2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)

The Inclusion Measure for Community Evaluation and Detection in Unweighted Networks

Nikolaos Koufos
Department of Computer Science & Engineering
University of Ioannina
Ioannina, Greece
nkoufos@cs.uoi.gr

Aristidis Likas
Department of Computer Science & Engineering
University of Ioannina
Ioannina, Greece
arly@cs.uoi.gr

***Abstract*—One of the most interesting problems in network analysis is community detection, i.e. the partitioning of nodes into communities, with many edges connecting nodes of the same community and comparatively few edges connecting nodes of different communities. We introduce a new quality measure to evaluate a partitioning of an undirected and unweighted graph into communities that is called inclusion. This quality measure evaluates how well each node is included in its community by considering both its existent and its non-existent edges. We have implemented a strategy that maximizes the inclusion criterion by moving each time a single node to another community. We also considered inclusion as a criterion for evaluating partitions provided by spectral clustering. In our experimental study, the inclusion criterion is compared to the widely used modularity criterion providing improved community detection results without requiring the a priori specification of the number of communities.**

***Index Terms*—social networks, community detection, modularity, inclusion**

I. Introduction

The detection of communities is of great significance in sociology, biology, computer science and other disciplines where complex systems are often represented as graphs or networks. A graph cluster or community is typically considered as a group of nodes with high connectivity among its members and low connectivity to nodes of different communities. The general methodology when trying to detect communities involves two main steps: i) Define a quality measure (objective function), that captures the main property of community structure: nodes in the same group have higher internal than external connectivity. ii) Use search methods so that the nodes are assigned to communities, through optimization of the objective function. In many cases, the exact optimization of the objective function leads to computationally hard problems. Therefore a common approach is to employ some kind of heuristic (e.g. greedy) algorithms or other approximation techniques. An alternative approach is to consider typical clustering methods (e.g. spectral clustering) to obtain partitions and then employ the quality measure to select the best among various partitions.

A popular measure that has been widely used to evaluate the quality of a a graph partition is *modularity* [1] . The main idea behind modularity is that, given a graph partition, it measures the difference between the number of edges that exist within a community and the expected number of edges of a random graph with the same degree distribution. More specifically, given a graph $G = (V, E)$ the modularity value Q of a partition $C = \{C_1, C_2, \ldots, C_m\}$ of G is defined as:

$$Q = \frac{1}{2s} \sum_{ij} (e_{ij} - \frac{d_i d_j}{2s}) \delta(C_i, C_j)$$

where i, j denote graph nodes, e_{ij} the weight of the edge between i and j, d_i is the sum of edge weights attached to node i, s is the sum of all edge weights, C_i denote the community of node i and $\delta(u, v)$ is 1 if $u = v$ and 0 otherwise.

It has been proved that exactly optimizing modularity is a NP-complete problem [2]. Among various approaches (e.g. agglomerative clustering, simulated annealing etc), a simple heuristic algorithm [3] (usually called *Louvain algorithm*) has gained wide acceptance due to its low complexity and good performance. The Louvain algorithm implements a 'greedy node movement' strategy: it computes for each node the change in modularity obtained by moving this node to another community and selects the node movement that mostly improves modularity. A notable property of modularity is that it can be used to compare partitions with different number of communities, thus modularity optimization be can used to automatically infer the number of communities.

II. Inclusion Measure

In this work we propose a new measure for the quality of a graph partition that we called *inclusion*. Assume we are given an *undirected* and *unweighted* graph $G = (V, E)$, where $n = |V|$ the number of nodes, $e_{ij} \in \{0, 1\}, i \neq j$ and $d_i > 0$ the degree of each node i ($i = 1, \ldots, n$). If $e_{ij} = 1$ we characterize the corresponding edge as *existent*, while if $e_{ij} = 0$ we characterize the edge as *non-existent*. Assume also a partitioning $C = \{C_1, C_2, , C_m\}$ of this graph into m communities C_i. As the name indicates, our measure for evaluating the quality of a partitioning focuses on *how well a node is included in its community*. Ideally, the node should be connected to all nodes in its community and should not contain edges to nodes in other communities. The ideal case occurs when the community structure contains totally disconnected subgraphs with each subgraph having full internal connectivity.

IEEE/ACM ASONAM 2018, August 28-31, 2018, Barcelona, Spain
978-1-5386-6051-5/18/$31.00 © 2018 IEEE

In this case, the inclusion of each node should have the highest possible value. Note that when a node is not connected to some nodes of its community or it is connected to nodes of other communities its inclusion should be much lower.

Let $e_i^1(in)$ the number of *existent* edges ($e_{ij} = 1$) that connect node i to nodes in its community and let $e_i^0(out)$ the number of *non-existent* edges ($e_{ij} = 0$) from node i to nodes outside its community. The ratio $I_i^1(in) = e_i^1(in)/d_i$ expresses the percentage of node i existent edges falling inside its community and becomes maximum (equal to 1) when the node is connected only to nodes in its community ($e_i^1(in) = d_i$). The ratio $I_i^0(out) = (e_i^0(out) + 1)/(n - d_i)$ expresses the percentage of node i non-existent edges going outside its community and becomes maximum (equal to 1) when all non-existent edges go to other communities ($e_0^i(out) = n-1-d_i$). The inclusion of node i (*node inclusion*) is defined as follows:

$$I_i = \frac{I_i^1(in) + I_i^0(out)}{2} = \frac{1}{2}(\frac{e_i^1(in)}{d_i} + \frac{e_i^0(out) + 1}{n - d_i})$$

In other words, the inclusion of each node takes into account the existing edges inside its community and the non-existing edges to the other communities. If all existent edges of node i are inside its community then $e_i^1(in) = d_i$ and $e_i^0(out) = n - 1 - d_i$ and inclusion becomes maximum, $I_i = 1$. If there are non-existent edges to nodes in the same community or existent edges to nodes outside the community then the node inclusion gets lower. In case where each node forms its own community, it holds that $I_i^1 = 0$ and $I_i^0 = 1$, thus inclusion gets much smaller, $I_i = 0.5$. The minimum value of inclusion is $I_i = 1/2(n - d_i)$ and occurs when $e_i^1(in) = e_i^0(out) = 0$.

The inclusion measure I (*graph inclusion*) of a partitioning C of graph G is defined as the average inclusion over all graph nodes:

$$I = \frac{1}{n}\sum_{i=1}^{n} I_i$$

Compared to modularity the inclusion measure exhibits two notable differences. The first is that inclusion not only promotes full connectivity inside a community, but also values the absence of edges among different communities. Thus it is a multi-objective criterion, while modularity focus only on the internal connectivity of a community. The second difference is that inclusion focuses primarily on evaluating nodes and not communities as happens with modularity.

As it can be observed in Fig. 1, the inclusion value tends to increase as the quality of the partition increases. In case (a) the graph is under-partitioned into three communities and the inclusion of the partition is I=0.85. In the second case, the graph is separated in four communities, which is the visually the best solution, and the inclusion is I=0.89. In the third case, the graph is over-partitioned into five communities and the inclusion is I=0.80. Is it clear that the quality of the three partitions aligns with the corresponding inclusion values and the maximization of inclusion can reveal the correct number of communities.

Another indicative example is that of a fully connected graph, thus for all edges $e_{ij} = 1$ and $d_i = n - 1$ for all nodes i. Obviously the best partitioning occurs when all nodes are in a single community. It can be easily shown that in such case the corresponding inclusion value is maximum: $I = I_i = 1$. On the hand, a bad solution occurs when each node forms its own community and the corresponding inclusion value is $I = I_i = 0.5$. In another example where the graph is partitioned in isolated fully connected subgraphs the inclusion of the optimal partition is $I = I_i = 1$.

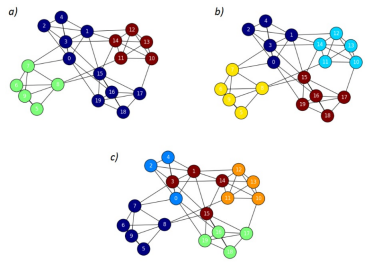

Fig. 1. (a) Graph partitioned into three communities, I = 0.85 b) Graph partitioned into four communities, I = 0.89 c) Graph partitioned into five communities, I = 0.80.

III. Inclusion for Community Detection

In order to exploit the inclusion measure for community detection search strategies should be developed aiming to provide community partitions of maximal inclusion.

A. Greedy Node Movement

A typical first approach to follow is the agglomerative strategy: starting each node in its own community, we iteratively merge two communities as long as the total inclusion of the new partition is increasing. However, it well-known that such agglomerative approach suffers from increased computational complexity. For this reason, inspired by the fast modularity approach (Louvain algorithm) [3], we implemented a strategy based on greedy node movement: each time a single node is allowed change community (ie. to move between communities) instead of merging whole communities. The initialization is the same as previously, with every node forming its own community. At each iteration, we calculate for every node i (currently in community C_k) and for every community C_l ($l \neq k$), the difference $\Delta I_i(C_k, C_l)$ in graph inclusion caused by moving i from C_k to C_l:

$$\Delta I_{1i} = \frac{1}{2}\sum_{j \in C_k, j \neq i} \{(1-e_{ij})(\frac{1}{n-d_j} + \frac{1}{n-d_i}) - e_{ij}(\frac{1}{d_j} + \frac{1}{d_i})\}$$

$$\Delta I_{2i} = \frac{1}{2}\sum_{j \in C_l} \{e_{ij}(\frac{1}{d_j} + \frac{1}{d_i}) - (1-e_{ij})(\frac{1}{n-d_j} + \frac{1}{n-d_i})\}$$

$$\Delta I_i(C_k, C_l) = \frac{1}{n}(\Delta I_{1i} + \Delta I_{2i})$$

Note that, since e_{ij} is zero or one, only one of the two terms needs to be computed in the above formulas. Based on the values of $\Delta I_i(C_k, C_l)$, we can decide an appropriate node

movement and this procedure is repeated until there is no possible single node movement with $\Delta I_i > 0$.

There are several strategies that could be followed to decide on the single node movement to be implemented at each iteration. One is to implement the best among all possible node movements A faster alternative adopted in our approach is to earlier accept movements that improve inclusion without examining all nodes. To implement such a strategy three decisions should be made: i) whether to examine nodes sequentially or to select nodes randomly, ii) whether to examine movements only to communities adjacent to the community of the examined node or movements to every other community, iii) whether for each node to examine all possible movements to other communities and find the best movement or to accept the first encountered movement that improves inclusion. We have implemented and compared the eight possible strategies to maximize inclusion based on single node movement between communities. Considering both clustering performance and computational cost we selected as more appropriate the strategy which at each step: i) examines all nodes sequentially, ii) allows node movement to adjacent clusters only, iii) accepts the first encountered movement that increases inclusion.

B. Spectral Clustering

As already mentioned, a convenient characteristic of the inclusion measure is that it can be used to compare among partitions with different number of communities. Therefore another approach to community detection is to produce several graph partitions using spectral clustering [5] and use inclusion as a quality measure to select the best partition. More specifically, for each graph, spectral clustering is executed for several values of the number of communities m (eg. for m=2 to 20), the inclusion of the partition for each m is computed and the partition of maximum inclusion is considered as the final result.

IV. Experimental Results

In order to empirically evaluate the proposed approach, we used several synthetic graphs as well as a real-world graph with known ground-truth partitions. In order to evaluate the quality of the compared methods, we computed the similarity of an obtained solution with the ground truth solution employing two commonly used measures: Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI). Both measures return values between zero and one and higher values indicate better clustering performance. We also present the number communities (clusters) of each solution for comparison with the number of communities in the ground truth solution.

It is well-known that methods based on modularity optimization suffer from the 'resolution limit' problem [4]. More precisely, modularity optimization might fail to detect clusters smaller than a scale number, which is mainly dependent on the graph size. This limitation is important because real world networks, often contain communities of various sizes. In order to test whether inclusion optimization can deal with the resolution limit problem, we created ring graphs containing small fully connected communities forming a ring with only one edge from one community to the next one (see Fig. 1). We compare our inclusion maximization method based on greedy node movement to the Louvain algorithm maximizing modularity [3]. In Table I we provide comparative performance results on ring graphs with different numbers of communities m and nodes per community L (thus the number of graph nodes is $n = m \times L$).

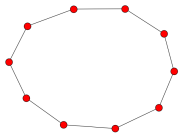

Fig. 2. A ring graph with m=10 fully connected communities. Each circle represents a community.

TABLE I
Compartative experimental results on ring graphs

	Inclusion			Modularity		
	Clusters	**NMI**	**ARI**	**Clusters**	**NMI**	**ARI**
m=10, L=3	10	1	1	5	0.836	0.524
m=15, L=4	15	1	1	8	0.872	0.587
m=30, L=5	30	1	1	16	0.899	0.617
m=50, L=6	50	1	1	25	0.907	0.616

From the results in Table I it is clear that, in contrast to modularity, the proposed inclusion measure does not suffer from the resolution limit problem and the inclusion optimization strategy always discovers the ground truth community structure, providing solutions with NMI and ARI equal to 1.

In order to produce synthetic graphs with specific properties, we implemented a function that creates a graph given the following parameters: number of nodes (n), number of communities (m), community size vector (CS) specifying the size of each community, internal edge probability vector (IEP) specifying for each community the probability of internal edge existence and, finally, external edge probability (EEP) which is the probability of each node to have an edge with nodes outside its community.

In all cases the external edge probability (EEP) was fixed to 0.15. For given values of n and m, by adjusting the parameters CS and IEP we created five categories of synthetic graphs with different characteristics regarding the distribution of community size and internal connectivity density: i) balanced and dense communities (B&D), ii) balanced communities of decreasing density (B&DD), iii) dense large communities and low density small communities (DL&SS), iv) decreasing size and decreasing density (DS&DD), v) increasing community size and decreasing density (IS&DD).

At first we compared our greedy inclusion maximization algorithm to the Louvain algorithm that maximizes modularity [3] in a similar way. For each of the above graph categories, we created: i) a set of 100 graphs with $n = 60$ nodes and $m = 4$ communities and ii) a set of 100 graphs with $n = 80$

and $m = 5$. For each graph set, we applied the two compared methods on all graphs. For each obtained partition we store the number of communities (graph clusters) as well as the NMI and ARI values. Average results (over the 100 graphs of each set) are presented in Table II. As it can be easily observed from

TABLE II
PERFORMANCE RESULTS ON SYNTHETIC GRAPHS USING GREEDY MAXIMIZATION OF INCLUSION (PROPOSED ALGORITHM) AND MODULARITY (LOUVAIN AGORITHM).

	n=60, m=4					
	Inclusion			Modularity		
	Clusters	**NMI**	**ARI**	**Clusters**	**NMI**	**ARI**
B&D	4	1	1	4	1	1
B&DD	4.01	0.995	0.995	3.99	0.996	0.994
DL&SS	4	0.998	0.999	3.48	0.954	0.955
DS&DD	4.09	0.975	0.982	3.14	0.906	0.901
IS&DD	3.96	0.994	0.993	3.71	0.968	0.963
	n=80, m=5					
	Inclusion			Modularity		
	Clusters	**NMI**	**ARI**	**Clusters**	**NMI**	**ARI**
B&D	5	1	1	4.99	0.999	0.998
B&DD	5.03	0.931	0.913	4.73	0.902	0.865
DL&SS	4.91	0.995	0.991	4.36	0.957	0.932
DS&DD	4.98	0.928	0.932	3.95	0.880	0.860
IS&DD	5.03	0.950	0.940	4.56	0.9225	0.895

Table II, for the first graph category (B&D) (relatively easy problems), both inclusion and modularity discover the ground truth solution. For the other graph categories where partitioning becomes harder, the superiority of inclusion is very clear. In some cases the performance of modularity decreases considerably, especially in what concerns the detected number of communities (clusters). On the contrary, the inclusion-based method consistently provides higher NMI and ARI values and also estimates very accurately the ground-truth number of communities.

We also considered a *real graph*, namely the *American College Football* dataset [6], with 112 nodes and 616 edges that represents a network of American football games between colleges during regular season Fall 2000. The ground-truth solution contains 12 communities. Optimizing inclusion yields a solution with 11 communities, NMI=0.91 and ARI=0.86, outperforming the modularity-based approach that gives a solution with 10 communities, NMI=0.89 and ARI=0.81.

Finally we compared inclusion and modularity when used as criteria to select the best among a set of solutions provided by applying *spectral clustering* [5] on the edge matrix. More specifically, for a given graph we ran the spectral clustering from m=2 to m=20 clusters and kept the partition that maximized modularity and inclusion respectively. For each of the five graph categories defined previously, we created synthetic sets containing 20 graph instances for n=1000 nodes and m=8 communities as well as for n=2000 and m=16. Average performance results are presented in Table III. The conclusions that can be drawn are analogous to those in Table II. As it can be observed from Table III, for the first graph category (B&D) both inclusion and modularity discover the ground truth solution while for harder partitioning problems inclusion is superior. For the difficult DS&DD problems inclusion leads to solutions of good quality, while the solutions selected using modularity are clearly inferior.

TABLE III
PERFORMANCE RESULTS ON SYNTHETIC GRAPHS USING SPECTRAL CLUSTERING FOR GRAPH PARTITIONING.

	n=1000, m=8					
	Inclusion			Modularity		
	Clusters	**NMI**	**ARI**	**Clusters**	**NMI**	**ARI**
B & D	8	1	1	8	1	1
B & DD	8	0.997	0.997	7.3	0.967	0.909
DL & SS	8	1	1	6.9	0.979	0.973
DS & DD	7	0.975	0.977	5	0.893	0.783
IS & DD	8	0.999	0.999	8	0.999	0.999
	n=2000, m=16					
	Inclusion			Modularity		
	Clusters	**NMI**	**ARI**	**Clusters**	**NMI**	**ARI**
B & D	16	1	1	16	1	1
B & DD	14.8	0.963	0.899	13	0.933	0.702
DL & SS	15.7	0.998	0.997	13.9	0.985	0.975
DS & DD	13.4	0.976	0.963	10.1	0.908	0.708
IS & DD	15.4	0.9818	0.942	13.8	0.960	0.862

V. CONCLUSIONS

In this work we have introduced inclusion for community detection and evaluation in unweighted graphs. Inclusion is node-centric, in the sense that is measures how well a node is included in its community, and promotes the absence of edges between nodes in different communities. We have also presented an approach to maximize inclusion based on greedy node movement between communities in analogy to the Louvain algorithm for maximizing modularity. We have experimentally shown that inclusion does not suffer from the resolution limit problem and that it clearly outperforms modularity, especially in hard partitioning cases.

Future work could focus on comparing inclusion to other related measures and also testing the approach on various community detection applications arising in biological, social and other types of networks. Another important research direction concerns the possible use of inclusion to detect communities in weighted graphs. In such a case, a considerable adaptation of the method would be necessary.

REFERENCES

[1] M. Newman and M. Girvan, "Finding and Evaluating Community Structure Networks," Phys. Rev. E, vol. 69, no.2: 026113, 2004.

[2] U. Brandes, D. Delling, M. Gaertler, R. Gorke, M. Hoefer, Z. Nikoloski and D. Wagner, "On Modularity Clustering," IEEE Trans. on Knowledge and Data Engineering, vol. 20, no. 2, 2008.

[3] V. Blondel, J. Guillaume, R. Lambiotte and E. Lefebvre, "Fast Unfolding of Communitties in Large Networks," Journal of Statistical Mechanics, vol. 10, P10008, 2008.

[4] S. Fortunato and M. Barthelemy, "Resolution limit in community detection," Proc. Nat. Acad. Sci. USA, vol. 104, no. 1, pp. 36–41, 2007.

[5] A. Ng, M. Jordan and Y. Weiss, "On Spectral Clustering: Analysis and an algorithm," Proc. NIPS 2001, vol. 2, pp. 849–856, 2001.

[6] M. Girvan and M. Newman, "Community structure in social and biological network," Proc. Nat. Acad. Sci. USA, vol. 99, pp. 7821-7826, 2002.