A new Pre-processing Strategy for Improving Community Detection Algorithms

A. Meligy
Professor of Computer Science, Faculty of Science, Menofia University, Egypt

Ahmed H. Samak
Asst. Professor of computer science, Faculty of Science, Menofia University, Egypt

Mai E. Saad
Researcher in Computer Science, Faculty of Science, Menofia University, Egypt

ABSTRACT
In the study of complex networks, a network is said to have community structure if it divides naturally into groups of nodes with dense connections within groups and only sparser connections between them. Detecting communities from complex networks has attracted attention of researchers in a wide range of research areas, from biology to sociology and computer science. In this paper, we introduce a new approach to make existing community detection algorithms execute with better results.

Our method enhancing community detection algorithms by applying a pre-processing step that exploits betweenness for nodes and edges, to maps unweighted graph onto a weighted graph. It has been tested in conjunction with four algorithms, namely the Louvain method, SOM algorithm, VOS clustering, and Danon algorithm. Experimental results show that our edge weighting strategies raises modularity for existing algorithms.

Keywords
Social Network, Community detection, Centrality measures, Modularity function.

1. INTRODUCTION
Social networks can be generally modeled by a graph $G = \langle V, E \rangle$, where $V = \{v_1, v_2, \cdots, v_n\}$ is the set of vertices and it represents the social actors, and E represents the set of edges connecting pairs of vertices and it represents the interactions between them [1].

For a better analysis of a social network, it is usually decomposed into subunits or communities, i.e., they naturally divided into groups of vertices with denser connections inside each group and fewer connections crossing groups. Members in each community of social networks usually contain users having similar characteristics that make them different from the others. The problem of community detection in a network is the gathering of network vertices into groups in such a way that nodes in each group are densely connected inside and sparser outside. The detection of community structure is extremely helpful since it help us understand the structures of given social networks. Communities are regarded as components of given social networks and they will clarify the functions and properties of the networks. There are several types of community detection algorithms that have been proposed. These algorithms are divided into two types: divisive and agglomerative.

Divisive algorithms detect inter-community links and remove them, so that the communities get disconnected from each other. The most popular algorithm is that proposed by Girvan and Newman [2]. In this algorithm, edges are removed according to their betweenness values. Agglomerative algorithms merge nodes into communities iteratively if their similarity is sufficiently high. The quality of the partitions resulting from these algorithms is often measured by a quality function evaluate how good a partition is. The most popularity quality function is the modularity of Newman and Girvan [3]. It can be written in several ways.

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(C_i, C_j) \quad (1)$$

where the sum runs over all pairs of vertices, A is the adjacency matrix, $k_i$is the degree of vertex i and m is the total number of edges of the network. The element of A_{ij} of the adjacency matrix is 1 if vertices i and j are connected, otherwise it is 0.The δ-function yields 1 if vertices i and j are in the same community, 0 otherwise.

Modularity can be rewritten as follows.

$$Q = \sum_{s=1}^{n_m}\left[\frac{l_s}{m} - \left(\frac{d_s}{2m}\right)^2\right] \quad (2)$$

where n_m is the number of communities, l_s is the total number of edges joining vertices of community s,and d_s is the sum of the degrees of the vertices of s.In the above formula, the first term of each sum is the fraction of edges of the network inside the community, whereas the second term represents the expected fraction of edges that would be there if the network were a random network with the same degree for each vertex.

The modularity of a partition is a scalar value between -1 and1. Large positive values of Q are expected to indicate good partitions. The modularity of the whole network, taken as a single community, is zero. Modularity has been used as quality function in many algorithms. In addition, modularity optimization is a popular method for community detection.

In [4], Fortunato and Barthélemy have shown that modularity optimization may fail to identify communities smaller than a scale which depends on the total size of the network and on the degree of interconnected-ness of the communities, which is called "resolution limit" problem. Clauset et al have proposed a fast agglomerative algorithm [5]. Starting from a set of isolated nodes, the links of the original graph are iteratively added such to produce the maximum possible increase in the modularityQ at each step. A technique for modularity maximization is presented by Duch and Arenas [6]. A fast modularity optimization method has been presented by Blondel et al [7]. In the following we will refer to it as LM. We present a detailed description of it in section 4.

Radicchi et al. have proposed an algorithm [8]. This algorithm removes links iteratively according to the value of their edge clustering coefficient. Cfinder [9] is a local algorithm proposed by Palla et al. It was the first paper that looks for communities which may overlap, i.e. share nodes. It

is based on the concept that the internal edges of a community are likely to form cliques due to their high density.

Spectral algorithm [10] introduced by Donetti and Munoz is a method based on spectral properties of the graph. The idea is that eigenvector components corresponding to nodes in the same community should have similar values, if communities are well identified. Donetti and Munoz focused on the eigenvectors of the laplacian matrix. The COPRA (Community Overlap PRopagation Algorithm) algorithm [11] relies on a label propagation strategy to find communities. COPRA is able to find both overlapping and non-overlapping communities. Vertices have labels that can propagate between neighboring vertices. Once labels have been propagated, it is expected that the vertices within a community have the same labels.

Other algorithms exploit the random walks technique for community detection. The κ-path edge centrality is a new measure of the centrality of an edge that is proposed in [12]. In this paper, the κ-path centrality $L^{\kappa}(e)$ of e, is defined as the sum, over all possible source vertex s, of the probability with which a message originated from s traverses e, assuming that the message traversals are only along random simple κ-paths. This new measure used in [13] [14] for enhancing community detection algorithms. Rosvall and Bergstorm introduced an efficient algorithm [15] that is based on random walk for detecting modules of networks. The modules can be detected by adopting a data compression perspective on the network (is denoted as informap). Studies on community detection can be found in an excellent surveys like [16] [17].

2. THE PROPOSED APPROACH

2.1 Overview of the proposed approach

In this paper, our goal is to present pre-processing steps, not to introduce another community detection algorithm. The pre-processing steps will improve existing community detection algorithms, and make it execute with better results.

Our approach has been designed to take as input a graph $G = \langle V, E \rangle$ with its adjacency matrix A, and benefits from betweenness to nodes and edges to produce a new matrix W such that w_{ij} is the weight of the edge connecting vertices i and j. Obviously, our approach takes as input an unweighted graph $G = \langle V, E \rangle$, and transfers it onto a weighted graph $G' = \langle V, E, W \rangle$. The community detection algorithm will run on G', using W as input matrix. As a final comment, we can conjugate our pre-processing steps with any community detection algorithm.

2.2 Centrality Measure in Social Networks

Within graph theory and network analysis, *betweenness centrality* is one of the most popular centrality measures.

Definition 1 (*Betweenness centrality of a Vertex*). Vertex Betweenness reflects the influence of a node over the flow of information between other nodes, especially in cases where information flow over a network primarily follows the shortest available path.

Given a graph *G = (V, E)*, the betweenness centrality $C_{B_n}(v)$ for the node $v \in V$ is defined as

$$C_{B_n}(v) = \sum_{S \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}} \quad (3)$$

Where s and t are nodes in V, σ_{st} is the number of shortest paths from s to t, and $\sigma_{st}(v)$ is the number of shortest paths from s to t passing through the node v. If there is no path joining s and t we conventionally set $\frac{\sigma_{st}(v)}{\sigma_{st}} = 0$. This may be normalised by dividing through the number of pairs of vertices not including v, which is $(n-1)(n-2)/2$ for undirected graphs [18].

Definition 2 *(Betweenness Centrality of an Edge).* The edge betweenness [2] measure assesses the centrality of an edge in a network. It is calculated by summing up the number of geodesics that pass the observed edge. The edges which connect different communities are usually expected to have higher betweenness.

The edge betweenness of an edge e in a graph *G = (V, E)*, is defined as the number of shortest paths passing through that edge. The betweenness centrality $C_{B_e}(e)$ of an edge $e \in E$ is defined as

$$C_{B_e}(e) = \sum_{S \neq t \in V} \frac{\sigma_{st}(e)}{\sigma_{st}} \quad (4)$$

as before, σ_{st} the number of shortest paths connecting s to t, and $\sigma_{st}(e)$ the number of shortest paths connecting s to t passing through the edge *e*. As in the previous case, if there is no path joining s and t we conventionally set $\frac{\sigma_{st}(e)}{\sigma_{st}} = 0$. Betweenness value can be normalised by $\big(n(n-1)\big)/2$ for undirected graphs [18].

2.3 Computing distances between graph vertices

The first step of our method is to calculate the betweenness centrality for each node, also calculate the edge betweenness centrality for each edge in the graph. Once the values of edge betweenness centrality have been computed, then we calculate the distance among each pair of connected nodes. This is done by using a L_2 distance (i.e., *the Euclidean distance*) calculated as

$$r_{ij} = \sqrt{\sum_{k=1}^{n} \frac{(C_{B_e}(e_{ik}) - C_{B_e}(e_{kj}))^2}{d(k)}} \quad (5)$$

where $C_{B_e}(e_{ik})$(resp., $C_{B_e}(e_{kj})$) is the edge betweenness centrality of the edge e_{ik}(resp.,e_{kj}) and $d(k)$ is the degree of the node.

2.4 The algorithm for Pre-processing steps for community detection

The details of our implementation are described as follows:

1: C_{B_n} ←Assign each node $v_n \in V$ with its betweenness centrality
2: $C_{B_n}{}^{*}$ ← Normalize C_{B_n}
3: C_{B_e} ←Assign each edge $e_m \in E$ with its edge betweenness centrality
4: $C_{B_e}{}^{*}$ ← Normalize C_{B_e}
5: r_{ij} ← Compute distance
6: Compute input matrix W: W← $A - r_{ij} + C_{B_n}{}^{*} + I$
7: $\mathcal{C}$ ← Community detection $\langle G, W \rangle$

3. APPLYING PREPROCESSING STEPS TO FIND COMMUNITIES

In this section we adopt four algorithms, namely the Louvain method (LM) [7], Self-organizing map (SOM) [19], visualization of similarities (VOS) [20] and Danon algorithm [21] to combine them with our pre-processing strategy and we

will describe each of these algorithms in detail. As discussed, our pre-processing steps can be conjugate with any existing community detection algorithm.

3.1 Louvain method-LM

The Louvain method has been introduced by Blondel et al. [7], for the general case of weighted graphs. LM consists of two phases that are repeated iteratively. The input of the algorithm is a weighted network of $\boldsymbol{\mathcal{N}}$ nodes. Initially, assign a different community to each node of the network. In this initial partition, there are many communities as there are nodes. Then, for each node $\boldsymbol{i}$, LM considers the neighbors $\boldsymbol{j}$ of $\boldsymbol{i}$ and evaluates the gain of modularity that would take place by removing $\boldsymbol{i}$ from its community and by placing it in the community of $\boldsymbol{j}$. The node $\boldsymbol{i}$ is then placed in the community for which the gain is maximum, but only if this gain is positive. If no positive gain is possible, $\boldsymbol{i}$ stays in its original community. This process is applied repeatedly and sequentially for all nodes until no further improvement can be achieved. This first phase stops when a local maximum of the modularity is attained, i.e. when no individual move can improve the modularity. The second phase of LM consists in building a new network whose nodes now the communities found during the first phase. To do so, the weights of the links between the new nodes are given by the sum of the weights of the links between nodes in the corresponding two communities. Links between nodes of the same community leads to self-loops for this community in the new network. Once this second phase is completed, it is then possible to reapply the first phase of the algorithm to the resulting weighted network and to iterate.

3.2 SOM algorithm

The SOM (self-organizing map) algorithm proposed by Zhenping et al. [19]. It is used for unsupervised network training. SOM method directly employs the adjacency matrix of a network as a feature data. The schema of the self-organizing map for detecting community structure, shown in figure 1, has an n input neurons corresponding to the nodes $v_1, v_2, \cdots, v_n$ in$G$, and m output neurons representing putative communities$\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_m$.

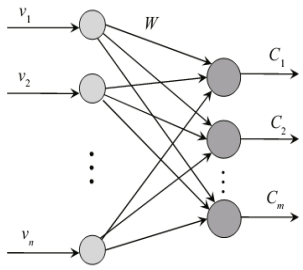

Figure 1 A two-layer self-organizing map for community detection

Let $A = [a_{ij}]_{n\times n}$ be the adjacency matrix of the network$G$, and $B = A + I$(I is a unit matrix). For each node v_i, the learning input $X = B_i \in R^n$ is a vector such that $b_{i,i} = 1$ and $b_{j,i} = 1$ if and only if v_j is adjacent to v_i in G, $b_{j,i} = 0$, otherwise. The connection weight matrix between input neurons and output neurons is$W = [w_{ij}]_{n\times m}$, where the weight w_{ij}expresses the possibility or membership degree that node v_i belongs to the community$\mathcal{C}_j$. If the input neuron v_i mapped into the output neuron$\mathcal{C}_j$, the connection between v_i and $\mathcal{C}_j$ should be reinforced. The discrimination function is the normalized correlation

$$\eta(W^j(k), X) = (W^j \otimes X)^T B(W^j \otimes X) \quad (6)$$

where W^j is the j-th column of the weighted matrix W. The winner neuron $\overline{c_j}$ with respect to the input vector X is selected by the following rule:

$$\bar{j} = \arg\max_j \eta(W^j(k), X). \quad (7)$$

The weights associated with the winner neuron are updated as

$$W^{\bar{j}}(k+1) = \frac{W^{\bar{j}}(k)+\alpha X}{\left\|W^{\bar{j}}(k)+\alpha X\right\|_\infty}, \quad (8)$$

and the weights associated with the non-winner neuron are updated as

$$W^j(k+1) = \frac{W^j(k)+\alpha(1-X)}{\left\|W^j(k)+\alpha(1-X)\right\|_\infty} \quad (9)$$

where α is the learning rate, and $j \neq \bar{j}$, $1 = (1,1,\cdots,1)^T$. After the training phase, the nodes of G are mapped into no more than m communities. The communities can be retrieved according to the final connection weight matrix W. For node v_i, it belongs to community c_j if

$$\bar{j} = \arg\max_j w_{ij}. \quad (10)$$

3.3 VOS Clustering

VOS (visualization of similarities) clustering technique proposed for the first time by Van Eck and Waltman in [20]. The communities obtained by VOS clustering are similar but not exactly the same as the ones obtained by the Louvain method, since in Louvain method modularity is optimized while in VOC clustering VOS quality function is optimized. The quality function V of VOS technique is:

$$V = \frac{1}{2m}\sum_{ij}[s_{ij} - \gamma]\delta(C_i, C_j) \quad (11)$$

where s_{ij} denotes the association strength of between vertex i and j, which is given by

$$s_{ij} = \frac{2m\, k_{ij}}{k_i k_j} \quad (12)$$

where k_{ij} denotes the number of edges between vertex i and j.

As discussed in [22] the quality function V of VOS clustering technique is equivalent to Newman's modularity Q when resolution parameter γ and edge weights are set to one.

3.4 Danon algorithm

Danon's greedy community detection agglomerative method has been introduced by Danon et al, [21]. This algorithm is a simple modification of the algorithm proposed by Newman for community detection [23]. A greedy method of Newman is an agglomerative hierarchical clustering method, where groups of vertices are successively joined to form larger communities such that modularity increases after the merging. This greedy optimization of modularity tends to form quickly large communities at the expenses of small ones, which often yields poor values of the modularity maxima. Danon et al. suggested normalizing the modularity variation ΔQ produced by the merger of two communities by the fraction of edges incident to one of the two communities, in order to favor small clusters. This trick leads to better modularity optima as compared to the original recipe of

Newman, especially when communities are very different in size.

4. EXPERIMENTAL RESULTS

In this section, we present the experimental results of our strategy .in particular; we assess the modularity of the partitions achieved by SOM algorithm, Louvain method, VOS clustering technique, and Danon algorithm with and without performing our pre-processing steps, and whether using pre-processing steps is capable of raising the modularity of a community detection algorithm.

4.1 Real networks

We chose 6 networks whose features are reported in Table 1 to perform our experiments.

Table 1. Real-world datasets

NO.	Dataset	No. nodes	No. edges	Ref.
1	Karate	34	78	[24]
2	Dolphins	62	159	[25]
3	Books US politics	105	441	[26]
4	Common adjective and noun adjacencies in David Copperfield (adjnoun)	112	425	[27]
5	College football	115	613	[2]
6	Jazz musicians	198	2742	[28]

Zachary's karate club: This is an undirected social network of friendship between 34 members of a karate club at US university. Edges connect individuals who were observed to interact outside the activities of the club.

Dolphin social network: Contains an undirected social network of frequent associations of 62 dolphins living in New Zealand. Edges were set between animals that were seen together more often than expected by chance.

Political Books: This dataset is the Amazon co-purchasing network with 105 books about US politics. Nodes are books and edges represent co-purchasing of books by the same buyers.

Common adjective and noun adjacencies in David Copperfield: The network of common adjective and noun adjacencies for the novel "David Copperfield" by Charles Dickens. Nodes represent the most commonly occurring adjectives and nouns in the book. Edges connect words that occur in adjacent position in the text of the book.

American college football dataset: This dataset contains 115 nodes representing teams. An edge exists between two vertices if there is match between two teams. More games happen among teams within the same community than teams from different community.

Jass musician network: This dataset is the collaboration network of jazz bands. There are 198 nodes representing the bands, and 2742 edges connecting the bands if there is at least one musician in common.

4.2 Results

To analyze modularity, we considered and we applied the SOM algorithm, Louvain method, VOS clustering technique, and Danon algorithm on the original datasets (which are unweighted graphs), and computed the modularity achieved by each algorithm on each of these datasets. After that, we pre-processed each of the datasets by applying our pre-processing steps. Therefore, each graph was transformed into a weighted graph. We re-applied the four algorithms on the modified datasets after applying our approach and re-computed the achieved modularity. The obtained results are reported on Table 2.For simplification, we refer to unweighted graph as UW, and use W referring to weighted graph.

Table 2. The modularity results on real networks

Dataset	LM UW	LM W	VOS UW	VOS W	SOM UW	SOM W	Danon UW	Danon W
Karate	0.4188	**0.5150 [+22.97%]**	0.4382	**0.6377 [+45.51%]**	**0.3438**	0.3428 [-0.29%]	0.4087	**0.5141 [+25.79%]**
Dolphins	0.5188	**0.5994 [+15.54%]**	0.5445	**0.6733 [+23.64%]**	**0.**4103	**0.4105 [+0.05%]**	**0.**5136	**0.5947 [+15.78%]**
Books US politics	0.4986	**0.5724 [+14.80%]**	0.5283	**0.6141 [+16.24%]**	0.3278	**0.3281 [+0.09%]**	0.5237	**0.5645 [+7.79%]**
adjnoun	0.2906	**0.4103 [+41.19%]**	0.5211	**0.6670 [+28.00%]**	0.2428	**0.2434 [+0.25%]**	**0.**2841	**0.4100 [+44.32%]**
College football	0.6046	**0.6510 [+7.67%]**	0.5236	**0.5781 [+10.39%]**	0.5090	**0.5199 [+2.14%]**	0.5661	**0.6225 [+9.95%]**
Jazz musicians	0.4431	**0.4617 [+4.20%]**	0.5492	**0.6077 [10.66%]**	**0.2448**	0.2445 [-0.12%]	0.4401	**0.4564 [+3.71%]**

5. CONCLUSION

This paper presented a new pre-processing step to make existing community detection algorithms execute with better results. Our pre-processing strategy enhancing community detection algorithms using betweenness for nodes and edges, to maps unweighted graph onto a weighted graph. It has been tested in conjunction with four algorithms (Louvain method, SOM algorithm, VOS clustering, and Danon algorithm) and applying to 6 Real networks. Experimental results show that our pre-processing strategy raises modularity for existing algorithms. The rate of increase in modularity after applying our preprocessing reached 45.51% in VOS and reached 41.19% in LM. In SOM algorithm the rate of increase in

modularity is reached 2.14%, and reached 44.32% in Danon algorithm.

6. REFERENCES

[1] F. Borko, "Hand book of Social Network Technologies and Applications" 2010, Springer.

[2] M. Girvan, M. E. J. Newman, "Community structure in social and biological networks", Proceedings of the National Academy of Sciences (PNAS), 99(12), 7821–7826, 2002.

[3] M. E. J. Newman, "Modularity and community structure in networks", Proceedings of theNational Academy of Sciences (PNAS), 103(23), 8577–8582, 2006

[4] S. Fortunato, M. Barthelemy, "Resolution limit in community detection" , Proceedings of the National Academy of Sciences (PNAS), 104(1), 36–41, 2007

[5] A. Clauset, M. E. J. Newman, C. Moore, " Finding community structure in very large networks ",Physical Review E, 70(066111), 1–6, 2004.

[6] J. Duch, A. Arenas," Community identification using extremal optimization", Physical Review E, 72 (027104) , (2005).

[7] V. Blondel, J. Guillaume, R. Lambiotte, E. Lefebvre, "Fast unfolding of communities in large networks, Journal of Statistical Mechanics", Theory and Experiment ,10, (P10008), (2008).

[8] F. Radicchi, C.Castellano , F. Cecconi , V.Loreto , D.Parisi ," Self-contained algorithms to detect communities in networks", Proceedings of the National Academy of Sciences,101:2658, USA 2004.

[9] G. Palla, I. Derenyi, I. Farkas, T. Vicsek. "Uncovering the overlapping community structure of complex networks in nature and society", Nature 435, 814 (2005).

[10] L. Donetti, MA. Munoz," Detecting network communities: a new systematic and efficient algorithm", Journal of Statistical MechanicsP10012, 2004.

[11] S. Gregory, "Finding overlapping communities in networks by label propagation". New Journal of Physics 12:103018, 2010.

[12] P. DeMeo, E. Ferrara, G. Fiumara, A. Ricciardello, "A novel measure of edge centrality in social networks", Knowledge-Based Systems 136–150. (2012).

[13] P DeMeo, E. Ferrara, G. Fiumara, and A. Provetti," Enhancing community detection using a network weighting strategy" . Information Sciences, 222:648-668, (2013).

[14] P DeMeo, E Ferrara, G Fiumara, and A Provetti. "Mixing local and global information for community detection in large networks". Journal of Computer and System Sciences, 2012.

[15] M. Rosvall, CT. Bergstorm, "Maps of random walks on complex networks reveal community structure", Proceedings of the National Academy of Sciences;105:1118–23 ,USA 2008.

[16] M. Porter, J. Onnela, P. Mucha, Communities in networks, Notices of the American Mathematical Society 56 (9) 1082–1097,(2009).

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

[18] Z. Wang, A. Scaglione, and R. J. Thomas, "Electrical CentralityMeasures for Electric Power Grid Vulnerability Analysis," presented at the 49th IEEE Conference on Decision and Control , pp.5792-5797, 15-17 Dec. 2010.

[19] Z. Li, R. Wang, X. Zhang, and L. Chen, "Self-organizing map of complex networks for community detection", Journal of Systems Science and Complexity, 23(5), 931-941, 2010.

[20] N.J. Van Eck, L.Waltman, "VOS: A new method for visualizing similarities between objects. In H.-J. Lenz & R. Decker (Eds.), Advances in data analysis", Proceedings of the 30th Annual Conference of the German Classification Society (pp. 299–306) , Springer (2007).

[21] L. Danon, A. Díaz-Guilera, A. Arenas," The effect of size heterogeneity on community identification in complex networks" , J. Stat. Mech. 11 (2006).

[22] L. Waltman, , N.J. van Eck, and E. Noyons, " A unified approach to mapping and clustering of bibliometric networks", Journal of Informetrics,. 4(4): p. 629-6352010.

[23] M. E. J. Newman," Fast algorithm for detecting community structure in networks", Phys. Rev. E 69 (6) 066133,(2004).

[24] W. Zachary, "An information flow model for conflict and fission in small groups", Journal of Anthropological Research, 33:452–473, 1977.

[25] D. Lusseau, K. Schneider, J. Oliver Boisseau, P. Haase, E. Slooten, and S. Dawson," The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations", Behavioral Ecology and Sociobiology, 54(4):396–405, 2003.

[26] V. Krebs, [http://www.orgnet.com/].

[27] M. E. J. Newman."Finding community structure in networks using the eigenvectors of matrices" ,Phys. Rev. E, 74:036104, 2006.

[28] P. Gleiser and L Danon." Community structure in jazz", Advances in Complex Systems, 06(04):565–573, 2003.