Combined method to detect communities in graphs of interacting objects[1]

Sofiya Y. Lobanova
BSc Program Student
Tikhonov Moscow Institute of Electronics and Mathematics
National Research University Higher School of Economics
Address: 20, Myasnitskaya Street, Moscow, 101000, Russian Federation
E-mail: s.lobanova@usabilitylab.net

Alexander A. Chepovskiy
Associate Professor,School of Applied Mathematics
Tikhonov Moscow Institute of Electronics and Mathematics
National Research University Higher School of Economics
Address: 20, Myasnitskaya Street, Moscow, 101000, Russian Federation
E-mail: aachepovsky@hse.ru

Abstract

In this paper, we propose and implement a method for detecting intersecting and nested communities in graphs of interacting objects of different natures. For this, two classical algorithms are taken: a hierarchical agglomerate and one based on the search for k-cliques. The combined algorithm presented is based on their consistent application. In addition, parametric options are developed that are responsible for actions with communities whose sizes are smaller than the given k, and also with single vertices. Varying these parameters allows us to take into account differences in the topology of the original graph and thus to correct the algorithm.

The testing was carried out on real data, including on a group of graphs of a social network, and the qualitative content of the resulting partition was investigated. To assess the differences between the integrated method and the classical algorithms of community detections, a common measure of similarity was used. As a result, it is clearly shown that the resulting partitions are significantly different. We found that for the approach proposed in the article the index of the numerical characteristic of the partitioning accuracy, modularity, can be lower than the corresponding value for other approaches. At the same time, the result of an integrated method is often more informative due to intersections and nested community structure.

A visualization of the partition obtained for one of the examples by an integrated method at the first and last steps is presented. Along with the successfully found set of parameters of the integrated method for small communities and cut off vertices in the case of social networks, some shortcomings of the proposed model are noted. Proposals are made to develop this approach by using a set of parametric algorithms.

Key words: graph, graph analysis, community detection, graph structure, social network analysis, big data.

Citation: Lobanova S.Y., Chepovskiy A.A. (2017) Combined method to detect communities in graphs of interacting objects. *Business Informatics*, no. 4 (42), pp. 64–73. DOI: 10.17323/1998-0663.2017.4.64.73.

[1] This work was supported by the Russian Foundation for Basic Research (projects No. 16-29-09546 and No. 16-07-00641)

Introduction

Today, a wide variety of real-world systems of many fields can be represented as complex networks or graphs, namely, sets of vertices and edges with non-trivial topological internals. Examples of such networks run from technology [1] to biological fields [2] and social [3] studies.

The notion of community is quite general and associated with object classification. Depending on the context, it may refer to the module, the class, the group, the cluster, etc. This notion is of great value for solving the community detection task for the Internet [4]. The issue is topical for the implementation of on-the-edge search engines, data filtration and further automated sorting.

Typical networks to study are popular social network services. Then the user accounts are taken for the nodes, and the connections between them are determined by the formal notions of "friendship", "subscription", etc., so that we obtain a graph of interacting objects. Such networks have the structure of implicit communities – groups of people united by some additional feature, for example: classmates, subscribers to a number of public persons or thematic publics, criminal elements.

The problem of community detection in graphs of interacting objects is of great value to network analysis. In the general case, this task is NP-complete, so the paper proposes to use a method with a number of terminal options since it is required to deal with graphs from various fields [5, 6]. These options are designed to possibly refine the resulting community structure and are based on the initial graph topology and corresponding field.

1. Community detection methods

The well-known GN algorithm proposed by Girvan and Newman in 2002 [7, 8] is among the pioneers of divisive algorithms. Initially all the graph vertices share one community. Each step of the GN algorithm starts with the edge betweenness [9] calculation for every edge in the graph [8] and then it removes those with the highest score. The algorithm repeats this procedure, and it splits the network into disconnected parts, which in their turn go through the same process. The loop breaks when there are no more edges in the graph or when the split reaches modularity's [10] maxima.

Modularity is a numerical characteristic of the correctness of a partition. Currently, this is the most popular measure of the quality of the partition, and it is used directly or indirectly in the study of complex networks. As a rule, a closer to real partitioning has larger values of modularity.

Modularity Q is defined on an existing community structure of a graph:

$$Q = \frac{1}{2m}\sum_{i,j}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\delta\left(c_i, c_j\right), \qquad (1)$$

where m the number of edges;

A_{ij} – the adjacency matrix element;

k_i – the degree of the vertex i;

c_i – the index of the class to which the vertex belongs;

$$\delta\left(c_i, c_j\right) = \begin{cases} 0, & c_i = c_j; \\ 1, & c_i \neq c_j. \end{cases}$$

The edge betweenness for a given edge is the fraction of the shortest paths passing this edge among all shortest paths:

$$c_B(e) = \sum_{\substack{s \neq t \\ s,v \in V}} \frac{\sigma_{st}(e)}{\sigma_{st}}, \qquad (2)$$

where e – the given edge;

V – the set of graph vertices;

σ_{st} – the number of the shortest paths between the vertices s and t;

$\sigma_{st}(e)$ – the number of the shortest paths between the vertices s and t passing through the edge e.

The calculation of edge betweenness $C_B(e)$ is computationally expensive, as was mentioned by Girvan and Newman [1, 8]. So in the case of studying large networks, it is better to apply different techniques [1].

The fast hierarchical approach was proposed by Blondel et al. in 2008 [11] for the general case of weighted graphs. The algorithm is agglomerative, so it combines current vertices to new supervertices (communities) step by step. The increase of modularity for such split must be maximal as well. At the end of each iteration, a new hierarchy level is reached. The loop breaks if there is no increase of modularity. The method is limited more by the processing memory than the time.

The resulting community structure may not make sense since the approach is based on the locals of vertices. That is why it is not always obvious if some intermediate split meets the reality. Moreover, the outcomes depend on the vertices order at the first step [12].

The aforementioned approaches mean that communities are not intersecting with each other. Furthermore, in real-world systems vertices are rarely belong to single community [13]. As a consequence, the resulting partition may not accurately describe the actual network.

Further on, for CPM we denote one of the well-known methods. In their work, Palla et al. [14, 15] presented a novel approach for detecting the overlapping community structure in complex networks.

In this method, finding the structure of overlapping communities is related to the problem of finding k-cliques in a network (complete subgraphs of size k). Given the clique size k, this algorithm first finds all k-cliques within the network. Then it associates two k-cliques if they share k - 1 nodes (*Figure 1*). Each community is made over a maximum set of adjacent k-cliques [15].

To measure the quality of such partitions, there are many modifications of classical modularity. In this paper, we consider the variant proposed in [13]:

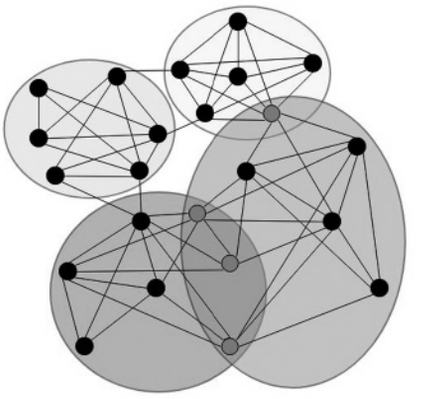

Fig. 1. Determined overlapping communities with the CPM algorithm for 4–cliques

$$Q_{ov} = \frac{1}{2m}\sum_{c\in C}\sum_{i,j\in c}\left[A_{ij} - \frac{k_i k_j}{2m}\right]\frac{1}{O_i O_j}, \qquad (3)$$

where $C = \{c_1, c_2, ..., c_{|C|}\}$ – the set of overlapping communities;

Q_i – the number of communities to which the vertex i belongs.

2. Description of the Combined method

In this paper, we consider a combination of the hierarchical Blondel algorithm with the CPM method of network partition into overlapping communities. By combination we mean a consistent application of the two methods. The idea is that at first our network is divided into large non-intersecting communities, and then we find smaller intersecting communities within them. Hereafter, we call this algorithm the "Combined method" (CM).

However, if Blondel's method identifies a finite community that does not split into parts in terms of a real network, then it is no longer useful to run CPM on it. Therefore, we apply the CPM algorithm to each separate c community identified by the Blondel method only if the percentage of vertices with high connection centrality does not exceed a predetermined value

$$\frac{\sum_{v \in c} \alpha(v)}{n_c} \le \beta(n_c), \alpha(v) = \begin{cases} c_B(v), c_B(v) \ge \tilde{\alpha} \\ 0, \quad c_B(v) < \tilde{\alpha} \end{cases}, \quad (4)$$

where n_C – number of vertices in c community; $c_B(v)$ – connection centrality of the vertex v; $\tilde{\alpha}$ and $\beta(n_c)$ – algorithm parameters, $0 \le \tilde{\alpha} < 1$; $0 < \beta \le 1$.

The relatively large connection centrality value characterizes the vertex as possibly connecting different communities. If the vertex is removed, the graph can be divided into several components. A similar consideration is used in the Girvan–Newman algorithm [1, 8]. The presence of a sufficient number of such peaks in the community may indicate that it has not been finalized completely, and further fragmentation by the Blondel algorithm is required. Therefore, the condition of applying CPM depends both on the value of connection centrality and on the percentage of vertices which have this value higher than some threshold $\tilde{\alpha}$. The threshold values for $c_B(v)$ and the vertex percentage are determined by the input parameters of algorithm $\tilde{\alpha}$ and β, and thus they are able to influence the final partition.

As a result of using CPM, vertices that are not included in cliques are cut off, and there is a problem of assigning them to the community. The solution is a system of two new parameters t_{opt} and m_{opt} of the Combined method. Depending on their values, not only the cut off vertices are redistributed, but also small communities. This allows us to partially influence the resulting structure, and in some cases – to avoid combining graph sheets or small communities into large, meaningless communities. This characteristic of the partition, which can be called "picking up junk", is intrinsic to many modularity optimization methods, including the Blondel algorithm [11, 16, 17].

To accurately define and illustrate the operation of the parameters t_{opt} and m_{opt}, let us consider $C_R = \{C_{R_1}, C_{R_2}, \ldots, C_{R_{rc}}\}$ the set of communities identified by the Blondel algorithm, where rc is their number. For convenience, we assume that if the individual community $C_{R_i} \in C_R$ does not satisfy the condition of applying the CPM, then the set $C_{CPM_i} = C_{R_i}$ duplicates it. Otherwise, C_{CPM_i} is defined as the set of communities obtained as a result of the CPM inside C_{CPM_i}:

$$C_{CPM_i} = \{C_{CPM_{i_1}}, C_{CPM_{i_2}}, \ldots, C_{CPM_{i_{mc_i}}}\}, \quad (5)$$

where mc_i number of communities in $C_{R_i} \in C_R$ that were identified by CPM algorithm.

Let us fix the result of CPM partition as C_f and denote the community structure after redistribution with the parameter i_{opt} as C_f^i. The parameters i_{opt} and m_{opt} are required when the algorithm starts, along with the size of the clique k and the limitations $\tilde{\alpha}$ and β. The community structure C_f is first transformed according to the value of the parameter i_{opt}, and then – according the value of the parameter m_{opt}.

The first parameter i_{opt} is responsible for the distribution of the cut off vertices of the CPM application and may have the following values (*Table 1*).

Table 1.

Description of the operation of i_{opt} parameter

None	Cut off vertices that do not belong to any community
i2c	Each cut off vertex is assigned to an individual (single) community within the community in which it was before applying CPM
i2r	Each cut off vertex is assigned to an individual (single) community that does not intersect with others
ia2c	A group of all cut off vertices that were in the same community prior to CPM application forms a community within it
ia2r	All the cut off vertices form one community that does not intersect with others

The second parameter m_{opt} works with small communities. The community is considered small if the number of vertices in it does not exceed the size of the clique k – the input parameter of the algorithm. This restriction is chosen because k is the minimum community size that can be indicated using the CPM method. The m_{opt} parameter can have the following values (*Table 2*).

Table 2.

Description of the operation of m_{opt} parameter

None	The structure of communities does not change
mdel	Small communities disband, but if the vertex belongs to other communities, then it remains in them. Otherwise, it will not belong to any community
m2c	All small communities that are in a common community (i.e., enclosed in it) are united into one common community that does not have enclosed communities
m2r	All small communities form one common community that does not intersect with others, even if the vertices belonged to other communities

Each of the transformations is done once, so when new small communities (possibly for *m2r*) appear, they will not be merged with the new common community. The resulting structure of C_f^* communities network is formed from C_f in accordance with the principles shown in *Tables 1* and *2*.

Separately, we should consider a combination of parameters with the values "*i2c / None*". In this case, when a small community is identified, it often turns out that all its members are also identified as individual communities, and large communities are divided into one larger part and several individual parts. To avoid a bunch of communities in the final structure, the authors use a modification: in the case described for small communities, individual communities disband, so only the smallest community remains. For a large community, it remains itself and the individual communities remain within it (that is, we remove the part of the internal partition except of those individual communities). *Figure 2* shows a possible result of the CM operation within one of the communities obtained in the first step. No similar result can be obtained by the separate application of Blondel and CPM algorithms. Meanwhile, in a number of cases in the graph analysis such a partition would be extremely informative.

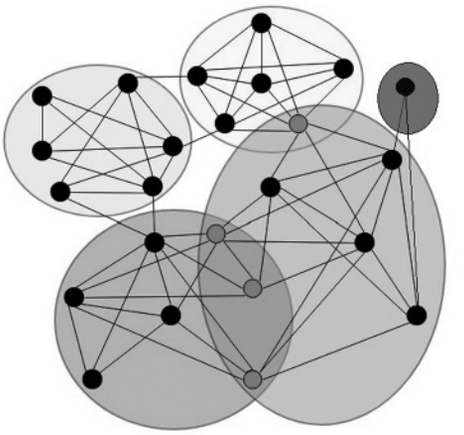

Fig. 2. Possible final partition within one of the communities obtained in the first step of CM

3. Tests

The results of the CM operation on different types of graphs were evaluated and compared with the separate results of the Blondel and CPM algorithms. In addition, comparison was made with the results of some classical algorithms: GN, quick greedy optimization of the modularity FG [18] and the CONGA algorithm (Cluster Overlap Newman-Girvan Optimized algorithm) [19]. We considered dependence of the CM results on the parameters k, $\tilde{\alpha}$, β, i_{opt}, m_{opt}. The size of the clique k is considered in the following range: $k \in [3, 10]$. For each partition, the value of Q_{ov} is calculated, and for the partitions obtained by CM, their similarity measure ω to other partitions must be calculated.

Collins and Dent [20, 21] proposed the ω (Omega Index) for estimating the similarity of two partitions, based on the number of pairs of vertices adjacent in the same number of communities. Let C_1 and C_2 be different partitions, K_1, K_2 – corresponding number of communities. Let $t_j(C)$ be the set of pairs of vertices occurring together in the C partition exactly j times. Then ω is defined as follows:

$$\omega(C',C'') = \frac{\omega_u(C',C'') - \omega_e(C',C'')}{1 - \omega_e(C',C'')}, \quad (6)$$

where $\omega_u(C',C'')$ – observed value,

$\omega_e(C',C'')$ – expected value:

$$\omega_u(C',C'') = \frac{1}{m}\sum_{j=0}^{\min(K_1,K_2)} \left| t_j(C') \bigcap t_j(C'') \right|,$$

$$\omega_e(C',C'') = \frac{1}{m^2}\sum_{j=0}^{\min(K_1,K_2)} \left| t_j(C') \right| \cdot \left| t_j(C'') \right|.$$

Value of $\omega(C',C'') = 1$ in the case of absolute equality of two partitions.

The authors studied the results of applying CM and selected classical algorithms on more than 70 graphs of interacting objects of various nature, such as: social networks, scientists citation networks, interactions of proteins, infrastructure networks. In a number of cases, for graphs from one group and with similar topological parameters, the most optimal values of the correcting parameters for CM have been identified. The software realization is written in Python 3.5, standard and open libraries, including classical algorithms, were used to work with graphs. Semi-automatic test blocks are implemented to find the best values for Q_{ov} parameter values. Visualization of the partitions obtained for the graphs is provided by third-party software.

Let us consider the results on ego-graphs obtained from the social network VKontakte (*Table 3*) in more detail. The ego-graphs in this case are graphs formed by a width search starting from a single vertex with a depth of 1 and adding all the edges between the vertices previously obtained. At the same time, as a starting point we took vertices with a degree in the range from about 100 to 500. This made it possible to obtain not only graphs with different topological properties, but ones which are also convenient for qualitative analysis of their sizes. The main objectives of this work are to study the content of the resulting partitions and to find the most suitable values of CM parameters.

Table 3.

Social network graphs for testing

	n	***m***	***avk***	***dst***	***avntr***	***avclu***	***maxcl***	***bet***	***clo***
G_1	105	792	15.1	0.1451	38.04	0.712	15	44.5	0.5434
G_2	120	1226	20.4	0.1717	52.39	0.668	14	49.3	0.5507
G_3	158	1352	17.1	0.1090	44.41	0.767	17	69.9	0.5309
G_4	158	1596	20.2	0.1287	63.80	0.638	21	68.4	0.5374
G_5	187	1873	20.0	0.1077	51.11	0.674	16	83.0	0.5308
G_6	229	1726	15.1	0.0661	26.67	0.633	14	106.5	0.5184
G_7	439	4143	18.9	0.0431	41.47	0.602	16	209.6	0.5117
G_8	461	4299	18.7	0.0405	49.36	0.665	17	220.7	0.5110

Notation: ***n*** – number of vertices; ***m*** – number of edges; ***avk*** – average degree of vertices; ***dst*** – connection density; ***avntr*** – average number of triangles; ***avclu*** – average cluster coefficient; ***maxcl*** – size of a maximal clique; ***bet*** – averaged betweenness centrality; ***clo*** – averaged closeness.

Table 4.

Test results

	$s(CM)$	Q_{ov}^{KA}	$s(BL)$	Q_{ov}^{BL}	$s(CP)$	Q_{ov}^{CP}	$s(GN)$	Q_{ov}^{GN}	$s(FG)$	Q_{ov}^{FG}	$s(CN)$	Q_{ov}^{CN}
G_1	38	0.213	4	0.320	4	0.167	17	0.250	3	0.315	3	0.245
G_2	21	0.367	4	0.385	4	– 0.013	20	0.329	4	0.371	4	0.136
G_3	28	0.650	4	0.614	5	0.390	15	0.581	5	0.581	5	0.494
G_4	39	0.327	4	0.342	3	0.016	47	0.175	3	0.313	4	0.100
G_5	36	0.048	5	0.369	3	– 0.008	27	0.250	5	0.325	110	0.011
G_6	51	0.224	5	0.465	4	– 0.029	22	0.408	6	0.393	4	0.273
G_7	90	0.501	7	0.572	8	– 0.149	54	0.536	8	0.509	5	0.304
G_8	99	0.349	7	0.569	11	–0.272	53	0.514	6	0.552	8	0.195

Notation: $s(CM)$ – number of communities identified by CM; $s(BL)$ – number of communities identified by the Blondel algorithm; $s(CP)$ – number of communities identified by CPM; $s(GN)$ – number of communities identified by GN; $s(FG)$ – number of communities identified by FG; $s(CN)$ – number of communities identified by CONGA; Each of Q_{ov}^{A} – value Q_{ov} for the partition, obtained by A algorithm.

In most cases, CM identified, as expected, a greater number of communities on these graphs (*Table 4*). For this group of graphs representing ego-graphs of the same social network, the most effective parameter option for the content of the partitions was the *i*2*c*/*m*2*r* and $k = 4$. A qualitative expert analysis of the composition of the groups singled out by the algorithms examined, especially of small communities of several vertices, showed the greatest efficiency especially for CM. For example, the Blondel, CPM and FG algorithms, as expected, left no large communities. And the smallest communities (including 1 element) identified by the GN algorithm for each of the graphs were not all meaningfully justified, and in some cases, taking into account the edges of the graph, they should be combined with the others.

It should be noted that the value of Q_{ov}^{KA} was not always higher than of other similar algorithms due to the fact that CM calculates another local extremum of the given quality measure, which is partially confirmed by the calculated values of the similarity measure ω of these partitions (*Table 5*). As you can see from the tables, not only the number of the communities obtained by CM and the other community algorithms differs greatly, but so do the sets of vertices combined into groups. After all, due to the peculiarities of the considered classical algorithms, some types of partitions cannot be obtained from their application, even theoretically. For example, the situation in *Figure 2* for CPM cannot be obtained. All combined, this shows that the partition obtained as a result of CM was essentially different from the others.

The partition obtained within the CM is hierarchical, with intersecting and nested communities, which gives an opportunity for more complete understanding of the structure of the investigated graph and better corresponds to the actual system of interaction of objects of some nature, in particular, users of social networks. As an illustration, let us consider the example for G_3 in more detail.

At the first step, the CM identifies four communities in the graph G_3: $C_{R_1}, C_{R_2}, C_{R_3}, C_{R_4}$, with the sizes 44, 41, 37, 36, respectively (*Figure 3*). This graph is very revealing, because the communities identified by the Blondel algorithm are of the similar size. Here C_{R_1} is a typical

Table 5.

The values of ω of the results of partitioning the graphs by CM and other algorithms

	$\omega(BL)$	$\omega(CPM)$	$\omega(GN)$	$\omega(FG)$	$\omega(CONGA)$
G_1	0.0811	–0.0478	0.0209	0.0729	0.0226
G_2	–0.0125	0.0139	0.0152	–0.0087	0.0655
G_3	0.0160	–0.0166	–0.0017	0.0043	0.0149
G_4	0.0459	–0.0077	–0.0098	0.0047	–0.0147
G_5	0.0651	–0.0058	0.0232	0.0133	–0.0055
G_6	0.0915	0.0003	0.0004	0.0159	0.0088
G_7	0.0452	0.0009	0.0061	0.0020	0.0066
G_8	0.0152	0.0137	–0.0070	0.0096	0.0079

Notation: $\omega(BL)$ – for partitioning a graph with CM and the Blondel algorithm; $\omega(CPM)$ – for partitioning a graph with CM and CPM algorithm; $\omega(GN)$ – for partitioning a graph with CM and GN algorithm; $\omega(FG)$ – for partitioning a graph with CM and FG algorithm; $\omega(CONGA)$ – for partitioning a graph with CM and CONGA algorithm.

"junk" community containing the original vertex of the ego-graph, leaf vertices and small communities. The other three communities are more meaningful, they are user groups, most of which have no connections with representatives of the other two communities.

Then, after applying the following CM steps on the graph G_3, the four communities are transformed into 28 as follows. Within the three meaningful dense communities, minimal changes occur. Thus, the algorithm found one vertex less connected to the others inside the C_{R_2}, for which, according to the *i2c/None* options, a separate community inside C_{R_2} was selected, and the remaining 40 vertices were put in the second internal community for C_{R_2}. The community C_{R_3} remained completely unchanged, and for C_{R_4} three vertices were found. They were separated into their individual internal communities, like in the C_{R_2} case, meanwhile creating a large remainder – a community of 33 vertices .

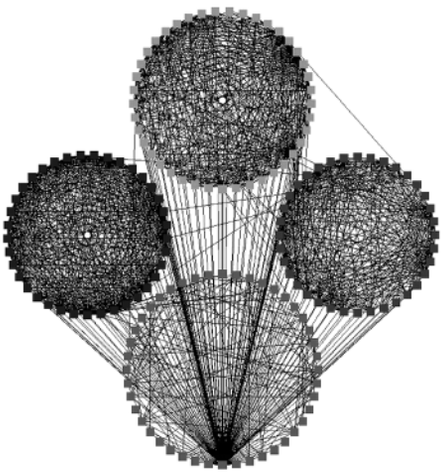

Fig. 3. The four communities identified by the Blondel algorithm in the graph G_3

The main changes occurred at the second step of the CM within C_{R_1}. Thus, in this community 4 others were identified. They intersected along the original vertex of the ego-graph. In addition, similar to the situation with the other two upper-level communities, 14 individual communities within the original community were identified (*Figure 4*).

In other ego-graph G_4 with the same number of vertices, but with a different structure (*Table 3*), 39 communities were identified after applying CM to it. And initially, at the first step, 4 large groups were also identified (here

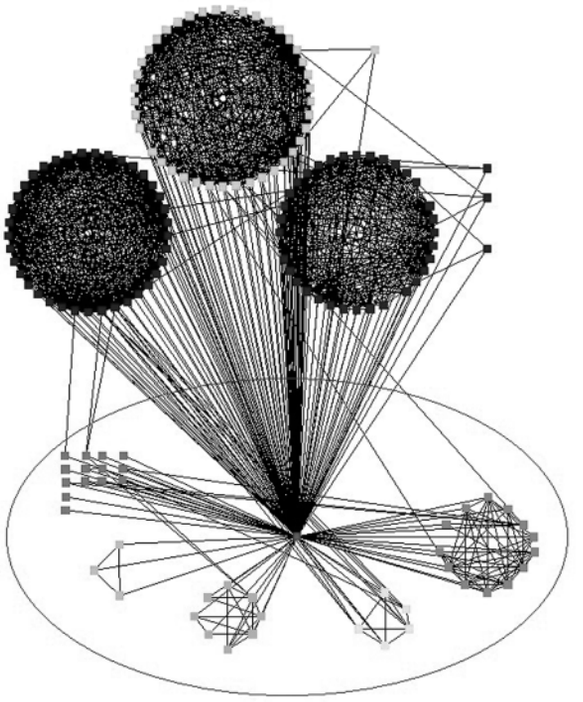

Fig. 4. Communities obtained by CM

$|C_{R_1}| = 50$, $|C_{R_2}| = 44$, $|C_{R_3}| = 44$, $|C_{R_4}| = 20$) inside of which already at the second step a search for intersecting nested communities is made. In this case, C_{R_2} and C_{R_3} did not change, but within the C_{R_1} and C_{R_4} 28 and 7 nested communities with intersections of different sizes were allocated, respectively.

It should be noted that according to the users of social networks whose ego-graphs were analyzed, other partitions obtained by the compared methods carried less information in themselves. However the Q_{ov} score was higher than for CM in a number of cases (*Table 4*). In particular, the results of algorithms that are components of CM were recognized as less informative and, for example, for GN the partitioning was partially superfluous.

Conclusion

A new algorithm, called by the authors the "Combined method" (CM), has been invented and presented to effectively identificate partitions of implicit communities in the graphs of interacting objects of various natures. We apply CM and well-known classical algorithms on the set of graphs from different industries which were taken from open sources. We studied the graph partitions into communities so obtained. Comparison of the modularity value Q_{ov} on the resultant partitions and the similarity measure ω between the result of CM and the remaining algorithms showed that in most cases there are significant differences in the resulting sets of communities. In addition, for a set of graphs from the social network, a meaningful analysis of the partitions obtained was carried out, and it showed that the result of CM with the options selected on this set more accurately identifies groups of interactions between users.

The proposed algorithm was run on graphs with different parameters. During the testing, it was found that the use of the parameters of the restrictions $\tilde{\alpha}$ and β on these graphs does not improve the quality of the resulting partition, since a result different from the results of applying the Blondel and CM algorithm (with no constraints) was achieved only in the case $\tilde{\alpha} > 0$ and $\beta = 0$. Further investigation was carried out at the values of the parameters $\tilde{\alpha} = 0$ and $\beta = 1$. Thus, the CPM method was run on all the Blondel communities generated by the algorithm.

The research showed that the most suitable values of additional parameters for social networks are *i2c/None* within the CM designations. At the same time, it cannot be stated that for a graph of interacting objects of a different nature this combination will be more effective than the others. Thus, the authors consider further studies on the design and implementation of parametric algorithms for the partition of communities. We also consider a possible use of automated tools to select suitable options and parameter values based on topological features and the original nature of graphs. ■

References

1. Guo Y., Xu K. (2007) An improved algorithm for finding community structure in networks with an application to IPv6 backbone network. *Frontiers of Computer Science in China*, vol. 1, pp. 459–467.
2. Wu F., Chen L., Wang J., Alhajj R. (2014) Biomolecular networks and human diseases. *BioMed Research International*, vol. 2014, pp. 1–2.
3. White H.C., Boorman S.A., Breiger R.L. (1976) Social structure from multiple networks. I. Blockmodels of roles and positions. *American Journal of Sociology*, vol. 81, no. 4, pp. 730–780.
4. Broder A.Z., Kumar S.R., Maghoul F., Raghavan P., Rajagopalan S., Stata R., Tomkins A., Wiener J. (2000) Graph structure in the Web. *Computer Networks*, vol. 33, no. 1–6, pp. 309–320.
5. Ahuja M.S., Singh J., Neha (2016) Practical applications of community detection. *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 6, no. 4, pp. 412–415.
6. Pennacchioli D., Coscia M., Pedreschi D. (2014) Overlap versus partition: Marketing classification and customer profiling in complex networks of products. Proceedings of 2014 *IEEE 30th International Conference on Data Engineering Workshops (ICDEW 2014). Chicago, 31 March – 4 April 2014*, pp. 103–110.
7. Newman M., Girvan M. (2004) Finding and evaluating community structure in networks. *Physical Review E*, vol. 69, no. 2, pp. 1–15.
8. Girvan M., Newman M. (2002) Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826.
9. Newman M. (2010) Networks: An introduction. Oxford: Oxford University Press.
10. Newman M. (2006) Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582.
11. Blondel V.D., Guillaume J.L., Lambiotte R., Lefebvre E. (2008) Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, no. 10, P10008.
12. Fortunato S. (2010) Community detection in graphs. *Physics Reports*, vol. 486, no. 3–5, pp. 75–174.
13. Shen H.W., Cheng X.Q., Cai K., Hu M.B. (2009) Detect overlapping and hierarchical community structure in networks. *Physica A: Statistical Mechanics and its Applications*, vol. 388, no. 8, pp. 1706–1712.
14. Palla G., Derényi I., Farkas I., Vicsek T. (2005) Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, vol. 435, no. 7043, pp. 814–818.
15. Palla G., Ábel D., Farkas I.J., Pollner P., Derényi I., Vicsek T. (2009) k-clique percolation and clustering. *Handbook of Large-scale Random Networks*, Springer, ch. 9, pp. 369–408.
16. Fortunato S., Barthelemy M. (2006) Resolution limit in community detection // *Proceedings of the National Academy of Sciences*, vol. 104, no. 1, pp. 36–41.
17. Orlov A.O., Chepovskiy A.A. (2016) Osobennosti algoritma Blondelya pri vyyavlenii soobshchestv v grafe sotsial'noy seti [Features of Blondel algorithm to reveal communities in social networks]. Proceedings of *International Scientific Conference of Moscow Institute of Physics and Techniques and Institute of Physical and Technical Information (SCVRT1516*). Moscow, Protvino: Institute of Physical and Technical Information, pp. 124–129 (in Russian).
18. Clauset A., Newman M., Moore C. (2004) Finding community structure in very large networks. *Physical Review*, E 70. 066111.
19. Gregory S. (2007) An algorithm to find overlapping community structure in networks. Proceedings of *Knowledge Discovery in Databases: PKDD 2007: 11th European Conference on Principles and Practice of Knowledge Discovery in Databases, Warsaw, Poland, 17–21 September 2007.* Berlin, Heidelberg: Springer, 2007, Vol. 4702. P. 91–102.
20. Collins L.M., Dent C.W. (1988) Omega: A general formulation of the Rand index of cluster recovery suitable for non-disjoint solutions. *Multivariate Behavioral Research,* vol. 23, no. 2, pp. 231–242.
21. Gregory S. (2011) Fuzzy overlapping communities in networks. *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2011, no. 02, pp. 1–18.