A Hierarchical Clustering Approach to Modularity Maximization

Wolfgang Gaul and Rebecca Klages

Abstract The problem of uncovering clusters of objects described by relationships that can be represented with the help of graphs is an application, which arises in fields as diverse as biology, computer science, and sociology, to name a few. To rate the quality of clusterings of undirected, unweighted graphs, modularity is a widely used goodness-of-fit index. As finding partitions of a graph's vertex set, which maximize modularity, is NP-complete, various cluster heuristics have been proposed. However, none of these methods uses classical cluster analysis, where clusters based on (dis-)similarity data are sought. We consider the lengths of shortest paths between all vertex pairs as dissimilarities between the pairs of objects in order to apply standard cluster analysis methods. To test the performance of our approach we use popular real-world as well as computer generated benchmark graphs with known optimized cluster structure. Our approach is simple and compares favourably w.r.t. results known from the literature.

1 Introduction

Graph clustering, sometimes also referred to as community structure detection in graphs, combines the research areas of graph theory (where binary, symmetric relations between objects are illustrated by undirected, unweighted edges between the vertices of a graph which represent the objects) and cluster analysis (where groups of vertices revealing special graph structures have to be found).

While in standard cluster analysis of dissimilarity data homogeneous clusters are sought that are heterogeneous among each other, in graphs we try to find tightly knit groups of vertices with few edges between these groups.

W. Gaul (✉) · R. Klages
Institute of Decision Theory and Management Science, Karlsruhe Institute of Technology (KIT), Kaiserstr. 12, 76128 Karlsruhe, Germany
e-mail: wolfgang.gaul@kit.edu; rebecca.klages@kit.edu

B. Lausen et al. (eds.), *Algorithms from and for Nature and Life*, Studies in Classification, Data Analysis, and Knowledge Organization, DOI 10.1007/978-3-319-00035-0_7,

A popular goodness-of-fit index to estimate and compare the quality of this kind of clusterings in graphs is called modularity, which was suggested in 2004 (see, e.g., Newman and Girvan 2004; Newman 2004a,b; Clauset et al. 2004). Other suggestions to measure graph clustering solutions are known (see, e.g., Brandes and Erlebach (Eds.) 2005), but will not be addressed here. A definition of modularity as well as a discussion concerning approaches using modularity are presented in Sect. 2. The application of shortest path dissimilarities in order to cluster graphs is motivated and explained in Sect. 3, where we also propose our approach, which consists of the following main steps: (1) computation of all shortest path lengths, (2) application of standard hierarchical clustering, (3) search for possible local improvements with the help of a vertex exchange algorithm. In Sect. 4 we show how our approach performs on benchmark graphs from the literature with known cluster structure. Finally, we give a summary as well as a brief outlook in Sect. 5.

2 Modularity as a Goodness-of-Fit Index

By $G = (V, E)$ we denote an undirected, unweighted graph with a set V of n vertices and a set E of m edges that link pairs of vertices, i.e., $e = (i, j) \in E$ with $i, j, \in V$, where no parallel edges (for each pair of vertices at most one edge exists) and no loops ($e = (i, i), i \in V$) are considered. $A = (A_{ij})$ with $A_{ij} = 1$, if $e = (i, j) \in E$, and $A_{ij} = 0$, otherwise, describes the adjacency information of the graph.

Formally, modularity is defined using the entries A_{ij} of the adjacency matrix A, the degrees k_i of vertices i (number of edges incident to i), and the underlying graph clustering, where c_i denotes the cluster that contains vertex i. In modularity calculations only relationships between vertices in the same cluster are considered which is achieved by using the Kronecker-Delta $\delta(c_i, c_j)$ (equal to 1 if $c_i = c_j$, and equal to 0, otherwise). Then, as formula for the modularity Q one can use

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

Note that every edge is incident to exactly two vertices, so $2m$ is equal to the sum of all vertex degrees. $[-0.5; 1]$ is a theoretical interval for values of Q (see, e.g., Brandes et al. 2007), which depend on the graph's structure, i.e., two clusterings in different graphs cannot be compared using modularity. In real-world graphs (Newman and Girvan 2004) state that optimized values of Q are often elements of the interval $[0.3; 0.7]$.

Modularity has been applied in quite a number of contributions, which shows the importance of this measure in scientific context. Originally introduced by Newman and Girvan (2004) as a new goodness-of-fit index along with a divisive hierarchical graph clustering procedure, Newman (2004a) suggested an agglomerative clustering method, which merges those two clusters in each agglomerative step whose fusion

causes the largest increase or smallest decrease of modularity. Brandes et al. (2008) showed that modularity maximization over all partitions of the vertex set V is NP-complete. Therefore, various heuristics to tackle this problem have been proposed. A modification of Newman's agglomerative algorithm was given by Schuetz and Caflisch (2008), which enables the fusion of more than two communities in each iteration step. Other hierarchical approaches were given by Radicchi et al. (2004), Xiang et al. (2008) as well as Mann et al. (2008). Algorithms similar to hierarchical clustering have been proposed by Arenas et al. (2007), Djidjev (2008), Zhu et al. (2008), and Blondel et al. (2008), who developed different procedures to coarsen the graph in question. Then, the coarsened copies of the original graph are either clustered or the coarsest version of the graph is defined as a clustering, whose cluster solutions are conveyed and refined to fit the original graph using iterative uncoarsening. Also, approaches using heuristics known from other fields of research have been applied, for example mathematical optimization (Duch and Arenas (2005) employ an extremal optimization procedure, Agarwal and Kempe (2008) express the problem with the help of linear and vector programming). The application of probabilitic flows on random walks in graphs (Rosvall and Bergstrom 2008) and matrix factorization (Ma et al. 2010) have also been suggested. A significant number of authors (see e.g.,Newman 2006) use spectral clustering algorithms (see Nascimento and de Carvalho (2010) for a recent overview).

Besides the development of various procedures that aim to find a partition of the vertex set with highest possible modularity, the concept has also been criticized. Fortunato and Barthélemy (2007) showed that there is a lower bound to the sizes of clusters that can be detected using heuristics which strive to maximize modularity. This lower bound depends on the number of vertices n and the interconnectedness of the clusters. Variations of modularity were proposed to avoid this weakness (see e.g., Li et al. (2008), who suggested a local measure that takes the density of subgraphs into account). However, the original definition of modularity is still widely used and extensions to weighted graphs (Newman 2004b) as well as to directed graphs (e.g., Arenas et al. 2007; Leicht and Newman 2008; Kim et al. 2010) have been presented. Good et al. (2010) review the performance of modularity maximization in practical contexts.

3 A New Heuristic to Find Clusters in Graphs

Given the many contributions in which modularity was used for community structure detection we considered the following idea to cluster a graph into subgraphs with closely connected vertices and comparatively few edges between different subgraphs: In a graph the important information is stored in the adjacency matrix $A = (A_{ij})$. While $A_{ij} = 1$ might be a reason to put vertex i and j into the same cluster, for a pair of vertices i and j with $A_{ij} = 0$ there is no information how similar i and j might be. They could have a common neighbour but they could also be in completely different areas of the graph. Therefore, we define the dissimilarity

between two vertices as length of a shortest path that connects them in the graph. If no path exists a sufficiently large constant is used to indicate this situation. Now, the solution of the underlying problem in terms of standard cluster analysis is straightforward and, as writing restrictions do not allow to describe the single steps of the new heuristic in mathematical terms, we give the following verbal explanations: First, we determine the lengths of the shortest paths between all pairs of vertices in the graph (see, e.g., Floyd 1962; Warshall 1962). Second, we apply a hierarchical clustering method to the shortest path length matrix computed in the first step and calculate the modularity for the clusterings given by the hierarchy. (Average and Weighted Average Linkage were well-suited for our data.) Third, an exchange algorithm called vertex mover (Schuetz and Caflisch 2008) is applied to the clustering in the hierarchy with highest modularity, which also provides the number of clusters needed to check whether improvements by exchange operations are still possible.

4 Performance on Benchmark Graphs

To test the performance of our approach we used several well-known undirected, unweighted real-world as well as computer generated benchmark graphs.

Example 1. The first example is a well-known real-world graph by Zachary (1977), who examined the relations between 34 members of a karate club. Two vertices of the graph are adjacent, if the corresponding people spent a significant amount of time together during the examination (see Fig. 1).

By chance there was a dispute between the principal karate teacher (vertex 33) and the administrator (vertex 1) of the club while Zachary studied the relations in this group, which caused the club to spilt into two subgroups. This real-life partition is depicted in Fig. 1 by a black line. The modularity of this spilt is 0.3715. Interestingly, a better Q for a solution with two clusters is 0.3718. Not only is this value only slightly larger, the clusters are also almost the same, just for vertex 10 the group membership has to be changed. This shows that modularity can successfully be used to predict the clusters of the split of this social network which separated into two groups. Our approach finds the two-cluster-solution with the better modularity mentioned above, which was also detected by Newman and Girvan (2004).

Additionally, our method finds the largest known modularity value for a clustering of this graph which is $Q = 0.4198$ as also reported by Duch and Arenas (2005). This value is obtained for the division into four clusters, which is also shown in Fig. 1, where the four different colors of the vertices indicate the four groups. These clusters happen to be subgroups of the real-life decomposition that Zachary (1977) observed.

Example 2. As real-world graphs known from the literature can be very specific and sometimes need lengthy explanations of the relationships that underly the

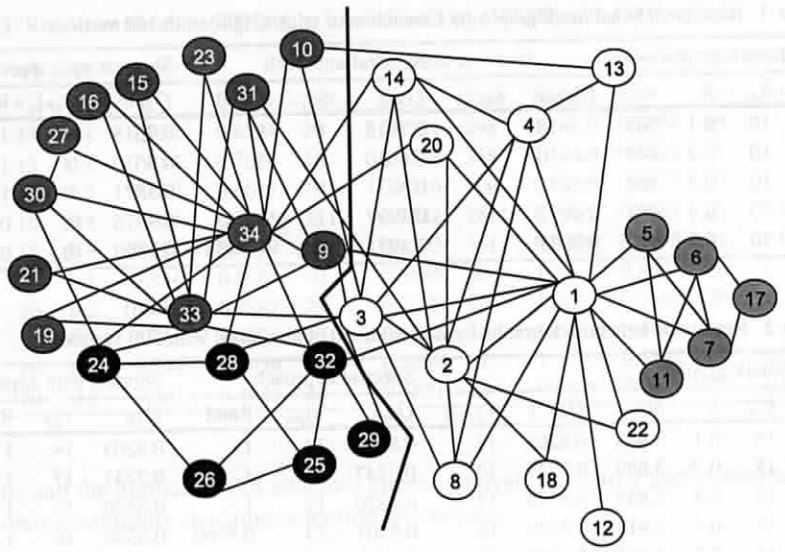

Fig. 1 The Zachary network of friendship ties between members of a karate club

described situation we tested our new method on a class of computer generated benchmark graphs with built-in community structures introduced by Lancichinetti et al. (2008). Although the authors argue that there is no guarantee that the bulit-in community structures constitute solutions with highest modularity these graphs provide the up-to-now best known benchmarks. They can be constructed for any choice of the following three parameters: n (the number of vertices), k_{av} (the average vertex degree), and μ (a mixing parameter which denotes the fraction of a vertex v's neighbours not in c_v). Heterogeneous vertex degrees are modeled by a power law distribution with parameter τ_1, heterogeneous cluster sizes are modeled by a power law distribution with parameter τ_2. To take into account restrictions for real-life graphs the authors propose $\tau_1 \in [2,3]$, $\tau_2 \in [1,2]$, and report on results of testgraphs with all four combinations of the extreme cases of τ_1 and τ_2 for which a very similar behaviour of the modularity calculation was found in all cases. Therefore we chose $\tau_1 = 2$ and $\tau_2 = 1$. So far we used different numbers of vertices $n \in \{100; 500; 1{,}000\}$ with adequate average degrees $k_{av} \in \{10; 15; 20\}$ and mixing parameters $\mu \in \{0.1; 0.2; 0.3; 0.4; 0.5\}$. Note that $\mu = 0.1$ indicates a strong cluster structure as 90 % of the neighbours of each vertex v are in the same cluster c_v, while in graphs with $\mu = 0.5$ half of the neighbours of each vertex are in other clusters than v. A minimal and a maximal value for the cluster sizes can also be selected. The software to construct these benchmarks is explained in a read-me file provided by the authors, in which the cluster sizes are chosen to be in the interval [20, 50], so we also used these values. In order to better analyze the results found by our method, we did not only compare our findings with the built-in community structures given by Lancichinetti et al. (2008), but also implemented the spectral approach proposed by

Table 1 Results on benchmark graphs by Lancichinetti et al. (2008) with 100 vertices

Benchmark graphs						Spectral approach			Shortest path approach		
n	k_{av}	μ	m	Q_{Bench}	$\mathscr{C}_{Bench}$	Q_{Spec}	$\mathscr{C}_{Spec}$	Rand	Q_{SP}	$\lvert\mathscr{C}_{SP}\rvert$	Rand
100	10	0.1	503	0.7618	8	0.7618	8	1	0.7618	8	1
100	10	0.2	494	0.6610	8	0.6610	8	1	0.6610	8	1
100	10	0.3	488	0.5871	9	0.5871	9	1	0.5871	9	1
100	10	0.4	488	0.4997	11	0.4997	11	1	0.4976	10	0.9836
100	10	0.5	495	0.3977	11	0.3871	8	0.9453	0.3899	10	0.9675

Table 2 Results on benchmark graphs by Lancichinetti et al. (2008) with 500 vertices

Benchmark graphs						Spectral approach			Shortest path approach		
n	k_{av}	μ	m	Q_{Bench}	$\mathscr{C}_{Bench}$	Q_{Spec}	$\mathscr{C}_{Spec}$	Rand	Q_{SP}	$\mathscr{C}_{SP}$	Rand
500	15	0.1	3,948	0.8203	14	0.8203	14	1	0.8203	14	1
500	15	0.2	3,889	0.7247	17	0.7247	17	1	0.7247	17	1
500	15	0.3	3,855	0.6320	17	0.6320	17	1	0.6320	17	1
500	15	0.4	3,918	0.5330	16	0.5207	17	0.9960	0.5330	16	1
500	15	0.5	3,853	0.4262	15	0.4037	12	0.9563	0.4229	14	0.9921
500	20	0.1	4,663	0.8187	15	0.8187	15	1	0.8187	15	1
500	20	0.2	5,041	0.7262	16	0.7262	16	1	0.7262	16	1
500	20	0.3	4,801	0.6167	13	0.6167	13	1	0.6167	13	1
500	20	0.4	5,065	0.5238	14	0.5238	14	1	0.5238	14	1
500	20	0.5	4,906	0.4202	13	0.4202	13	1	0.4170	12	0.9913

Newman (2006) to see which results a known algorithm obtains on these graphs. Of course, comparisons to other modularity optimizing techniques could be performed. We selected spectral clustering because of the recent overview of Nascimento and de Carvalho (2010).

In the Tables 1–3 we present a comparison between the modularity value Q_{Bench} of the built-in community structures $\mathscr{C}_{Bench}$ to the modularity values Q_{Spec} of solutions $\mathscr{C}_{Spec}$ found by our implementation of Newmans spectral method and to Q_{SP} of clusterings $\mathscr{C}_{SP}$ constructed by our own SP (Shortest Path) approach. With $|\mathscr{C}|$ as cardinality of a clustering $\mathscr{C}$, the numbers $|\mathscr{C}_{Bench}|$, $|\mathscr{C}_{Spec}|$, and $|\mathscr{C}_{SP}|$ of the clusters of the three solutions are given along with the Rand indices (see, e.g., Hubert and Arabie 1985) comparing the clusterings of the spectral procedure and of our method with the benchmark solution.

From the 25 (n, k_{av}, μ, m) benchmark graphs in the Tables 1–3 in three cases ((100, 10, 0.4, 488), (500, 20, 0.5, 4,906), and (1,000, 20, 0.4, 9,731)) the spectral approach performed slightly better while in seven cases ((100, 10, 0.5, 495), (500, 15, 0.4, 3,918), (500, 15, 0.5, 3,853), (1,000, 15, 0.3, 7,609), (1,000, 15, 0.4, 7,631), (1,000, 15, 0.5, 7,571), and (1,000, 20, 0.5, 9,581)) our shortest path approach was in front. In the other 15 cases both approaches showed identical outcomes. These are convincing results that the enrichment of the adjacency information by shortest path

Table 3 Results on benchmark graphs by Lancichinetti et al. (2008) with 1,000 vertices

Benchmark graphs						Spectral approach			Shortest path approach		
n	k_{av}	μ	m	Q_{Bench}	$\mathscr{C}_{Bench}$	Q_{Spec}	$\mathscr{C}_{Spec}$	Rand	Q_{SP}	$\mathscr{C}_{SP}$	Rand
1,000	15	0.1	7,930	0.8594	29	0.8594	29	1	0.8594	29	1
1,000	15	0.2	7,858	0.7624	31	0.7624	31	1	0.7624	31	1
1,000	15	0.3	7,609	0.6617	30	0.6516	30	0.9969	0.6617	30	1
1,000	15	0.4	7,631	0.5615	29	0.5522	28	0.9965	0.5583	25	0.9901
1,000	15	0.5	7,571	0.4639	30	0.4293	16	0.9396	0.4555	21	0.9755
1,000	20	0.1	9,834	0.8585	30	0.8585	30	1	0.8585	30	1
1,000	20	0.2	10,017	0.7582	29	0.7582	29	1	0.7582	29	1
1,000	20	0.3	9,765	0.6622	30	0.6622	30	1	0.6622	30	1
1,000	20	0.4	9,731	0.5659	31	0.5659	31	1	0.5596	25	0.9868
1,000	20	0.5	9,581	0.4633	30	0.4581	24	0.9875	0.4609	23	0.9880

lengths and the application of standard cluster analysis leads to a useful alternative to known community structure detection techniques.

5 Conclusion

Against the background that finding a partition of a graph's vertex set with maximal modularity is a NP-complete problem, we proposed the application of standard cluster analysis methods developed for dissimilarity data to the problem of graph clustering. As dissimilarities between pairs of objects, in our case vertices, are needed, we transformed the adjacency matrix into a matrix of shortest path lengths between all vertex pairs of the graph. From all clusterings of the hierarchy that was computed by a standard agglomerative cluster procedure we chose the one with highest modularity as starting solution for an exchange algorithm. On several benchmark graphs we obtained promising results showing that our approach compares favorably with findings from the literature. A next challenge is to transfer the ideas presented in this paper to directed graphs.

References

Agarwal, G., & Kempe, D. (2008). Modularity-maximizing graph communities via mathematical programming. *European Physical Journal B, 66*, 409–418.

Arenas, A., Duch, J., Fernández, A., & Gómez, S. (2007). Size reduction of complex networks preserving modularity. *New Journal of Physics, 9*, 176.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., & Lefebvre, E. (2008). Fast unfolding of community hierarchies in large networks. *Journal of Statistical Mechanics, 10*, P10008.

Brandes, U., & Erlebach, T. (Eds.). (2005). Network analysis: methodological foundations. In *Lecture notes in computer science* (Vol. 3418). Berlin/Heidelberg: Springer.

Brandes, U., Delling, D., Gaertler, M., Goerke, R., Hoefer, M., Nikoloski, Z., & Wagner, D. (2007). On finding graph clusterings with maximum modularity. In *Lecture notes in computer science* (Vol. 4769, pp. 121–132). Berlin/Heidelberg: Springer.

Brandes, U., Delling, D., Gaertler, M., Goerke, R., Hoefer, M., Nikoloski, Z., & Wagner, D. (2008). On modularity clustering. *IEEE Transactions on Knowledge and Data Engineering, 20*(2), 172–188.

Clauset, A., Newman, M. E., & Moore, C. (2004). Finding community structure in very large networks. *Physical Review E, 70*, 066111.

Djidjev, H. N. (2008). A scalable multilevel algorithm for graph clustering and community structure detection. In *Lecture notes in computer science* (Vol. 4936, pp. 117–128). Berlin/Heidelberg: Springer.

Duch, J., & Arenas, A. (2005). Community detection in complex networks using extremal optimization. *Physical Review E, 72*, 027104.

Floyd, R. W. (1962). Algorithm 97: shortest path. *Communications of the ACM, 5*(6), 345–345.

Fortunato, S., & Barthélemy, M. (2007). Resolution limit in community detection. *Proceedings of the National Academy of Sciences, 104*(1), 36–41.

Good, B. H., de Montjoye, Y.-A., & Clauset, A. (2010). The performance of modularity maximization in practical contexts. *Physical Review E, 81*, 046106.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification, 2*, 193–218.

Kim, Y., Son, S.-W., & Jeong, H. (2010). LinkRank: finding communities in directed networks. *Physical Review E, 81*, 016103.

Lancichinetti, A., Fortunato, S., & Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. *Physical Review E, 78*, 046110.

Leicht, E. A., & Newman, M. E. (2008). Community structure in directed networks. *Physical Review Letters, 100*, 118703.

Li, Z., Zhang, S., Wang, R.-S., Zhang, X.-S., & Chen, L. (2008). Quantitative function for community detection. *Physical Review E, 77*, 036109.

Ma, X., Gao, L., Yong, X., & Fu, L. (2010). Semi-supervised clustering algorithm for community structure detection in complex networks. *Physica A, 389*, 187–197.

Mann, C. F., Matula, D. W., & Olinick, E. V. (2008). The use of sparsest cuts to reveal the hierarchical community structure of social networks. *Social Networks, 30*, 223–234.

Nascimento, M. C., & de Carvalho, A. C. (2010). Spectral methods for graph clustering – a survey. *European Journal of Operational Research, 211*(2), 221–231.

Newman, M. E. (2004a). Fast algorithm for detecting community structure in networks. *Physical Review E, 69*, 066133.

Newman, M. E. (2004b). Analysis of weighted networks. *Physical Review E, 70*, 056131.

Newman, M. E. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical Review E, 74*, 036104.

Newman, M. E., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E, 69*, 026113.

Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., & Parisi, D. (2004). Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences, 101*(9), 2658–2663.

Rosvall, M., & Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences, 105*, 1118–1123.

Schuetz, P., & Caflisch, A. (2008). Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement. *Physical Review E, 77*, 046112.

Warshall, S. (1962). A theorem on Boolean matrices. *Journal of the ACM, 9*(1), 11–12.

Xiang, J., Hu, K., & Tang, Y. (2008). A class of improved algorithms for detecting communities in complex networks. *Physica A, 387*, 3327–3334.

Zhu, Z., Wang, C., Ma, L., Pan, Y., & Ding, Z. (2008). Scalable community discovery of large networks. In *Proceedings of the 2008 ninth international conference on web-age information management*, Zhangjiajie, China, pp. 381–388.

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of Anthropological Research, 33*(4), 452–473.