Structural Vulnerability Analysis of Overlapping Communities in Complex Networks

Md Abdul Alim[‡*], Nam P. Nguyen[†*], Thang N. Dinh[◇] and My T. Thai[‡]
[‡]Department of Computer & Information Science & Engineering, University of Florida, USA.
[†]Department of Computer & Information Sciences, Towson University, USA.
[◇]Department of Computer Science, Virginia Commonwealth University, USA.
[*]: Co-first authors. Email: {alim, mythai}@cise.ufl.edu, npnguyen@towson.edu, tndinh@vcu.edu.

***Abstract*—Many complex networks commonly exhibit community structure in their underlying organizations, i.e., they contain multiple groups of nodes having more connections inside a group and less interactions among groups. This special structure not only offers key insights into understanding the network organization principles but also plays a vital role in maintaining the normal function of the whole system. As a result, any significant change to the network communities, due to element-wise failures, can potentially redefine their organizational structures and consequently lead to the malfunction or undesirable corruption of the entire system. Therefore, identifying network elements that are essential to its community structure is a fundamental and important problem. However, to the best of our knowledge, this research direction has not received much been attention in the literature.**

In this paper, we study the structural vulnerability of *overlapping* complex network communities to identify nodes that are important in maintaining the complex structure organization. Specifically, given a network and a budget of k nodes, we want to identify k critical nodes whose their exclusions transforms the current network community structure. To effectively analyze this vulnerability on overlapping communities, we propose the concept of *generating edges* and provide an optimal algorithm for detecting the Minimal Generating Edge Set (MGES) in a network community. We suggest *genEdge*, an effective solution based on this MGES. Empirical results on both synthesized networks with known community structures, and real data including Reality cellular data, Foursquare and Facebook social traces confirm the efficacy of our approach.

I. Introduction

Many complex networks in reality, such as transportation, communication and social networks, expose to be extremely vulnerable under element-wise attacks. In some scenarios, the failures of only a few key nodes are enough to bring the whole network operation down to its knees [1]. This vulnerability can further be propagated to a wider population, leading to a crippling consequence. Therefore, understanding both the impact of element failures on the network structure and also the inner and interdependency among those structural components is important. It is of particular interest to explore how the non-participation of a single node, or a collection of nodes in general, can significantly change the structure of the network components as well as how these components would affect each other in cases of failures. However, the large scale and dynamical property of networks in practice make the assessment of this structural vulnerability a fundamental yet challenging issue [2].

In this paper, we investigate how the failures of crucial nodes in the network will affect its *overlapping* community structure. We are interested in identifying nodes whose removals trigger a significant reorganization of the overlapped network community structure. Formally, given the input network and a positive number k, want to find out a set S of k nodes whose removal maximally transforms the current network community structure to a totally different one, i.e., the new community structure resulted from the removal of S is of least similarity to the original one, evaluated via the Normalized Mutual Information [3] measure.

In an application perspective, the awareness of community structure vulnerability is extremely beneficial, especially for social-aware methods in mobile ad-hoc (MANETs) and online social networks (OSNs). To give a sense of its effects, consider message forwarding protocols in MANETs. Because social-based forwarding strategies in MANETs rely on the nodes in each community to forward the message [4][5], the awareness of this vulnerability can help to either design routing protocols that do not overload those crucial nodes, or to design an effective backup plan when some of them may fail at the same time. In worm containment application in OSNs [6][7], this knowledge can provide helpful insights into the protection of those sensitive nodes, if they are indeed high influential users, once worms spread out in the network. The identification of nodes whose removal triggers a massive transformation of the social community structure, as a result, is extremely important for analyzing the social network structure, especially when network communities can overlap with each other.

In order to develop an effective solution for this challenging research topic, it is essential to understand the core behaviors of network communities when their nodes are excluded from the structure. The first work of Nguyen et al. [8] has suggested potential conditions for the transformation of the network community structure, and have proposed multiple heuristic algorithms based on the modularity contribution of communities in the network. These approaches, while are efficient in analyzing disjoint structures, face nonnegligible limitations when applied to real networks displaying overlapping community structures. As a result, the need for an effective algorithm that can assess the vulnerability of the general network structures is of desire. To this end, we present a mathematical analysis on possible conditions that can lead

to the maximal transformation of overlapping community structures. This is a more general analysis than in the case of disjoint community structure. Based on these findings, we define the concepts of the "generating edge" and the "Minimal Generating Edge Set" (MGES), and utilize these features to explore the key weakness of the network communities. We then propose *genEdge*, a node identification strategy built on top of the optimal solution of the MGES of each community.

In summary, our contributions in this work are: (1) We study the analysis of structural vulnerability to assess the impact of nodes' failures on the network community structure. (2) We analyze potential conditions that can lead to the minimization of NMI on overlapped network community structures. Based on that, we suggest the concept of *generating edges* of a community and provide an optimal solution for finding a MGES. We propose *genEdge*, a node selection strategy based on the MGES solution; (3) We conducted experiments on both synthesized data with known community structures and real world traces. Empirical results reveal that *genEdge* outperforms other node selection strategies in terms of solution quality as well as in reference to different underlying community detection algorithms.

(Organization): Section II presents preliminaries on graph notations, the concept of NMI and the problem formulation. The analysis of NMI measure is presented in section III. Section IV discusses our node selection strategy. Sections V and VI discuss the experimental results and the work related to our study. Finally, we conclude our paper in section VII.

II. Problem formulation

In this section, we first define the graph notations that will be used thoroughly in this paper. We then describe *Normalized Mutual Information (NMI)* [3], a concept in Information Theory, as a metric to assess the difference between community structures before and after the removal of important nodes. Finally, we revisit the Community Structure Vulnerability problem [8] on overlapping community structures.

For the sake of simplicity, let $G = (V, E)$ be an undirected unweighted graph representing a network where V is the set of $|V| = N$ nodes (e.g., users), and E is the set of $|E| = M$ links (e.g., friendships). For any node $u \in V$ and a set $C \subseteq V$, let $N(u), d_u$ and d_u^C be the set of all neighbors of u, its degree in G and its degree in C, respectively. Let $n_C = |C|$ be the number of nodes and m_C be the number of edges in C.

Denote by $\mathcal{A}$ the specific community detection algorithm that will be applied on G, and by $X = \{X_1, X_2, ..., X_{c_X}\}, Y = \{Y_1, Y_2, ..., Y_{c_Y}\}$ the two (possibly overlapped) community structures of c_X and c_Y communities detected by $\mathcal{A}$ *before* and *after* the removal of a set S of k nodes in G, respectively. Mathematically, X and Y are represented as $X = \mathcal{A}(G)$ and $Y = \mathcal{A}(G[V \backslash S])$, where $G[V \backslash S]$ is the subgraph induced by $V \backslash S$ on G. For any index $i = 1, ..., c_X$ and $j = 1, ..., c_Y$, let $x_i = |X_i|$, $y_j = |Y_j|$, and $n_{ij} = |X_i \cap Y_j|$. Finally, let $\bar{x} = \sum_{i=1}^{c_X} x_i$, $\bar{y} = \sum_{j=1}^{c_Y} y_j$ and $\bar{n} = \sum_{i=1}^{c_X} \sum_{j=1}^{c_Y} n_{ij}$ be the total size of communities in X and Y, and the total number of common nodes shared between X and Y, respectively.

In order to evaluate how much the network community structure changes before and after the removal of important nodes, we utilize the concept of Normalized Mutual Information suggested in [3]. Given two structures X and Y, $NMI(X, Y)$ is 1 if X and Y are identical and is 0 if X and Y are totally separated, and the higher the NMI score, the more similarity between X and Y. As a result, NMI is a well-suited metric dedicated for certifying the quality of community structures discovered by different detection algorithms. The effectiveness of this widely-accepted measure has also been extensively verified in the literature [9]. Formally, $NMI(X, Y)$ is defined as

$$NMI(X, Y) = \frac{2I(X, Y)}{H(X) + H(Y)},$$

where $H(X)$, $H(Y)$ and $I(X, Y)$ are the entropy of structures X and Y, and the Mutual Information conveyed between them, respectively. More details about NMI formulation will be elaborated in our analysis in section III.

Finally, the Structural Vulnerability Analysis of the Network Community structure is formulated as [8].

***Definition** 1:* Given a network represented by an undirected and unweighted graph G, a specific community detection algorithm $\mathcal{A}$, and a positive integer $k \leq N$, we seek for a subset $S \subseteq V$ such that

$$S = \underset{S' \subseteq V, |S'| = k}{\operatorname{argmin}} \{NMI(\mathcal{A}(G), \mathcal{A}(G[V \backslash S']))\}.$$

This formulation requires the community detection algorithm $\mathcal{A}$ as an input parameter. Because there is not yet a universal agreement or accepted definition of a network community, this input is necessary in the sense that different algorithms with different objective functions might favor different sets of nodes, and thus, a good solution set for one community detection algorithm may not be good for the others. However, when there is a clear objective function for finding community structure, such as maximizing Modularity Q [9] or the total internal density [5], this requirement can be lifted. Nevertheless, the node selection strategy that relies more on the input network and less on the community detection algorithm is always of desire.

III. Analysis of NMI measure

In this section, we investigate the possible conditions on sizes and the number of communities that can potentially lead to either the global or local minimization of $NMI(X, Y)$. We stress that these conditions are by no means universal or exhaustive since some of them might not hold true simultaneously, given the input parameters. Indeed, what we hope for is these conditions would provide us key insights into the selection of important nodes to maximally separate X and Y. In the coming paragraphs, we first discuss the NMI formulation in a greater detail, and then analyze it in terms of *overlapping* community structures.

A. NMI formulation

To find $NMI(X,Y)$ [3] where $X = \{X_1, X_2, ..., X_{c_X}\}$ and $Y = \{Y_1, Y_2, ..., Y_{c_Y}\}$, we start out by considering community assignments X_i and Y_j, where X_i and Y_j indicate the community labels of a node t in X and Y, respectively. Without loss of generality, we can also assume that the labels X_i and Y_j are also values of two random "variables" X and Y (here we reuse notations X and Y to denote the two random variables), with joint distribution $P(X_i, Y_j) = P(X = X_i; Y = Y_i) = n_{ij}/(N-k)$, and individual distribution $P(X_i) = P(X = X_i) = x_i/N$, $P(Y_j) = P(Y = Y_j) = y_j/(N-k)$.

The entropy of X and Y is defined as [10] $H(X) = -\sum_{i=1}^{c_X} \frac{x_i}{N} \log \frac{x_i}{N}$, and $H(Y) = \frac{1}{N-k}\big(\bar{y}\log(N-k) - \sum_{j=1}^{c_Y} y_j \log y_j\big)$. Note that X can be derived straightforwardly based on $\mathcal{A}$ and G, and thus, quantities x_i's can also be inferred from these input parameters. Therefore, we simply consider x_i's and $H(X)$ as constants in this paper.

The Mutual Information $I(X,Y)$ [10] of X and Y is $I(X,Y) = \sum_{i=1}^{c_X}\sum_{j=1}^{c_Y} \frac{n_{ij}}{(N-k)} \log \frac{Nn_{ij}}{x_i y_j}$. This measure is symmetric and it tells us how much we know about variable (or structure) Y if we already know about variable X, and vice versa. However, as indicated in [3][9], Mutual Information itself is not ideal as a global similarity metric since any subpartition of a given community structure X would result in the same mutual information with X, even though they can possibly be very different from each other. As a result, [3] introduces the Normalized Mutual Information which can overcome that limitation. Formally, NMI of X and Y is defined as $NMI(X,Y) = \frac{2I(X,Y)}{H(X)+H(Y)}$, or equivalently,

$$\frac{2\sum_{i=1}^{c_X}\sum_{j=1}^{c_Y} n_{ij} \log \frac{Nn_{ij}}{x_i y_j}}{(N-k)H(X) + \bar{y}\log(N-k) - \sum_{j=1}^{c_Y} y_j \log y_j}. \quad (1)$$

B. Minimizing NMI in an overlapped community structure

In comparison with the disjoint structure, the minimization of $NMI(X,Y)$ is much more complicated when network communities of X and Y can overlap with each other. In particular, the conditions $\cup_{i=1}^{c_X} X_i = V$ and $\cup_{j=1}^{c_Y} Y_j = V \backslash S$ still hold in this case; however, $X_i \cap X_{i'}$ and $Y_j \cap Y_{j'}$ might not be empty for some $i, i' = 1, ..., c_X$ and $j, j' = 1, ..., c_Y$. These facts indicate that $\bar{x} \geq N, \bar{y} \geq N-k$ and $\bar{n} \geq N-k$.

Our analysis strategy in this case is similar to the prior one as we also strive for maximizing the denominator while minimizing the numerator of $NMI(X,Y)$ (eq. 1). Because $\bar{n} \geq N-k$, the minimization of the top term $I(X,Y)$ no longer depends only on x_i's. One way to work around this issue is to investigate the relative correlation between the total community size $\bar{y}$ and the number of communities c_Y. Let $\alpha_{\mathcal{A}} = \frac{\bar{y}}{c_Y}$ be the ratio between these two quantities, or in other words, the averaged community size. The denominator of $NMI(X,Y)$ is evaluated as

$$\bar{y}\log(N-k) - \sum_{j=1}^{c_Y} y_j \log y_j \leq \bar{y}\log(N-k) - \alpha_{\mathcal{A}} \log \alpha_{\mathcal{A}},$$

with equality holds when all y_j's are equal to each other. To further maximize this denominator, we need $\bar{y}$ to be as large as possible while keeping $\alpha_{\mathcal{A}}$ as small as possible, i.e., the new community structure Y should contain more and more communities as to increase c_Y as well as to lower down $\alpha_{\mathcal{A}}$.

Due to the dependence on the specific detection algorithm $\mathcal{A}$, this optimization on the correlation between $\bar{y}$ and c_Y might not be globally achieved. However, a coarse analysis between $\bar{y}$ and c_Y can relatively be conducted in the following senses: if we assume that $\bar{y}$ is within a constant factor of the total number of actual nodes $(N-k)$, i.e., $\bar{y} \leq a_0(N-k)$ for some constant $a_0 > 1$, we can then increase the value of the RHS by breaking as many communities as possible while keeping them having the size (i.e., enlarge c_Y and keep y_j's are all the same), which helps to reduce the impact of $\alpha_{\mathcal{A}} \log \alpha_{\mathcal{A}}$. This observation, though relative, agrees with what we achieved in the case of disjoint community structure. In an unfortunate case where $\bar{y}$ is not known to be within any constant factor of $(N-k)$, the observation might not hold since both $\bar{y}$ and c_Y can be arbitrary large and thus, $\alpha_{\mathcal{A}} \log \alpha_{\mathcal{A}}$ could still be relatively small. Next, applying Log Sum Theorem [10] on the numerator yields

$$I(X,Y) = \sum_{ij} n_{ij} \log \frac{Nn_{ij}}{x_i y_j} \geq \bar{n} \log \frac{N\bar{n}}{\bar{x}\bar{y}},$$

with equality holds when $\frac{Nn_{ij}}{x_i y_j}$ is a constant for all $i = 1, ..., c_X$ and $j = 1, ..., c_Y$. Thus, one can try to minimize $I(X,Y)$ by deleting nodes in such a way that $\bar{n}$ is maximized and $\bar{y}$ is minimized while making sure that $\frac{Nn_{ij}}{x_i y_j}$ is a constant. As a result, this minimization of $I(X,Y)$ is a multiple-objective optimizations problem which may not have a feasible solution. However, if we assume that the later condition is imposed, i.e., $\frac{Nn_{ij}}{x_i y_j} = \beta_{\mathcal{A}}$ for some constant $\beta_{\mathcal{A}} > 0$, then $n_{ij} = \frac{\beta_{\mathcal{A}} x_i y_j}{N}$, and thus $\bar{n} = \frac{\beta_{\mathcal{A}}}{N}\bar{x}\bar{y}$. This reduces the above inequality to

$$I(X,Y) \geq \frac{\bar{x}}{N}\beta_{\mathcal{A}}\bar{y}\log \beta_{\mathcal{A}} N.$$

The right hand side of the inequality advises that, in order to minimize $I(X,Y)$, *the total size of network communities should not be too large while the overlapping ratio of every community should be equal to each other and be as small as possible*. These conditions make sense because such uniformly distributed community structure would not impose any huge-sized overlapped communities, and thus, provide almost minimum information prior community organization.

IV. Method

In the following paragraphs, we consider the scenario when maximizing the internal density [5] is the objective function for finding network communities, i.e., communities of G are assumed to have optimized internal densities. To this end, we present genEdeg, an algorithm for analyzing the structural vulnerability of complex networks that is independent of the underlying community detection algorithm $\mathcal{A}$. Our solution strategy will try to break larger communities to as many small

ones as possible while looking for them to have the relatively same size with small overlapping ratios.

A. Intuitions

The idea of our strategy is based on the following intuition: since communities in X are optimized for their internal density, they are likely to contain strong substructures that are tightly connected which form the cores of these communities. As a result, the removal of crucial nodes in a core might potentially break the community into smaller modules. Moreover, as nodes in a core are tightly connected, there should be some edge that generate them, i.e., all nodes in the core are incident to both endpoints of this edge. Inspired by this intuition, our strategy works towards the identification of these generating edges of a community, and then seek for the minimum set of generating edges that composes the original communities.

Let D be a subset of V. Denote by

$$\Psi(D) = \frac{2m_D}{n_D(n_D-1)}$$

the internal density of D and by

$$\tau(D) = \frac{n_D(n_D-1)}{2}^{-\frac{2}{n_D(n_D-1)}}$$

the threshold function on the internal density of D, respectively. For any nodes $u, v \in D$, if edge (u, v) is not in E, we call it a missing edge in D. In addition, we call an edge in D "negative" if it is incident to a missing edge in D, and "positive" otherwise. We define the concept of *generating edges* of D as follow

***Definition** 2:* (Generating edge) For any edge (u, v) in D, if $D = (D \cap N(u) \cap N(v)) \cup \{u, v\}$ and $\Psi(D) \geq \tau(D)$, we call (u, v) a generating edge of D. We further call D a local core generated by (u, v) and write $gen(u, v) = D$.

For any community C of G, a set $L \subseteq E$ is called a "generating edge set" of a C if $\cup_{(u,v)\in L} gen(u, v) = C$. Since C can be generated by different generating edge sets and we are constrained on the node budget, we would intuitively seek for the generating edge set of minimal cardinality.

***Definition** 3:* (Minimum Generating Edge Set) Given a community C of G, the MGES problem seeks for a generating edge set L^* of C with the smallest cardinality.

The cores generated by edges in a MGES of a community C of G are tightly connected and they all together compose C. As a result, if we delete an endpoint of every edge in a MGES, C will be broken into smaller modules with the number of modules is at least the number of edges in a MGES (Lemma 1). Since our goal is to break the current community structure X into as many new communities as possible, the removal of crucial nodes defined by edges in a MGES will be a good heuristic for this purpose. But first and foremost, we need to characterize all MGESs in the current community structure X based only on the input network G. Lemma 2 realizes the location of the generating edge(s) of a local core in a community C: they have to be adjacent to nodes with the highest degree in C. Based on this result, we present in Alg. 1 a procedure that can correctly find the MGES of a given community C (Theorem 1).

Algorithm 1 An optimal algorithm for finding the MGES

Input: Network $G = (V, E)$ and a community $C \in X$;
Output: Minimum generating edge set L^* of C;

0. Mark all nodes as "unassigned" and $L^* = \emptyset$.
1. Remove all negative edges in C. If any edge(s) survive, they are candidate for generating edges in their corresponding communities, include them to L^*, go to step 2. Else, go to step 3.
2. Reconstruct local cores based on generating edges found in step 1. Mark all nodes in those communities as "assigned". Discard generating edges in L^* that fall into any newly constructed communities. Return if all edges are assigned.
3. Find the set U as in Lemma 2. Find the edge in $NE(U)$ that can generate a local community having the largest size. Include this edge to L^* and mark all nodes in the new local community as "assigned". Ties are broken randomly. Return if all edges are assigned.
4. If there are still unassigned nodes, say the set $I \subseteq C$, construct $G' = G[(I \cup N(I)) \cap C]$. Go to back to step 1.

***Lemma** 1:* Let L^* be a MGSE of a community C. The removal of an endpoint in every edge of L^* will break C into at least $|L^*|$ subcommunities.

Proof: Clearly, the removal of an endpoint of every edge in L^* will degrade the internal density of each core since the endpoint of the generating edge is of full degree in its core. Now, if the number of subcommunities resulted in the node removal is less than $|L^*|$, it means there are at least two cores that are merged together. That is there are cores c_1 and c_2 are merged together even with less internal density. This should not be the case since otherwise, they have to be identified as a single core at the first place. Their combination, as a result, implies that C has a MGES of size less than $|L^*|$, which raises a contradiction to the assumption that L^* is a MGES of C. ■

***Lemma** 2:* Let C be a subset of V, $U = \{u \in C | d_u^C$ is the highest in $C\}$ and $NE(U) = \{(u, v) | u \in U$ or $v \in U$ but not both $\}$. Then, $|NE(U) \cap L^*| \geq 1$.

Proof: After each refreshment in step 2, let u be the node with the highest indegree in C. After step 1 of Alg. 1, all negative edges are deleted since they do not contribute to the actual generating set L^*. As such, edges incident to u are not negative. This in turn implies that they are candidates for generating edges. Now, iterate through all edges incident to u and choose the one that generates the biggest-sized core. This edge should be in the list L^*. ■

***Theorem** 1:* Let d_C be the maximum in-degree of a node in C. Alg. 1 takes $O(d_C|C|)$ time in the worst case scenarios and returns an optimal solution for MGES problem.

Proof: Since every time Lemma 2 makes sure that at least one edge should be added to L^* and the procedure terminates when no edges left, the Alg. 1 should terminate. Moreover, it is

verifiable that Alg. 1 takes time as most the number of edges in C, which is $O(d_C|C|)$. Also, due to the intense internal density of a core, every time an edge is added into L^*, that edge actually generates the largest core possible. The proof follows from this fact, Lemma 2 and the exhaustive property of Alg. 1. ∎

Algorithm 2 $genEdge$ - A node selection strategy based on generating edges

Input: Network $G = (V, E)$, $X = \mathcal{A}(G)$;
Output: A set $S \subseteq V$ of k nodes;

1. Use Alg. 1 to find $L^*_{X_i}$ for all communities X_i's in X.
2. Sort all communities X_i's in X by their sizes of MGSEs.
3. Sort all nodes in G by the number of generating edges that they are incident to in X_i. If there is a tie, sort them by their degrees in G.
4. Return top k nodes in step 3.

With the optimal solution of MGES taken into account, we next suggest a heuristic for selecting important nodes following the guidelines suggested in section III. In particular, our heuristic Alg. 2 selects nodes in a greedy manner, starting from communities that have large-size MGESs. Moreover, in the MGES of each community C, we give priority to nodes that are incident to more generating edges since their removals will break C into more subcommunities.

V. Experimental results

In this section, we show the empirical results of our node selection strategy on both synthesized networks with known community structures and real-world social traces including the Reality mining cellular dataset [11], Facebook [12] and Foursquare [13] social networks. In order to certify the performance of our approach, we compare the results obtained by the following methods: *High degree centrality* ($highDeg$) selects top k nodes in G with the highest degrees, $betweenness$ *centrality* ($betweenness$) selects top k nodes in G with the highest $betweenness$es (where the $betweenness$ of a node u is the number of shortest paths in G that pass through u), *Generating edges* ($genEdge$) - our strategy described in Alg. 2, and finally, *Node Importance* ($nodeImp$) [14] selects top k nodes by their importance to the community structure.

We first examine the effect of the underlying community detection methods by comparing results obtained by $AFOCS$ [5], $Blondel$ [15] and $Oslom$ [16] algorithms to the embedded groundtruths. In particular, we set X to be the groundtruth community structure and when S is removed from the network $NMI(X, Y)$ is reported, where $Y = AFOCS(G[V \backslash S])$, $Y = Blondel(G[V \backslash S])$ and $Y = Oslom(G[V \backslash S])$, respectively. These methods have been empirically certified in the literature to the best algorithms for finding non-overlapping and overlapping community structure [9]. Verifying our strategy on synthesized networks not only certifies its performance but also provides us the confidence to its behaviors when applied to real-word traces.

A. Results on synthesized networks

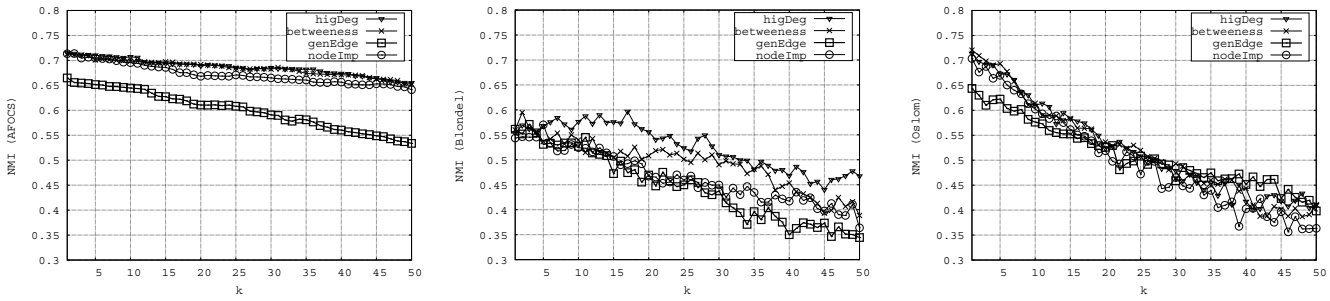
Of course, the best way to evaluate our approach is to validate them on real-world traces with known community structures. Unfortunately, we often do not know that structures beforehand, or such structures cannot be easily mined from the network topologies. Although synthesized networks might not reflect all the statistical properties of real ones, they can provide us the known ground-truths via planted communities and the ability to vary other network parameters such as sizes, densities and overlapping levels, etc.

Setup: We use the well-known LFR overlapping benchmark [9] to generate test networks. The number of nodes are $N = 2500$ and 5000, the mixing parameter $\mu = 0.15$, the community sizes $c_{min} = 10$ and $c_{max} = 50$ for $N = 2500$ and $c_{min} = 30$ and $c_{max} = 100$ for $N = 5000$. At every k nodes are removed from the network, the network community structure is reidentified and compared to the original embedded one (or the ground-truth). The overlapping threshold β in $AFOCS$ is set at 0.7 and all tests are averaged on 100 runs for consistency.

1) Solution quality: We first evaluate the performance of all aforementioned node selections strategies on different community detection algorithms $AFCOS$, $Blondel$ and $Oslom$, respectively. Because the ground-truth communities on synthesized networks are given a priori, comparisons through NMI scores among these strategies as well as among detection algorithms are therefore valid, and the lower NMI scores a strategy obtains, the more effective it seems to be. In addition, the higher the remaining NMI measure a detection algorithm obtains after the node removal, the more resistant to node vulnerability it seems to be.

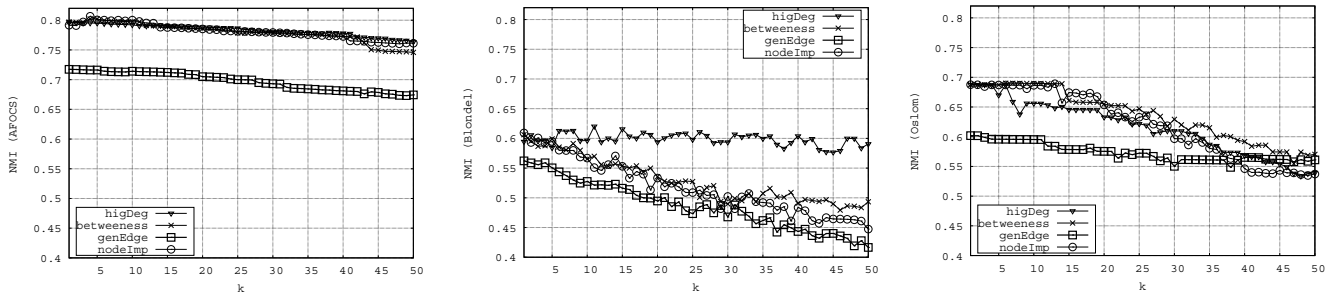
The quality of node selection solutions, are reported in figures 1 and 2. In a general trend, NMI scores tend to drop down quickly as more nodes are removed from the network when $N = 2500$; however, they degrade much slower in networks with $N = 5000$. The first observation revealed in those figures is that our approach $genEdge$ achieves the best (lowest) NMI scores on almost all test cases. In average, on networks with 2500 nodes, $genEdge$ is 14% better than both $highDeg$ and $betweenness$, and is 12% better than $nodeImp$ on $AFOCS$ algorithm; and is 19%, 11% and 5% better than $highDeg$, $betweenness$, and $nodeImp$ on $Blondel$ algorithm (figure 1(a), 1(b)). On $Oslom$ algorithm, $genEdge$ differs insignificant with $highDeg$ and $betweenness$ with 1.5% and 1.4% better, and is only lagged behind $nodeImp$ with 3% lower NMI scores. On network with 5000 nodes, $genEdge$ still outperforms other strategies with 12% lower NMI scores than the others on $AFOCS$ algorithm, and with 23%, 8% and 6% lower NMI scores than $highDeg$, $betweenness$ and $nodeImp$ on $Blondel$ algorithm, and finally, with 7%, 10% and 8% better than the others on $Oslom$ algorithm (figure 2). These results imply that $genEdge$ performs excellently with competitive results on different community detection algorithm in comparison with other strategies.

The second observation we obtain from figures 1 and 2



(a) NMI scores by $AFOCS$ (b) NMI scores by $Blondel$ (c) NMI scores by $Oslom$

Fig. 1. Comparison among different node selection strategies on synthesized networks with $N = 2500$ nodes



(a) NMI scores on $AFOCS$ (b) NMI scores on $Blondel$ (c) NMI scores on $Oslom$

Fig. 2. Comparison among different node selection strategies on synthesized networks with $N = 5000$ nodes

is that the top-of-the-list node seems to be essential to the network community structure. The removal of only this node from the network brings the NMI scores to as low as 0.7 - 0.8 on $AFOCS$ (figure 1(a), 2(a)), to 0.58 - 0.6 on $Blondel$ algorithm (figure 1(b), 2(b)), and to 0.7 on $Oslom$ algorithm. Furthermore, the top 15-20 nodes are also vital to the network community structure detected by $Oslom$ and $Blondel$ since their destruction brings the NMI scores down to 0.5, the threshold where the community structure become stochastic and fuzzy to recognize. The NMI values on $AFOCS$ algorithm, on the other hand, do not suffer from this destruction as they only come close to 0.5 when almost $k = 50$ nodes are removed from the networks with $N = 2500$ nodes (figure 1(a)).

Finally, the last observation inferred from figures 1 and 2 is that, among the three community detection algorithms, $AFOCS$ algorithm obtains the highest remaining NMI values when the same number of nodes is removed from the networks. In other words, $AFOCS$ was able to detect the community structure which was of the most similarity to the ground-truth communities. As we discussed above, this observation implies that $AFOCS$ seems to be the detection algorithm which is more resilient to node vulnerability than the other algorithms. Therefore, we employ $AFOCS$ as the main community detection algorithm to further analyze network communities of real-world traces.

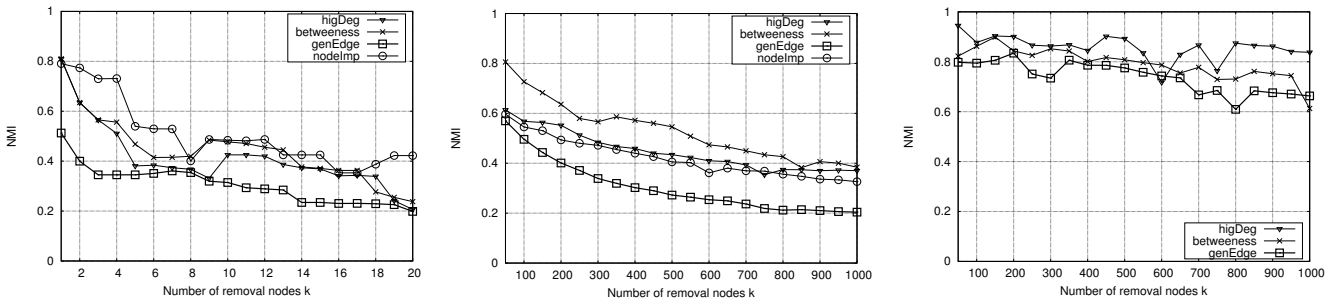
TABLE I
STATISTIC OF SOCIAL TRACES

Data	N	M	Avg. Deg	Max. Com. Size
Reality	100	3100	62	35
Facebook	63731	1.5M	23.50	33425
Foursquare	47260	1.1M	49.13	30381

B. Results on real-world traces

We further present the empirical results on real-world networks including Reality mobile phone data [11], Facebook [12] and Foursquare [13] datasets. The overview of these datasets is summarized in Table I.

Reality Mining dataset provided by the MIT Media Lab. This dataset contains communication, proximity, location, call, and activity information from 100 students at MIT over the course of the 2004-2005 academic year. *Facebook* dataset contains friendship information (i.e., who is friend with whom and wall posts) among New Orleans regional network on Facebook, spanning from Sep 2006 to Jan 2009. To collect the information, the authors created several Facebook accounts, joined each to the regional network, started crawling from a single user and visited all friends in a breath-first-search fashion. *Foursquare* dataset contains location and activities of 47260 users on Foursquare social network on May 2011 - Jul 2011. To collect the data, we created several Foursquare accounts, joined to the network, started crawling from a single user and visited all friends also in a breadth-first-search fashion.



(a) Reality (b) Foursquare (c) Facebook

Fig. 3. NMI scores on Reality mining data, Foursquare and Facebook networks obtained by $AFOCS$ ($k = 50...1000$)

On Reality Mining dataset, we set $k = 1...20$ and report result in figure 3(a). It reveals from this figure that community structure in this dataset is extremely vulnerable to node attacks since the removal of only 2 nodes, found by $genEdge$ is enough to make the new community structure significantly differ from the original one as it brings down the NMI values to 0.4. In comparison with other node selection methods, $genEdge$ still performs excellently and is about 14% - 17% better than the others. We note that the first node identified by $genEdge$ is indeed crucial to the community structure of this network since it immediately brings down NMI score to 0.6 while the other does not seem to discover this important feature. Furthermore, when too many nodes are removed from the network, the network does not seem to contain communities anymore or the community structure become extremely fuzzy as NMI values converge down to around 0.2. This is understandable since this dataset is of small size with a very high average node degree.

On larger networks Facebook and Foursquare, we set k from 50 nodes to 1000 nodes (only 2.1% and 1.5% number of nodes of Foursquare and Facebook networks) with a 50-node increment at a time. The numerical results are reported in figure 3. In general, NMI values of all methods degrade quickly on Foursquare networks, and tend to decrease slower on Facebook networks. As more nodes are excluded from the network, $genEdge$ still achieves the best performance on both networks with significantly lower NMI values than the other methods. Specifically, on Foursquare with high average degree and internal community density, the removal of nodes incident to the most generating edges in $genEdge$ significantly leads to the separation of network community structure as NMI scores drop down to 0.2 in $genEdge$. On Facebook network, the similarity between the original and new community structure seem to retain fairly high even all 1000 nodes are removed. This implies that community structure in Foursquare network is also extremely vulnerable to node removal attacks, while the mature Facebook network does not seem to suffer this threat. One possible reason for this is since Facebook contains a giant community with low average degree, it therefore requires much more effort in order to break that giant community apart.

In summary, the experiments on both synthesized and real-work social network confirm the effectiveness of our proposed method based on generating edges. The empirical results also confirm that, $genEdge$ outperforms other heuristic methods on other community detection methods such as $AFOCS$, $Blondel$ and $Oslom$ algorithms.

VI. Related work

Community structure and complex network vulnerability are the two major and well-developed areas of networking research. Surveys on community structure detection algorithms as well as methods for assessing network vulnerabilities can be found in the work of Fortunatos et. al. [17], and Grubesic et. al. [18], respectively. However, assessing the vulnerability of network community structure has so far been an untrodden area.

The first attempt on this research direction [8] has suggested potential conditions for the transformation of the network community structure, and have proposed multiple heuristic algorithms based on the modularity contribution of communities in the network. These approaches, while are efficient in analyzing disjoint structures, face limitations when applied to real networks displaying overlapping community structures. As a result, the need for an effective algorithm that can assess the vulnerability of the general network structures is of desire. Aside from that, a large body of work has been devoted to find the node roles within a community by a link-based technique together with a modification of node degree [19], by using the spectrum of the graph [14], by using a within-module degree and their participation coefficient [20], or by the detection of key nodes, overlapping communities and "date" and "party" hubs [21]. However, none of these approaches discuss how the community structure would change in the failure of those important nodes, especially in terms of NMI measure.

The vulnerability of network function and structure has been examined under the node centrality metrics, such as high degree and betweenness centrality, as well as under the average shortest path which tries to signify the lengths of shortest distances between node pairs [18], under the pairwise connectivity metric whose goal aims to break the network's pairwise connectivity down to a certain level [1], or under the available number of compromised $s - t$ flows [22], etc.

VII. Conclusions

We assess the structural vulnerability of complex network community structures. We propose the concept of *generating edges* of a community, provide an optimal algorithm for detecting the MGES in a community, and then suggest *genEdge*, a node selection strategy based on this MGES. In comparison with established results on disjoint community structure, we analyze possible conditions that can lead to the minimization of the Normalized Mututal Information on the general overlapped community structure. Our analysis provides more general and insightful keys into understanding the vulnerability of overlapped network community structure. We conducted experiments on both synthesized and real networks. Empirical results confirm the efficacy of our suggested approach.

Acknowledgment

Md Abdul Alim and My T. Thai are partially supported by the National Science Foundation under CAREER Award grant number 0953284, by the DTRA Young Investigator Program under grant number HDTRA1-09-1-0061. Nam Nguyen is partially supported by the Faculty Development and Research Committee Awards, Towson University.

References

[1] Thang N. Dinh, Ying Xuan, My T. Thai, Panos M. Pardalos, and Taieb Znati. On new approaches of assessing network vulnerability: hardness and approximation. *IEEE/ACM Trans. Netw.*, 20(2):609–619, April 2012.

[2] Karsten Peters, Lubos Buzna, and Dirk Helbing. Modelling of cascading effects and efficient response to disaster spreading in complex networks. *IJCIS*, 4(1/2):46–62, 2008.

[3] L. Danon, A. Diaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(09):P09008, 2005.

[4] Pan Hui, Jon Crowcroft, and Eiko Yoneki. Bubble rap: social-based forwarding in delay tolerant networks. In *Proceedings of the 9th ACM international symposium on Mobile ad hoc networking and computing*, MobiHoc '08, pages 241–250, New York, NY, USA, 2008. ACM.

[5] Nam P. Nguyen, Thang N. Dinh, Sindhura Tokala, and My T. Thai. Overlapping communities in dynamic networks: their detection and mobile applications. In *Proceedings of the 17th annual international conference on Mobile computing and networking*, MobiCom '11, pages 85–96, New York, NY, USA, 2011. ACM.

[6] N.P. Nguyen, Ying Xuan, and M.T. Thai. A novel method for worm containment on dynamic social networks. In *MILCOM 2010*, pages 2180 –2185, 31 2010-nov. 3 2010.

[7] Zhichao Zhu, Guohong Cao, Sencun Zhu, S. Ranjan, and A. Nucci. A social network based patching scheme for worm containment in cellular networks. In *INFOCOM 2009, IEEE*, pages 1476 –1484, april 2009.

[8] Nam P. Nguyen, Md Abdul Alim, Yilin Shen, and My T. Thai. Assessing network vulnerability in a community structure point of view. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM '13, pages 231–235, New York, NY, USA, 2013. ACM.

[9] A. Lancichinetti and S. Fortunato. Community detection algorithms: A comparative analysis. *Phys. Rev. E*, 80(5):056117, Nov 2009.

[10] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley-Interscience, 1991.

[11] Nathan Eagle and Alex (Sandy) Pentland. Reality mining: sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268, March 2006.

[12] 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.

[13] Foursquare Data. sites.google.com/site/namnpuf/original_foursquare.7z. In *Collected data*, 2012.

[14] Yang Wang, Zengru Di, and Ying Fan. Identifying and characterizing nodes important to community structure using the spectrum of the graph. *PLoS ONE*, 6(11):e27418, 11 2011.

[15] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[16] Andrea Lancichinetti, Filippo Radicchi, Jos J. Ramasco, and Santo Fortunato. Finding statistically significant communities in networks. *PLoS ONE*, 6(4):e18961, 04 2011.

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

[18] T. H. Grubesic, T. C. Matisziw, A. T. Murray, and D. Snediker. Comparative approaches for assessing network vulnerability. *Inter. Regional Sci. Review*, 31, 2008.

[19] Jerry Scripps, Pang-Ning Tan, and Abdol-Hossein Esfahanian. Node roles and community structure in networks. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, WebKDD/SNA-KDD '07, pages 26–35, New York, NY, USA, 2007. ACM.

[20] Roger Guimera and Luis A. Nunes Amaral. Functional cartography of complex metabolic networks. *Nature*, 433(7028):895–900, feb 2005.

[21] Istvan A. Kovacs, Robin Palotai, Mate S. Szalay, and Peter Csermely. Community landscapes: An integrative approach to determine overlapping network module hierarchy, identify key nodes and predict network dynamics. *PLoS ONE*, 5(9):e12528, 09 2010.

[22] Timothy C. Matisziw and Alan T. Murray. Modeling s-t path availability to support disaster vulnerability assessment of network infrastructure.