Slovak University of Technology in Bratislava
Faculty of Mechanical Engineering
17th Conference on Applied Mathematics **APLIMAT 2018**

Proceedings

GRAPHLET COUNTING

HOČEVAR Tomaž (SI)

Abstract. Graphlet analysis describes a graph's topology by observing frequencies of 4- or 5-node induced subgraphs (graphlets) in the graph. We developed a graphlet counting approach based on a system of linear equations that relate the graphlet counts. For sparse large networks, such as those appearing in bioinformatics, the resulting algorithm is an order of magnitude faster than the existing approaches.

Keywords: graphlets, counting, network, bioinformatics

Mathematics subject classification: Primary 92C42, 05C60; Secondary 68R10, 68W01

1 Introduction

Networks are a very general modeling tool. All we need is a set of objects and a binary relation between them and we can build a network. Therefore, networks arise in a variety of research areas, from social sciences [2] and linguistics [3] to chemistry, pharmacology and bioinformatics [23]. We can analyze such networks to discover interesting properties that translate to new understandings of the underlying process.

One such approach is graphlet analysis. Graphlets are small connected patterns. Examples of 4-node graphlets are a path (P_4), a cycle (C_4), a cycle with a diagonal (called a diamond), etc. They are typically observed as induced subgraphs of the network. As small patterns present in the network, they represent a summary of the network's local structure. For example, denser networks would contain more diamonds and cliques, while sparser ones would contain a higher proportion of paths and cycles. Some other types of networks might differentiate themselves by a higher or lower presence of some other graphlet.

We can use graphlets to summarize the entire structure of the network by counting how many times each graphlet appears in it [22]. The other option is more node-centric and can be used to describe the structure around a particular node [16]. Let us describe a node by a vector of how many times it participates in each graphlet. To be even more precise, we can introduce the notion of orbits. We can say that there are two types or roles of nodes in a star graph: the central node and leaves. More formally, the orbits are equivalence classes of nodes under the action of the automorphisms of each

graphlet. There are 11 orbits in 4-node graphlets. Therefore, we can assign a vector of orbit counts to each node that tells us, how many times does a node occur in the role of each orbit.

Usually, we observe graphlets consisting of up to 4 or 5 nodes. Because the number of graphlets of larger sizes grows very quickly (Table 1), it is not feasible to observe larger ones. Therefore, the graphlet frequency vectors would contain mostly zeros. Besides, larger graphlets already span a large portion of the network (small-world effect), which defeats the purpose of observing the local structure of a network.

k	2	3	4	5	6	7	8	9	10
graphlets	1	2	6	21	112	853	11117	261080	11716571

Tab. 1. The number of k-node graphlets.

If we use graphlet counts as a signature of a network's topology, how well does this actually describe the network? An interesting related problem is the reconstruction conjecture [13, 20]—are two graphs with the same multiset of vertex-deleted subgraphs isomorphic? A multiset of vertex-deleted subgraphs is equivalent to the frequencies of all $(n-1)$-node graphlets in a graph with n nodes. What about smaller graphlets? It is not hard to find a counterexample for 5-node graphlets. Graphs in Figure 1 have the same number of all 5-node graphlets but are not isomorphic.

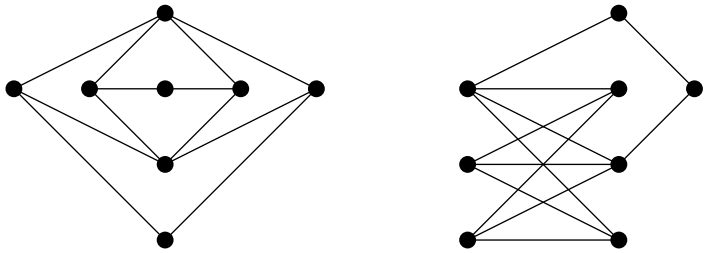

Fig. 1. Two nonisomorphic graphs with equal 5-node graphlet counts.

1.1 Applications

Graphlet analysis originates from bioinfomatics. There it was first used to propose a better random model for protein-protein interaction (PPI) networks. Pržulj et al. [22] proposed the relative graphlet frequency distance: a function that compares the local structure of two networks by comparing the relative frequency of each graphlet in the networks. They used their distance measure to show that the graphlet distribution in PPI networks is more similar to the graphlet distribution in random geometric graphs (induced by a set of randomly placed points in 3D space that are close enough) than to random networks that are modeling scale-free graphs.

Node-centric applications of graphlet analysis in bioinformatics are based on the assumption that the network topology of the modeled process is related to the function of objects corresponding to nodes. Let us consider the most prominent example of PPI networks. We model proteins with nodes and their interactions with edges. We assume that proteins with similar functions should form similar patterns of connections to neighbouring nodes in the network. Therefore, a similar local topology of two nodes is an indication that they might have similar protein functions or properties and should be investigated further. The goal of graphlet analysis in bioinformatics is often to narrow the search space of possible candidates that have to be further investigated experimentally; speeding-up the research process and

reducing the expenses. Graphlets have been successfully employed to predict protein functions [16] and aid in discovery of cancer-related [14] or age-related genes [17].

Besides the already mentioned applications, they also have other, more indirect uses. One such example is their use in algorithms for network alignment. Given two similar networks, we would like to align or pair the nodes from both networks to maintain the adjacency. The alignment should be such that two nodes in the first network are adjacent exactly when their corresponding aligned nodes in the second network are adjacent. Because there is often no exact solution, we have to allow some mismatches. Exact penalties for such mismatches and the optimization function are a matter of application. In any case, we can make use of vectors of orbit counts for each node as a heuristic for aligning nodes that have a similar local topology. This led to development of several graphlet-based network alignment algorithms: GRAAL [11], H-GRALL [15], MI-GRAAL [12].

Finally, let us mention some other non-biological applications of graphlet analysis. Juszczyszyn et al. [9] observed graphlets in email-based social networks. Zhang et al. [28] used graphlets as an additional feature in the problem of aerial image categorization, while Mugan et al. [18] employed them for the task of entity resolution.

2 Graphlet Counting

Graphlet counting tools used in bioinformatics rely on exhaustive enumeration of all k-node graphlets. Enumeration of graphlets is a challenging problem itself. Ideally, it should require a computational time that is proportional to the actual number of all graphlets. Various approaches aim to achieve this by optimizing isomorphism testing [26], exploiting graphlet symmetries [27], simultaneously counting all graphlets [24], etc. However, we're usually not interested in their actual occurrences but only in their counts. It is clear that we can count graphlets at least as fast as we can enumerate them; but can we count them faster than we can enumerate them?

Let us consider the most simple case, a triangle. Surprisingly, we can count the number of triangles in an arbitrary graph with n nodes in $o(n^3)$ [8]. Let A represent a graph's adjacency matrix. Then, j-th element in i-th row of A^3 contains the number of paths of length 3 between from node i to j. It is well known that we can multiply two matrices in $o(n^3)$. The number of triangles is equivalent to $\frac{1}{6}\sum_{i=1}^{n}(A^3)_{i,i}$. Every cycle will be considered once for each of its three nodes and once for each direction, together six times. The solution for detecting larger cliques by Nešetřil and Poljak [19] reduces the problem to detecting a triangle in an auxiliary graph with $O(n^{k/3})$ nodes, resulting in a $o(n^3)$ algorithm.

We can count cliques faster than we can enumerate them. What about other graphlets? Kloks [10] presented a system of equations that relates all 4-node graphlets and can be set up and solved in $O(n^\omega)$, where ω represents the exponent in the time complexity of matrix multiplication. Eq. 1 is one of his equations (A denotes the adjacency matrix of graph $G = (V, E)$ and C the adjacency matrix of its complement).

$$\sum_{(x,y)\in E}(AC)_{x,y}(CA)_{x,y} = 4\#C_4 + \#P_4 \tag{1}$$

All mentioned approaches focus on solving a pattern counting problem in arbitrary dense graphs and rely on fast matrix multiplication techniques. However, in practical applications we encounter mostly sparse graphs, where it is often not possible to even construct an adjacency matrix, let alone multiply it.

2.1 ORCA

We developed ORCA (ORbit Counting Algorithm) [5] with the goal of outperforming existing enumeration techniques on sparse networks. It is based on a system of carefully chosen equations that can be set up faster than we can enumerate all k-node graphlets. In fact, it requires enumeration of only $(k-1)$-node graphlets. ORCA computes all orbit counts for every node in the network. From this more detailed statistic we can easily compute graphlet counts for each node and the total number of graphlets in the network.

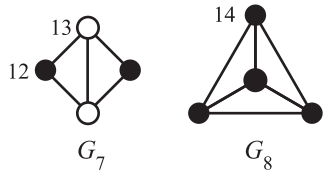

Fig. 2. Graphlets G_7 and G_8.

$$o_{12} + 3o_{14} = \sum_{y,z:\, y<z, G[\{x,y,z\}] \cong G_2} (c(y,z) - 1) \tag{2}$$

Equation 2 is an example of one such relation for a given node x in graph G that establishes a connection between orbits 12 and 14 of 4-node graphlets. The sum runs over all graphlets G_2 (C_3, a cycle with 3 nodes) in G that contain x as one of the nodes. We refer to the other two nodes as y and z. There are $c(y,z) - 1$ common neighbours of nodes y and z (without x), which together with x, y and z induce either a subgraph equivalent to graphlet G_7 with x in orbit 12 or graphlet G_8 with x in orbit 14. It turns out that we consider each occurrence of x in orbit 14 three times.

The system of equations for counting 4-node graphlets was designed on paper through trial and error of finding a system of independent equations that is efficient to set up. The rank of this system of equations is by 1 smaller than the number of orbits. Therefore, we have to know or compute one of the orbit counts. It turns out that we can do this efficiently (in sparse real-world network) for complete graphlets.

Moving on to 5-node graphlets required a more systematic approach (there are 58 orbits in 5-node graphlets). We observed possible extensions of 4-node graphlets by attaching a new node to some subset of its nodes. Each such extension leads to a candidate equation. We also showed that a similar approach as for counting node-orbits can be used for counting edge-orbits [6]. The developed algorithm has been used as a benchmark for other parallel [1] and approximate [4] graphlet counting methods.

A generalization to larger graphlets is of more interest from theoretical than practical perspective. We determined the conditions for construction of the required equations for the ORCA algorithm. For every node x in every graphlet G there has to exist another node y such that:

1. $d(y) \leq k-2$,
2. $G \setminus \{y\}$ is a connected graph,
3. if $d(y) = k-2$, the neighbours of y induce a connected graph,

where $d(y)$ represents the degree of node y. Such node y will represent the node by which we will try to extend a smaller graphlet with $k-1$ nodes. The first condition ensures that we are not concerned with common neighbours of too large sets of nodes, which are a part of precomputation. The second condition ensures that the algorithm will have to enumerate $(k-1)$-node graphlets and not some disconnected patterns. The last condition restricts the algorithm's space complexity and prevents memory accesses from dominating the run time. The algorithm's time complexity for counting k-node graphlets is $O(ed^{k-3} + T_k)$; e represents the number of edges, d is the maximum degree of a node and T_k is the time required to count cliques of size k.

We proved that such nodes indeed exist [7] for every noncomplete graphlet with at least 4 nodes. The only exception is a cycle with 4 nodes, C_4, which can be handled separately. There is a simple strategy that accomplishes this. From the nodes that are farthest away from x (suppose they are at distance u) pick the one with the lowest degree. If it doesn't satisfy the conditions, pick the node with the lowest degree among those at distance $u-1$.

Can we adapt this approach for the dynamic setting of the graphlet counting problem? We are given interleaved queries of three types: add an edge, remove an edge, compute orbit counts for a given node. The algorithm already handles the computation of orbit counts for each node independently, we just have to efficiently update the numbers of common neighbours (i.e. values $c(a,b)$), which are used in equations. This can be done and results in a dynamic version of ORCA which is as fast as the original version when used to add all edges into an empty graph and than compute the orbit counts for all nodes. Another version of dynamic graphlet counting dispenses with the last query type (obtaining orbit counts for a given node) and instead constantly maintains the number of graphlets in the network after each addition or removal of an edge.

2.2 Application: Generating Random Graphs

How do we describe a general topology of a real-life network? By picking a random model that generates similar networks; social networks, for instance, are described as scale-free. This also enables us to generate an arbitrary number of topologically similar networks. Such ensembles can be used, for example, to evaluate the performance of a newly designed method on a larger set of networks. Another related concept are network motifs [25], where an ensemble of random networks is used as a null model to discover over-represented patterns (motifs) in the original network. Because graphlet counts represent a summary of a network's topology, we attempted to generate random networks that have a prescribed (or at least very similar) number, X_i, of each 4-node graphlet.

$$F(G) = \sum_{i=0}^{8} |\log(N_i(G)+1) - \log(X_i+1)| \tag{3}$$

The process of generating a random network with a given graphlet distribution was formulated as an optimization procedure. We specified an optimization function, $F(G)$, based on the relative graphlet frequency distance, but removed the "relative" part (Eq. 3). A simple hill-climbing optimization picks random edges to add or remove from the network and checks the similarity of the modified network to the target distribution of graphlets. If the similarity improves, we keep the change, otherwise we revert it. Using the dynamic version of ORCA that is capable of maintaining graphlet counts under such modifications of the network makes the process feasible for the current size of PPI networks.

	$F(G)$	G_0	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8
start	2.761	11000	42251	13492	167375	27308	1462	95264	25235	9321
finish	0.082	11626	418249	24796	11976090	9379476	387973	2749137	285672	22358
target	0.0	11626	418270	18225	11110782	9383348	445186	2216446	285736	22376

Tab. 2. Graphlet counts obtained by starting with a random geometric network.

The similarity of the obtained random network depends on the choice of the network that we start from. We attempted to approximate the graphlet counts observed in a PPI network of bacteria E. Coli by starting with a random geometric network (Table 2). The starting and target networks both contained around 3000 nodes and 11000 edges. The optimization process requires less than an hour for 1.5 million iterations. The resulting network that we obtained through optimization of 4-node graphlet counts had a surprisingly similar distribution of 5-node graphlet counts (Figure 3). This is a preliminary result that requires further investigation. Intuitively, the number of 2-node graphlets (the number of edges) does not tell us much about the number of 3-node graphlets. On the other hand, 4-node graphlets seem to restrict the space of 5-node graphlets much more. It would make sense that this influence grows with the size of graphlets.

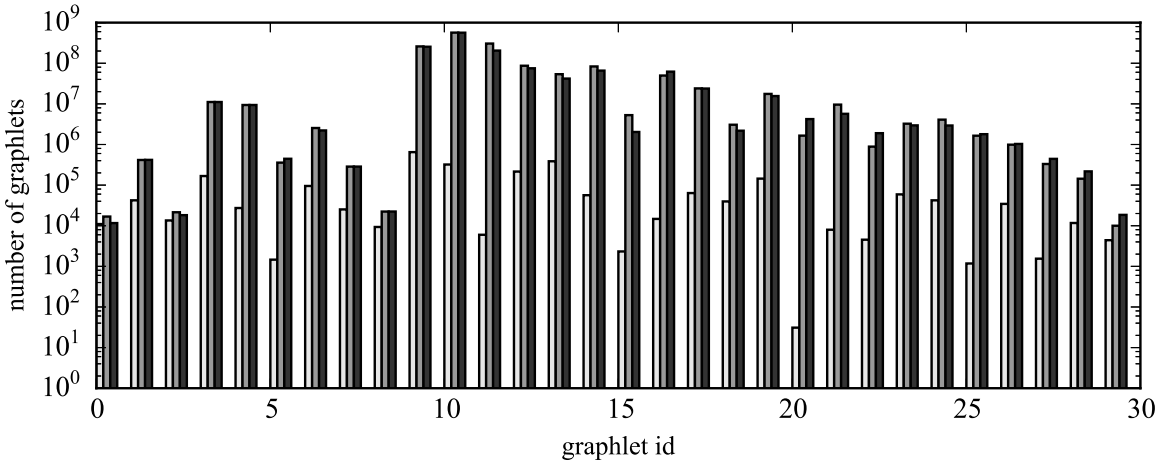

Fig. 3. The 5-node graphlet counts obtained by optimizing 4-node counts. White bars correspond to the distribution in the starting graph, gray bars corresponds to the obtained (generated) graph and black bars to the target graphlet distribution of a PPI network.

3 Conclusion

Bioinformatics is a rapidly evolving field with a growing amount of data due to recent technological advances. Simple improvements such as parallelization are often not enough; instead, new algorithmic and mathematical insights are required. We developed ORCA, which is based on combinatorial observations that establish relations between individual graphlet counts. These are employed to design an efficient graphlet counting algorithm that outperforms other approaches, which are based on exhaustive enumeration.

However, there is still a lot of room for improvement. Ortmann et al. [21] have already surpassed our method for counting 4-node graphlets. We know that the system of equations must have nonnegative integral solutions. Could we somehow incorporate this property into a more efficient algorithm?

Several applications would benefit from an efficient graphlet counting method in weighted graphs. Can we approach this with a similar method or does it require something new?

Acknowledgement

The research presented in this paper was funded by the Slovenian research agency grants J2-5480 and P2-0209.

References

[1] AHMED, N. K., NEVILLE, J., ROSSI, R. A., DUFFIELD, N.: *Efficient Graphlet Counting for Large Networks*, in *2015 IEEE International Conference on Data Mining*, IEEE, nov 2015, pp. 1–10.

[2] BORGATTI, S. P., MEHRA, A., BRASS, D. J., LABIANCA, G.: *Network Analysis in the Social Sciences*, Science, vol. 323, no. 5916, (2009), pp. 892–895.

[3] CANCHO, R. F. i., SOLE, R. V.: *The small world of human language*, Proceedings of the Royal Society B: Biological Sciences, vol. 268, no. 1482, (2001), pp. 2261–2265.

[4] ELENBERG, E. R., SHANMUGAM, K., BOROKHOVICH, M., DIMAKIS, A. G.: *Distributed Estimation of Graph 4-Profiles*, in *Proceedings of the 25th International Conference on World Wide Web - WWW '16*, ACM Press, New York, New York, USA, 2016, pp. 483–493.

[5] HOČEVAR, T., DEMŠAR, J.: *A combinatorial approach to graphlet counting*, Bioinformatics, vol. 30, no. 4, (2014), pp. 559–565.

[6] HOČEVAR, T., DEMŠAR, J.: *Computation of Graphlet Orbits for Nodes and Edges in Sparse Graphs*, Journal of Statistical Software, vol. 71, no. 10.

[7] HOČEVAR, T., DEMŠAR, J.: *Combinatorial algorithm for counting small induced graphs and orbits*, PLOS ONE, vol. 12, no. 2.

[8] ITAI, A., RODEH, M.: *Finding a Minimum Circuit in a Graph*, SIAM Journal on Computing, vol. 7, no. 4, (1978), pp. 413–423.

[9] JUSZCZYSZYN, K., KAZIENKO, P., MUSIAŁ, K.: *Local Topology of Social Network*, in *KES '08 Proceedings of the 12th international conference on Knowledge-Based Intelligent Information and Engineering Systems, Part II*, Zagreb, 2008, pp. 97–105.

[10] KLOKS, T., KRATSCH, D., MÜLLER, H.: *Finding and counting small induced subgraphs efficiently*, Information Processing Letters, vol. 74, no. 3-4, (2000), pp. 115–121.

[11] KUCHAIEV, O., MILENKOVIĆ, T., MEMISEVIĆ, V., HAYES, W., PRŽULJ, N.: *Topological network alignment uncovers biological function and phylogeny.*, Journal of the Royal Society, Interface / the Royal Society, vol. 7, no. 50, (2010), pp. 1341–1354.

[12] KUCHAIEV, O., PRŽULJ, N.: *Integrative network alignment reveals large regions of global network similarity in yeast and human.*, Bioinformatics (Oxford, England), vol. 27, no. 10, (2011), pp. 1390–1396.

[13] MCKAY, B. D.: *Small graphs are reconstructible.*, Australasian Journal of Combinatorics, vol. 15, (1997), pp. 123–126.

[14] MILENKOVIĆ, T., MEMISEVIĆ, V., GANESAN, A. K., PRŽULJ, N.: *Systems-level cancer gene identification from protein interaction network topology applied to melanogenesis-related functional genomics data.*, Journal of the Royal Society, Interface, vol. 7, no. 44, (2010), pp. 423–437.

[15] MILENKOVIĆ, T., NG, W. L., HAYES, W., PRŽULJ, N.: *Optimal network alignment with graphlet degree vectors.*, Cancer informatics, vol. 9, (2010), pp. 121–137.

[16] MILENKOVIĆ, T., PRŽULJ, N.: *Uncovering biological network function via graphlet degree signatures.*, Cancer informatics, vol. 6, (2008), pp. 257–273.

[17] MILENKOVIĆ, T., ZHAO, H., FAISAL, F. E.: *Global Network Alignment In The Context Of Aging*, in *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics - BCB'13*, ACM Press, New York, New York, USA, 2013, pp. 23–32.

[18] MUGAN, J., CHARI, R., HITT, L., MCDERMID, E., SOWELL, M., QU, Y., COFFMAN, T.: *Entity resolution using inferred relationships and behavior*, in *2014 IEEE International Conference on Big Data (Big Data)*, IEEE, 2014, pp. 555–560.

[19] NEŠETŘIL, J., POLJAK, S.: *On the complexity of the subgraph problem*, Commentationes Mathematicae Universitatis Carolinae, vol. 26, no. 2, (1985), pp. 415–419.

[20] O'NEIL, P. V.: *Ulam's Conjecture and Graph Reconstructions*, The American Mathematical Monthly, vol. 77, no. 1, (1970), pp. 35–43.

[21] ORTMANN, M., BRANDES, U.: *Quad census computation: Simple, efficient, and orbit-aware*, in *Lecture Notes in Computer Science*, vol. 9564, Springer International Publishing, 2016, pp. 1–13.

[22] PRŽULJ, N., CORNEIL, D. G., JURISICA, I.: *Modeling interactome: scale-free or geometric?*, Bioinformatics, vol. 20, no. 18, (2004), pp. 3508–3515.

[23] PUJOL, A., MOSCA, R., FARRÉS, J., ALOY, P.: *Unveiling the role of network and systems biology in drug discovery*, Trends in Pharmacological Sciences, vol. 31, no. 3, (2010), pp. 115–123.

[24] RIBEIRO, P., SILVA, F.: *G-tries: an efficient data structure for discovering network motifs*, in *Proceedings of the 2010 ACM Symposium on Applied Computing - SAC '10*, ACM Press, New York, New York, USA, 2010, pp. 1559–1566.

[25] SHEN-ORR, S. S., MILO, R., MANGAN, S., ALON, U.: *Network motifs in the transcriptional regulation network of Escherichia coli.*, Nature genetics, vol. 31, no. 1, (2002), pp. 64–8.

[26] STOICA, A., PRIEUR, C.: *Structure of Neighborhoods in a Large Social Network*, in *2009 International Conference on Computational Science and Engineering*, IEEE, 2009, pp. 26–33.

[27] WERNICKE, S.: *Efficient detection of network motifs.*, IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM, vol. 3, no. 4, (2006), pp. 347–59.

[28] ZHANG, L., HAN, Y., YANG, Y., SONG, M., YAN, S., TIAN, Q.: *Discovering discriminative graphlets for aerial image categories recognition.*, IEEE transactions on image processing, vol. 22, no. 12, (2013), pp. 5071–5084.

Current address

Hočevar Tomaž, PhD.
Faculty of Computer and Information Science
University of Ljubljana
Večna pot 113, 1000 Ljubljana, Slovenia
E-mail: tomaz.hocevar@fri.uni-lj.si