Innovative Methods and Measures in Overlapping Community Detection

Nazar Buzun and Anton Korshunov

Institute for System Programming Russian Academy of Sciences
Moscow 109004, Russia
{nazar,korshunov}@ispras.ru
http://www.ispras.ru

Abstract. Cluster structure is one of the main features of social graphs. Many algorithms have been proposed in recent years that are capable of revealing fuzzy communities. But a lot of them tend to degrade in some special cases, for example when nodes assigned to more than two groups. Taking into account that such highly overlapping membership is rather common for many social networks, it becomes obvious that there is a need for flexible techniques and detecting the scope of their effective applicability for various network configuration parameters. This article focuses on the resistance to cluster's growth intersection with emphasis on local fitness function's optimization. The testing of the modern fuzzy clustering methods and generalized classical approaches is performed. Depending on the scale of fuzziness the conclusion is provided about the applicability of certain algorithm classes with common methodology and their representatives.

Keywords: community detection, fuzzy clustering, social networks, social graph mining, local optimization

1 Introduction

Networks are natural representations of various complex systems from society, biology, engineering and other fields. The set of networks is characterized by mesoscopic organisational level inside groups of vertices, which comprise units with a big number of links. Such units are referred to as clusters (or communities or modules).

The universal definition of community partition is stated here only in a qualitative form. It is due to a big variety of formal community detection problem statements and different final goals in particular applications. So far, the problem of partition quality estimation appears to be non-trivial.

In the recent years this research domain has been focused on social and natural networks, whose internal structure cannot be detected by classical clustering algorithms. In these areas analogues of communities are the lists of friends and subscribers, friends circles in Google+ and some social interest groups.

One can figure out several applications of useful information obtained from the network partitioning into communities: system functional units detection;

identification of the community vertices similarity; vertices from a community can be classified in accordance with their position (leaders, linking ones and so on); convenient method of system visualization; vertices' attributes learning on the basis of general attributes of communities which include them. Furthermore, one can specify several methods of machine learning: classification, recommendation, prediction, filtration of non-typical elements (where case division into modular units is a sub-problem). In addition, it's appropriate to mention issues of optimal storage, placement and compression of data; analysis of information distribution; influence inside the global networks.

In spite of the applied problems variety, let us sort out the most general requirements to the methods. Here we also regard some important features of social networks structure.

- The vertex could be found in more than one communities with various degrees of belonging (*fuzzy clusters*) [2-10,24,27]
- Communities may have a *hierarchical structure* [4,6,8,11,22,24] that is required for the efficient management in large-scale organizations, and its presence stresses the stability of the system [12].
- In addition high density of edges doesn't indicate the cluster. Therefore, in order to cut-off "pseudo-communities" a probability of a particular subgraph configuration (*"statistical significance"*) is calculated, under assumption of random edges distribution hypothesis (for the given values of vertices degree) [9,13]. For this purpose it makes sense to look for "significant" subgraphs by taking into account *weak* links [8]. The link (edge) between nodes assumed to be weak if it is not a part of a triangle.
- In some cases (such as for defining attributes of vertices) one need to manipulate vertices and edges with *additional parameters* [1,2,14]. But the majority of current algorithms take only one input parameter like weight of a link.
- While searching for implicit individual user communities (circles of friends, *egomunities*) the execution time and access to graph structure are often limited. Usually in such a case only second friends' neighborhood is known.
- One may also put an additional problem of studying the *community dynamics* [15].

This article focuses on identifying overlapping communities in large networks ($n = 10^8, m \sim n$) with a high coefficient of intersection ($r \sim 10$). Here P is a set of communities $G = (E, V), m = |E|, n = |V|, r = \sum |P_i|/n$. These characteristics are inherent for many real networks. In the Fig.1 one could find more specific communities settings as a vertexes degree function in social Facebook's subgraphs.

We are going to discuss a variety of modern algorithms that is initially characterized by the ability to identify fuzzy communities. In addition, several universal generalizations of classical algorithms will be proposed for the case of graphs with overlapping clusters. The first purpose of this research is to determine of algorithm classes in accordance with their basic ideas. The second aim is to identify the most relevant methods of fuzzy (overlapping) clustering and ways to assess the quality of graph partition.

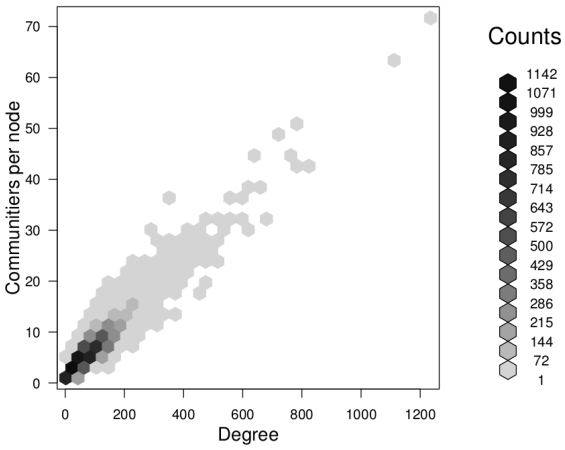

Fig. 1. Social network properties. Coefficient of intersection in Facebook's subgraphs. Shows dependence of communities count that includes node and number of its neighbors. Right column illustrates amount of vertices in the point with certain color.

2 Graph clustering methods overview

Considered clustering methods is divided into classes according to the generality of underlying principles and specification of community definition.

2.1 Null graph model

In methods within this class the given configuration of edges is compared with their uniform distribution for each vertex in graph. For this reason one should use a probabilistic graph model (null model). In its definition the expectation of the nodes degree is fixed. Follows the edge existence probability *(i,j)* is defined as a composition of node's degrees divided by the doubled count of edges: $P_{ij} = \frac{k_i k_j}{2m}$. The classic variant here is to maximize the target **modularity** function and its modifications [16-21], that characterize sum of differences between the total number of edges in the community and its mathematical expectation:

$$Q = \frac{1}{2m} \sum_{c \in P} \sum_{i,j \in c} [A_{ij} - Pr(A_{ij} = 1)],$$

where P - communities set, A - adjacency matrix.

Similarly, instead of edges triangles and more larger cliques could be taken into account. Considering that a link between vertices is weak if it is not a triangle's edge, the community is optimized to increase amount of internal triangles. At the same time the count of adjacent triangles that have exactly two nodes inside the community should be reduced [8,20].

Originally modularity measure was introduced to describe disjoint partitions, but there are some generalizations for the case of overlapped communities [17,21]. Additionally it is worth mentioning its quantum-mechanical modification [18,19], that allows to improve the *resolution limit* and to give it an energetic sense. So it becomes a hamiltonian for a set of particles with various spin values (*spinglass* [19]).

A more common approach is the detection of "significant" clusters. In this case algorithms tend to include in each module those nodes that are most strongly connected to each other. Such cluster type should have a low probability of gathering better interacted users according to random graph model. But due to correlations it is rather complicated to calculate the statistics of the internal connections. Really it is more practical to fix inner community structure and calculate the statistics for the external vertices. This inform us of how much of users for some group are compatible with the null model distribution (*oslom*) [9].

Alternatively one could define probability of link existence to be proportional to the number of communities to which the link belongs (*moses*) [5] and then find the maximum likelihood. This model doesn't account for node degree distribution.So it leads to worse results in some cases but it is rather stable in implementations with high fuzziness.

2.2 Random walks

Here we have three most common methods.

infomap [10,22]: In this case, the clusters are formed to minimize the description length of a random walk in the graph. One of the code length's estimators is entropy that is widely used in various information theory branches. Based on it, [22] propose to consider the following function as a partition quality measure:

$$L(P) = qH(Q) + \sum_i p_i H(P_i),$$

where q - probability that the random walker switches module, p_i- fraction of within module movements, H(Q)- entropy of module names, H(P_i)- entropy of inner module movements including its exit code, i - module number.

walktrap [23]: Here the formation of communities is based on the following proposition: Let the vertices i, j belong to the same cluster, then

$Pr(k \to i, t) \approx Pr(k \to j, t)$ for all $k \in V$, where Pr - transition matrix from a random walk process.

betweenness [9]: Using the measure called "betweenness" on the set of edges (the higher runs count along the edge during a random walk, the greater is measure value). Edges with a high "betweenness" are naturally considered as links between communities (*conga, GN*) [3].

2.3 Local expansion

In a local study and formation of the cluster is generally considered the ratio between the amount of interior edges or triangles and the exterior ones (*cohesion*

[8], GCE [7]). In some approaches link density is additionally compared with its possible maximum. And all such optimizations are usually done disregarding the rest of the graph structure. So the distinguishing feature of this class is an iterative addition of new nodes to the cluster and removal of the existing ones independently of any other clusters. Communities can also be formed on the basis of similarity to a complete graph or a set of connected cliques with different sizes (*CFinder, GCE*) [7]. Besides that, above-mentioned *"statistical significance"* can be used as a local characteristic of similarity between a subgraph and real community. There is also a set of methods in this section which allow independent subgraphs detection to provide high-value influence of vertices within the module (*moduland* [6]). For this class methods a selection of intersecting communities is rather natural, but at the other hand there are some difficulties with the subsequent formation of the final partition in the graph.

2.4 Agent based model

In this case an epidemic process is generated that usually represent a speakers-listener model (*copra* [3], *slpa* [27]). During the execution we should fix a listener node and start gathering information from each of its neighbours. So every such node could save recommendations per each module from received messages. After that it could give an advice to others basing on obtained experience. Here we don't have to define any functional for community, we only spread labels between nodes according to pairwise interaction rules.

There are two types of execution of such epidemic process: synchronous and asynchronous. Synchronous type is more preferable because it prevents monster communities and is easily parallelized. But at the other side it may trigger oscillation phenomenon which should be calmed down with colouring phase (linked nodes get different colours and aren't handled synchronously). .

2.5 R_n metric space

Another elegant approach is to assign coordinates to vertices in the graph [26]. Such coordinates are components of the eigenvectors for the normalized Laplacian matrix L.

$$L_{ij} = \begin{cases} 1, & i = j \\ -\frac{1}{\sqrt{k_i k_j}}, & i - edge - j \\ 0, & else \end{cases}$$

This method of clustering is very useful if one wants to take in account some additional attributes of the vertices.

Summing up the review we can distinguish such methods as spinglass, infomap, wolktrap that have the highest rates of *Normalized Mutual Information* [24] (for the case of **disjoint** communities) with a relatively short execution time and the possibility of parallel execution [25].

3 Generalization methods in the case of overlapping communities

Unfortunately, not all algorithms from the considered classes support fuzzy clusterization. That is why methods of their generalization are required.

3.1 Static

Using the measure of "betweenness" on the set of nodes, one can divide each vertex with high value into two ones connected by an edge. Thus after clustering of the modified graph some user parts could be included into different modules and consequently perform overlapping communities.

The alternative is a generation of line graphs (where edges are turned to vertices and vertices are turned to zero or several edges) and successive edge clustering.

3.2 Dynamic

Introducing membership coefficients for the vertices (which are equal to probabilities of being the member of the particular community), one then assigns a vertex to several classes simultaneously during the algorithm's run. As a first approximation for the membership coefficient one can use the following functions:

– Individual contribution to increase of objective function:

$$Pr(V_i \in P_k) \sim Q(V_i \in P_k) - Q(P_k \backslash V_i) = \triangle Q_{ik}$$

– Probability of being at the particular energy level:

$$Pr(V_i \in P_k) = e^{-\beta Q(V_i \in P_k)} / \sum_S e^{-\beta Q(V_i \in P_S)},$$

where Q - objective function, - value that is inversely proportional to the overlapping coefficient.

It worth noticing that introduction of membership coefficients often improves partition into non-overlapping communities. The main idea here is that by setting probabilities of vertex transition to other communities (staying with some probability in the original one) we let other vertices know about their behaviour tactics. Thereby the following expression can be used in order to set up the coefficients in this case:

$$Pr(V_i \in P_k) \sim \triangle Q_{ik} - \min_h(\triangle Q_{ih}),\ V_i \in P_h$$

$$Pr(V_i \in P_{hmax}) \sim 0.1$$

4 Implementation methods

Let us try sort out several implementation ways without binding ourselves to a particular community detection algorithm.

1. Greedy algorithm (used by the majority of the algorithms mentioned above): Originally each vertex is a community itself. Then at the each step of the algorithm every vertex selects the communities to be appended to by comparing relative increases of objective function. A completion phase in this implementation is a clustering of obtained modules, whose unions improve final graph partition.
2. Central vertices: In this method one sets several central users. So the others are gradually attached to them by selecting the closest cluster.
3. Recursive graph partition into two or more parts: In the beginning vertices are randomly partitioned. Then those of them which give the maximal objective function increase are relocated.
4. In the case of local optimization one can recommend to apply the following scheme:

   Single-cluster analysis → Internal structure validation → Clusters consolidation → Membership coefficient computation ↻

   At single-cluster analysis stage each community either gets new nodes or loses those nodes weakly connected with the rest of vertices in the community. So to reach an extremum in this process we should define a function $F(n_{in}, m_{in}, m_{ext}, k_{in})$ depended on internal nodes count, internal and external edges set, links between the community, and the considered node.

   Also, one could use order statistics to work with ranks defined vertex-community closeness. Then to optimize cluster structure one search for a minimum of rank distribution value: $min[F_q(r_q)]$, where q - order number of rank. For this purpose one of null graph models should be chosen (Girvan and Newman [16] or Molloy and Reed [28], for example). If the first stage has a probabilistic character it repeats several times. The final cluster contains those vertices that appear to be included into the group more than fixed times. The considered subgraph is significant cluster if the single-cluster analysis yields a non-empty subgraph in more than definite percent of iterations.

   At the following step of clusters consolidation we may unite some closely located modules or divide them to more small parts. In this case the following measures are usually used:

   common nodes fraction: $\frac{|P_i \cap P_j|}{\min(P_i, P_j)}$

   edges density: $dQ = (\sum A_{ij} - Exp(\sum A_{ij}))$,

   where Q - objective function, P - communities set, A - adjacency matrix.

   Another way is to run a single-cluster analysis on the subgraph of two modules that are to be united or separated.

5 Testing

LFM [1] benchmark algorithm is used as a generator of networks with overlapping cluster structure. In order to investigate the algorithms performance for various degree of community overlapping, two sets of test graphs with predefined partition were generated. The following variables were given to the generator as input parameters: n - number of nodes, k - vertex degree average, k_{max} - maximal value of vertex degree, $|Pi|$ - number of nodes in a cluster, τ_1 - value of the exponent of power law distribution for vertex degree, τ_2 - value of the exponent of power law distribution $|Pi|$, μ - averaged normalized vertex degree inside parent community, on - number of vertices owned by more than one community, om - number of communities containing fixed vertex. Parameters of the graphs from the first community differ by the 'om' value, from the second community - by the 'on' value.

For a comparison of partitions obtained by different methods (Fig 2: fig.1, fig.2, fig.4), let us introduce measure Normalized Mutual Information (I_{norm}) [24] based on the following assumption: if two graph partitions are similar then there is a little information required to obtain the first partition when the second one is known.

$$I(X,Y) = H(X) - H(X|Y)$$

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

where H - Shannon entropy

In addition to graph partitions from the first set, let us calculate modularity generalized for fuzzy clustering.

From the results, one can conclude that for the case of considerable overlapping only following methods show acceptable performance: **oslom** [9], **moses** [5], **gce** [7], which are representatives of the local optimization class. In particular, first two of them prove the effectiveness of exploiting statistical significance as an individual (local) characteristic of cluster structure. It is also worth to notice the effectiveness of overlapping edge clusterization, that can be applied to the networks of small and moderate size. For the networks of large size with insignificant overlapping one can exploit methods of complexity no more than $O(n^{\alpha}), \alpha \in [1,2]$ - fuzzy infomap [10], gce [7], spinglass [18] generalization, slpa [27].

Besides this, after analyzing plots of modularity values (Fig 2: fig3), its worth to emphasize a discrepancy of NMI partition quality while increasing 'on'. Therefore, modularity provides impartial partition estimate only if overlapping coefficient 'r' is small.

We also have tested the same methods in local community detection task. Our purpose here was to identify fractions of global network clusters that are the friend circles. We deals only with the second area of the fixed central user, that is

[1] http://sites.google.com/site/andrealancichinetti/files

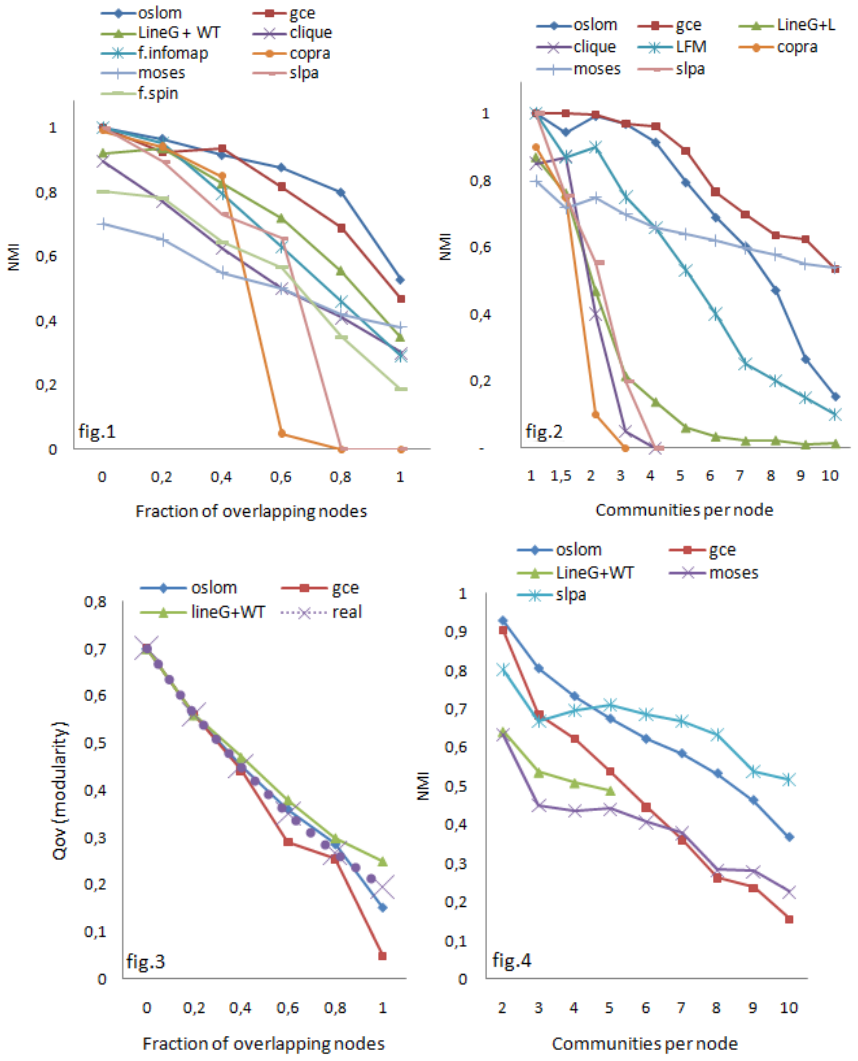

Fig. 2. Testing algorithms of overlapping community detection.
fig.1,fig.3: $n = 2000, k = 15om, kmax = 45om, |P_i| \in [15, 60], \tau_1 = 2, \tau_2 = 0, \mu = 0.2, on \in \{0, 1000, 2000\}$, $om \in \{1, 1.5, 2, 3, ..., 9, 10\}$
fig.2: $n = 1000, k = 20, kmax = 50, |P_i| \in [20, 100], \tau_1 = 2, \tau_2 = 1, \mu = 0.3, on \in \{0, 200, 400, ..., 1000\}$, $om = 2$
fig.4: $n = 4000, k = \max(3om, 10), kmax = 3k, |P_i| \in [20, 80], \tau_1 = 2, \tau_2 = 1, \mu = 0.3, on = 800$, $om \in \{2, 3, ..., 10\}$

the graph information about friends and connections between them. Such kind of local communities (egomunities) could be obtained from the corresponding global ones generated the same way as mentioned above.

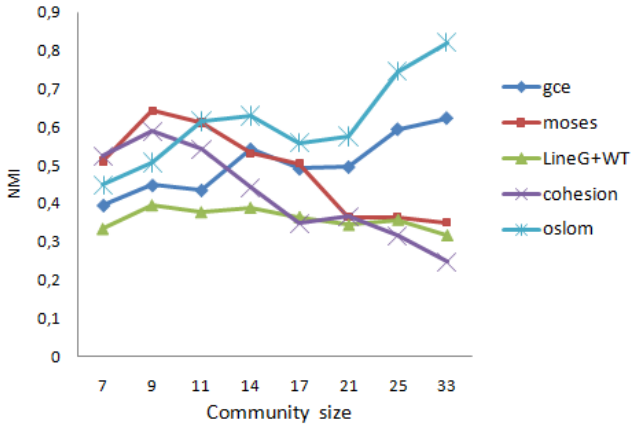

Fig. 3. Testing algorithms of user's second neighborhood egomunity(local community) detection. $n \in [30, 250], |P_i| \in [7, 33], \tau_1 = 2, \tau_2 = 0, \mu = 0.2, om = 6$

In case of local tests attention is drawn to the instability of using "statistical significance" [9] with the small circles of friends. So here in some situations (Fig.3) algorithms that do not use null graph model [16,28] work more efficiently: *cohesion* [8]. On the other hand *moses* [5] utilizing alternative random graph model is quite suitable in such cases. But if the size of the friends neighborhood is rather large the methods similar to *oslom* [9] have higher NMI scores.

6 Conclusion

In summary, several basic features of social and natural graphs were pointed out and algorithms were divided into five classes. Also several different types of their generalization were proposed, and main variants of their implementation were provided. Artificially created networks were used to compare an applicability of the most modern methods. We tested the methods with various network generator parameters. The most effective ones were identified for the particular overlapping coefficient values.

One of the plausible directions of further research is an investigation of weak and strong features of the discussed algorithm classes depending on graph properties and application goals. Herewith all features of social networks mentioned in the beginning of the paper will also be taken into account. In particular, among considerable enough problems are: hierarchical structure detection and methods of its assessment, clustering of graphs with attributes (ordered graphs [14]) on a set of vertices and edges. The last is a task of the highest priority for unknown attributes prediction. Also accumulation of the results of the conducted experiments may possibly result in the development of supervised graph analyser. It will determine at which parts of a graph it would be possible to effectively apply a particular method.

References

1. Lei Tang. 2010. Learning with Large-Scale Social Media Networks. Ph.D. Dissertation. Arizona State University, Tempe, AZ, USA. Advisor(s) Huan Liu. AAI3425805
2. Zhang S, Wang RS, Zhang XS. 2007. Identification of overlapping community structure in complex networks using fuzzy c-means clustering. Physica A 374: 483490.
3. Gregory S. 2007. An algorithm to find overlapping community structure in networks. Berlin, Germany: Springer-Verlag. pp 91102. https://www.cs.bris.ac.uk/~steve
4. Y Ahn, JP Bagrow, S Lehmann. 2010. Link communities reveal multi-scale complexity in networks. Nature 466, 761764.
5. AF McDaid, NJ Hurley. 2010. Using Model-based Overlapping Seed Expansion to detect highly overlapping community structure. In: ASONAM 2010. http://sites.google.com/site/aaronmcdaid/moses
6. Kovacs IA, Palotai R, Szalay MS, Csermely P. 2010. Community landscapes: An integrative approach to determine overlapping network module hierarchy, identify key nodes and predict network dynamics. PLoS ONE 5: e12528
7. Lee C, Reid F, McDaid A, Hurley N. 2010. Detecting highly overlapping community structure by greedy clique expansion. Poster at KDD 2010.
8. A. Friggeri, G. Chelius, and E. Fleury. 2011. Egomunities, Exploring Socially Cohesive Person-based Communities. NRIA, Research Report RR-7535, 02 2011
9. A. Lancichinetti, F. Radicchi, J. Ramasco, S. Fortunato. 2011. Finding Statistically Significant Communities in Networks. PLoS ONE 6(4): e18961. http://santo.fortunato.googlepages.com/inthepress2
10. AV Esquivel, M Rosvall. 2011. Compression of flow can reveal overlapping modular organization in networks. Phys. Rev. X 1, 021025 (2011). https://sites.google.com/site/alcidesve82

11. Clauset A, Moore C, Newman MEJ. 2008. Hierarchical structure and the prediction of missing links in networks. Nature 453: 98101.
12. Simon H. 1962. The architecture of complexity. Proc Am Phil Soc 106: 467482.
13. Lancichinetti A, Radicchi F, Ramasco JJ. 2010. Statistical significance of communities in networks. Phys Rev E 81: 046110
14. Gregory S. 2011. Ordered community structure in networks. Physica A: Statistical Mechanics and its Applications (December 2011)
15. Mucha PJ, Richardson T, Macon K, Porter MA, Onnela J. 2010. Community Structure in Time-Dependent, Multiscale, and Multiplex Networks. Science 328: 876.
16. M. E. J. Newman. 2006. Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E 74, 036104 (2006)
17. V. Nicosia,G. Mangioni,V. Carchiolo,M. Malgeri. 2008. Extending modularity definition for directed graphs with overlapping communities. J. Stat. Mech. P03024 (2009).
18. J. Reichardt, S. Bornholdt. 2008. Statistical Mechanics of Community Detection. Phys. Rev. E 74 (1) (2006) 016110
19. P. Ronhovde, Z. Nussinov. 2009. Multiresolution community detection for megascale networks by information-based replica correlations. Phys. Rev. E 80 (1) (2009) 016109
20. A. Arenas, A. Fernandez, S. Fortunato, S. Gomez. 2008. Motif-based communities in complex networks. J. Phys. A 41 (22) (2008) 224001.
21. A Lazar, D Abel, T Vicsek. 2009. Modularity Measure of Networks With Overlapping Modules. IOP Publishing, Pages: 18001
22. M Rosvall, CT Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. Proc Natl Acad Sci USA 105: 11181123. `http://www.tp.umu.se/~rosvall/code.html`
23. P Pons, M Latapy. 2005. Computing communities in large networks using random walks. Sci. 3733 (2005) 284293.
24. A. Lancichinetti, S. Fortunato, J. Kertesz. 2009. Detecting the overlapping and hierarchical community structure in complex networks. New J. Phys. 11, 033015, 2009
25. Santo Fortunato. 2009 Community detection in graphs. Physics Reports , 486, 75 174
26. L Donetti, M.A. Mutoz. 2004. Detecting network communities: a new systematic and efficient algorithm. J. Stat. Mech. P10012 (2004).
27. Xie, J., Szymanski, B. K., and Liu, X. 2011. Slpa: Uncovering overlapping communities in social networks via a speaker-listener interaction dynamic process. In IEEE ICDM 2011 Workshop on DMCCI
28. M. Molloy, B. Reed. 1995. A critical point for random graphs with a given degree sequence. Random Structures and Algorithms, Vol. 6, no. 2 and 3 161-179.