Proceedings of the World Congress on Engineering 2021
WCE 2021, July 7-9, 2021, London, U.K.

Multiple Communities of Ego in Social Networks

Günce Keziban Orman

***Abstract*—Communities in social networks have the tendency to overlap. Their overlapping parts carry important knowledge about the studied system. For this reason overlapping community identification arises as an important task in social network analysis. Up to now, several different methods have been proposed but suffer from being sensitive to not only the number of nodes in overlapping parts of communities, but also the number of communities which overlap. In this work, we propose a new algorithm merging multiple communities around each ego nodes. It is designed for detecting highly overlapping communities at small size. These types of communities correspond to small social groups that share members such as families, close friends or coworkers. Experiments are conducted on artificial and real-world networks. Performances are evaluated quantitatively and qualitatively by comparing foremost state-of-art algorithms. The results show that our algorithm overcomes previous issues of overlapping community detection.**

***Index Terms*—Overlapping Community, Node Disjoint Path, Algorithm Comparison**

I. INTRODUCTION

Community detection is one of the most attractive subjects in social network analysis [1]. A community corresponds to a group of nodes with denser inner links and sparser outer links [1]. Communities are widely studied for discovering functionally related objects, finding interactions between modules, inferring missing attribute values and predicting unobserved connections, etc.[2]. The applications are numerous, such as: recommendation systems, viral marketing or sentiment analysis. Right now, majority of the existing approaches considers communities as disjoint node groups [1]. Recently, a high interest to finding overlapping communities arises [3], [4], [5]. An overlapping community structure can be defined as groups of nodes in which any two groups might share common nodes.

In [6], the authors underline that communities as functional objects share some common nodes. Those overlapping nodes have higher link density than non-overlapping parts. Hence, finding them gives us important knowledge about the functional roles of those communities and their nodes. Raey et al. show that overlapping communities are significant features in social networks [7]. Finding them serves to discover the dynamics of social interactions. For instance, in a social environment, one person can belong to multiple communities at the same time, e.g. his coworkers, his social friends or his family members. Overlapping parts of different communities may represent similar sides of those social groups. Or, people belonging to only one community may be identified by a single interest. Moreover, overlapping amount of two communities gives an idea about their future state, i.e. those groups might merge in near future. As a result, overlapping community detection becomes an important task in social network analysis.

We encounter several different approaches which are dedicated to the detection of overlapping community structure [5], [8], [4], [3], [9], [10], [11], [12]. In [3], the authors reveal common limits of state-of-art algorithms; they are sensitive to the overlapping level of the communities. More explicitly, their performance is affected by both the number of nodes which are at the overlapping parts of the communities and also the number of communities which overlap. Some other drawbacks can be listed as first, some of the existing methods put every node into at least one community [4], [13]. However, in reality, networks may contain community-less nodes which correspond to noise or outliers. Second, the size of communities are in general large [4]. But, one needs to find smaller communities which corresponds to small friendship groups or family member in real-world social environments. In this work, we propose a new overlapping community detection algorithm that overcomes mentioned drawbacks of previously described algorithms. This algorithm let the user regulate the size and the cohesiveness of the communities. It is not sensitive to the overlapping level of the communities and also extracts a specific type of outlier or noise.

In real-world social environments, we encounter multiple cohesive groups around each specific person. Some of those groups are his family members, his coworkers, his social friends and so on. Each person can also contact individually with some unique people without belonging to any group. In the algorithm, we modelled this social life phenomenon. We concentrate on each ego node and look for all social cohesive groups around each ego. At that point we consult the notion of being *k-connected* of the nodes for defining cohesive groups. Our algorithm discovers overlapping communities by merging similar *k-connected* node groups around each ego. The two main contributions of this article is first analytically describing a new algorithm for finding overlapping community structure fro social network and second evaluating the performance of this algorithm by comparing it with the performance of foremost overlapping community detection algorithms. We consider some prominent methods such as OSLOM [14], GCE [5], MOSES [8], COPRA [4] and EGO-BASED [9] at this comparative study. In the following section, we give the definitions and methods related to the details of proposed algorithm. In section III, we describe the LFR model and accuracy results of all algorithms. Finally, in the last section, we give a brief conclusion and explain future aspects of this work.

II. METHOD

Given a plain network $G=(E,V)$, V is a set of *n nodes* and E is a set of *m links*. For each node i in V, its *ego-centered network at radius* d_i, $G_i^{d_i}$, is the sub-network of G

Manuscript received March 7, 2021; revised March 24, 2021. This work was supported by the Galatasaray University Research Fund (BAP) within the scope of project number 18.401.004, and titled "Sıralı Sistemlerde Tahmin ve Çıkarım".

G.K. Orman is an Assitant Professor of Computer Engineering Department, Galatasaray University, Istanbul, Turkey, (korman@gsu.edu.tr).

ISBN: 978-988-14049-2-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

centered on node i and surrounded with d_i^{th} level neighbors of i. Note that if $d_i = 1$, all the nodes in G_i^1 are the direct neighbors of i. If d_i is equal to the diameter of G then, $G_i^{d_i}$ is as same as G. Two *paths* between two nodes i and j are *node independent*, a.k.a *node disjoint*, if they do not have any common internal nodes except i and j [15]. We say i and j are k-connected if there is at least $k \in [0, \infty[$ different node disjoint paths from i to j. We define an *ego-centered k-connected node group* $C_{ip} \subseteq V_i^{d_i}$ of ego i as a group of nodes in which each pairs of nodes are *k-connected*. Note that $\cup_p C_{ip}$ is equal to $V_i^{d_i}$ depending on the value of k. If k is at its lowest value, $\cup_p C_{ip} = V_i^{d_i}$. However, if k is at its maximal value, several nodes may not belong to any group. Note that in an ego-centered k-connected node group, $\forall k > 1$, the nodes stay connected even if ego is removed from the network because they are all *k-connected* before the removal. To find such groups, we propose an algorithm which at first eliminate the nodes which are not *k-connected* with the ego. We then extract each connected components after removing ego.

Algorithm 1 Finding Ego-centered k-Connected Node Groups

Require: $G_i^{d_i}$, k, i
Ensure: $C_i = C_{i1}, \ldots, C_{ip}$
1: **for** $j \in V_i^{d_i}$ **do**
2:   **if** $nodeDisjointPathNumber(i, j) < k$ **then**
3:     $G_i^{d_i} = remove(G_i^{d_i}, j)$
4:   **end if**
5: **end for**
6: $G_i^{d_i} = remove(G_i^{d_i}, i)$
7: $C_i = extractConnectedComponents(G_i^{d_i})$
8: $C_i = insertEgoToAll(C_i, i)$

We give pseudo code of finding ego-centered k-connected node groups in algorithm 1. It takes minimum number of node disjoint paths (k), ego-centered network ($G_i^{d_i}$) and ego (i) itself as input. It first eliminates the nodes whose node disjoint path numbers to the ego are less than k (lines 1, 2, 3, 4 and 5). Here, function `nodeDisjointPathNumber(...)` in line 2 calculates the number of disjoint paths. We use push relabel max flow algorithm [16] for this function. The complexity of this step is $O(|V_i^{d_i}|^3)$. We calculate it for all nodes, so it makes $O(|V_i^{d_i}|^4)$. Functions `remove(...)` and `insertEgoToAll(...)`, given in lines 6 and 8 respectively, can be computed in constant time. Extracting connected components after removal of ego (given in line 7 as function `extractConnectedComponents(...)`) can be done with BFS. Running time of this step is $O(|V_i^{d_i}| + |E_i^{d_i}|)$. So overall running time of this algorithm is $O(|V_i^{d_i}|^4 + |V_i^{d_i}| + |E_i^{d_i}|) \sim O(|V_i^{d_i}|^4)$.

A. Ego Based Merged Overlapping Communities

We can extract same or similar node groups for different egos if they share many common neighbors. In this case, we need to merge similar groups to detect final community structure. Previously, Rees et al. merged the groups that match all nodes but one node of smaller group [9]. However, if the sizes of node groups are high, although the groups are similar, they can be ignored. That is why; we propose to take into account the rate of the similar part to non similar part of the node groups. Let us note that S_1 and S_2 are two node groups and $sim(S_1, S_2) = |S_1 \cap S_2| / |S_1|$ is their similarity.

Algorithm 2 Merging Node Groups

Require: S, *threshold*
Ensure: M
1: $sort(S)$
2: **for** $i \in 1 \ldots |S|$ **do**
3:   **for** $j \in i \ldots |S|$ **do**
4:     **if** $sim(S_i, S_j) \geq threshold$ **then**
5:       $S_j = union(S_i, S_j)$
6:       $S_i = \emptyset$
7:     **end if**
8:   **end for**
9: **end for**

The pseudo code of merging process is given in algorithm 2. It takes the set S, whose members are the node groups found for entire network, and the minimum ratio of the similarity (*threshold*) as the input parameters. It outputs overlapping community structure M. It firstly sorts S according to the size of node groups in increasing order (line 1). Computing size of each group can be done in constant time. We apply simple radix sort which requires $O(|S|)$ where $|S|$ is the total number of node groups. Then, the procedure checks if the similarity of each pair of node groups, S_i and S_j is greater than given threshold (line 4). Similarity can be computed in $O(min(|S_i|, |S_j|))$ by using hash table for storing the elements of one of the node groups. In case of being sufficiently similar, these two groups are merged (line 5) and the groups S_i and S_j are updated. Merging phase can also be done in $O(min(|S_i|, |S_j|))$ with hash table. The algorithm continues until each pair of node groups is processed. The average size of a node group can be given $n/|S|$. So, the overall time complexity of merging phase is $O(|S| + 2 \times (n/|S|) \times |S|^2) \sim O(n \times |S|)$

The algorithm Ego Based Merged Overlapping Communities, EMOC, at first extracts ego-centered network of each node i at radius d_i. This can be done in constant time by using the adjacency matrix representation of the graph. It then finds k-connected node groups related to each ego by using algorithm1 and creates the global set S of node groups over entire network. Finally, it applies algorithm 2 and creates the overlapping community structure. Total time complexity of these two steps is $O(|V_i^{d_i}|^4 + n \times |S|)$. We assume that $|V_i^{d_i}|^4 \gg n \times |S|$. It results $O(|V_i^{d_i}|^4)$. This complexity highly depends on the density of network and the chosen radius value d_i to create ego-centered networks. If network is sparse and $d_i = 1$, then, size of ego-centered network will be $\log n$ [9]. So, time complexity will be $O((\log n)^4)$. In case of dense network, or high values of d_i, complexity can be $O(n^4)$. Indeed, considering each node as an ego gives the opportunity of paralleling the computation. Thus, in practice, this time complexity can be reduced too much.

EMOC considers three parameters: (1) d_i, radius of ego-centered network to adjust the size of the communities. As this parameter can be constant for whole network, it can also change from one node to another, (2) k, number of node disjoint paths, to regulate the cohesiveness of node groups and

Table I
LFR NETWORK GENERATION PARAMETER VALUES

	Parameters	Values
1	μ	$\{0.1, 0.3, 0.5\}$
2	(c_{min}, c_{max})	$\{(5,25),(10,50),(20,100)\}$
3	O_n	$\{50, 100, 500\}$
4	O_m	$\{2,3,4,5,6,7,8,9,10\}$

(3), *threshold*, to decide the minimum ratio of node groups similarity. Adjusting values of these parameters requires topological analysis of the studied network. Nevertheless, in the most basic form, one can set constant $d_i = 1$, $k = 2$ and $threshold = 0.8$ for considering first-level ego-centered network, minimum cohesiveness inside the groups and high similarity of the groups. Note that decreasing the value of *threshold* may result high overlapping of the communities.

III. RESULTS

We made experiments on artificial and real-world networks to see the performance and the limits of EMOC. We generate artificial networks with predefined overlapping community structure by using LFR model [17] which is the most realistic artificial network generator in the literature. It allows the user to adjust many network properties by its generation parameters. It is used for community detection algorithms performance evaluation commonly [3]. Usually, the authors generate a set of artificial networks having different topological properties by changing the values of LFR generator parameters. Then, they examine the performance of the algorithms by applying them on generated networks. The version of LFR we consult at this work generates plain networks with predefined overlapping community structure. We manipulate following parameters; (c_{min}, c_{max}), μ, O_n and O_m. This allow us to control generated communities' sizes, the ratio of the links inside of the same communities that a node has, total number of nodes which belong to more than one community and maximum number of communities that a node can belong in overlapping structure respectively. The values of those parameters are given in table I. We generate networks with $n = 1000$. Three parameters of EMOC are set to $d_i = 1$, $k = 2$ and $threshold = 0.8$. The accuracy of EMOC is determined by a modified version of normalized mutual information (NMI) for overlapping communities [18]. As the traditional one, modified NMI is used commonly for this issue. It takes 0 if two compared overlapping structure is totally dissimilar and it takes 1 if they are exactly same. We compare EMOC with five foremost overlapping algorithms: GCE [5], OSLOM [14], COPRA [4], MOSES [8] and EGO-BASED [9]. In this section, we at first concentrate on quantitative performance evaluation of algorithms. Then we evaluate the algorithms' by the qualitative properties of estimated community structures. Finally, we give the results on real-world networks.

A. Quantitative Performance on Artificial Networks

In figure 1, NMI scores of the algorithms with O_m increase for different μ values is given. For $\mu = 0.1$, the case of having cohesive communities with good separation, all the algorithms exhibit good performance (NMI > 0.8, see figure 1 top-left). When $O_m \leqslant 5$, OSLOM and MOSES are the most

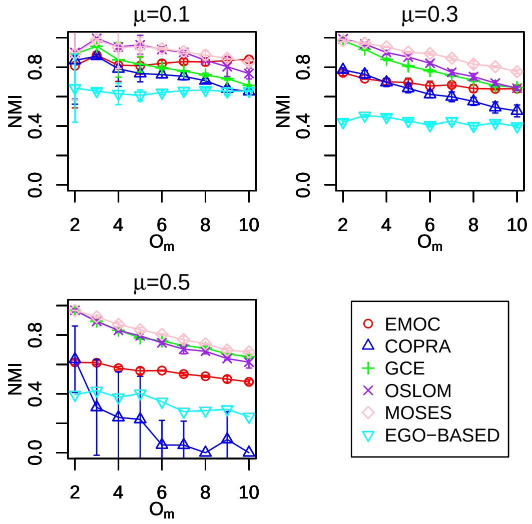

Figure 1. NMI Results of six algorithms for $(c_{min}, c_{max}) = (5,25)$, $O_n = 50$. Top-left plot is for well-separated communities ($\mu = 0.1$). Top-right plot is for medium-separated communities ($\mu = 0.3$). Bottom-left plot is for few-separated communities ($\mu = 0.5$)

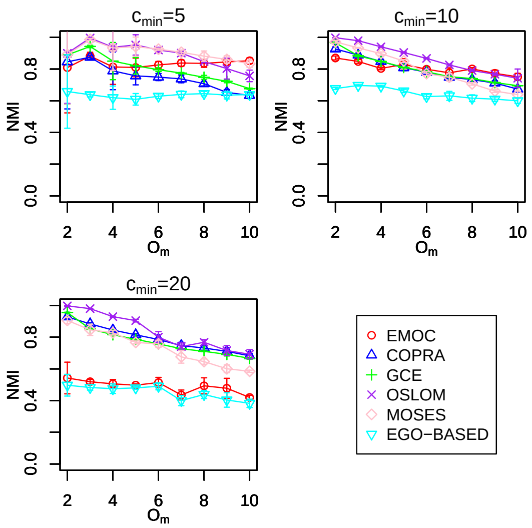

Figure 2. NMI Results of six algorithms for $\mu = 0.1$, $O_n = 50$. Top-left plot is for small communities ($(c_{min}, c_{max}) = (5,25)$). Top-right plot is for medium communities ($(c_{min}, c_{max}) = (10,50)$). Bottom-left plot is for large communities ($(c_{min}, c_{max}) = (20,100)$)

performing ones. GCE, EMOC and COPRA follow them. EGO-BASED seems the least performing one. The two ego based methods are less sensitive to community cohesiveness. Among them, EMOC seems more performing than EGO-BASED. The performance of EMOC, EGO-BASED and COPRA decreases with increase of μ. However, EMOC and EGO-BASED still can keep their performance stable to the increase of O_m. COPRA does not exhibit a robust behavior especially when $\mu = 0.5$ (figure 1 bottom-left).

We show NMI results of the algorithms according to different community sizes on different plots in figure 2. For all algorithms, the easiest case occurs when the network has small communities whose sizes change between 5 and 25 (figure 2 top-left). For larger communities the algorithms performance is still good (figure 2 top-right). However, we observe a linear decrease for all algorithms except ego based ones with the increase of O_m. Two ego based methods keep a

ISBN: 978-988-14049-2-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

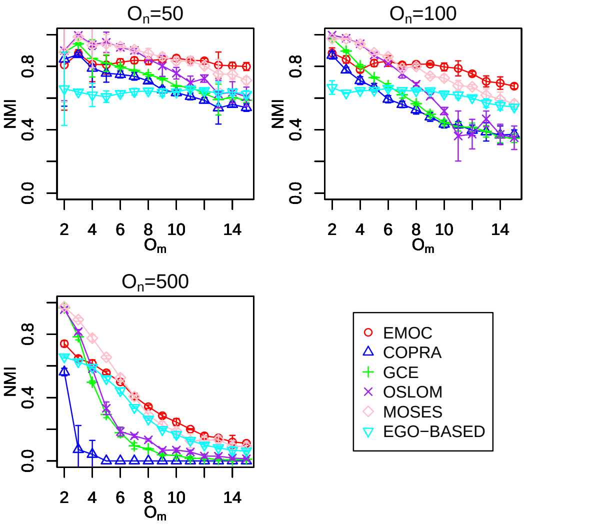

Figure 3. NMI Results of six algorithms for $\mu = 0.1$, $(c_{min}, c_{max}) = (5, 25)$. Top-left plot is for low level of overlapping ($O_n = 50$). Top-right plot is for medium level of overlapping ($O_n = 100$). Bottom-left plot is for high level of overlapping ($O_n = 500$)

stable performance. Among them, EMOC results are as good as other algorithms while EGO-BASED is one step backward than them. Especially, when $O_m > 8$, EMOC and OSLOM perform the best.

In figure 3, we represent NMI results of the algorithms for networks generated with $O_n = 50$, 100 and 500 in plots top-left, top-right and bottom-left respectively. The easiest case for all algorithms is $O_n = 50$. When the numbers of overlapping nodes increase to 100, we observe a visible linear decrease on the performance of all algorithms except ego based ones. EMOC's performance is as similar as the case of $O_n = 50$. Among the expansion based methods, MOSES exhibit better performance than the others even for high O_m values. It is claimed that MOSES is successful for detecting highly overlapping structures [8]. As seen in top-right plot, EMOC performs as well as MOSES for $O_m \geqslant 6$. Its performance is higher than MOSES for $O_m = 9$ and 10. In case that the half of the nodes in the network overlaps (figure 3 bottom-left), the decrease of the performance of all algorithms with the increase of O_m becomes more visible. The performance of the algorithms OSLOM, GCE and MOSES decrease logarithmic. Among them, MOSES has a smoother decreasing trend. COPRA has a sudden decrease. Even for low O_m values, its performance is worse than the others. Two ego based methods have similar performance trends and values.

By overall observation of the results for every parameter combination, the performance of GCE, OSLOM and MOSES are good and similar with each other when communities do not overlap too much. However, their performances are affected by overlapping density (controlled by O_n) and diversity (controlled by O_m). In contrast to this fact, two ego based methods seems more stable than the others when meeting with the changes on overlapping level. Among them EMOC exhibits better results than EGO-BASED in many cases. Especially, if the communities are well-separated and their sizes are small, EMOC results are as good as the most performing algorithms.

B. Qualitative Performance on Artificial Networks

In this section, we evaluate estimated community structures qualitatively by taking into account LFR reference community sizes as well. Estimated O_m with O_m increases for different O_n are shown in figure 4. It is an expected behavior that reference structure has a linear increase with the increase of O_m. For any algorithm, we also expect such a linear increase.

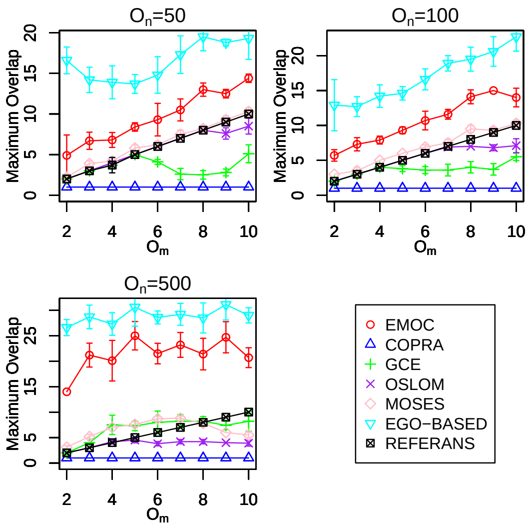

Figure 4. Maximum number of overlapping communities found by each algorithm and LFR reference for $\mu = 0.1$, $(c_{min}, c_{max}) = (5, 25)$. Top-left plot is for low level of overlapping ($O_n = 50$). Top-right plot is for medium level of overlapping ($O_n = 100$). Bottom-left plot is for high level of overlapping ($O_n = 500$)

COPRA does not find any overlapping community for any case. MOSES and OSLOM find community structure with the most similar maximum overlapping community to the reference one for any case. However, when half of the nodes overlap ($O_n = 500$), they differentiate from the reference. Especially, OSLOM seems not to find as many overlapping communities as the reference one for this case. GCE cannot find as many overlapping communities as the reference one when $O_m > 5$ for $O_n = \{50, 100\}$. Two ego-centric methods find much more overlapping communities than the reference one for any case. Among them, EMOC has a more linear trend. EMOC is designed for finding small and highly overlapping communities. These graphical results confirm its compatibility for this aim.

In figure 5, estimated values of O_n are shown. COPRA does not find any overlapping nodes for any O_n. Estimated values are different from the reference one when half of the nodes overlap. For other two cases ($O_n = \{50, 100\}$), OSLOM finds as much overlapping nodes as the reference one. Two ego-centric methods result much more overlapping nodes than the reference one. Among them EGO-BASED puts all nodes into more than one community for any case while EMOC results %30 of overlapping nodes in the network. In the reference community structure, LFR puts all nodes into at least one community. Thus, we expect that the algorithms should not result any community-less nodes. Community-less node numbers for different O_n and O_m values are shown in figure 6. Except GCE and MOSES, all the algorithms put all the nodes into at least one community.

ISBN: 978-988-14049-2-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

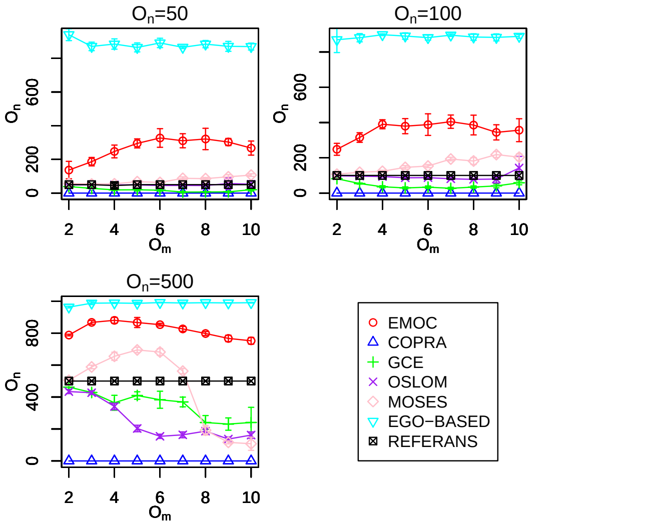

Figure 5. Number of overlapping nodes found by each algorithm and LFR reference for $\mu = 0.1$, $(c_{min}, c_{max}) = (5, 25)$. Top-left plot is for low level of overlapping ($O_n = 50$). Top-right plot is for medium level of overlapping ($O_n = 100$). Bottom-left plot is for high level of overlapping ($O_n = 500$)

However those two algorithms output some community-less nodes. Especially, GCE finds many community-less nodes. Their numbers are affected both by O_m and O_n. When there is too much overlapping node with many overlaps between communities ($O_m > 6$ and $O_n = 500$), MOSES results many community-less nodes. We want to remind here that NMI is sensitive and positively affected by the number of communities. Those two algorithms' NMI score are high to be compared with other algorithms. However, here we discover that they might result many community-less nodes. Hence, the reliability of quantitative comparison of the algorithms might be open to criticism. One single measure is not efficient to explain the performance of the algorithms but it should be supported by qualitative analysis which enlightens the properties of the algorithms.

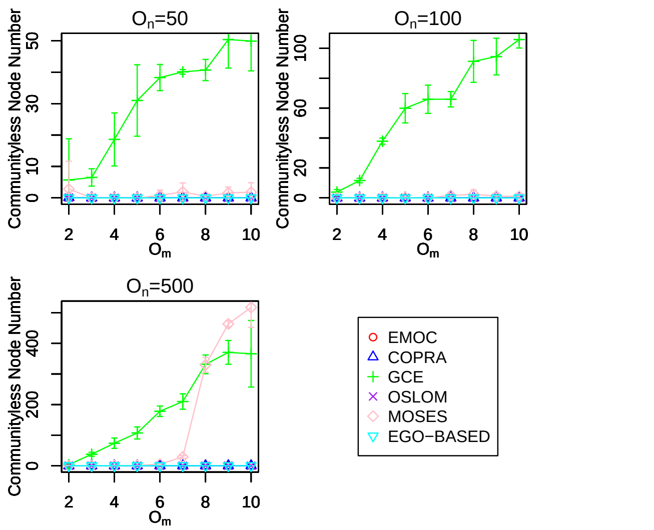

Figure 6. Number of community-less nodes found by each algorithm and LFR reference for $\mu = 0.1$, $(c_{min}, c_{max}) = (5, 25)$. Top-right plot is for medium level of overlapping ($O_n = 100$). Bottom-left plot is for high level of overlapping ($O_n = 500$)

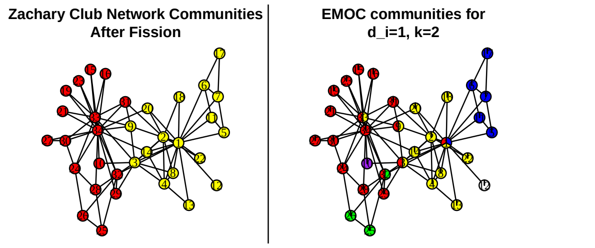

Figure 7. Zachary Karate Club Network. The different colors of nodes represent belonging to different communities. Nodes with multiple colors belong to multiple communities. Left and right plots represent ground-truth and EMOC communities for $d_i = 1$, $k = 2$ and $threshold = 0.8$ respectively.

C. Results on Real-World Networks

We apply our algorithm on two real-world networks. The first one is well-known Zachary karate club network [19]. This network is created by observing Zachary club members for 2 years. It shows the relations of 34 club members. Club members split into two groups because of political conflict between karate trainer John (node #34) and club president Mr. Hi (node #1). There are two natural communities whose leaders are those two members. One can see these communities represented by different colors in left plot of figure 7. The faction of each club member is also declared as strong or weak connection with one of the leaders or none.

We find 4 communities (shown in right plot of figure 7). Natural communities of Zachary network seem to be split by EMOC. Union of red and green communities of EMOC substantially correspond John's group. Likewise, yellow and blue communities of EMOC constitute Mr. Hi's group. There are 5 overlapping nodes (#1, 3, 9, 32, 33). Mr. Hi (node #1) belongs to three communities that 2 of them correspond to his group after split. Evaluating centrality scores, overlapping nodes have the highest betweenness centrality. Overlapping nodes are either lying in-between two groups or they are embedded in the core center of the groups. The most interesting result is about node #9. Its betweenness score is not as high as others. This node was a weak supporter of John before the split but he joined Mr. Hi's group afterwards [19]. EMOC puts him into two communities that each of them correspond to the groups of different leaders in reality. Two nodes (#10 and 12) are not placed into any community. Regarding their topological position, node #12 is connected only with node #1 (Mr. Hi). Node #10 has only two connections. The faction of this node is marked as *none*. Node #10's two friends are Mr. Hi's strong supporter and John himself. His idea about leaders is neutral and his friends are homogeneous. As a result, although he is placed in one of the groups in reality, we cannot claim that he is embedded there. Briefly, it is seen that EMOC finds consistent communities with real groups in Zachary Karate network. Overlapping nodes have important topological situation. Community-less nodes can be either non-effective or easily affected by other people. Second real-world network we deal with is Facebook Network [20]. This network is a combination of 10 ego-centered networks that each of them includes the social circles of ten different Facebook users. There are 4039 nodes corresponding 10 ego and their friends and 88234 links representing the friendship relation of them. This network is ego-centralized

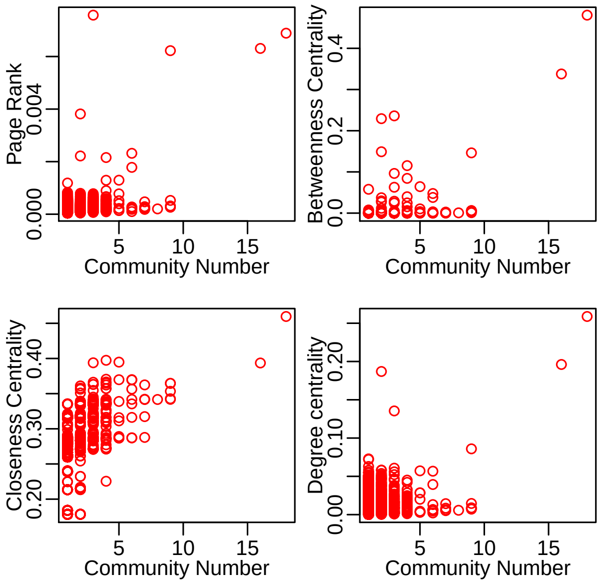

Figure 8. Relation between nodal topological measures and the numbers of communities that Facebook users belong. Each red circle in the plots corresponds to the scores of one node. Top-left, top-right, bottom-left, bottom-right plots show the relation between community numbers and page rank, betweenness, closeness and degree centralities respectively

by its construction. We apply EMOC with same parameter values. We examine a possible relation between numbers of communities that a node belongs and topological properties. For this reason, we represent in figure 8 the scores of page rank, betweenness, closeness and degree centralities with the numbers of overlapping communities for each node. As it is case for Zachary karate network, in Facebook, the most overlapping nodes (#108 and 1685, points at the top-right corners of each plot) are the most central ones. Their page rank score is also high. Those nodes correspond to two egos having hub position in the network. Some nodes belonging to low numbers of community have high page rank scores. Looking at them in more detail, we notice that those nodes are not central themselves but they have direct connection with important hubs. As a result, their page rank score is high because of knowing important people.

Considering two real-world network experiments, EMOC in general finds small communities. Some of them corresponds to highly interconnected node groups which do not have relation between each other except connecting to the same hub. In general the most overlapping nodes are those hubs. Other highly overlapping nodes are bridges connecting different communities. The community-less nodes are different types of outliers such as the people having unexpected behavior or people who have no more connections than one in whole network.

IV. Conclusion

In this work, we propose an algorithm EMOC to find small overlapping communities and evaluate its accuracy on the artificial networks by comparing its results with foremost algorithms. The results show that EMOC is performing to find small and well-separated communities. Those types of communities are family members or close friend groups in real-worlds social environments. Its performance is not sensitive to overlapping density and diversity. Hence, it can be used for detecting communities in any overlapping level. It can find contextually accurate communities in real-world social networks. In its estimated community structure, overlapping nodes corresponds to hubs which are meeting points of different node groups or bridges binding different communities. The community-less nodes are different types of outliers. Some prominent perspectives appearing from this study are examining the roles of different parameters of EMOC on the topology of detected communities, applying EMOC on different types of social networks and interpreting the results and developing a strategy to automatically and dynamically determine EMOC parameter values according to the topological positions of the nodes. We also want to modify EMOC for weighted and directed networks.

References

[1] S. Fortunato, “Community detection in graphs,” *Phys. Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.
[2] J. Yang, J. McAuley, and J. Leskovec, “Community detection in networks with node attributes,” in *ICDM*, 2013, pp. 1151–1156.
[3] J. Xie, S. Kelley, and B. K. Szymanski, “Overlapping community detection in networks: The state-of-the-art and comparative study,” *ACM Comput. Surv.*, vol. 45, no. 4, pp. 43:1–43:35, Aug. 2013.
[4] S. Gregory, “Finding overlapping communities in networks by label propagation,” *New Journal of Physics*, vol. 12, no. 10, p. 103018, 2010.
[5] C. Lee, F. Reid, A. McDaid, and H. Neil, “Detecting highly overlapping community structure by greedy clique expansion,” in *SNA-KDD10*, 2010.
[6] J. Yang and J. Leskovec, “Overlapping communities explain core-periphery organization of networks,” *Proceedings of the IEEE*, vol. 102, no. 12, pp. 1892–1902, Dec 2014.
[7] F. Reid, A. McDaid, and N. Hurley, “Partitioning breaks communities,” in *ASONAM*, 2011, pp. 102–109.
[8] A. McDaid and N. Hurley, “Detecting highly overlapping communities with model-based overlapping seed expansion,” in *ASONAM*, 2010, pp. 112–119.
[9] B. Rees and K. Gallagher, “Overlapping community detection by collective friendship group inference,” in *ASONAM*, 2010, pp. 375–379.
[10] X. Chen and J. Li, “Community detection in complex networks using edge-deleting with restrictions,” *Physica A: Statistical Mechanics and its Applications*, vol. 519, pp. 181–194, 2019.
[11] H. Jin, W. Yu, and S. Li, “Graph regularized nonnegative matrix tri-factorization for overlapping community detection,” *Physica A: Statistical Mechanics and its Applications*, vol. 515, pp. 376 – 387, 2019.
[12] T. Chakraborty, S. Ghosh, and N. Park, “Ensemble-based overlapping community detection using disjoint community structures,” *Knowledge-Based Systems*, vol. 163, pp. 241 – 251, 2019.
[13] J. Xie, B. Szymanski, and X. Liu, “Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process,” in *ICDMW*, Dec 2011, pp. 344–349.
[14] A. Lancichinetti, F. Radicchi, J. Ramasco, and S. Fortunato, “Finding statistically significant communities in networks,” *PLoS ONE*, vol. 6, no. 4, p. e18961, 2011.
[15] D. R. White and F. Harary, “The cohesiveness of blocks in social networks: Node connectivity and conditional density,” *Sociological Methodology*, vol. 31, no. 1, pp. 305–359, 2001.
[16] A. V. Goldberg and R. E. Tarjan, “A new approach to the maximum-flow problem,” *J. ACM*, vol. 35, no. 4, pp. 921–940, 1988.
[17] A. Lancichinetti and S. Fortunato, “Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities,” *Phys. Rev. E*, vol. 80, p. 016118, Jul 2009.
[18] A. Lancichinetti, S. Fortunato, and J. Kert??sz, “Detecting the overlapping and hierarchical community structure in complex networks,” *New Journal of Physics*, vol. 11, no. 3, p. 033015, 2009.
[19] W. W. Zachary, “An information flow model for conflict and fission in small groups,” *Journal of Anthropological Research*, vol. 33, pp. 452–473, 1977.
[20] J. J. McAuley and J. Leskovec, “Learning to discover social circles in ego networks.” in *NIPS*, 2012, pp. 548–556.

ISBN: 978-988-14049-2-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)