Applied Mathematics E-Notes, 21(2021), 615-621 © ISSN 1607-2510
Available free at mirror sites of http://www.math.nthu.edu.tw/∼amen/

Detecting Communities Under Constraints In Directed Acyclic Networks*

Suzana Antunović[†], Damir Vukičević[‡]

Received 19 August 2020

Abstract

Community detection is one of the fundamental problems in complex networks theory with applications in many different branches of science. Many available algorithms for community detection in directed acyclic networks do not include analysis of the resulting set of communities, and those that do, mostly focus on factors like the number of communities and community stability, not on relations between communities. In this paper, we present an algorithm that, given the topological ordering of a directed acyclic network, produces an optimal division (in terms of modularity) for that ordering which allows the establishment of an ordering on the resulting set of communities. The algorithm is based on recursively placing of the vertices into appropriate communities, thus respecting the order of the vertices, and resulting in division with optimal modularity.

1 Introduction

The study of networks, in the form of mathematical graph theory, is one of the fundamental pillars of discrete mathematics [12]. A special class of networks that occur widely in natural and man-made settings are directed acyclic networks [7]. The term refers to the finite directed graph that has no directed cycles. Equivalently, directed acyclic graph is a directed graph that has a topological ordering [2], a linear ordering of its vertices in such way that for every directed edge the starting vertex of the edge occurs earlier in the sequence than the ending vertex of the edge. Directed acyclic graphs have many applications in scheduling for systems of tasks with ordering constraints [23], may be used to represent a network of processing elements [21], Bayesian networks [22], family trees [9], citation graphs [20]. Within the field of complex networks, the problem of community detection has received wide attention. It relates to finding a natural division of the network into groups of vertices such that there are many edges within the community, and several (less) edges between communities [15]. Community detection has proved to be a problem of remarkable subtlety, computationally challenging and with deep connections to other areas of research [3, 5]. There have been many different approaches to solving this problem including hierarchical clustering [13], clique based methods [18], optimization techniques [10], edge-betweenness analysis [5] etc. During the last few decades the most popular community detection methods are based on maximizing modularity [10, 14, 16]. In recent years, the focus has shifted to community detection in directed acyclic networks. Taking the direction of edges into account produces more constraints on the process of community detection. Some of the methods for resolving the problem include generalizing the form of modularity for directed networks [17, 8, 24], extending the clique percolation method [19], layering [25], the game theoretical approach [6] and many others.

In this paper, we are interested is finding the optimal, in terms of modularity, division of the network into set of communities under the following condition. Let G be a directed acyclic network with n vertices and m directed edges and let it hold $x_1 \prec x_2 \prec ... \prec x_n$ (sign "$\prec$" denotes that vertex x_i comes before vertex x_j in the topological ordering).

[*] Mathematics Subject Classifications: 05C85, 68R10, 90C35, 05C20.

[†] Department of Mathematics, Faculty of Civil Engineering, Architecture and Geodesy, University of Split, Croatia

[‡] Department of Mathematics, Faculty of Science, University of Split, Croatia

We are interested in finding communities $C_1, C_2, ..., C_k$ in such way that it holds:

$$\text{if } x_i \prec x_j, x_i \in C_p \text{ and } x_j \in C_q, \text{ then } C_p \prec C_q \text{ or } C_p = C_q.$$

In other words, if all the vertices in community C_i appear earlier in the topological ordering of vertices than all the vertices in community C_j, the community C_i "appears" before community C_j in the community order. Consider, for instance, devising a curriculum for a certain course, or devising a new college major program. One has to take into account that every educational unit has its prerequisites and that the curriculum needs to be arranged in a certain order. This algorithm allows the educational units to be grouped in chapters or courses that can be taught consecutively. The algorithm can also be applied in any process that can be represented as a directed acyclic graph and divided into consecutive communities. It is based on recursively placing the vertices into appropriate community choosing the one with the highest modularity increase (if possible) and respecting the imposed condition on community order. The algorithm was developed for and tested on curriculum networks described in [1], where educational units are grouped into chapters to be taught consecutively.

2 Algorithm for Detecting Set of Communities

Every directed acyclic graph has at least one topological ordering of the vertices. For simplicity, we assume that the vertices are labeled $l_u \in \{1, ..., n\}$ where label l_u represents the position of the vertex u in the topological ordering. The measure used to evaluate the quality of community division is modurity defined as follows. For a directed network G with n vertices and m directed edges represented by an adjacency matrix $\mathbf{A}$, let $d^{in}(i)$ and $d^{out}(i)$ be in–degree and out–degree of a vertex $i \in V(G)$. Let vertex i belong to the community l_i. Modularity for directed networks is defined as [10]

$$Q_d = \frac{1}{m} \sum_{1 \leq i,j \leq n} \left[A_{ij} - \frac{d^{in}(j) d^{out}(i)}{m} \right] \delta(l_i, l_j) \tag{1}$$

where $\delta(l_i, l_j)$ is Kronecker's delta. Modularity measures the actual ratio of edges within the community reduced by the expected value in the null–model, where the division into communities is the same, but the edges between the vertices are placed randomly [11].

The algorithm works as follows. Initially, we consider each vertex to belong to a separate community. Starting from the last vertex in the ordering, for each vertex, we consider placing it into communities that have been obtained as the best solutions in the previous steps. We introduce the following notations: let r_k be the optimal solution obtained in step k of the algorithm (during the placing of the vertex k) and let z_{ij} be the community consisting of vertices $i, j \in V(G)$. The algorithm begins from the last vertex in the ordering (the one with the largest label n). It holds $r_n = z_n$. Moving on to the vertex labeled $l_u = n-1$, we consider the change in modularity obtained by combining it with the optimal solution from the previous step or by remaining in separate communities. Specifically, we consider the cases $z_{(n-1)n}$ and $z_{(n-1)} + r_n$ (the "+" sign indicates that there are two separate communities). In the next step, we consider vertex labeled $n-2$ and the modularity increase for cases $z_{(n-2)(n-1)n}$, $z_{(n-2)(n-1)} + r_n$ i $z_{(n-2)} + r_{(n-1)}$, where $r_{(n-1)}$ denotes the optimal solution obtained in the step $n-1$. In general, for placing the vertex labeled k we consider the following cases:

- $z_{k(k+1)...n}$
- $z_{k(k+1)...(n-1)} + r_n$
- $z_{k(k+1)...(n-2)} + r_{(n-1)}$

  $\vdots$
- $z_k + r_{(k+1)}$

The change in modularity ΔQ_d caused by placing two vertices into the same community can be calculated as follows. Let the vertex i change the existing label l_i to the new label l_j. The change in modularity caused by this change follows from the equation (1) and is calculated as

$$\Delta Q_d(ij) = \frac{d_i^j}{m} - \left[\frac{d^{out}(i)S_{in}(j) + d^{in}(i)S_{out}(j)}{m^2}\right] \tag{2}$$

where

- d_i^j is the total number of neighbours of i with label j,
- $S_{in}(j)$ is the total in–degree of vertices labeled j (the sum of all $d^{in}(u)$ such that $l_u = j$),
- $S_{out}(j)$ is the total out–degree of vertices labeled j (the sum of all $d^{out}(u)$ such that je $l_u = j$).

In each step, we choose the optimal solution that results in the largest non-negative change in modularity, i.e. largest $\Delta Q_d \geq 0$. The algorithm ends when all vertices are placed into the appropriate communities. The initial topological ordering can be given as an input to the algorithm or it can be obtained during the execution of the algorithm. The way in which vertices are placed ensures that the condition of a valid community order is met and that the solution obtained is optimal under the given condition since we look at the optimal solutions from previous cases at each step. Pseudo code of the algorithm with a given topological ordering is displayed in the Algorithm 1.

Algorithm 1 Algorithm for detecting communities under constraints

```
assign to each vertex $i$ a unique numerical label $l_i \in \{1, 2, ..., n\}$ indicating its place in the topological ordering of the vertices
set $r_n = z_n$
while there are vertices that have not been considered do
    for each vertex $k$ such that $l_k \in \{n-1, ..., 1\}$ do
        calculate $\Delta Q_d$ for each case $z_{k(k+1)...n}$, $z_{k(k+1)...(n-1)} + r_n$, ... , $z_k + r_{(k+1)}$
        assign $r_k$ to the case with the highest $\Delta Q_d \geq 0$
        add vertex $k$ to the appropriate community in accordance with the solution obtained
    end for
end while
```

Let us consider the complexity of the algorithm. To allocate the vertex k into the appropriate community, it is necessary to consider $n - k + 1$ cases. For each of these cases, it is necessary to calculate the modularity change that is calculated by the formula (2), which requires going through all the neighbours of the current vertex. Let us denote with $d_k = d^{in}(k) + d^{out}(k)$. In total, it takes $(n - k + 1)d_k$ operations to correctly place the vertex k. Summing up through all the vertices gives:

$$\begin{aligned}
&\sum_{k=1}^{n}(n-k+1)d_k = nd_1 + (n-1)d_2 + (n-2)d_3 + ... + 2d_{n-1} + d_n \\
&= (d_1 + d_2 + d_3 + ... + d_n) + (d_1 + d_2 + d_3 + ... + d_{(n-1)}) + ... + (d_1 + d_2) + d_1 \\
&\leq 2m + 2m + ... + 2m \leq 2mn.
\end{aligned}$$

From the foregoing considerations it follows that the total complexity of the algorithm is equal to $O(nm)$. The algorithm produces an optimal divison into a set od validly ordered communities for a given topological ordering of vertices. Let us consider in details how the algorithm works on an example of a very simple directed acyclic network shown on Figure 1.

A step-by-step demonstration is shown below. The optimal solution in terms of modularity increase is denoted by a red rectange. In this example, the Algorithm starts from the topological ordering $1 \prec 2 \prec 3 \prec 4 \prec 5 \prec 6$. Starting form the last vertex, the process goes as follows.

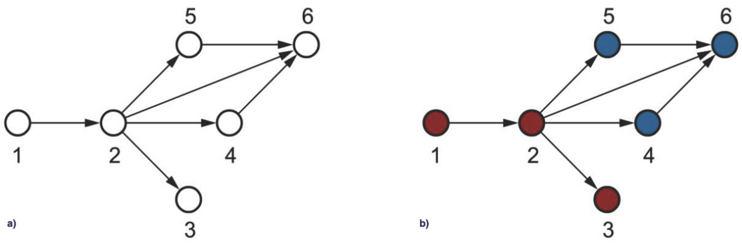

Figure 1: **The effect of the algorithm on a simple directed acyclic network.** A simple example of directed acyclic network with $n = 6$ vertices and $m = 7$ directed edges. **b)** Division into 2 consecutive communities obtained by the proposed algorithm. Vertices in community C_1 are denoted in red, vertices in community C_2 are denoted in blue.

$\boxed{i=6}$	$\boxed{z_6}$	$[6] \rightarrow r_6$
$\boxed{i=5}$	$\boxed{z_{56}}$	$[5\ 6] \rightarrow r_5$
	$z_5 + r_6$	$[5]\ [6]$
$\boxed{i=4}$	$\boxed{z_{456}}$	$[4\ 5\ 6] \rightarrow r_4$
	$z_{45} + r_6$	$[4\ 5][6]$
	$z_4 + r_5$	$[4]\ [5\ 6]$
$\boxed{i=3}$	z_{3456}	$[3\ 4\ 5\ 6]$
	$z_{345} + r_6$	$[3\ 4\ 5]\ [6]$
	$z_{34} + r_5$	$[3\ 4]\ [5\ 6]$
	$\boxed{z_3 + r_4}$	$[3]\ [4\ 5\ 6] \rightarrow r_3$
$\boxed{i=2}$	z_{23456}	$[2\ 3\ 4\ 5\ 6]$
	$z_{2345} + r_6$	$[2\ 3\ 4\ 5]\ [6]$
	$z_{234} + r_5$	$[2\ 3\ 4]\ [5\ 6]$
	$\boxed{z_{23} + r_4}$	$[2\ 3]\ [4\ 5\ 6] \rightarrow r_2$
	$z_2 + r_3$	$[2]\ [3]\ [4\ 5\ 6]$
$\boxed{i=1}$	z_{123456}	$[1\ 2\ 3\ 4\ 5\ 6]$
	$z_{12345} + r_6$	$[1\ 2\ 3\ 4\ 5]\ [6]$
	$z_{1234} + r_5$	$[1\ 2\ 3\ 4]\ [5\ 6]$
	$\boxed{z_{123} + r_4}$	$[1\ 2\ 3]\ [4\ 5\ 6] \rightarrow r_1$
	$z_{12} + r_3$	$[1\ 2]\ [3]\ [4\ 5\ 6]$
	$z_1 + r_2$	$[1]\ [2\ 3]\ [4\ 5\ 6]$

3 Experiments and Results

3.1 Data Sets

The algorithm was tested on curriculum networks, directed acyclic networks where vertices represent educational units, which are described in [1]. Directed edge from vertex u to vertex v means that unit u is a

Table 1: **Basic statistics for curriculum networks.** *Notation:* number of vertices n, number of directed edges m, largest in–degree d_{in}, largest out–degree d_{out}, average degree d_{avg}, average shortest path length l for pairs of connected vertices, clustering coefficient C.

Network	$\boldsymbol{n}$	$\boldsymbol{m}$	$\boldsymbol{d_{in}}$	$\boldsymbol{d_{out}}$	$\boldsymbol{d_{avg}}$	**l**	**C**
Number set $\mathbb{Q}$	47	254	17	26	5.404	2.011	0.254
Elementary functions	84	502	27	51	5.976	2.132	0.255
Integral	223	655	15	28	2.941	3.899	0.084
Physics	31	49	4	8	1.581	1.575	0.049
Primary production	28	93	9	14	3.321	2.135	0.183
Data processing	54	197	12	22	3.648	1.744	0.338

Table 2: **Comparison of the results obtained using the Algorithm with the results suggested by the experts who compiled the curriculum networks.** *Notation*: number of vertices n, number of directed edges m, number of communities N_c, value of modularity Q_d calculated for the proposed network division.

			Expert		*Algorithm*	
	n	m	Q_d	N_c	Q_d	N_c
Number set $\mathbb{Q}$	47	254	0.311	5	0.377	4
Elementary functions	84	502	0.239	6	0.286	8
Integral	223	655	0.455	10	0.484	10
Data processing	54	197	0.389	6	0.430	6
Primary production	28	93	0.237	3	0.259	3
Physics	31	49	0.238	6	0.375	4

prerequisite in learning and understanding unit v, i.e., should be studied before the unit v. In each network, every educational unit $u \in V$ has a unique numerical label $p(u) \in \{1, 2, ..., n\}$ indicating the order in which it is taught which corresponds to its place in the topological ordering. Community division in this context results in dividing educational units into chapters that can be learned or taught consecutively. The networks are named after the key concept whose understanding is set as a learning objective for that area. While tested on curriculum networks, the algorithm has been given a topological ordering as an input. Some basic statistics for curriculum networks can be found in Table 1. More details can be found in [1]. Measures used are defined in [15].

3.2 Results

The results given by the algorithm were compared to results given by the authors of each network. Precisely, the author of each of the networks gave a community division they think is the best way of arranging educational units into consecutive chapters. Results were given in Table 2. Community division given by the algorithm gives higher modularity scores for each of the proposed networks. Since the algorithm comes down to modularity optimization, it is worth mentioning the resolution limit problem introduced in [4]. Their results imply that modularity optimization algorithms might miss important substructures of a network by clustering smaller communities to form a larger community, although the problem is most likely to occur for communities with a number of internal links of the order of $\sqrt{2m}$ or smaller. The resolution limit of modularity does not depend on particular network structure, but results only from the comparison between the number of links of the interconnected communities and the total number of links of the network [4]. As suggested in paper [4], we constrained the modularity optimization to each single community obtained for every network used and confirmed that the resolution limit did not have a significant impact on the community division.

4 Discussion

Since the algorithm was applied to a particular type of networks, it was important to interpret the results in logical and methodical sense. An analysis of both divisions leads to the conclusion that the algorithm provides meaningful divisions into communities. The divisions obtained by the algorithm have logical interpretations and valuable tips can be drawn from the results that could help the assemblers to better compile and distribute the material. Experts who have created the networks consider that the divisions obtained are meaningful and valid.

The algorithm does not require a community size specification, although the tendency is to divide the network into approximately equal smaller communities. Since the testing was carried out on very small networks, the resolution limit did not have a strong effect on the results. Nevertheless, one should be careful when using the algorithm on larger networks. The algorithm can be modified by enforcing the community size and number requirement depending on the type of network that is being analyzed and the various needs of different researchers.

5 Conclusion

In this paper we present an algorithm for detecting communites that need to be arranged in a certain order. The algorithm gives great results in terms of modularity, but it also satisfies logical and methodological requirements of the network. It can be used in different situations and settings, ranging from computer science, physics, mathematics, biology, etc. Thus, in addition to increasing modularity, the algorithm provides logically meaningful divisions.

The algorithm gives a division regarding a specific topological ordering of the vertices. Since there can be many different orderings of a network, the future work may include the stability analysis of the obtained communities regarding different valid orderings of vertices, testing the algorithm on different types of networks and considering the impact of the resolution limit on communities obtained for larger networks.

References

[1] S. Antunović and D. Vukičević, Detecting communities in directed acyclic networks using modified LPA algorithms, Proceedings of the 2nd Croatian Combinatorial Days, (2019), 1–14.

[2] J. Bang–Jensen, Digraphs: Theory, Algorithms and Applications, vol. 5, Springer-Verlag, 2008.

[3] S. Fortunato, Community detection in graphs, Phys. Rep., 486(2010), 75–174.

[4] S. Fortunato and M. Barthélemy, Resolution limit in community detection, PNAS, 104(2007), 36–41.

[5] M. Girvan and M. E. J. Newman, Community structure in social and biological networks, Proc. Natl. Acad. Sci., 99(2002), 7821–7826.

[6] A. Jonnalagadda and L. Kuppusamy, Mining Communities in Directed Networks: A Game Theoretic Approach. In: Abraham A., Muhuri P., Muda A., Gandhi N. (eds) Intelligent Systems Design and Applications. ISDA 2017. Advances in Intelligent Systems and Computing, vol 736. Springer, Cham., 2018.

[7] B. Karrer and M. E. J. Newman, Random graph model for directed acyclic networks, Phys. Rev. E, 80(2009).

[8] Y. Kim S.-W. Son and H. Jeong, Finding communities in directed networks, Phys. Rev. E, 81(2010), 016103.

[9] B. B. Kirkpatrick, Haplotypes versus genotypes on pedigrees, Algorithms for Molecular Biology, 6(2011).

[10] E. A. Leicht and M. E. J. Newman, Community structure in directed networks, Phys. Rev. Lett, 100(2008).

[11] X. Liu and T. Murata, Advanced modularity-specialized label proagation algotirhm fordetecting communities in networks, Physica A., 389(2010), 1493–1500.

[12] M. E. J. Newman, The structure and function of complex networks, SIAM Review, 45(2003), 167–256.

[13] M. E. J. Newman, Detecting community structure in networks, Eur. Phys. J. B., 38(2004), 321–330.

[14] M. E. J. Newman, Modularity and community structure in networks, Proceedings of the National Academy of Sciences, 103(2006),8577–8582.

[15] M. E. J. Newman, Networks: An Introduction, Oxford Univ. Press, Oxford, 2010.

[16] M. E. J. Newman and M. Girvan, Finding and evaluating community structure in networks, Phys. Rev. E., 69(2004).

[17] V. Nicosia, G. Mangioni, V. Carchiolo and M. Malgeri, Extending the definition of modularity to directed graphs with overlapping communities, J.Stat. Mech, 9(2009).

[18] G. Palla, I. Derényi, I. Farkas and T. Vicsek, Uncovering the overlapping community structure of complex networks in nature and society, Nature, 435(2005), 814–818.

[19] G. Palla, I. J. Farkas, P. Pollner, I. Derényi and T. Vicsek, Directed network modules, New J. Phys. 9, 186(2007).

[20] D. J. de S. Price, Networks of Scientific Papers, Science, 149(1965), 510–515.

[21] S. Sapatnekar, Timing, 1^{st} ed., Springer, Boston, 2004.

[22] I. Shmulevich and E. R. Dougherty, Probabilistic Boolean Networks: The Modeling and Control of Gene Regulatory Networks, SIAM, 2010.

[23] S. S. Skiena, The Algorithm Design Manual, 2^{nd} ed., Springer-Verlag, London, 2009.

[24] L. Speidel, T. Takaguchi and N. Masada, Community detection in directed acyclic graphs, European Physical Journal B, 88(2015).

[25] V. Vasiliauskaite and T. S. Evans, Making communities show respect for order, Appl. Netw. Sci. 5, 15(2020).