J Comb Optim
DOI 10.1007/s10878-013-9665-1

A near-optimal adaptive algorithm for maximizing modularity in dynamic scale-free networks

Thang N. Dinh · Nam P. Nguyen · Md Abdul Alim · My T. Thai

Abstract We introduce A^3CS, an adaptive framework with approximation guarantees for quickly identifying community structure in dynamic networks via maximizing Modularity Q. Our framework explores the advantages of the power-law distribution property found in many real-world complex systems. The framework is scalable for very large networks, and more excitingly, possesses approximation factors to ensure the quality of its detected community structure. To the best of our knowledge, this is the first framework that achieves approximation guarantees for the NP-hard *Modularity maximization* problem, especially on dynamic scale-free networks. To certify our approach, we conduct extensive experiments in comparison with other adaptive methods on both synthesized networks with known community structures and real-world traces including ArXiv e-print citation and Facebook social networks. Excellent

T. N. Dinh (✉) · N. P. Nguyen · M. A. Alim · M. T. Thai
Deparment of Computer and Information Science and Engineering,
University of Florida, Gainesville, FL 32611, USA
e-mail: tdinh@cise.ufl.edu

N. P. Nguyen
e-mail: nanguyen@cise.ufl.edu

M. A. Alim
e-mail: alim@cise.ufl.edu

M. T. Thai
e-mail: mythai@cise.ufl.edu

Present address:
T. N. Dinh
Department of Computer Science, Virginia Commonwealth University, Richmond, VA 23284, USA

Present address:
N. P. Nguyen
Computer and Information Sciences Department, Towson University, Towson, MD 21252, USA

Published online: 15 October 2013

empirical results not only confirm our theoretical results but also promise the practical applicability of A^{3}CS in a wide range of dynamic networks.

Keywords Adaptive approximation algorithm · Community structure · Modularity · Social networks

1 Introduction

Many complex systems in practice, despite their diversity in physical infrastructures and internal interactions, appear to display some ubiquitous features, such as the small-world and scale-free phenomena (Newman 2003), the power-law degree distribution, i.e., the fraction of nodes having degree k is proportional to $k^{-\gamma}$ where γ is the exponent parameter (Barabasi et al. 2000, 2002), and more importantly, the property of containing community structure. That is, they contain multiple groups of nodes with more interactions within a group and less connections among groups. For instance, a community in biology networks often consists of proteins, genes or subunits with functional similarity. In online social networks (OSNs), a community can be illustrated as a group of users having a common interest, such as music, movies or photography. Detecting this special structure finds itself extremely useful in deriving social-based solutions for many network problems, such as forwarding and routing strategies in communication networks (Dinh et al. 2009; Nguyen et al. 2011; Hui et al. 2011), Sybil defense (Yu et al. 2006; Viswanath et al. 2009), worm containment on cellular networks (Zhu et al. 2009), and sensor reprogramming (Pásztor et al. 2010). In the network visualization perspective, the detection of community structure is extremely helpful since it only displays core network components and their mutual interactions, hence presents a more compact and understandable description of the network as a whole (Noack 2009).

Perhaps the most notable feature of complex networks that applies in a wide range of practical systems is their dynamical property. Many types of complex networks in reality often evolve heavily over time and frequently experience topological changes during their evolution. In the sense of OSNs, such as Facebook, Twitter or Google+, these changes are often introduced by users joining in or leaving a particular group or community, by friends and friends connecting together, or by new people making friend with each other. Though any of these events seems to have a little effect on the local structure of the network on one hand; the network dynamics over a long period of time on the other hand, may significantly transform the current community structure to a totally different one, thus raises a natural need of reidentification. However, the rapid and unpredictably changing topologies of these networks makes it an extremely complicated yet challenging problem.

Although the multiple execution of any static community detection algorithm (Clauset et al. 2004; Blondel et al. 2008) can be utilized to find the new community structure whenever the network evolves, repeatedly doing so will incur extremely heavy computational cost as well as is time consuming, especially on large scale networks. A better, much efficient and less time consuming way to accomplish this task is to adaptively update the network communities from the previously discovered structures based only on the network changes, which helps to avoid the hassle of recomputing from scratch. This reassembles the concept of *adaptive algorithms* for detecting community structure in dynamic social networks. Of course, the processing time of any good adaptive algorithm should be propositional to only the amount of changes introduced during the network evolution.

Many approaches have been suggested for dynamic networks in this line of adaptive algorithms (Nguyen et al. 2011; Lancichinetti et al. 2011; Lin et al. 2008; Dinh et al. 2009 (see Related work for a review). However, they encounter the following crucial limitations: (1) their execution time is not proportional to only the amount of changes in the network. As a result, they are not capable for very large scale social networks, and more important (2) *they do not possess any performance guarantee to ensure the quality of detected communities*. This is the most challenging feature one can ask for an algorithm of this kind since local adaptive procedures may not be able to reflect changes in network topology over a long duration, as observed in Nguyen et al. (2011). In addition, most of these methods do not take into account the advantages of the power-law distribution, which also is a key property of complex networks.

In this paper, we offer A^{3}CS [1], an ***adaptive algorithm with approximation guarantee*** for quickly identifying community structure in OSNs via maximizing *Modularity*—a widely accepted measure in community detection field. The exciting features that differentiate A^{3}CS from the other adaptive algorithms are (1) it possess approximation factors to ensure the quality of the detected community structures, and (2) it explores the advantages of power-law distribution, is scalable for very large networks and can be easily extended to directed networks with the same performance guarantee.

In particular, A^{3}CS is optimal up to a constant factor $\rho \approx \frac{\zeta(\gamma)}{\zeta(\gamma-1)}$ where $\zeta(\gamma) = \sum_{i=1}^{\infty} \frac{1}{i^{\gamma}}$ is the Rieman Zeta function. The constant factor is applicable when the network's power exponent $\gamma > 2$, which is the most popular scenario (Barabasi et al. 2002; Faloutsos et al. 1999). To the best of our knowledge, our proposed algorithm is the first approach that achieves approximation guarantees for the NP-hard *Modularity maximization* problem in dynamic scale-free networks. Finally, we conduct extensive experiments in comparison with other methods on both synthesized networks with known community structures and real-world traces including ArXiv e-print citation and Facebook social networks. Excellent empirical results not only confirm our theoretical results but also the practical applicability of our proposed framework A^{3}CS in a wide range of OSNs.

Related work. The design and analysis of adaptive algorithms to detect network community structure have attracted a lot of attention recently, and many methods have been proposed in the literature. For instance, algorithms based on optimizing local gained modularity (Nguyen et al. 2011), based on nonnegative matrix factorization (Lin et al. 2008), by compression of network modules (Dinh et al. 2009) or by finding groups of nodes that have significant statistical features in the network (Lancichinetti et al. 2011). However, designing adaptive algorithms that possess approximation ratios to guarantee their performance has not much been achieved.

[1] Adaptive Approximation Algorithm for Community Structure detection

The modularity maximization problem is shown to be NP-hard (Brandes et al. 2008) and APX-hard (DasGupta and Desai 2012). 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 Agarwal and Kempe (2008) in which they rounded the fractional solution of a linear programming (LP). However, no approximation algorithms were provided. In Tantipathananandh and Berger-Wolf (2009), the authors provided constant-factor approximation algorithms for identifying communities in dynamic social networks. We note that the concept of communities considered in the paper is rather different with the usual notion of communities and clusters. The observed grouping structures of the network at each time points are given. And the communities across several time points are identified in order to minimize switching group costs of individuals in the network. Thus, the proposed approaches are inapplicable to most biological, social and communication networks when the structure in each time steps is unknown.

Some clustering problems also yield algorithms with good performance guarantees (Bansal et al. 2002; Giotis and Guruswami 2006), 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.

Modularity has several known drawbacks. For example, Fortunato and Barthelemy (Fortunato and Barthelemy 2007) has shown the resolution limit, i.e., maximizing modularity methods fail to detect communities smaller than a scale, the resolution limit only appears when the network is substantially large (Lancichinetti and Fortunato 2009). Another drawback is modularitys highly degenerate energy landscape (Good et al. 2010), which may lead to very different yet equally high modularity partitions. However, finding community structure via modularity maximization is still the common choice for many networks of interests including biological networks and social networks. The method proposed by Blondel et al. (2008) to optimize modularity is one of the best performing algorithms according to the LFR benchmark (Lancichinetti and Fortunato 2009). Approximation algorithms for maximizing modularity are first studied in DasGupta and Desai (2012) for networks with maximum degree at most $\frac{\sqrt{5}n}{16\ln n}$ and in Dinh and Thai (2011) for scale-free networks.

Organization. We present the preliminaries in Sect. 2. The main algorithm and its time complexity is presented in Sect. 3. The analysis of the adaptive approximation factor on scale-free networks are presented in Sect. 4. In Sect. 5, we perform extensive experiments on both synthesized networks and real-world traces. Finally, we conclude our paper in Sect. 6.

2 Preliminaries

In this section, we present the dynamic graph model that represents a dynamic complex network, and the quality measurement for the goodness of the detected community structure. Furthermore, we introduce the notion of *Adaptive Approximation Algorithm*.

2.1 Network model

A network is represented by an undirected unweighted graph $G = (V, E)$ with $n = |V|$ vertices and $m = |E|$ edges. The *adjacency matrix* of G is denoted by $\boldsymbol{A} = \left(A_{ij}\right)$, 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.

2.2 Community structure and quality measurement

Given a community structure $\mathcal{C} = \{C_1, C_2, \ldots, C_l\}$ where $C_i \subseteq V$ is the ith community in the network, the *modularity* (Newman 2006), denoted by Q, is defined as

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

where $\delta_{ij} = 1$ if i and j are in the same community, and 0 otherwise.

The modularity values can be either positive or negative and the higher (positive) modularity values indicate a stronger community structure. The *maximizing modularity problem* asks to find a division which maximizes the modularity value.

The modularity can be equivalently defined as

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

where E_t is the number of edges that both 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. In addition, we can infer from (2) that $Q \leq 1$.

2.3 Dynamic network model

A dynamic network $\mathcal{G}$ is represented by a series of time dependent network snapshots $\mathcal{G} = \{G^{(0)}, G^{(1)}, \ldots, G^{(s)}\}$, where $G^{(t)} = (V^{(t)}, E^{(t)})$ is the snapshot of the network at the time point $0 \leq t \leq s$. The change between two consecutive snapshots $G^{(t)}$ and $G^{(t-1)}$ is denoted by $\Delta G^{(t)} = (\Delta V^{(t)}, \Delta E^{(t)})$ where $\Delta V^{(t)} = V^{(t)} \ominus V^{(t-1)}$ and $\Delta E^{(t)} = E^{(t)} \ominus E^{(t-1)}$. Here, the notation $\ominus$ denotes the symmetric difference between two sets. Equivalently, the dynamic network can also be given by the original snapshot $G^{(0)}$ and the subsequence changes in the network, i.e., $\mathcal{G} = (G^{(0)}, \Delta G^{(1)}), \ldots, \Delta G^{(s)})$.

Adaptive Community Detection Problem. (ACD) Given a dynamic network $\mathcal{G} = (G^{(0)}, \Delta G^{(1)}, \ldots, \Delta G^{(s)})$, we need to find community structure in each network snapshot ***adaptively***. That is the community structure at time point t is detected base on the community structure at the time point $t-1$ and the change in the network $\Delta G^{(t)}$ rather than recomputing the community structure in the snapshot $G^{(t)}$ from scratch.

2.4 Adaptive approximation algorithm

Adaptive Algorithm. In general, an adaptive algorithm attempts to reduce the computational cost of finding the new solution (the new community structure in our case) when changes occur to the network topology. In this manner, the new solution is found by updating the current solution according to the changes only, rather than recomputing a new solution from scratch. An adaptive algorithm processes the input changes in a serial fashion, assuming *these changes arrive in batch* whereas an *offline algorithm* is provided with the whole network information in the first place (for example, community detection in a static network).

Note that an adaptive algorithm differs from an *online algorithm* in the availability of the input. In an online algorithm, the input is available *piece-by-piece* in their orders while in an adaptive algorithm, the input arrives in batch without any particular order. That is an online algorithm knows the exact order in which edges and vertices arrive or get removed; however an adaptive algorithm is only provided a set of changed edges and nodes are given at each time point as a whole. Therefore, an online algorithm can be regarded as a special case of an adaptive algorithm in which the order of input are defined beforehand.

Adaptive Approximation Algorithm. In a ρ*-adaptive approximation algorithm*, the solution at any time point t will not be less (or more, depending on the objective is minimized or maximized) than a factor ρ times the optimum solution, provided the whole input from beginning to the time point t available at once. For example, a ρ-adaptive approximation algorithm for the ACD problem will find at any time point t a community structure with modularity at least $\rho Q_{opt}^{(t)}$, where $Q_{opt}^{(t)}$ is the maximum modularity of any community structure in $G^{(t)}$, the graph at time point t. The factor ρ is called the *adaptive approximation factor*, which represents the relative performance guarantee. 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 the performance guarantee.

3 A^{3}CS: adaptive algorithm for community structure in dynamic networks

In this section, we present A^{3}CS, the adaptive algorithm to detect community structure in dynamic networks together with its analysis on time complexity and several heuristics to further enhance the algorithm.

3.1 Low-degree following principle

Intuitively, a node in the network should be in the same community with at least one of its neighbors. Regarding maximizing modularity, this can be proved rigorously as in the following lemma.

Lemma 1 (Dinh et al. 2009) *Every non-isolated node must be in the same community with one of its neighbor, in order to maximize modularity* Q.

Therefore, if we randomly group a vertex u with one of its neighbor i.e., assigning them to a same community, the chance that we make the same choice as in the optimal community structure is at least $1/k_i$. Thus, the lower the degree of i, the higher the probability of making the optimal move. This motivates the underlying principle of our A^{3}CS algorithm: randomly grouping/joining *low-degree nodes* with one of their neighbors.

3.2 Algorithm descriptions

A^{3}CS, presented in Algorithm 1, is a meta-algorithm that first calls A-Base algorithm to find the community structure $C^{(0)}$ of the first network snapshot $G^{(0)}$, then iteratively finds community structure $C^{(t)}$ at time point t by invoking the A-Adaptive algorithm (Algorithm 3). The two algorithms A-Base and A-Adaptive construct the community structure via assigning values for two arrays $label[i]$ and $follow[i]$.

Algorithm 1. A^{3}CS—Adaptive Approx. Alg. for CS

1. $C^{(0)} =$ A-Base$(G^{(0)})$
2. **for** $t = 1$ to s **do**
3. $\quad C^{(t)} =$ A-Adaptive$(C^{(t-1)}, \Delta G^{(t)})$

The meaning of $label$ and $array$ is as follow. Each node i is labeled with either $leader$, $follower$, or unlabeled (also denoted with $\varnothing$). For a node i labeled with $follower$, $follow[i]$ is the name of the leader that i follows. Precisely, we have

$$follow[i] = \begin{cases} i & \text{if } label[i] = leader \\ j \neq i & \text{if } label[i] = follower \,\&\, i \text{ follows } j \\ \varnothing & \text{if } label[i] = \varnothing \end{cases}$$

At any time point t, the community structure is given by the union of two types of communities: (1) all *followers* that follow the same *leader* are assigned into the same community; and (2) each unlabeled node forms a singleton community of size one.

At the heart of the proposed algorithms, the assigned labels satisfy the important properties stated in the following lemma.

Lemma 2 *At the end of the algorithms A-Base and A-Adaptive, the following properties hold.*

1. *All* low-degree *nodes i.e., nodes with degree at most d_0 for some predefined constant $d_0 > 0$, are labeled either with $leader$ or $follower$.*
2. *All* followers *are* low-degree *nodes.*
3. *Each* leader *is followed by at least one* follower*; and each* follower *follows exactly one* leader. *Thus* followers *will not follow each other or unlabeled nodes.*

The intuition to this lemma will be explained through the presentation of A-Base and A-Adaptive.

Algorithm 2. A-Base

$label[i] = \varnothing, follow[i] = \varnothing \; \forall i = 1..n$
Sorted nodes in non-decreasing order of degree.
for each vertex i with $k_i \leq d_0$ **do**
  if $label[i] = \varnothing$ **then**
    FOLLOW_NEIGHBOR(i)
Return $C^{(0)} = \langle follow \rangle$

A-Base. (Alg. 2) This algorithm finds the community structure of $G^{(0)}$ via labeling nodes in the network. Nodes are first sorted in a non-decreasing order of degree, and then, each *low-degree* and unlabeled node i selects one of its neighbors to follow using the FOLLOW_NEIGHBOR algorithm, (Alg. 4a) in which the $label[i]$ and the $follow[i]$ are assigned accordingly. We can verify that all the properties in Lemma 2 hold at the end of A-Base.

Algorithm 3. A-Adaptive ($C^{(t-1)}$, $\Delta G^{(t)}$)

1. **for each** edge $(u, v) \in \Delta E^{(t)}$ **do**
2.   Update degree of nodes u and v
3. **for each** vertex i appears in $\Delta G^{(t)}$ **do**
4.   **if** $(k_i \leq d_0) \& (label[i] = \varnothing)$ **then**
5.   FOLLOW_NEIGHBOR(i)
6.   **else if** $(k_i > d_0) \& (label[i] = follower)$ **then**
7.   UNFOLLOW(i)
8. Return $C^{(t)} = \langle follow \rangle$

A-Adaptive. (Alg. 3) This algorithm finds the community structure at time point t based on $C^{(t-1)}$ and $\Delta G^{(t)}$—the previous community structure and the changes in the network. After updating the node degrees (lines 1–2), the algorithm checks all nodes that appear in $\Delta G^{(t)}$ and corrects all possible "mis-labeling" caused by the degree changes. Two "mis-labeling" cases are (1) low-degree and unlabeled nodes as the results of removing edges (or adding new nodes), and (2) follower nodes with the degrees higher than d_0 as the results of adding new edges/nodes. The two "mis-labeling" cases are corrected using FOLLOW_NEIGHBOR and UNFOLLOW algorithms, as shown in lines 4 to 7.

Algorithm 4a. FOLLOW_NEIGHBOR(i **)**

$label[i] = follower$
if $\exists j \in N(i) : label[j] \neq follower$ **then**
    if $label[j] = \varnothing$ **then** $label[j] = leader$
    $follow[i] = j$
else
  Select a random $j \in N(i)$
  UNFOLLOW(j)
  $follow[i] = j, label[j] = leader$
Update the modularity value.

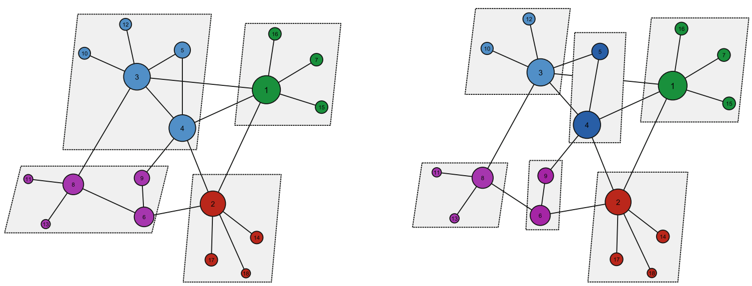
FOLLOW_NEIGHBOR. (Alg. 4a) This is the fundamental procedure in A^{3}CS. Given a node i, the algorithm identifies a neighbor j so that i can follow j without violating the properties of Lemma 2. Lines 3 and 4 explore the case when we can find a non-follower neighbor j of i. When all neighbors of i are followers, we first use the UNFOLLOW algorithm to make a neighbor j of i unlabeled or labeled it with $leader$, and only then we can let i follow j (lines 6–8).

Algorithm 4b. UNFOLLOW(i)

1. Let $j = follow[i]$, $label[i] = \varnothing$
2. **if** j has no followers **then**
3. **if** $k_j \le d_0$ **then**
4. $follow[j] = i$, $label[j] = follower$
5. $label[i] = leader$
6. **else** $label[j] = \varnothing$
7. Update the modularity value.

UNFOLLOW. (Alg. 4b) As briefly mentioned, the algorithm UNFOLLOW is invoked when we need to stop a node i from following its current leader j. This can usually be done by simply unlabeling i. The interesting case happens when i is the only follower of j and unlabeling i will make j a leader without followers (opposing the third property in Lemma 2). We handle this case by either unlabeling j or swapping i and j's labels together with making j follow i (lines 3–6).

Figure 1a shows an example of A-Base algorithm on a network snapshot with $d_0 = 2$. Comparing to the community structure with the maximum modularity value (found by solving the Integer Programming in Agarwal and Kempe (2008)), community structure produced by A-Base has more communities of smaller sizes. However, by merging those small communities using local search method (Newman 2006), we are able to obtain the optimal community structure on the right.



(a) Community structure with the maximum Q **(b)** A^{3}CS algorithm

Fig. 1 Community structures found in a sample network: **a** The optimal community structure found by solving the Integer Programming formulation in Agarwal and Kempe (2008) with the mathematical software package CPLEX. **b** The community structure found by the A^{3}CS algorithm with $d_0 = 2$. *Leader nodes* are drawn larger within each community

Note that we keep A^3CS algorithm simple to simplify the analysis on the adaptive approximation factor in Sect. 4. In Sect. 3.4, we shall present optimization techniques to further enhance the solution quality (without affecting the adaptive approximation factor).

3.3 Time complexity

We provide the time complexities for A-Base and A-Adaptive, two major components in A^3CS. We can verify that FOLLOW_NEIGHBOR(i) and UNFOLLOW(i) both take $O(k_i)$ times. In the worst case, the A-Base touches all nodes in $G^{(0)}$. Thus the time complexity of A-Base is given in the following lemma.

Lemma 3 *The time complexity of A-Base (Alg. 2) is* $O(|V^{(0)}| + |E^{(0)}|)$.

Similarly, A-Adaptive will touch all nodes in $\Delta G^{(t)}$ in the worst case; hence the algorithm has the following time complexity.

Lemma 4 *A-Adaptive (Alg. 3) has a* linear time complexity *w.r.t. the total degree of nodes in* $\Delta G^{(t)}$

The time complexity of A-Adaptive , the adaptive part of A^3CS, is highly desirable and we probably cannot hope for an adaptive algorithm with better time complexity. The time complexity does not involve any global parameters such as $|V^{(t)}|$, the number of nodes or $|E^{(t)}|$, the number of edges in the network at time point t. This is extremely helpful in the case of a very large network with billion of nodes/edges in which the changes only happen within a small local part of the network. The existing methods for dynamic community structure (Lin et al. 2008; Lancichinetti et al. 2011) involve at least $\Omega(n)$ time complexities whenever the community structure need to be updated. Thus they are far more time-consuming than an adaptive algorithm such as A^3CS and QCA (Nguyen et al. 2011).

3.4 Parameter selection and further optimization

3.4.1 Automatic selection of d_0

Selecting parameter d_0 is an important part of A^3CS. For the analysis of the adaptive approximation ratio in Sect. 4, 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 to maximize modularity Q within A-Base. This can be done by trying all possible values of d_0 from 1 to $n_0 = |V^{(0)}|$, and selecting d_0 that maximizes Q. This approach can be done without increasing time complexity of *A-Base*. Recall that nodes are sorted in a non-decreasing order of their degrees. Therefore, if we first set $d_0 = n_0$ and apply FOLLOW_NEIGHBOR on all unlabeled nodes, we will eventually iterate through all possible values of d_0. All we need to do is to remember the vertex i that associates with the maximum modularity and select $d_0 = k_i$.

Lemma 5 *Automatic selection of the best* d_0 *can be done in* $O(|V^{(0)}| + |E^{(0)})|)$.

3.4.2 Further optimization

We can further optimize the $\mathsf{A}^3\mathsf{CS}$ algorithm without changing its properties stated in Lemma 2. First, we can derandomize the selection of neighbor inside FOLLOW_NEIGHBOR 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 (Blondel et al. 2008; Dinh et al. 2009). We then apply local search (Newman 2006) on the abstracted network to increase the overall modularity.

4 Adaptive approximation factor in scale-free networks

We analyze the proposed $\mathrm{A}^3\mathrm{CS}$ algorithm to show that it yields a constant *adaptive approximation algorithm factor* for the *Adaptive Community Detection Problem* (ACD) in scale-free networks. Let $Q_{\text{opt}}^{(t)}$ be the maximum of modularity values over all possible divisions of the network at time point t, our algorithm finds in polynomial-time a division with the modularity at least $\rho\, Q_{\text{opt}}^{(t)}$ for some constant $0 < \rho < 1$.

4.1 Networks with power-law degree sequences

We consider in this section scale-free networks with the power exponents $\gamma > 2$. This class covers a wide range of scale-free networks of interest, since typically $2 < \gamma < 3$ (Barabasi et al. 2002; Albert et al. 2000; Faloutsos et al. 1999). For examples, scientific collaboration networks has γ in the range $2.1 < \gamma < 2.45$, Word Wide Web with γ for in-degree and out-degree of 2.1 and 2.45, respectively; Internet at router and intra-domain level with $\gamma = 2.48$ and so on.

Let us consider the network at time point t. Denote by n_t and m_t the number of nodes and edges in $G^{(t)}$. In our network, the number of vertices of degree k is $\lfloor \frac{e^{\alpha_t}}{k^\gamma} \rfloor$ where e^{α_t} is the normalization factor as in the $P(\alpha, \gamma)$ model (Aiello et al. 2000). Due to the power-law distribution, we can infer that the maximum degree in the network is $e^{\frac{\alpha_t}{\gamma}}$ (since for $k > e^{\frac{\alpha_t}{\gamma}}$, the number of edges will be less than 1). The number of vertices and edges are

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

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

where $\zeta(\gamma) = \sum_{i=1}^{\infty} \frac{1}{i^\gamma}$ is the Riemann Zeta function (Aiello et al. 2000) which converges absolutely for $\gamma > 1$ and diverges for all $\gamma \le 1$. 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_t, 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 value of community structure $C^{(t)}$ at time point t, discovered by A^3CS is at least $\frac{\zeta(\gamma)}{\zeta(\gamma-1)} - \epsilon$. Thus, A^3CS is a $\left(\frac{\zeta(\gamma)}{\zeta(\gamma-1)} - \epsilon\right)$-adaptive approximation algorithm for the ACD problem, where $\epsilon > 0$ is an arbitrary small constant.*

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

We first bound $E(C^{(t)})$ using properties of Lemma 2. First, all vertices with degree at most d_0 must be labeled, and at least half of them must be labeled with *follower* since one leader is followed by at least one follower. Second, edges between the *followers* and their leaders have both endpoints inside the same community. From Eq. 3, we have $E(C^{(t)}) \geq \frac{1}{2}e^{\alpha_t} \sum_{i=1}^{d_0} i^{-\gamma}$.

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

$$E(C^{(t)}) \geq \frac{1}{2}e^{\alpha_t} \sum_{i=1}^{d_0} i^{-\gamma} \geq \frac{1}{2}e^{\alpha_t}(\zeta(\gamma) - \epsilon). \tag{4}$$

We now give an upper bound for the volume of a community based on the degree of its *leader*. Let node i be a *leader* within a community. Node i is followed by at most k_i *followers* of degree at most d_0. Thus, the volume, i.e., the total degree of nodes in the community, is bounded by

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

Therefore, we have:

$$\begin{aligned} Q(C^{(t)}) &\geq \frac{e^{\alpha_t}\,(\zeta(\gamma) - \epsilon)}{2m_t} - \sum_{i \in L} \frac{(2k_i d_0)^2}{4m_t^2} \\ &\geq \frac{e^{\alpha_t}\,(\zeta(\gamma) - \epsilon)}{2m_t} - \sum_{i=1}^{n_t} \frac{4d_0^2 k_i^2}{4m_t^2} = \frac{n_t - \epsilon e^{\alpha_t}}{2m_t} - 8d_0^2 D, \end{aligned}$$

where

$$D = \sum_{i=1}^{n_t} \frac{k_i^2}{8m_t^2} = \sum_{k=1}^{e^{\frac{\alpha_t}{\gamma}}} \frac{e^{\alpha_t}}{k^\gamma} \frac{k^2}{8m_t^2} = \frac{e^{\alpha_t}}{8m_t^2} \sum_{k=1}^{e^{\frac{\alpha_t}{\gamma}}} k^{2-\gamma}. \tag{5}$$

Since $\gamma > 2$, we have $k^{2-\gamma} < 1$, from Eq. 5 for sufficient large n we have

$$\begin{aligned} Q(C^{(t)}) &\geq \frac{\zeta(\gamma) - \epsilon}{\zeta(\gamma - 1)} - 8d_0^2 \frac{e^{\alpha_t}}{8m_t^2} e^{\frac{\alpha_t}{\gamma}} \\ &\geq \frac{\zeta(\gamma) - \epsilon}{\zeta(\gamma - 1)} - \frac{4d_0^2}{\zeta(\gamma - 1)^2 e^{\alpha_t(1-1/\gamma)}} \\ &\geq \frac{\zeta(\gamma) - \epsilon}{\zeta(\gamma - 1)} - \frac{4d_0^2}{n_t^{(1-1/\gamma)}} \geq \frac{\zeta(\gamma)}{\zeta(\gamma - 1)} - \epsilon. \end{aligned} \tag{6}$$

Since $Q_{\text{opt}}^{(t)} < 1$, A^{3}CS is an $\left(\frac{\zeta(\gamma)}{\zeta(\gamma-1)} - \epsilon\right)$-adaptive approximation algorithm for the ACD problem.

For scale-free networks with $\gamma > \gamma_0 \approx 2.23$, by Theorem 1, the modularity value is at least 0.3. Here, γ_0 is found by solving the equation $\zeta(\gamma) - 0.3\zeta(\gamma - 1) = 0$. 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 $\frac{\zeta(2.48)}{\zeta(1.48)} \approx 0.5$, which means A^{3}CS is an $\frac{1}{2}$-adaptive approximation algorithm in that case.

4.2 Relaxed power-law network models

We extend the analysis to a broader class of networks in which the degree sequence might deviates from the power-law distribution. Specifically, we consider the graphs that satisfy the following two common characteristics of real-world scale-free networks: (1) the network is sparse, i.e., $m_t \leq cn_t$ for some constant $c > 0$, and (2) the network does not have "super" giant hubs, vertices of degree $\Omega(m_t)$ (Aiello et al. 2000, 2001; Ferrante 2006; Bianconi and Barabasi 2001). For networks satisfying those two properties, the following theorem holds.

Theorem 2 *For a sufficient large network satisfying $m_t \leq cn_t$ for some constant c and the maximum degree $\Delta = o(m_t)$, A^{3}CS returns a community structure $C^{(t)}$ whose modularity is at least $\frac{2c-1}{4c^2}$. Thus, A^{3}CS is an $\frac{1}{2(c+1)}$-adaptive approximation algorithm for the ACD problem.*

Proof The key idea is to select $d_0 = \lfloor 2c \rfloor$. Since, the network cannot have more than $n_t \lceil 2c \rceil^{-1}$ vertices of degree $\lceil 2c \rceil$ or higher, the number of vertices with degree at most $\lfloor 2c \rfloor$ is at least $n_t(1 - \lceil 2c \rceil^{-1})$. Using the same arguments in the proof of Theorem 1, the number of edges with both ends inside the same community, is then at least $\frac{n_t}{2}(1-\lceil 2c\rceil^{-1})$ and the volume of the community with vertex i as the *leader* is bounded by $2k_i d_0$. Therefore,

$$Q(C^{(t)}) \geq \frac{n_t}{2m_t}\left(1-\frac{1}{\lceil 2c\rceil}\right) - \sum_{i\in L}\frac{2d_0^2k_i^2}{4m_t^2}$$

$$\geq \frac{1}{2c}\left(1-\frac{1}{\lceil 2c\rceil}\right) - \sum_{i=1}^{n}\frac{2d_0^2k_i^2}{4m_t^2}. \tag{7}$$

The rest is to prove that the second term in the above formula is only $o(1)$. Let Δ be the maximum degree. Since $\sum_{i=1}^{n_t} k_i = 2m_t$, we have $\sum_{i=1}^{n_t} k_i^2 \leq \sum_{i=1}^{n_t} k_i\Delta = 2m_t\Delta$. Using the hypothesis $\Delta = o(m_t)$, we have

$$\sum_{i=1}^{n_t}\frac{2d_0^2k_i^2}{4m_t^2} \leq \frac{2d_0^2 2m_t\Delta}{4m_t^2} = d_0^2\frac{\Delta}{m_t} = o(1).$$

Thus, the modularity $Q(C^{(t)})$ is at least

$$\frac{1}{2c}\left(1-\frac{1}{\lceil 2c\rceil}\right) - o(1) > \frac{2c-1}{4c^2}.$$

Since, $Q_{\text{opt}}^{(t)} < 1$ and $\frac{4c^2}{2c-1} < 2(c+1)$, A^{3}CS is a $\frac{1}{2(c+1)}$-approximation algorithm for the ACD problem.

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.

Remarks. The simplicity of A^{3}CS enables its extension to directed networks with only subtle changes. As long as the in-degree (or the out-degree) sequence follows a power-law distribution, the algorithm will yield an constant adaptive approximation factor. The key difference to the undirected case is that we either follow only incoming links or only outgoing edges. As the consequence, the adaptive approximation factor is reduced by half. In addition, the following mechanism in the algorithm can be easy implemented in a distributed manner which is a huge advantage for ad hoc networks.

5 Experimental results

In this section, we first validate the performance our A^{3}CS on different synthesized networks with known community structures (or groundtruths), and then present the empirical results on popular real world traces arXiv eprint citation (data 2003) and Facebook social networks (Viswanath et al. 2009). To certify the performance of our algorithms, we compare A^{3}CS to other adaptive community detection methods including (1) QCA framework suggested by Nguyen et al. (2011), (2) FacetNet algorithm proposed by Lin et al. (2008), (3) MIEN algorithm proposed by Dinh et al. (2009), and (4) OSLOM method suggested by Lancichinetti et al. (2011). Our experiments are performed on a PC with an Intel Xeon 2.93Ghz CPU and 12 GB RAM.

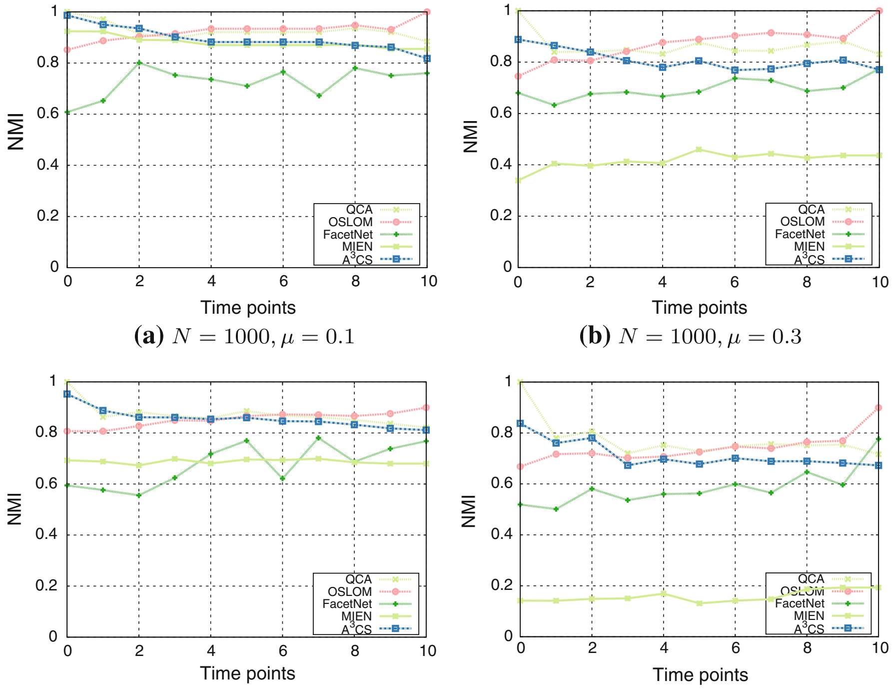
5.1 Results on synthesized networks

Certainly, the best way to evaluate our approach is to validate them on real networks with known community structures. Unfortunately, we often do not know that structures beforehand, or such structures cannot be easily mined from the network topology. Although synthesized networks might not reflect all the statistical properties of real ones, they provide us known ground truths via planted communities, and the ability to vary other parameters such as sizes, densities and overlapping levels, etc. Testing community detection methods on generated data has become an usual practice that is widely accepted in the field. Hence, comparing A^3CS with other dynamic methods on synthesized networks not only certifies its performance but also provides us the confidence to its behaviors on real world traces.

Setup. We use the well-known LFR benchmark to generate 40 networks with 10 snapshots. Parameters are: the number of nodes $N = \{1000, 5000\}$, the mixing parameter $\mu = \{0.1, 0.3\}$ controlling the overall sharpness of the community structure. To quantify the similarity between the identified communities and the ground truth, we adopt a well known measure in Information Theory called normalized mutual information (NMI). NMI has been proven to be reliable and is widely used in testing community detection algorithms. Basically, $NMI(U, V)$ equals 1 if structures U and V are identical and equals 0 if they are totally separated, and the higher NMI the better. We want to demonstrate (1) quality of communities detected by A^3CS (and other methods) through NMI scores, and (3) the modularity values achieved by A^3CS in comparison with those of the groundtruths.

Results. The NMI and Modularity values are reported in Figs. 2 and 3, respectively. As depicted in their subfigures, NMI and modularity values obtained by A^3CS, in general, are very high and competitive with those of OSLOM and QCA while are much better than those produced by MIEN and FacetNet methods. In average, NMI scores achieved by A^3CS are only about 5.29 and 5.95 % lag behind QCA and OSLOM, and are from 16.1 and 24.8 % better than FacetNet and MIEN on networks with $N = 2500$ and $N = 5000$ nodes, respectively. Moreover, the performance of FacetNet and MIEN seems to be unstable as their NMI scores degrade quickly, especially when the network community structure becomes stochastic and unclear (as $\mu = 0.3$ in Fig. 2b, d). The NMI scores of our framework (as well as QCA and OSLOM), on the other hand, appear to stay wealthy and do not seem to be strongly affected by the unclearness of network community structure, even when $\mu = 0.3$. This implies that network communities revealed by our A^3CS framework are highly similar to those contained in the groundtruths, and also are highly competitively with those obtained by other methods.

In terms of the main objective function modularity, the values obtained by A^3CS, QCA and OSLOM methods are very similar to each other and differ insignificantly from those of the groundtruths, whereas the values attained by MIEN and FacetNet are much lower, especially on networks with unclear community structure of $\mu = 0.3$. In average, the modularity values of A^3CS tend to tangle with those of QCA while NMI scores are just about 2–3 % less than the groundtruths, and almost the same as those of QCA and OSLOM while are better than MIEN ($\approx$8.3 %) and particularly FacetNet ($\approx$12.67 %). Note that the good behaviors of OSLOM and QCA are not



(a) $N = 1000, \mu = 0.1$ **(b)** $N = 1000, \mu = 0.3$

(c) $N = 5000, \mu = 0.1$ **(d)** $N = 5000, \mu = 0.3$

Fig. 2 Normalized mutual information (NMI) scores on synthesized networks with known communities

really surprising since they are current best adaptive algorithms for dynamic networks; however, the highly competitively performance A^{3}CS is indeed very impressive and strongly encouraged, especially when A^{3}CS is the adaptive algorithms with approximation ratios to guarantee its performance. Moreover, as we will show next, our A^{3}CS is more scalable for larger networks than QCA or OSLOM methods as it is significantly less time consuming. These experiments on generated network conclude the quality of communities detected by A^{3}CS and give us the confidence to its behavior in real-world traces.

5.2 Results on real-world traces

We next present the results of A^{3}CS algorithms on real world dynamic traces including arXiv e-print citation (data 2003), and Facebook social networks (Viswanath et al. 2009). Due to the lack of community groundtruths corresponding to these traces, we report the performance of the aforementioned algorithms in reference to the static method proposed by Blondel et al. (2008), whose goal also aims to optimize Q and whose performance has been verified in the literature. In particular, we will show the following quantities (1) modularity values, (2) the quality of the identified network communities through NMI scores, and (3) the processing time of our A^{3}CS in comparison with other methods. The above networks possess to contain strong community

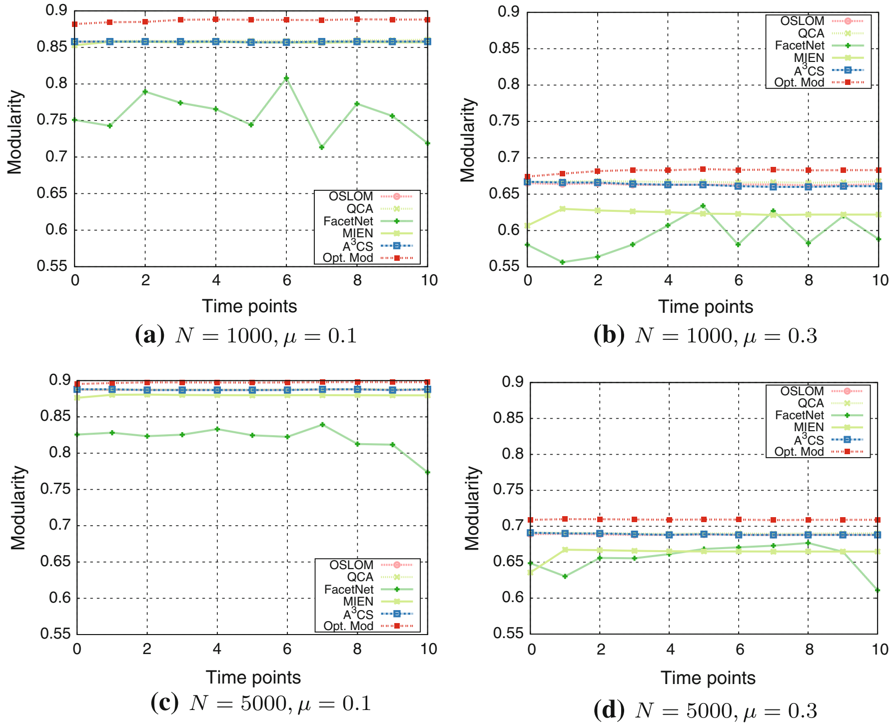


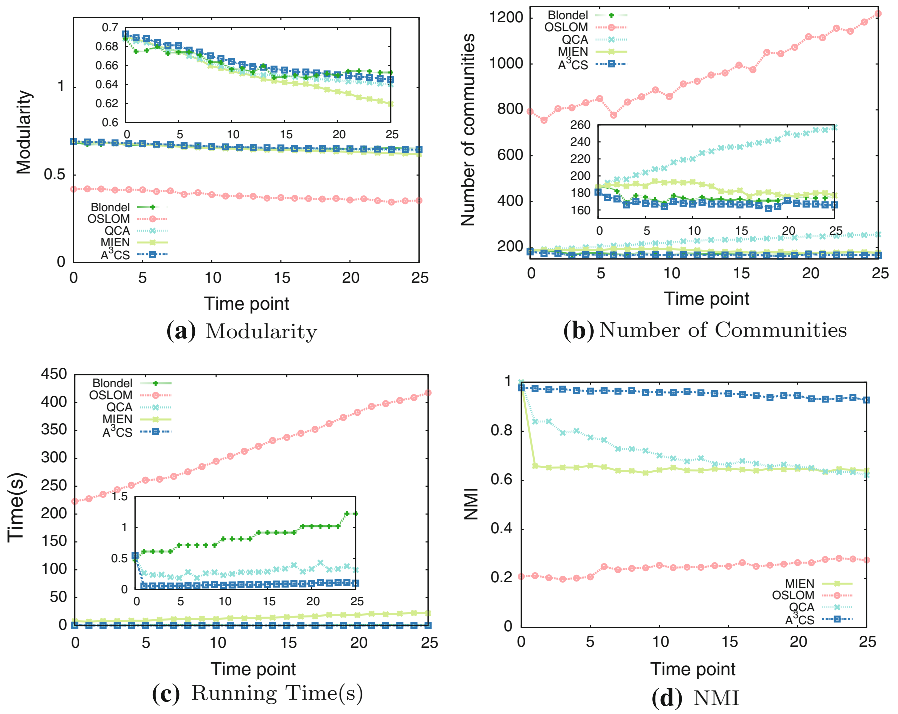
Fig. 3 Modularity values on synthesized networks with known communities

structures due to their high modularities, which was the main reason for them to be chosen.

To construct the snapshots, time information in each network is first extracted and a portion of the network data (usually the first snapshot) is then collected to form the basic community structure. All adaptive methods take into account that basic structure and run on the network changes whereas the Blondel method is executed on the whole snapshot at each time point. In this experiment, FacetNet method does not appear to complete the tasks in a timely manner, and is thus excluded from the plots. We use the same data and settings as in our previous paper (Nguyen et al. 2011).

5.2.1 ArXiv e-print citation network

It reveals from Fig. 4a that in general modularities obtained by A^{3}CS are highly competitive, if not to say the highest, with those of QCA, MIEN and the static Blondel methods, while are far better than those obtained by OSLOM. In particular, the modularity values produced by A^{3}CS differ insignificantly from those attained by QCA (3–4 % better than QCA) while cover from 97 % up to 100 % that of Blondel method, are from 6 to 10 % higher than MIEN at the end of the network evolution and are at least 1.5× better than OSLOM.



(a) Modularity **(b)** Number of Communities **(c)** Running Time(s) **(d)** NMI

Fig. 4 Simulation results on ArXiv e-print citation network. *The inner charts* zoom in the values of A^3CS, QCA, MIEN, and Blondel when necessary

In this citation networks, the numbers of communities discovered by our A^{3}CS framework are relatively similar to those found out by Blondel method, while are much less than the number of communities detected by MIEN and QCA (Fig. 4b). This probably explains why the modularities of A^{3}CS and Blondel methods are fairly similar to each other. On the other hand, since MIEN and QCA share the same objective of optimizing modularity Q, their local adaptive procedures may over-partition the network into more smaller communities; however, they are potentially subcommunities of A^{3}CS and Blondel methods, which reason why they still attain high modularities. This quantity detected by OSLOM method, on the other hand take off with more than 1200 and thus, may not be optimized for high modularity values.

A second observation (Fig. 4c) shows that A^{3}CS and QCA outperform all other dynamic methods as well as the static Blondel method on the running time: they require as much as nothing to complete analyzing each network snapshot, while the Blondel asks for ≈1 s to complete updating the network structure. In this network, MIEN asks for more than ≈15s in average while OSLOM consumes a huge amount of time with at least 222s and at most 464s to complete its tasks. In addition, higher NMI scores of A^{3}CS than those of QCA, MIEN and especially OSLOM methods (Fig. 4d) imply that network communities identified by our approach are not only of high similarity to the ground truth but also more precise than that detected by other methods, while the computational cost and the running time are significantly reduced.

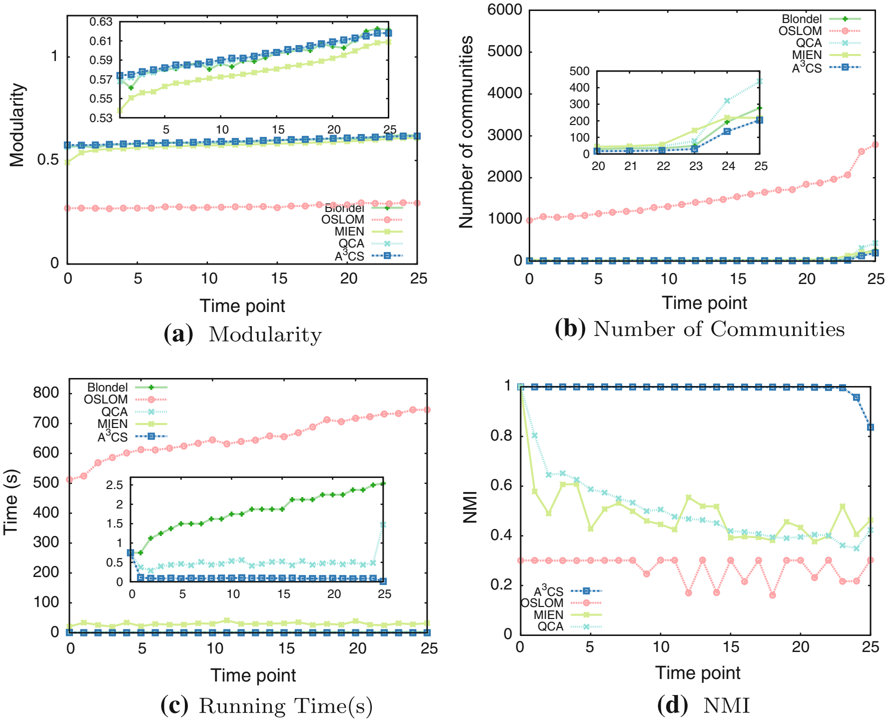


Fig. 5 Simulation results on Facebook social network. *The inner charts* zoom in the values of A^3CS, QCA, MIEN, and Blondel when necessary

5.2.2 Facebook social network

The evaluation depicted in Fig. 5a reveals that A^{3}CS achieves competitive modularity values in comparison with the Blondel and QCA methods, and again, far better than those obtained by MIEN and OSLOM methods, especially in comparison with OSLOM whose perform was nice on synthesized networks. In the general trend, the line representing A^{3}CS results closely approximates that of QCA and the static method Blondel with more stability. Moreover, the two final modularity values at the end of the experiment are relatively the same, which means that our adaptive method performs competitively with the static method running on the whole network, even at the very end of the network evolution.

Figure 5c describes the running time of the three methods on this Facebook data set. As one can see from this figure, A^{3}CS only takes its first second to analyze the basic structure and almost no time to update each network snapshot. In this larger network, QCA takes little more time to discover the network structures, however, it was not as significant as Blondel method which require around 2.4 s. MIEN, and especially OSLOM method, consume a huge amount of time to update the network structure as the network evolve. This reveal that, OSLOM, despite its nice outcomes on synthesized networks, may not be ideal for analyzing communities on a large network as Facebook data, and neither is MIEN.

In terms of similarity to the ground truth implied by the static method, A^3CS again shows it superiority when consistently achieved near perfect NMI scores even when the network has changed significantly over a long period of time. NMI score, obtained by other methods, tend to degrade significantly as the time point increase. This means a refreshment of network community structure is required for other adaptive methods after a long enough duration. While this refreshment is crucial to QCA, MIEN as well as OSLOM, A^3CS bypasses this need since it appear to be strongly stable over time, which is quite an advantage of an adaptive algorithm for detecting network community structure.

In conclusion, high NMI and modularity scores together with decent executing times on all test cases confirm the effectiveness of our A^3CS adaptive framework, especially when applied to real world networks where a centralized algorithm, or other dynamic algorithms, may not be able to detect a good network community structure in a timely manner.

6 Conclusion

In this paper, we propose the first adaptive approximation algorithms for the finding community structure problem in dynamic scale-free networks. In scale-free networks with $\gamma > 2$, our algorithm guarantees a $\left(\frac{\zeta(\gamma)}{\zeta(\gamma-1)} - \epsilon\right)$-adaptive approximation factor. Our algorithm is simple yet efficient and can detect high quality community structure in a very short amount of time. Due to the simplicity, the algorithm can also be extended to work for directed networks and can be implemented in a distributed manner. One of our future work is to analyze the algorithm under the light of other community structure measurements.

Acknowledgments This work is partially supported by NSF CAREER AWARD 0953284 and HDTRA-1-10-1-0050.

References

Agarwal G, Kempe D (2008) Modularity-maximizing graph communities via mathematical programming. Eur Phys J B 66:409–418

Aiello W, Chung F, Lu L (2000) A random graph model for massive graphs. In: STOC '00. ACM, New York, NY, USA

Aiello W, Chung F, Lu L (2001) Random evolution in massive graphs. In: Handbook of massive data sets. Kluwer Academic Publishers, Norwell

Albert R, Jeong H, Barabasi A (2000) Error and attack tolerance of complex networks. Nature 406:378–482

Bansal N, Blum A, Chawla S (2002) Correlation clustering. In: Annual IEEE symposium on foundations of computer science (FOCS), vol 0, p 238. doi:10.1109/SFCS.2002.1181947

Barabasi A, Albert R, Jeong H (2000) Scale-free characteristics of random networks: the topology of the world-wide web. Phys A 281:69–77

Barabasi AL, Jeong H, Nda Z, Ravasz E, Schubert A, Vicsek T (2002) Evolution of the social network of scientific collaborations. Phys A 311:590–614

Bianconi G, Barabasi AL (2001) Competition and multiscaling in evolving networks. EPL (Europhysics Letters) 54(4):436. http://stacks.iop.org/0295-5075/54/i=4/a=436

Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. J Stat Mech Theory Exp 2008(10):P10008

Brandes U, Delling D, Gaertler M, Gorke R, Hoefer M, Nikoloski Z, Wagner D (2008) On modularity clustering. IEEE Trans Knowl Data Eng 20(2):172–188

Clauset A, Newman MEJ, Moore C (2004) Finding community structure in very large networks. Phys Rev E 70(6):066111

DasGupta, B, Desai D (2012) On the complexity of newman's community finding approach for biological and social networks. J Comput Syst Sci 79(1):50–67. doi:10.1016/j.jcss.2012.04.003

Data A (2003) www.cs.cornell.edu/projects/kddcup/datasets.html. KDD Cup 2003

Dinh TN, Thai MT (2011) Finding community structure with performance guarantees in scale-free networks. In: SocialCom/PASSAT, pp 888–891

Dinh TN, Xuan Y, Thai MT (2009) Towards social-aware routing in dynamic communication networks. IPCCC

Faloutsos M, Faloutsos P, Faloutsos C (1999) On power-law relationships of the internet topology. In: Proceedings of the conference on applications, technologies, architectures, and protocols for computer communication, SIGCOMM '99, pp 251–262. ACM, New York, NY, USA. doi:10.1145/316188.316229

Ferrante A (2006) Hardness and approximation algorithms of some graph problems

Fortunato S, Barthelemy M (2007) Resolution limit in community detection. Proc Natl Acad Sci USA 104(1):36–41

Giotis I, Guruswami V (2006) Correlation clustering with a fixed number of clusters. Theory Comput 2(1):249–266. doi:10.4086/toc.2006.v002a013

Good BH, de Montjoye YA, Clauset A (2010) Performance of modularity maximization in practical contexts. Phys Rev E 81, 046,106. doi:10.1103/PhysRevE.81.046106

Hui P, Crowcroft J, Yoneki E (2011) Bubble rap: social-based forwarding in delay-tolerant networks. IEEE Trans Mobile Comput 10(11):1576–1589. doi:10.1109/TMC.2010.246

Lancichinetti A, Fortunato S (2009) Community detection algorithms: a comparative analysis. Phys Rev E 80(5), 056117. doi:10.1103/PhysRevE.80.056117

Lancichinetti A, Radicchi F, Ramasco JJ, Fortunato S (2011) Finding statistically significant communities in networks. PLoS ONE 6, e17249

Lin Y, Chi Y, Zhu S, Sundaram H, Tseng BL, Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. WWW (2008)

Newman MEJ (2003) The structure and function of complex networks. SIAM Rev 45(2):167–256

Newman MEJ (2006) Modularity and community structure in networks. Proc Natl Acad Sci USA 103(23):8577–8582

Nguyen N, Dinh T, Xuan Y, Thai M (2011) Adaptive algorithms for detecting community structure in dynamic social networks. In: INFOCOM, 2011 Proceedings IEEE, pp 2282–2290. doi:10.1109/INFCOM.2011.5935045

Noack A (2009) Modularity clustering is force-directed layout. Phys Rev E 79, 026,102. doi:10.1103/PhysRevE.79.026102

Pásztor B, Mottola L, Mascolo C, Picco G, Ellwood S, Macdonald D (2010) Selective reprogramming of mobile sensor networks through social community detection. In: Proceedings of EWSN, vol 5970, pp 178–193. Springer, Berlin

Tantipathananandh C, Berger-Wolf T (2009) Constant-factor approximation algorithms for identifying dynamic communities. In: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '09. ACM, New York, NY, USA, pp 827–836. doi:10.1145/1557019.1557110.

Viswanath B, Mislove A, Cha M, Gummadi KP (2009) On the evolution of user interaction in facebook. In: 2nd ACM SIGCOMM Workshop on Social Networks

Yu H, Kaminsky M, Gibbons PB, Flaxman A (2006) Sybilguard: defending against sybil attacks via social networks. In: Proceedings of the ACM SIGCOMM 2006 conference, SIGCOMM '06, pp 267–278. ACM, New York, NY, USA. doi:10.1145/1159913.1159945.

Zhu Z, Cao G, Zhu S, Ranjan S, Nucci A (2009) A social network based patching scheme for worm containment in cellular networks. In: INFOCOM 2009, IEEE, pp 1476–1484. doi:10.1109/INFCOM.2009.5062064.