Community Structure Detection in Multipartite Networks

A new fitness measure

Noémi Gaskó
Centre for the Study of Complexity
Babeş-Bolyai University
gaskonomi@cs.ubbcluj.ro

Florentin Bota
Centre for the Study of Complexity
Babeş-Bolyai University
botaflorentin@cs.ubbcluj.ro

Mihai Suciu
Centre for the Study of Complexity
Babeş-Bolyai University
mihai-suciu@cs.ubbcluj.ro

Rodica Ioana Lung
Centre for the Study of Complexity
Babeş-Bolyai University
rodica.lung@econ.ubbcluj.ro

ABSTRACT

Community structure detection algorithms are used to identify groups of nodes that are more connected to each other than to the rest of the network. Multipartite networks are a special type of network in which nodes are divided into partitions such that there are no links between nodes in the same partition. However, such nodes may belong to the same community, making the identification of the community structure of a multipartite network computationally challenging. In this paper, we propose a new fitness function that takes into account the information induced by existing links in the network by considering shadowed connections between nodes that have a common neighbor. The existence of a correct fitness function, i.e. one whose optimum values correspond to the community structure of the network, enables the design and use of optimization-based heuristics for solving this problem. We use numerical experiments performed on artificial benchmarks to illustrate the effectiveness of this function used within an extremal optimization based algorithm and compared to existing approaches. As a direct application, a multipartite network constructed from a direct marketing database is analyzed.

CCS CONCEPTS

•Computing methodologies → Discrete space search;

KEYWORDS

community structure, multipartite networks, fitness function

ACM Reference format:
Noémi Gaskó, Florentin Bota, Mihai Suciu, and Rodica Ioana Lung. 2017. Community Structure Detection in Multipartite Networks . In *Proceedings of GECCO '17, Berlin,Germany, July 15-19, 2017,* 7 pages.
DOI: http://dx.doi.org/10.1145/3071178.3071295

GECCO '17, Berlin,Germany

DOI: http://dx.doi.org/10.1145/3071178.3071295

1 INTRODUCTION

The network community structure detection problem has gained a lot of attention in recent years due to its large applicability. The problem consists in finding groups of nodes in a network that are more densely connected to each other than to other nodes in the network [3] and it has been extensively studied for unipartite and bipartite networks. However, there are very few methods and results extending to multipartite networks, in spite of the fact that numerous applications might benefit from such methods as many living systems or phenomena can be modeled as multipartite networks [1, 10]. A general approach to the community structure detection problem consists in the optimization of a fitness function that is supposed to reflect the modularity of the network, i.e. its optimum value corresponds to a community structure. Currently there does not exist a function that can be used for any type of network and correctly capture various types of structures, but there are some successful ones such as the modularity [16], the modularity density [8], the community fitness [7], etc.

We propose an extension of the community fitness defined in [7] for identifying communities in multipartite networks by considering that nodes that have a common neighbor are also connected to each other and thus creating *shadowed* links in the network helping in the discovery of the community structure. We use numerical experiments performed on synthetic benchmarks to test the efficiency of this function. As an application we analyze a marketing database by converting it into a multipartite network in which nodes are variable categories or quantiles for numerical variables and observations build links between different categories. In this way each variable creates a partition set in the network and we can study possible connections between variables; the community structure of such a network reveals connections among variable categories.

2 RELATED WORK

Community structure detection methods have recently become the focus of research due to the relevant information revealed by such structures. However, there is little research performed on community structure detection methods for multipartite networks. There is a body of work related to bipartite networks, but very few extensions.

For unipartite networks, one of the major approaches to community structure detection converts the search problem into an

optimization one by using a fitness function that is supposed to illustrate the modular structure of the network. One of the most popular, and debated, fitness function is the *modularity* [15, 16]. The modularity Q_N basically compares a community structure for a network to the corresponding structure when considering a random network. It can be computed as:

$$Q_N = \sum_{C \in \mathcal{C}} \left(\frac{m_C}{m} - \left(\frac{D_C}{2m} \right)^2 \right), \tag{1}$$

where m is the number of edges, m_C is the number of edges connecting vertices in community C, and D_C is the sum of the degrees of all vertices in community C. A higher modularity value indicates a better solution. In [3] it is stated that multipartite networks can be approached by the same methods as unipartite networks reducing the multipartite graph to unipartite ones, usually by considering a different network for each partition. While this approach is adequate, apart from the fact that some information may be lost in the process, in some cases we are interested in the community structure in the sense that we want to find nodes from different - or the same - partitions that are more connected to each other than to the rest of the network.

In this context it was noticed that Newmann's modularity described above does not reflect any more the difference to a random network because a random multipartite network does not present any links between nodes in the same partitions. To take into account this fact, Barber [2] proposed the following modularity formula for bipartite networks:

$$Q_B = \sum_{C \in \mathcal{C}} \left(\frac{m_C}{m} - \frac{R_C \times B_C}{m^2} \right), \tag{2}$$

where m is the number of edges, m_C is the number of edges connecting vertices in community C, R and B represent the two disjoint sets of vertices, R_C is the sum of the degrees of vertices from R in community C, B_C is the sum of the degrees of vertices from B in community C. In [13] it is shown that maximizing Barber's modularity is NP-hard.

Regarding multipartite networks, approaches vary depending on the type of network and solution concept. There are very few methods that take into account the multipartite structure: in [18] a ranking method is proposed for star networks; in [12] the problem of finding communities among joint networks is approached; in [9] a composite modularity is defined and maximized to reveal the community structure;the weakness of this modularity [10] is not being able to handle communities with many-to-many correspondences for multipartite multi-relational networks, the authors propose an information compression method inspired from [17] for this problem.

3 MULTIPARTITE COMMUNITY SCORE

The specificity of the multipartite network requires taking into account that, while nodes belonging to the same partition may not be directly connected to each other, they may be indirectly connected by common neighbors in other partitions. Such induced connections may also be responsible for the underlying community structure and should be taken into account when evaluating potential solutions. In this context we propose the Multipartite

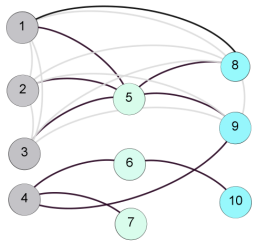

Figure 1: A simple tripartite network. The degree of node 5 is 5, the modified degree with $\alpha = 0.05$ is 5.5. If we consider community $C = \{1, 2, 3, 5\}$, $f_{0.05}(C) = 0.64$.

Community Fitness M by extending the *community fitness* in [7] to include such links. Since these links cannot be considered 'real' network links (if that were the case they would have been added at the construction of the network) we call them *shadowed* links and add them to the node degree by using a *multipartite sensibility factor* α.

Thus, if we consider $G = (V, E)$ with V a set of vertices and E a set of links connecting two elements of V, G is a multipartite network if there exists a partition $\mathcal{V} = \{V_l\}_{l=\overline{1,r}}$ over V such that within each partition there are no links between any pair of nodes. r denotes the number of partitions. If $r = 2$ we have a bipartite network, if $r = 3$ we have a tripartite network, etc. Any link in E therefore connects nodes from different partitions.

To construct the multipartite community fitness we first define the *modified node degree* $d_{i,\alpha}$ as

$$d_{i,\alpha} = d_i + \alpha \cdot \frac{d_i(d_i - 1)}{2} \tag{3}$$

where d_i is the degree of node i in G. Thus we actually take into account the fact that all nodes connected with i can be considered also connected to each other; the strength of these connections is controlled by α.

Figure 1 illustrates a small tripartite network with 10 nodes. The degree of node 5 in this network is 5, and its modified degree $d_{5,0.05}$ is 5.5 computed by taking into account the 10 *shadowed* links connecting nodes 1, 2, 3, 8, and 9.

If we consider a community $C \subset V$ then the inner degree of a node $i \in V$, $k_{in}(i|C)$ is computed as

$$k_\alpha(i|C) = k(i, C) + \alpha \cdot \frac{k(i, C)(k(i, C) - 1)}{2} \tag{4}$$

where $k(i, C)$ is the number of links node i has with other nodes in C. In the example illustrated in Figure 1, if we consider $C = \{1, 2, 3, 5\}$, the inner degree of node 5 is $k_{0.05}(5|C) = 3 + 0.05 \cdot 3 = 3.15$.

The fitness of community C is computed as:

$$f_\alpha(C) = \frac{\sum_{i \in C} k_\alpha(i|C)}{\sum_{i \in C} d_{i,\alpha}}, \tag{5}$$

as the ratio of the modified total inner degree of the nodes in community C and the total degree of nodes in C. A higher fitness value can be considered to indicate a better community. Considering the same community $C = \{1, 2, 3, 5\}$ in Figure 1, its fitness $f_{0.05}(C) = \frac{1+1+1+3.15}{2.05+1+1+5.5} = 0.64$.

The fitness of a *community structure* C, i.e. a partition set over the set of nodes, is computed as the average of the fitnesses of all the communities $C \in \mathcal{C}$:

$$M_\alpha(\mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} f_\alpha(C), \tag{6}$$

where $|\mathcal{C}|$ is the number of communities.

For weighted networks we use the same approach but considering in all formulas the weighted degree, or weighted inner degree of the nodes, and the sum of the fitnesses of each community in (6).

Let us consider the tripartite network in Figure 1 and two communities: $C_1 = \{1, 2, 3, 5, 8, 9\}$ and $C_2 = \{4, 6, 7, 10\}$. In this case $Q_N = \left(\frac{6}{10} - \left(\frac{13}{20}\right)^2\right) + \left(\frac{3}{10} - \left(\frac{7}{20}\right)^2\right) = 0.355$ and $M_{0.05}(\mathcal{C}) = \frac{1}{2}(f_{0.05}(C_1) + f_{0.05}(C_2))$, where $f_{0.05}(C_1) = 0.923$, $f_{0.05}(C_2) = 0.847$, so $M_{0.05}(\mathcal{C}) = 0.885$.

3.1 Method

Maximizing the fitness M should uncover a multipartite community structure in a similar manner a fitness function such as the modularity uncovers the community structure of unipartite networks. To test this hypothesis we use an efficient method based on extremal optimization for community structure detection, called *NoisyEO* [11] to maximize the fitness M.

The algorithm is described in Algs. 1 and 2. *NoisyEO* evolves pairs of individuals s and s_{best} by re-assigning random values to the weakest components in s and replacing s_{best} whenever a better solution has been found. s and s_{best} encode possible covers as vectors of size N equal to the number of nodes in the network. For a node i, $s(i)$ represents its community. The fitness of a node is computed as the difference between the fitness of its community and the fitness of its community when the node is removed, i.e. as the contribution of the node to the fitness of its community. The only modification to the original *NoisyEO* algorithm is made at line 3 in Alg. 2 where we use the modified community fitness M in (6).

Algorithm 1 *NoisyEO* algorithm

Parameters:

- Population size - *popsize*;
- Probability of shift - p_{shift};
- Number of generations between switching networks - G;
- Total number of shifts - $NrShifts$;
- Expected minimum and maximum number of communities.

1: Randomly initialize *popsize* pairs of configurations (s, s_{best}).
2: *noise*=false;
3: **repeat**
4: **if** *noise* **then**
5: Induce noise with probability $p^{(*)}_{shift}$;
6: Randomly reinitialize each s_{best} in population;
7: **else**
8: perform search on the original network;
9: **end if**
10: *noise*=not *noise*;
11: Linearly decrease k until the middle of the search; after that set $k = 1$;
12: **for** G generations **do**
13: Apply κEO (s, s_{best}) for all pairs (s, s_{best}) - Alg. 2;
14: **end for**
15: **until** $G * NrShifts >$ Maximum number of generations;
16: Return s_{best} with highest ***fitness***.

(*) Modify network by randomly deleting/adding links with probability p_{shift} which decreases linearly from an initial value to 0 during the search.

Algorithm 2 κEO(s, s_{best}) iteration

1: For the current configuration s evaluate $u_i(s)$, the fitness function corresponding to node $i \in \{1, \ldots, n\}$.
2: find the κ worst components and replace them with a random value;
3: **if** $M(s) > M(s_{best})$ **then**
4: set $s_{best} := s$.
5: **end if**

4 NUMERICAL EXPERIMENTS

To illustrate the efficiency of the measure we perform numerical experiments on synthetic benchmarks and on a set of multipartite networks constructed from a real-world marketing database.

4.1 Experimental set-up

Parameter Settings. For *NoisyEO* we used the following parameters: population size 30, $p_{shift} = 1$, $G = 45$, total number of shifts 30, expected number of communities between 2 and 8.

To evaluate the efficiency of the multipartite community score M, we run *NoisyEO* to maximize also the modularity Q_N (1). For bipartite networks, we test also the maximization of the Barber modularity Q_B (2). As *NoisyEO* is not tuned for any type of modularity, differences in results arise from the different fitness functions used. For bipartite networks we also compare results with the Fast projection (FP) method in [5] using the software provided by the authors.

The multipartite community score M uses a sensibility factor α. We tested for $\alpha \in \{0, 0.025, 0.05\}$. When $\alpha = 0$ the multipartite aspect of the network is completely ignored.

Benchmarks. In order to test the efficiency of the approach and compare its performance with existing methods we need a set of multipartite networks with known community structures. Currently there are no standard benchmarks in the literature, but there are simple ways to generate multipartite graphs with community structures. We have chosen the following method to generate a multipartite unweighted graph: first we divide randomly the set of N nodes in r partitions, and in c communities by using a Monte Carlo approach that assigns approximatively equal number of nodes in each partition and in each community. After that, for each pair of nodes we create a link with probability p_{in} if they belong to the same community and with probability p_{out} if they are in different communities; if two nodes belong to the same partition no link is created. In this manner we construct multipartite networks with community structures of various degrees of difficulties. Similar approaches can be found in the unipartite community structure detection literature [7].

Considering uniform probability distributions, we can compute the mixing degree (the average ratio between number of outer links and total degree of the nodes [6]) as:

$$\mu = \frac{p_{out}(cr - c - r + 1)}{p_{in}(r-1) + p_{out}(cr - r - c + 1)} = \frac{p_{out}(c-1)}{p_{in} + p_{out}(c-1)} \quad (7)$$

which means that for our synthetic networks μ does not depend on the number of partitions but on the number of communities.

We considered all combinations between $p_{in} = 0.7, 0.8, 0.9$, and $p_{out} = 0.1, 0.2, 0.3$ and 7 sets of networks with $r = 2, 3, 4, 5, 6, 7,$ and 8 partitions respectively; all sets have 3 communities. For each set we report in Table 1 the corresponding μ values for an easier interpretation. For each setting we generated 10 networks.

Table 1: μ values for the synthetic benchmarks

μ	$p_{out} = 0.1$	$p_{out} = 0.2$	$p_{out} = 0.3$
$p_{in} = 0.9$	0.18	0.30	0.40
$p_{in} = 0.8$	0.20	0.33	0.42
$p_{in} = 0.7$	0.22	0.36	0.46

To evaluate the results we computed the normalized mutual information (NMI [7]) values obtained in each run for each network by comparing the output of the algorithm with the real community structure. A NMI value of 1 indicates that the two are identical. When comparing results, the higher the NMI value, the better. Statistical significance of differences between methods is evaluated by using a Wilcoxon sum-rank test with a significance level of 0.05.

The bank marketing data is retrieved form the UCI database[1] [14] and is used for classification purposes. It has 17 variables and 45211 observations. We use this dataset to illustrate the construction and analysis of a real-world multipartite dataset. To construct the network we proceed as follows: for categorical variables we consider each category as a network node; for numerical variables we consider as nodes the quantiles of the distribution of that variable. We obtain a weighted network which indicates connections between categories of different variables.

4.2 Results

In this section the results obtained when using the multipartite community fitness M for the synthetic and bank data are presented.

Bipartite networks. Figure 2 presents boxplots of NMI values obtained for bipartite networks; results presented here are obtained with $\alpha = 0.05$. The considered synthetic benchmarks can be divided on three levels of difficulty on p_{out} values, table 1. For the most simple ones, with $p_{out} = 0.1$ there are no statistical differences between results, all variants identify the community structure (except M, for $p_{in} = 0.7$ in some runs). For $p_{out} = 0.2$, M and Q_B yield best results, except for $p_{in} = 0.7$ where M outperforms all other methods. The same happens for all sets having $p_{out} = 0.3$, where we can see no statistical difference between using Barber's modularity and Q_N.

Multipartite networks. Table 2 presents results obtained for 2, 3, 4, 5-partite networks for different α values. Numerical results obtained using $\alpha = 0$ and Newmann's modularity Q_N are used as

[1] https://archive.ics.uci.edu/ml/datasets/Bank+Marketing

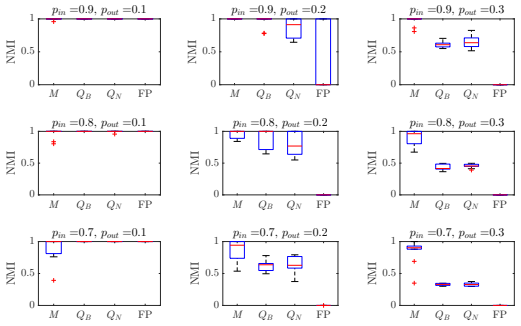

Figure 2: Results obtained for bipartite networks by maximizing the multipartite community fitness M, the Newman modularity Q_N and the Barber modularity Q_B, compared also with Fast projection (FP).

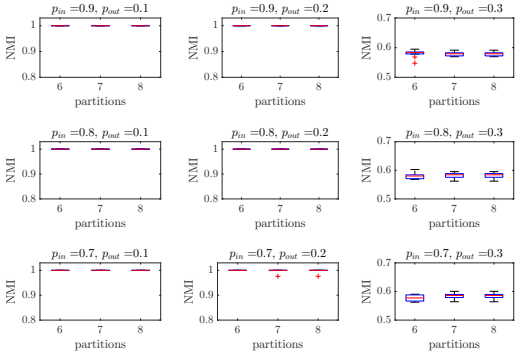

Figure 3: NMI best modularity for 6-partite, 7-partite, and 8-partite synthetic networks with 256 nodes.

baseline, as they do not take into account network partitions. We expect results obtained with M and a positive value of α to be better than both. A * is used to indicate statistical significance of differences in results based on the Wilcoxon sum-rank test. All methods that are not statistical different than the best one are marked. From the tested values, $\alpha = 0.05$ yields the best results on these networks.

Figure 3 presents boxplots of NMI values obtained for the 256 nodes networks and 6,7, and 8 partitions of nodes. We find that the number of partitions does not influence the results, and that the most challenging networks are those having $p_{out} = 0.3$.

Remarks related to NMI values. In all experiments, *NoisyEO* reported the individual having the best fitness value considering the function to be maximized, i.e. M, Q_N or Q_B. For technical purposes we also recorded the best NMI in the final population, which does not always coincide with the NMI of the best fitness values. It is the case that in many runs *NoisyEO* did in fact compute a correct cover, but the individual having best NMI value did not have the best fitness function value. To illustrate this we present

Table 2: Numerical results obtained for different α values on the sets of 2,3,4,5 - partite networks

p_{in}	p_{out}	$\alpha = 0$	$\alpha = 0.025$	$\alpha = 0.05$	Q_N
bipartite network					
0.9	0.1	0.67±0.20	0.91±0.19*	0.99±0.01*	1.00±0.00*
0.9	0.2	0.69±0.12	0.97±0.05*	1.00±0.00*	0.86±0.15
0.9	0.3	0.64±0.08	0.95±0.07*	0.96±0.07*	0.65±0.09
0.8	0.1	0.48±0.09	0.74±0.25	0.96±0.07*	0.99±0.01*
0.8	0.2	0.56±0.11	0.71±0.17	0.95±0.06*	0.77±0.17
0.8	0.3	0.45±0.09	0.63±0.21	0.90±0.11*	0.45±0.03
0.7	0.1	0.38±0.04	0.50±0.15	0.89±0.19	1.00±0.00*
0.7	0.2	0.44±0.12	0.53±0.16	0.85±0.16*	0.63±0.15
0.7	0.3	0.34±0.05	0.48±0.15	0.84±0.19*	0.33±0.02
tripartite network					
0.9	0.1	0.67±0.14	1.00±0.00*	1.00±0.00*	1.00±0.00*
0.9	0.2	0.81±0.17	0.99±0.02*	0.99±0.02*	0.95±0.09*
0.9	0.3	0.64±0.09	0.99±0.01*	0.99±0.01*	0.66±0.06
0.8	0.1	0.52±0.10	0.99±0.01*	0.99±0.01*	1.00±0.00*
0.8	0.2	0.58±0.12	0.98±0.05*	1.00±0.00*	0.85±0.11
0.8	0.3	0.57±0.13	0.91±0.10*	0.98±0.02*	0.44±0.03
0.7	0.1	0.48±0.12	0.88±0.12	0.99±0.01*	0.97±0.06*
0.7	0.2	0.40±0.07	0.75±0.27	0.99±0.01*	0.67±0.13
0.7	0.3	0.45±0.10	0.67±0.26*	0.75±0.22*	0.35±0.01
4-partite network					
0.9	0.1	0.73±0.18	1.00±0.00*	1.00±0.00*	1.00±0.00*
0.9	0.2	0.97±0.06*	1.00±0.00*	1.00±0.00*	0.95±0.07*
0.9	0.3	0.72±0.19	0.99±0.01*	0.95±0.14*	0.64±0.08
0.8	0.1	0.67±0.14	1.00±0.00*	1.00±0.00*	1.00±0.00*
0.8	0.2	0.52±0.10	1.00±0.00*	1.00±0.00*	0.90±0.10
0.8	0.3	0.57±0.16	0.99±0.01*	0.99±0.01*	0.45±0.05
0.7	0.1	0.46±0.13	0.95±0.08*	1.00±0.00*	1.00±0.00*
0.7	0.2	0.44±0.09	0.96±0.06*	0.95±0.11*	0.62±0.10
0.7	0.3	0.44±0.14	0.88±0.12*	0.83±0.18*	0.35±0.02
5-partite network					
0.9	0.1	0.75±0.16	1.00±0.00*	1.00±0.00*	1.00±0.00*
0.9	0.2	0.86±0.21	0.98±0.03*	1.00±0.00*	0.95±0.08*
0.9	0.3	0.95±0.10*	1.00±0.00*	1.00±0.00*	0.60±0.07
0.8	0.1	0.60±0.13	1.00±0.00*	1.00±0.00*	1.00±0.00*
0.8	0.2	0.51±0.09	1.00±0.00*	1.00±0.00*	0.89±0.14
0.8	0.3	0.53±0.10	0.99±0.01*	0.95±0.12*	0.45±0.04
0.7	0.1	0.46±0.13	0.99±0.01*	1.00±0.00*	1.00±0.00*
0.7	0.2	0.46±0.11	0.95±0.14*	1.00±0.00*	0.66±0.08
0.7	0.3	0.43±0.05	0.88±0.16*	0.86±0.20*	0.37±0.01

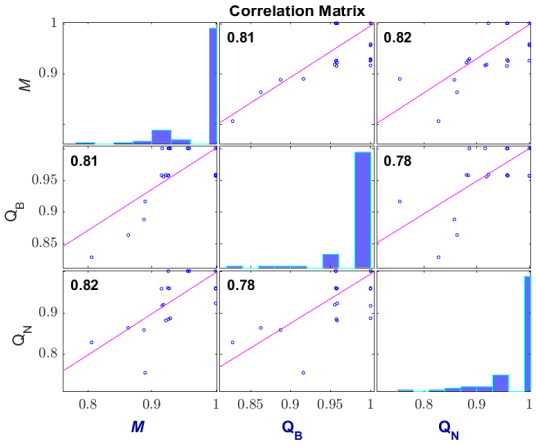

Figure 4: Correlations between best NMI values in the final population for bipartite networks. Histograms show a high count of NMI values equal to 1.

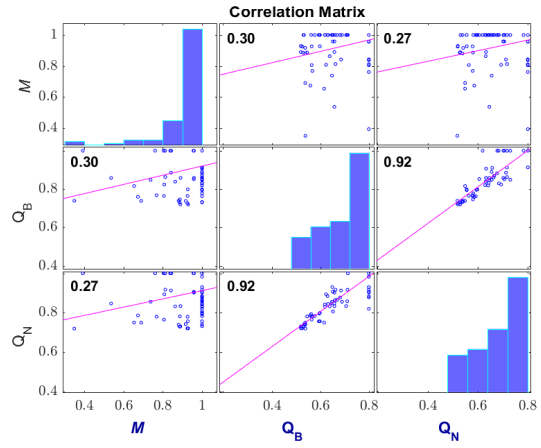

Figure 5: Correlations between NMI values of individuals having best fitnesses in the final population for bipartite networks. There are less NMI values equal to 1 than in Figure 4.

two correlation matrices based on the results obtained for bipartite networks: in Figure 4 we show the correlation between best NMI values obtained using each fitness function, and in Figure 5 the NMI values of the individuals having the best fitness value. The histograms in the diagonal show that the distributions of NMI values differ significantly, with more NMI values of 1 in the Figure 4. Correlation values in Figure 4 also indicate that to obtain the best NMI of 1 any of the fitness functions could be used.

The correlation matrix of NMI values of individuals having the best fitness values in the final population shows that the modularity functions are strongly correlated, with 0.92 and similar distributions, while M exhibits a different behavior, but with more NMI values of 1. These graphs show that, while experiments performed on synthetic data do show better results obtained with our multipartite community fitness, there is still room for improvement, as there are situations in which the maximum value of M does not identify the individual having the best NMI value in the population.

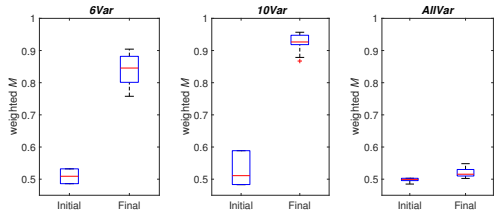

Figure 6: Results obtained for the three bank data networks. Weighted M values for the best individual in the initial population (random) and final population.

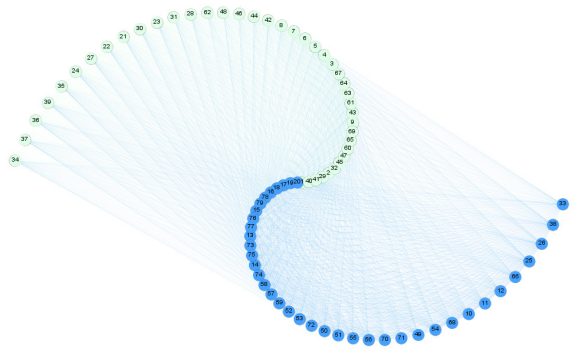

Figure 7: The community structure detected in the Bank dataset, *6Var*.

Bank marketing data. To illustrate the analysis of the bank data we constructed three weighted networks: one containing all variables in the dataset (*AllVar*, 785 nodes, 31549 edges), one that contains only data related to the customer (age-job-marital-education-default-balance-housing-loan-duration-y *10 Var*, 87 nodes, 2845 edges) and one with binary variables removed (age-job-marital-education-balance-duration *6Var*, 79 nodes, 2217 edges). The number of variables determines the number of partitions. We performed 10 independent runs of *NoisyEO* for each network. Modularity values obtained in the 10 runs are represented as boxplots in Figure 6, including initial values: we can see that for the *6Var* and *10var* the search yields a significant increase in modularity, while for the *AllVar* network there is no difference, suggesting that the algorithm was not able to improve the random structure generated in the first iteration. Solutions with the highest modularities are represented in Figures 7-8.

To interpret results, for example, for the *6Var* network, the algorithm divided people aged below 53 year old from the others. At the job category the algorithm placed in the same community *housmaids*, *retired*, and those with *job-unknown*. By marital status divorced are separated from single and married. Education *primary* is separated form all other levels. This results show how the community structure can reveal connections about categories and help analyze big sets of data.

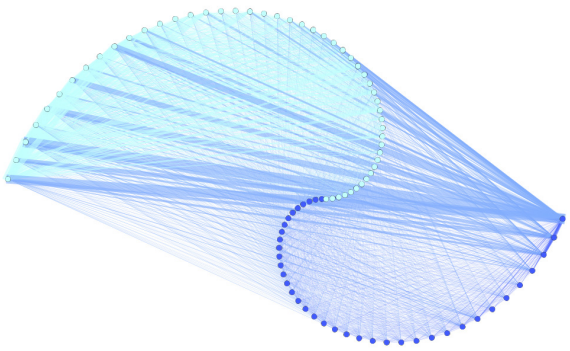

Figure 8: The community structure detected in the Bank dataset, *10Var*.

5 CONCLUSIONS

The problem of community structure detection in multipartite networks can be approached by considering *shadowed* connections between nodes having a common neighbor, even if they belong to the same partition. We propose a fitness function that takes into account these connections and show that it can be used to identify communities on synthetic benchmarks and on a real world application. We can compare results with existing approaches for bipartite networks; numerical results illustrate the efficiency of this approach and identify a new manner of analyzing large datasets.

ACKNOWLEDGEMENT

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS - UEFISCDI, project number PN-II-RU-TE-2014-4-2332.

REFERENCES

[1] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6 (2005), 734–749. DOI: http://dx.doi.org/10.1109/TKDE.2005.99 arXiv:3

[2] Michael J Barber. 2007. Modularity and community detection in bipartite networks. *Physical Review E* 76, 6 (2007), 066102.

[3] Santo Fortunato. 2010. Community detection in graphs. *Physics Reports* 486, 3-5 (feb 2010), 75–174. DOI: http://dx.doi.org/10.1016/j.physrep.2009.11.002

[4] Santo Fortunato and Marc Barthelemy. 2007. Resolution limit in community detection. *Proceedings of the National Academy of Sciences* 104, 1 (2007), 36–41. DOI: http://dx.doi.org/10.1073/pnas.0605965104 arXiv:http://www.pnas.org/content/104/1/36.full.pdf+html

[5] M. Kheirkhahzadeh, A. Lancichinetti, and M. Rosvall. 2016. Efficient community detection of network flows for varying Markov times and bipartite networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* 93, 3 (2016). DOI: http://dx.doi.org/10.1103/PhysRevE.93.032309

[6] Andrea Lancichinetti and Santo Fortunato. 2009. Benchmarks for testing community detection algorithms on directed and weighted graphs with overlapping communities. *Phys. Rev. E* 80 (Jul 2009), 016118. Issue 1. DOI: http://dx.doi.org/10.1103/PhysRevE.80.016118

[7] Andrea Lancichinetti, Santo Fortunato, and János Kertész. 2009. Detecting the overlapping and hierarchical community structure in complex networks. *New Journal of Physics* 11, 3 (2009), 033015.

[8] Zhenping Li, Shihua Zhang, Rui-Sheng Wang, Xiang-Sun Zhang, and Luonan Chen. 2008. Quantitative function for community detection. *Phys. Rev. E* 77 (Mar 2008), 036109. Issue 3. DOI: http://dx.doi.org/10.1103/PhysRevE.77.036109

[9] Xin Liu, Weichu Liu, Tsuyoshi Murata, and Ken Wakita. 2014. A framework for community detection in heterogeneous multi-relational networks. *Advances in Complex Systems* 17, 06 (2014), 1450018. DOI: http://dx.doi.org/10.1142/S0219525914500180

[10] Y Liu, T Yang, L Fu, and J Liu. 2015. Community Detection in Multi-Partite Multi-Relational Networks Based on Information Compression. *Journal of Computational Information Systems* 11, 2 (2015), 693–700. DOI: http://dx.doi.org/10.12733/jcis13086

[11] Rodica Ioana Lung, Mihai Suciu, and Noémi Gaskó. 2017. Noisy extremal optimization. *Soft Computing* 21, 5 (2017), 1253–1270. DOI: http://dx.doi.org/10.1007/s00500-015-1858-3

[12] Prakash Mandayam Comar, Pang-Ning Tan, and Anil K Jain. 2012. A framework for joint community detection across multiple related networks. *Neurocomputing* 76, 1 (2012), 93–104.

[13] A. Miyauchi and N. Sukegawa. 2015. Maximizing Barber's bipartite modularity is also hard. *Optimization Letters* 9, 5 (2015). DOI: http://dx.doi.org/10.1007/s11590-014-0818-7

[14] Sérgio Moro, Paulo Cortez, and Paulo Rita. 2014. A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems* 62 (2014), 22–31. DOI: http://dx.doi.org/10.1016/j.dss.2014.03.001

[15] Mark EJ Newman and Michelle Girvan. 2004. Finding and evaluating community structure in networks. *Physical review E* 69, 2 (2004), 026113.

[16] M. E. J. Newman. 2006. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences* 103, 23 (2006), 8577–8582. DOI: http://dx.doi.org/10.1073/pnas.0601602103 arXiv:http://www.pnas.org/content/103/23/8577.full.pdf+html

[17] Martin Rosvall and Carl T. Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences* 105, 4 (2008), 1118–1123. DOI: http://dx.doi.org/10.1073/pnas.0706851105 arXiv:http://www.pnas.org/content/105/4/1118.full.pdf+html

[18] Yizhou Sun, Yintao Yu, and Jiawei Han. 2009. Ranking-based Clustering of Heterogeneous Information Networks with Star Network Schema. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*. ACM, New York, NY, USA, 797–806. DOI: http://dx.doi.org/10.1145/1557019.1557107