# An Adaptive Approximation Algorithm for Community Detection in Dynamic Scale-free Networks

Thang N. Dinh, Nam P. Nguyen, and My T. Thai
Dept. of Comp. & Info. Sci. & Eng.
University of Florida, Gainesville, Florida 32611
Email: {tdinh, nnguyen, mythai}@cise.ufl.edu

***Abstract*—We introduce A$^3$CS, an adaptive framework with approximation guarantees for quickly identifying community structure in dynamic networks via maximizing Modularity Q. Our framework explores the advantages of power-law distribution property, 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 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 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.**

***Index Terms*—Adaptive approximation algorithm, Community structure, Modularity, Social networks;**

## I. Introduction

Many complex networks 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 [15], the power-law degree distribution, i.e., the fraction of nodes having degree $k$ is proportional to $k^{-\gamma}$ where $\gamma$ is the exponent parameter [2][3], 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. Community detection, as a result, is the classification of network nodes into communities so that the network's natural structure is properly displayed. 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 [8], [17], [11], or sybil defense [19], [18].

Complex networks in reality, such as OSNs, 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.

Several approaches have been suggested for community structure detection in dynamic networks [17][13][14][7]. However, they encounter the following crucial limitations: (1) their execution time is not proportional to only the network changes. 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 [17]. In addition, most of these methods do not take advantage of power-law distribution, which also is a common property of complex networks.

In this paper, we propose A$^3$CS, an ***adaptive algorithm with approximation guarantee*** for quickly identifying community structure in OSNs via maximizing *Modularity* - a widely accepted measure in community detection field [12]. 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 $\rho \approx \frac{\zeta(\gamma)}{\zeta(\gamma-1)}$ when the network's power exponent $\gamma > 2$, which is the most popular scenario [9][3]. To the best of our knowledge, our proposed algorithm is the first approach that achieves approximation guarantees for the NP-hard Modularity maximization problem [5], especially on dynamic 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 employment 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 [17], based on nonnegative matrix factorization [14], by compression of network modules [8] or by finding groups of nodes that have significant statistical features in the network [13]. However, designing adaptive algorithms that possess approximation ratios to guarantee their performance has not much been studied.

## II. Preliminaries

### A. 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 $\boldsymbol{A} = (A_{ij})$, 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 $k_i$.

### B. 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 $i^{th}$ community in the network, the *modularity* [16], denoted by $Q$, 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 0 otherwise. The modularity values can be either positive or negative and the higher modularity values indicate stronger community structure. The *maximizing modularity problem* asks to find a division which maximize the modularity value.

### C. 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.

### D. Adaptive Approximation Algorithm.

*Adaptive Algorithm.* In an adaptive algorithm, 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).

*Adaptive Approximation Algorithm.* In a $\rho$-*adaptive approximation algorithm*, the solution at any time point $t$ will be within a factor $\rho$ times the optimum objective value, provided the whole input from beginning to the time point $t$ available at once. For example, a $\rho$-adaptive approximation algorithm for the ACD problem will find at any time point $t$ a community structure with modularity at least $\rho Q^{(t)}_{opt}$, where $Q^{(t)}_{opt}$ is the maximum modularity of any community structure in $G^{(t)}$. The factor $\rho$ is called the *adaptive approximation factor*, and is less than one for maximization problems and greater than one for minimization problems.

## III. 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.

### A. 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 three cases a) $follow[i] = i$ if $label[i] = leader$ b) $follow[i] = j \neq i$ if $label[i] = follower$ & $i$ follows $j$, otherwise c) $follow[i] = label[i] = \emptyset$. 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 1:* 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**
1. $label[i] = \varnothing, follow[i] = \varnothing \ \forall i = 1..n$
2. Sorted nodes in non-decreasing order of degree.
3. **for each** vertex $i$ with $k_i \leq d_0$ **do**
4. **if** $label[i] = \varnothing$ **then**
5. FOLLOW_NEIGHBOR(i)
6. 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 1 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 to 2), the algorithm checks all nodes that appear in $\Delta G^{(t)}$ and corrects all possible "mis-labeling" caused by the degree changes. Two cases of "mis-labeling" are 1) low-degree but unlabeled nodes resulted from removing edges (or adding new nodes), and 2) follower nodes with degree higher than $d_0$ resulted from 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**( $i$ )
1. $label[i] = follower$
2. **if** $\exists j \in N(i) : label[j] \neq follower$ **then**
3. **if** $label[j] = \varnothing$ **then** $label[j] = leader$
4. $follow[i] = j$
5. **else**
6. Select a random $j \in N(i)$
7. UNFOLLOW( j )
8. $follow[i] = j, label[j] = leader$
9. 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 1. 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 to 8).

**Algorithm 4b. UNFOLLOW**( $i$ )
1. Let $j = follow[i], label[i] = \varnothing$
2. **if** $j$ has no followers **then**
3. **if** $k_j \leq d_0$ **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 1). We handle this case by either unlabeling $j$ or swapping $i$ and $j$'s labels together with making $j$ follow $i$ (lines 3 to 6).

### *B. Time complexity*

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 below.

*Lemma 2:* 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. Thus we have the following time complexity.

*Lemma 3:* 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$^3$CS, 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 very large networks of billion of nodes/edges in which the changes only happen within a small local part of the network. In comparison, existing methods for dynamic community structure [14], [13] involve at least $\Omega(|V^{(t)}| + |E^{(t)}|)$ time complexities whenever the community structure need to be updated. Hence they are far more time-consuming than an adaptive algorithm such as *A$^3$CS* and QCA [17].

### *C. Parameter Selection & Further Optimization*

*1) Automatic selection of $d_0$:* Selecting parameter $d_0$ is an important part of A$^3$CS. For the analysis of the adaptive approximation ratio in subsection III-A, it is sufficient to select $d_0$ as a large constant that relies only on $\gamma$. 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 4:* Automatic selection of the best $d_0$ can be done in $O(|V^{(0)}| + |E^{(0)})|)$.

*2) Further Optimization:* We can further optimize the A$^3$CS algorithm without changing its properties stated in Lemma 1. 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 [4], [8]. We then apply local search [16] on the abstracted network to increase the overall modularity.

### D. Performance Guarantee in Power-law Networks

In this subsection we consider power-law networks with the power exponents $\gamma > 2$. This network class covers a wide range of scale-free networks of interest, since the power exponents $\gamma$ in known scale-free networks typically satisfy $2 < \gamma < 3$ [3][1][9]. The performance of the A$^3$CS algorithm in networks with power-law degree sequences is stated in the following theorem.

*Theorem 1:* For scale-free networks with $\gamma > 2$, the modularity value of community structure $C^{(t)}$ at time point $t$, discovered by A$^3$CS is at least $\frac{\zeta(\gamma)}{\zeta(\gamma-1)} - \epsilon$. Thus, A$^3$CS 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.

According to Newman and Girvan [10], modularity values between $0.3$ and $0.7$ indicate a strong community structure and higher values are rare. For scale-free networks with $\gamma > \gamma_0 \approx 2.23$, by Theorem 1, the modularity value is at least $0.3$. 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 a $\frac{1}{2}$-adaptive approximation algorithm in that case.

**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 follow a power-law distribution, we can obtain the 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 algorithm can be easy implemented in a *distributed* manner which is a huge advantage for ad-hoc networks.

## IV. 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 [6] and Facebook social networks [18]. 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.* [17], (2) FacetNet algorithm proposed by Lin *et al.* [14], (3) MIEN algorithm proposed by Dinh *et al.* [8], and (4) OSLOM method suggested by Lancichinetti *et al.* [13].

### A. Results on synthesized networks

**Setup.** We use the well-known LFR benchmark [12] 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)* [12]. We want to demonstrate (1) quality of communities detected by A$^3$CS (and other methods) through NMI scores, and (2) the modularity values achieved by A$^3$CS in comparison with those of the ground truths.

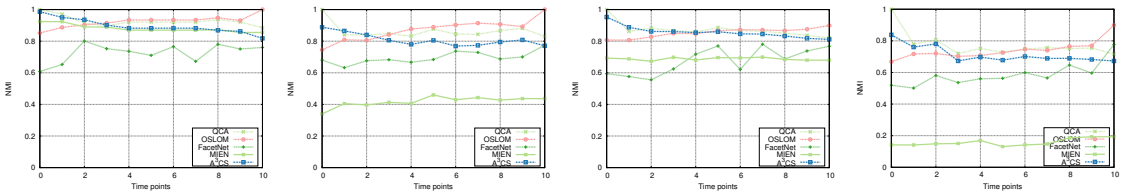
**Results.** The NMI and Modularity values are reported in Figs. 1 and 2, respectively. NMI and modularity values obtained by A$^3$CS, in general, are very high and competitive with those of OSLOM and QCA, while are much better than those produced by MIEN and FacetNet methods. In average, NMI scores achieved by A$^3$CS are only about 5% 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 subfigures 1b and 1d). The NMI scores of our framework, on the other hand, appear to stay wealthy, even when $\mu = 0.3$. This implies that network communities revealed by our A$^3$CS framework are highly similar to those contained in the ground truths, and also are highly competitively with those obtained by other methods.

In terms of modularity, the values obtained by A$^3$CS, QCA and OSLOM methods are very similar to each other and differ insignificantly from those of the ground truths, 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$^3$CS tend to tangle with those of QCA while NMI scores are just about 2%-3% less than the ground truths. Note that the good behaviors of OSLOM and QCA are not 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.

### B. Results on real-world traces

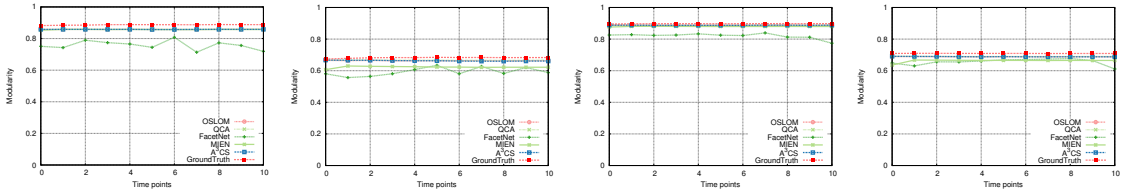
We next present the results of A$^3$CS algorithms on real world dynamic traces including arXiv e-print citation [6], and Facebook social networks [18]. Due to the lack of community ground truths corresponding to these traces, we report the performance of the aforementioned algorithms in reference to the static method proposed by Blondel *et al.* [4], whose goal also aims to optimize $Q$ and whose performance has been verified in the literature [12]. The network snapshots are constructed following [17]. 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.

*1) Modularity:* It reveals from Fig. 3a and 4a that in general modularity values 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



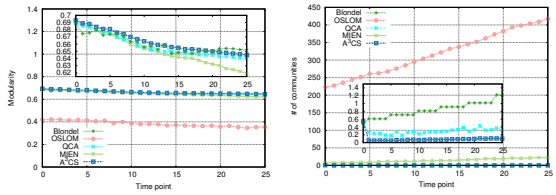

(a) $N = 1000, \mu = 0.1$ (b) $N = 1000, \mu = 0.3$ (c) $N = 5000, \mu = 0.1$ (d) $N = 5000, \mu = 0.3$

Fig. 1: Normalized Mutual Information (NMI) scores on synthesized networks with known communities



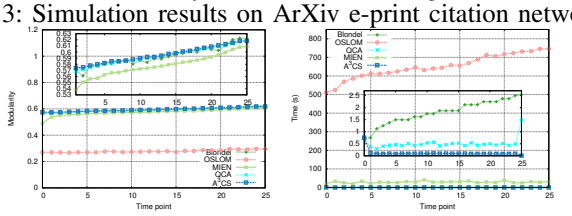

(a) $N = 1000, \mu = 0.1$ (b) $N = 1000, \mu = 0.3$ (c) $N = 5000, \mu = 0.1$ (d) $N = 5000, \mu = 0.3$

Fig. 2: Modularity values on synthesized networks with known communities




(a) Modularity (b) Running Time(s)

Fig. 3: Simulation results on ArXiv e-print citation network.




(a) Modularity (b) Running Time(s)

Fig. 4: Simulation results on Facebook social network.

by OSLOM. In particular, the modularity values produced by A$^3$CS are around 97% to 100% those of Blondel method, and from 6% to 10% higher than those of MIEN, and are up to 1.5x better than OSLOM.

*2) Running time:* Figures 4b and 3b describe the running time of the three methods on the arXiv and Facebook data sets. As shown in these figures, A$^3$CS outperforms all other dynamic methods as well as the static Blondel method on the running time: it requires as much as nothing to complete analyzing each network snapshot. MIEN and OSLOM methods consume a huge amount of time to update the network structure as the network evolve. This reveals that, OSLOM may not be ideal for analyzing communities on large networks such as Facebook.

In conclusion, high NMI and modularity scores together with extremely fast executing times on all test cases confirm the effectiveness of our A$^3$CS algorithm, 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.

## V. ACKNOWLEDGEMENT

This work is partially supported by NSF CAREER 0953284.

## REFERENCES

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

[2] 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.

[3] 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.

[4] 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.

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

[6] A. data, "www.cs.cornell.edu/projects/kddcup/datasets.html," *KDD Cup 2003*, Feb 2003.

[7] T. N. Dinh, I. Shin, N. K. Thai, M. T. Thai, and T. Znati, "A general approach for modules identification in evolving networks," in *Dynamics of Information Systems*, M. J. Hirsch, P. M. Pardalos, and R. Murphey, Eds. Springer New York, 2010, vol. 40, pp. 83–100.

[8] T. N. Dinh, Y. Xuan, and M. T. Thai, "Towards social-aware routing in dynamic communication networks," *IPCCC*, 2009.

[9] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," ser. SIGCOMM '99, '99, pp. 251–262.

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

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

[12] A. Lancichinetti and S. Fortunato, "Community detection algorithms: A comparative analysis," *Phys. Rev. E*, vol. 80, 2009.

[13] A. Lancichinetti, F. Radicchi, J. J. Ramasco, and S. Fortunato, "Finding statistically significant communities in networks," *PLoS ONE 6*, 2011.

[14] Y. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng, "Facetnet: a framework for analyzing communities and their evolutions in dynamic networks," *WWW*, 08.

[15] M. E. J. Newman, "The Structure and Function of Complex Networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.

[16] ——, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, 2006.

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

[18] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in facebook," in *2nd ACM SIGCOMM Workshop on Social Networks*, 2009.

[19] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman, "Sybilguard: defending against sybil attacks via social networks," ser. SIGCOMM '06. New York, NY, USA: ACM, 2006, pp. 267–278.