.

[7] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," *IEEE FOCS*, vol. 0, p. 238, 2002.

[8] A. Barabasi, R. Albert, and H. Jeong, "Scale-free characteristics of random networks: the topology of the world-wide web," *Physica A*, vol. 281, 2000.

[9] A. L. Barabasi, H. Jeong, Z. Nda, E. Ravasz, A. Schubert, and T. Vicsek, "Evolution of the social network of scientific collaborations," *Physica A: Statistical Mechanics and its Applications*, vol. 311, 2002.

[10] G. Bianconi and A.-L. Barabasi, "Competition and multiscaling in evolving networks," *EPL*, vol. 54, no. 4, p. 436, 2001.

[11] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, 2008.

[12] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski, and D. Wagner, "On modularity clustering," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 20, no. 2, 2008.

[13] G. Caldarelli, A. Capocci, P. De Los Rios, and M. A. Muñoz, "Scale-free networks from varying vertex intrinsic fitness." *Phys. Rev. Lett.*, vol. 89, no. 25, Dec. 2002.

[14] M. Charikar and A. Wirth, "Maximizing quadratic programs: Extending Grothendieck's inequality," *FOCS*, 2004.

[15] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," *SIAM Reviews*, 2007.

[16] B. DasGupta and D. Desai, "On the complexity of Newman's community finding approach for biological and social networks," *Journal of Computer and System Sciences*, pp. 50–67, 2013.

[17] T. N. Dinh, Y. Xuan, and M. T. Thai, "Towards social-aware routing in dynamic communication networks," in *IEEE IPCCC*, 2009, pp. 161–168.

[18] T. N. Dinh, Y. Shen, and M. T. Thai, "The walls have ears: Optimize sharing for visibility and privacy in online social networks," in *CIKM*. ACM, 2012.

[19] T. Dinh and M. Thai, "Finding community structure with performance guarantees in scale-free networks," in *SOCIALCOM*, oct. 2011, pp. 888 –891.

[20] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," ser. SIGCOMM '99. New York, NY, USA: ACM, 1999, pp. 251–262.

[21] A. Ferrante, "Hardness and approximation algorithms of some graph problems," 2006.

[22] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, 2007.

[23] S. Fortunato and C. Castellano, "Community structure in graphs," *Encyclopedia of Complexity and Systems Science*, 2008.

[24] M. Girvan and M. E. Newman, "Community structure in social and biological networks." *PNAS*, vol. 99, no. 12, 2002.

[25] B. H. Good, Y.-A. de Montjoye, and A. Clauset, "Performance of modularity maximization in practical contexts," *Phys. Rev. E*, vol. 81, p. 046106, Apr 2010.

[26] R. Guimera and L. A. Nunes A., "Functional cartography of complex metabolic networks," *Nature*, vol. 433, no. 7028, pp. 895–900, Feb. 2005.

[27] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Trans. on Mob. Com.*, vol. 10, no. 11, pp. 1576 –1589, nov. 2011.

[28] J. Jiang, R. Wang, and Q. A. Wang, "Network model of deviation from power-law distribution in complex network," *The European Physical Journal B - Condensed Matter and Complex Systems*, vol. 79, pp. 29–33, 2011, 10.1140/epjb/e2010-10230-x.

[29] R. Kannan, S. Vempala, and A. Veta, "On clusterings-good, bad and spectral," in *FOCS*. IEEE Computer Society, 2000, pp. 367–.

[30] E. A. Leicht and M. E. J. Newman, "Community structure in directed networks," *Phys. Rev. Lett.*, vol. 100, p. 118703, Mar 2008.

[31] M. E. J. Newman, "Modularity and community structure in networks," *PNAS*, vol. 103, 2006.

[32] N. P. Nguyen, T. N. Dinh, S. Tokala, and M. T. Thai, "Overlapping communities in dynamic networks: their detection and mobile applications," ser. MobiCom '11. ACM, 2011, pp. 85–96.

[33] N. P. Nguyen, T. N. Dinh, Y. Xuan, and M. T. Thai, "Adaptive algorithms for detecting community structure in dynamic social networks," in *IEEE INFOCOM*, april 2011, pp. 2282 –2290.

[34] A. Noack, "Modularity clustering is force-directed layout," *Phys. Rev. E*, vol. 79, p. 026102, Feb 2009.

[35] B. Pásztor, L. Mottola, C. Mascolo, G. Picco, S. Ellwood, and D. Macdonald, "Selective Reprogramming of Mobile Sensor Networks through Social Community Detection," in *EWSN*, vol. 5970, 2010, pp. 178–193.

[36] P. Ronhovde and Z. Nussinov, "Multiresolution community detection for megascale networks by information-based replica correlations," *Phys. Rev. E*, vol. 80, p. 016109, Jul 2009.

[37] M. Rosvall and C. T. Bergstrom, "Maps of random walks on complex networks reveal community structure," *Proceedings of the National Academy of Sciences*, vol. 105, no. 4, pp. 1118–1123, 2008.

[38] C. Tantipathananandh and T. Berger-Wolf, "Constant-factor approximation algorithms for identifying dynamic communities," ser. KDD '09. ACM, 2009.

[39] V. Vazirani, *Approximation algorithms*. Springer, 2001.

[40] B. Viswanath, A. Post, K. P. Gummadi, and A. Mislove, "An analysis of social network-based sybil defenses," in *SIGCOMM '10*. ACM, 2010, pp. 363–374.

[41] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "Sybilguard: defending against sybil attacks via social networks," in *SIGCOMM*. ACM, 2006, pp. 267–278.

[42] Y. Zhao, Y. Xie, F. Yu, Q. Ke, Y. Yu, Y. Chen, and E. Gillum, "Botgraph: large scale spamming botnet detection," in *NSDI*. Berkeley, CA, USA: USENIX Association, 2009, pp. 321–334.

[43] Z. Zhu, G. Cao, S. Zhu, S. Ranjan, and A. Nucci, "A social network based patching scheme for worm containment in cellular networks," in *INFOCOM 2009, IEEE*, april 2009, pp. 1476 –1484.

Thang N. Dinh (S'11) received the B.S. degree in Information Technology from Vietnam National University, Hanoi, Vietnam in 2007. He is currently a PhD candidate at the Department of Computer and Information Science and Engineering, University of Florida, under the supervision of Dr. My T. Thai. His research focuses on designing optimization methods and approximation algorithms for complex networking systems such as communication networks, social networks, mobile ad hoc network, and biological networks.

My T. Thai (M'06) received the Ph.D. degree in computer science from the University of Minnesota, Minneapolis, in 2005. She is an Associate Professor at the Computer and Information Science and Engineering Department, University of Florida. Her current research interests include algorithms and optimization on network science and engineering.

Dr. Thai has engaged in many professional activities, serving many conferences, such as being the PC chair of IEEE IWCMC 12, IEEE ISSPIT 12, and COCOON 2010. She is an Associate Editor of JOCO, IEEE Transactions on Parallel and Distributed Systems, and a series editor of Springer Briefs in Optimization. She has received many research awards including a Provosts Excellence Award for Assistant Professors at the University of Florida, a DoD YIP, and an NSF CAREER Award.