NETWORK COMMUNITY DETECTION BASED ON IMPROVING VERTEX COORDINATES

Lai Van Trung*, Nguyen Thi Thanh Giang
University of Information and Communication Technology, Thai Nguyen University, Vietnam

ARTICLE INFORMATION

Journal: *Vinh University Journal of Science Natural Sciences, Engineering and Technology*
p-ISSN*: 3030-4563*
e-ISSN: *3030-4180*

Volume: 53
Issue: 2A
***Correspondence:** lvtrung@ictu.edu.vn*

Received: *05 February 2024*
Accepted: *17 April 2024*
Published: *20 June 2024*

Citation:
Lai Van Trung, Nguyen Thi Thanh Giang (2024). Network community detection based on improving vertex coordinates. ***Vinh Uni. J. Sci.*** *Vol. 53 (2A), pp. 81-88 doi: 10.56824/vujs.2024a016a*

ABSTRACT

In recent years, with the strong development of information technology, detecting communities in large real networks is a very important issue which is of interest to many scientists. Community detection in large real networks with millions of nodes is often difficult. To solve this problem, many online community search algorithms have been proposed with many different approaches. One of the approaches is to coordinate the vertices of the graph and build a reasonable distance between those vertices. It has been observed that vertices in the same community have approximately the same probability of reaching other vertices through a random walk. Based on this principle, the authors propose a way to coordinate vertices and build distances between vertices in the graph that reduces computational complexity compared to existing techniques. This approach involves representing peaks as vectors and using the K-means++ algorithm for community detection, whose effectiveness is evaluated through experimental results presented.

Keywords: Community detection; random walk; coordinates; distance; modularity.

1. Introduction

In recent times, network analysis has attracted the attention of many scientists due to its wide application in many different fields such as social research [1], biology [2-3] and computer networks [4]. Mathematically, a network is represented as a graph consisting of vertices connected by edges. Real-life networks are often large and sparse, with hundreds of thousands or even millions of vertices, and they often have a community structure. These communities can be thought of as groups of vertices that are tightly interconnected internally but more loosely connected to the rest of the network. Exploring these community structures has become a fascinating area of research because they provide valuable insights into the original network. Defining a community in a graph is mathematically complex. There have been many different approaches proposed by scientists, specifically as follows: The approach towards coordinateizing vertices and constructing the distance between the vertices of the graph based on the

idea that two vertices belonging to the same community will have a small distance betwween them [5-6]; The approach is to maximize the number of edges within a community and minimize the number of edges between communities [7-9]; The approach using linear algebra [10-11]; The approach uses clustering coefficients with the idea of two certain people knowing each other, there is a high probability that those two people will have friends in common [12-13]; The approach towards building probabilistic models [14].

In this paper, based on the idea of distance developed by Latapy and Pons [6], we introduce a new coordinateization method and construct the distance between vertices in the graph. This approach significantly reduces computational complexity compared to previously established methods [6]. Based on the random walk, each vertex in the graph is presented as an h-dimensional vector and the K-means++ algorithm is used for community detection. Furthermore, some experiments are conducted using random graph generation models and real-world datasets to investigate the effectiveness of the proposed algorithm.

2. Latapy and Pons distance

Consider an undirected graph $G = (V, E)$, where $V = \{v_1, v_2, \dots, v_n\}$ is the set of vertices and and E is the set of edges of the graph, $A = (A_{ij})_{n\times n}$ is the adjacency matrix of the graph G, where $A_{ij} = 1$ if between two vertices i và j there is a connecting edge; $A_{ij} = 0$ if there is no connecting edge connection between those two vertices. The degree of vertex i, denoted by d_i, is defined as the number of vertices adjacent to vertex i. For simplicity, in this paper unweighted graphs is consider. However, it is possible to develop to weighted graphs by considering $A_{ij} \in \mathbb{R}^+$ instead of $A_{ij} \in \{0,1\}$. Consider a random walk on graph G [15, 16], which is a random process that describes a path consisting of random walks $X_0, X_1, \dots X_t, \dots$ on G. Each step is moving from a certain vertex to its adjacent vertices randomly with equal probability. The probability of moving from vertex v_i to v_j is $P_{ij} = \frac{A_{ij}}{d_i}$, so the transition probability matrix after one step is $P = (P_{ij})_{n\times n}$ and $P^t = (P_{ij}^t)_{n\times n}$ is the transition probability matrix after a step of a random walk on G. Denote D as the degree matrix of vertices in the graph, that is $D_{ij} = d_i$ if $i \neq j$ và $D_{ij} = 0$ if $i = j$, then $P = D^{-1}A$.

Consider an acyclic random walk on a finite graph G with n vertices. According to the convergence theorem of Markov chains [16], we have a transition probability matrix P satisfying $\lim_{k\to\infty} P^k = P_\infty$, with $(P_\infty)_{ij} = \emptyset_j$, where $\emptyset_j$ is the j-component of the stationary distribution $\emptyset = (\emptyset_1, \emptyset_2, \dots, \emptyset_n)$. If G is an undirected graph then $a_{ij} = a_{ji}$, we have $\emptyset_i = \frac{d_i}{2m}$, $\forall\, i = 1,2,\dots,n$ with $2m = \sum_{i,j=1}^{n} A_{ij}$. In [6], Latapy and Pons found that, if two vertices i and j belong to the same community, the probabilities of going randomly from vertex i and j to arbitrary vertex k are approximately the same, that is $P_{ik}^t \simeq P_{jk}^t$, for every vertex k. From there, the distance between i and j is defined as follows:

$$r_{ij} = \sqrt{\sum_{k=1}^{n} \frac{(P_{ik}^t - P_{jk}^t)^2}{d_k}} = \left\| D^{-\frac{1}{2}} P_{i\bullet}^t - D^{-\frac{1}{2}} P_{j\bullet}^t \right\|, \quad (1)$$

where $\|\bullet\|$ is the inner Euclidean norm $\mathbb{R}^n$ and $P_{i\bullet}^t$ is the second row i of the matrix P^t. From the above distance formula, each vertex i can be coordinated by a vector $D^{-\frac{1}{2}} P_{i\bullet}^t$ or

$$Coord(i) = D^{-\frac{1}{2}} P_{i\bullet}^t = \left\{ \frac{P_{i1}^t}{d_1^{1/2}}, \frac{P_{i2}^t}{d_2^{1/2}}, \dots, \frac{P_{in}^t}{d_n^{1/2}} \right\}. \quad (2)$$

Based on the above idea, we propose a new way to coordinate vertices and define the distance between vertices of the graph presented in the following section.

3. Recommended distance

Definition: With k being any vertex, the distance between two vertices i and j is determined by the following formula:

$$\Re_{ij} = \sqrt{((P_{ki}^t - \emptyset_i) - (P_{kj}^t - \emptyset_j))^2} \quad (3)$$

Thus, each vertex i in the graph can be considered as a one-dimensional vector with coordinates as follows:

$$Coord(i) = (P_{ki}^t - \emptyset_i). \quad (4)$$

To improve the accuracy of the distance proposed by formula (3), any h components are randomly selected within $P_{i\bullet}^t$ and coordinate each vertex by h dimensional vector:

$$Coord(i) = (P_{ki_1}^t - \emptyset_{i_1}, P_{ki_2}^t - \emptyset_{i_2}, \dots, P_{ki_h}^t - \emptyset_{i_h}). \quad (5)$$

Next, the relationship of the distance formula (3) with the spectrum of the transition probability matrix P will be considered, thereby proving that when two vertices i and j belong to the same community, the distance between them will be small.

Lemma *([6, Lemma 1])* *The eigenvectors of the matrix P are real and satisfy:*

$$1 = \lambda_1 > \lambda_2 \geq \cdots \geq \lambda_n \geq -1.$$

Furthermore, there exists a family of orthogonal vector $(S_\alpha)_{1 \leq \alpha \leq n}$ *so that each vector* $v_\alpha = D^{-\frac{1}{2}} S_\alpha$ *and* $u_\alpha = D^{\frac{1}{2}} S_\alpha$ *is a right eigenvector and a left eigenvector corresponding to the eigenvalue* λ_α *and satisfy:*

$$\forall \alpha, P v_\alpha = \lambda_\alpha v_\alpha, P^T u_\alpha = \lambda_\alpha u_\alpha \text{ and } \forall \alpha, \forall \beta, v_\alpha^T u_\beta = \delta_{\alpha,\beta}.$$

From Lemma 2 above, the following Theorem is proven:

Theorem: *Distance* $\Re$ *is related to the spectrum of matrix P by:*

$$\Re_{ij}^2 = \left(\sum_{\alpha=2}^{n} \lambda_\alpha^t v_\alpha(k)(u_\alpha(i) - u_\alpha(j))\right)^2 \quad (6)$$

in which $(\lambda_\alpha)_{1\le\alpha\le n}$ *and* $(v_\alpha)_{1\le\alpha\le n}$ *are the eigenvalues and right eigenvectors of the matrix* P*, respectively.*

Prove: From Lemma 2, the spectrum of the matrix $\boldsymbol{P}$ is analyzed as follows:

$$P = \sum_{\alpha=1}^{n} \lambda_\alpha v_\alpha u_\alpha^T \text{ and } P^t = \sum_{\alpha=1}^{n} \lambda_\alpha^t\, v_\alpha u_\alpha^T$$

By normalization, for each vertex k we have $v_1(k) = \frac{1}{\sqrt{\sum_{l=1}^{n} d(l)}}$; for each vertex i we also have $u_1(i) = \frac{d(i)}{\sqrt{\sum_{l=1}^{n} d(l)}}$.

Therefore we have $v_1(k)u_1(i) = \frac{d(i)}{\sum_{l=1}^{n} d(l)} = \phi_i$.

So

$$\begin{aligned} P_{ki}^{i} &= \sum_{\alpha=1}^{n} \lambda_\alpha^t v_\alpha(k)u_\alpha(i) = v_1(k)u_1(i) + \sum_{\alpha=2}^{n} \lambda_\alpha^t v_\alpha(k)u_\alpha(i) \\ &= \phi_i + \sum_{\alpha=2}^{n} \lambda_\alpha^t v_\alpha(k)u_\alpha(i) \end{aligned} \tag{7}$$

Substituting (7) into formula (3) we get $\Re_{ij}^2 = (\sum_{\alpha=2}^{n} \lambda_\alpha^t v_\alpha(k)u_\alpha(i) - \sum_{\alpha=2}^{n} \lambda_\alpha^t v_\alpha(k)u_\alpha(j))^2$

Or $\Re_{ij}^2 = (\sum_{\alpha=2}^{n} \lambda_\alpha^t v_\alpha(k)(u_\alpha(i) - u_\alpha(j)))^2$. So (6) has been proven. In the following, the proposed distance-based network community search algorithm will be presented.

Input data: Undirected graph $G = (V, E)$; k is the number of clusters.

Output: k community clusters have been partitioned.

Step 1: Calculate the matrix P^t (with t sufficiently large)

Step 2: Coordinate the vertices of the graph $Coord(i) = (P_{ki_1}^t - \emptyset_{i_1}, P_{ki_2}^t - \emptyset_{i_2}, \dots, P_{ki_h}^t - \emptyset_{i_h})$.

Step 3: Use K-means++ algorithm for community detection.

The proposed coordinateization method significantly reduces the computational complexity compared to the coordinateization method of Latapy and Pons. Specifically, according to the coordinateization method of Latapy and Pons, the distance must be calculated based on all components of vector $P_{i\bullet}^t$ and the calculation time for each of these vectors is $O(tm)$. To calculate all vectors $P_{i\bullet}^t$, $(i = 1,2,\dots,n)$, the computation time will becomes $O(tmn)$. Meanwhile, our proposed distance calculation only needs to consider h components of the vector$P_{i\bullet}^t$, so the computational complexity is $O(tmh)$, with h no larger than the number of communities in the network under consideration.

To evaluate the effectiveness of the proposed distance-based algorithm, the proposed coordinateization of vertices from formula (5) and the coordinateization according to Pons and Latapy by formula (2) have been used in the experiments, then K-means++ algorithm is used to detect the communities. The experiments are performed on

real Email network data and randomly generated graph data. The Modularity function is then used to evaluate the quality of the generated clusters.

4. Experimental results

Modularity function is used to evaluate the clustering quality of network community search algorithms. Whichever clustering algorithm has a greater modularity value will have better clustering quality. The Modularity function on undirected graphs is defined in [17] as follows: $Q = \frac{1}{2m}\sum_{i=1,j=1}^{n}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta(C_i, C_j)$, in which A_{ij} is the element located on the intersection of row i and column j of the adjacency matrix A; k_i, k_j is the degree of i and j; C_i, C_j are the communities containing i and j respectively and $\delta(C_i, C_j) = \begin{cases} 1 \; when \; C_i = C_j, \\ 0 \; when \; C_i \neq C_j \end{cases}$

In the first experiment, the real Email network dataset, introduced in [18], was used. This is a social network constructed from email communications within a medium-sized university with about 1700 employees. Email networks provide an accurate and non-intrusive description of the flow of information within human organizations. The algorithm with the proposed coordinateization method and the coordinateization method by Latapy and Pons were used. The results obtained are shown in Figure 1.

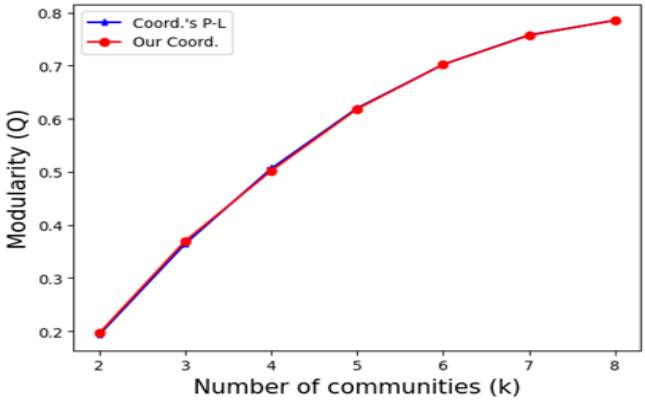

Figure 1: *Comparison of Modularity between the proposed coordinate algorithm and coordinateization by Latapy and Pons on real data Email network*

Besides testing with real data sets, another experiment on the dataset which is a random generating graph established using a Gaussian random partition generator [19] was conducted. The number of peaks is randomly selected within the range (1000, 2000), and for each cluster, the average number of peaks is randomly selected within (80, 120). The parameter is set to 0.7. Three different values of 0.01, 0.05, and 0.1 have been tested, corresponding to graphs with clear community structure, moderately clear community structure, and unclear community structure. The results are illustrated in Figure 2.

From the conducted experiments, it is clear that the coordinate method proposed in this study proves to be equally effective as the Latapy and Pons coordinate method in a variety of situations, and the computational complexity is significantly reduced.

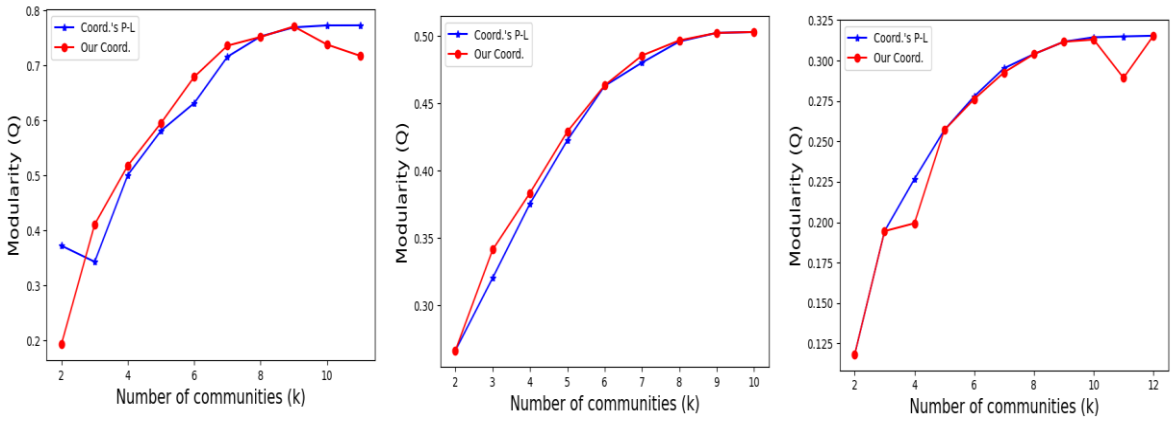

Figure 2: *Comparison of Modularity between the proposed coordinate algorithm and coordinateization by Latapy and Pons on randomly generated graph data*

5. Conclusion

This article has presented the use of random walks on graphs, proposed a new coordinate method and constructed the distance between vertices in the graph. The proposed coordinateization method has significantly reduced the computational complexity compared to the coordinateization method of Latapy and Pons, and at the same time the effectiveness is equivalent. This is also an important result for continuing research on finding network communities on large real-world networks.

REFERENCES

[1] MC. Gonzalez, HJ. Herrmann, J. Kertesz and T. Vicsek. "Community structure and ethnic preferences in school friendship networks," *Physical A 379*, 307-316, 2007. DOI: 10.1016/j.physa.2007.01.002

[2] J. F. Rural *et al.*, "Towards a proteome-scale map of the human protein-protein interaction network", *Nature* 437,1173, 2005. DOI: 10.1038/nature04209

[3] U. Stelzl *et al.*, "A Human ProteinProtein Interaction Network," *A Resource for Annotating the Proteome*, Cell 122: 957-968, 2005. DOI: 10.1016/j.cell.2005.08.029

[4] C. Moore, "The Computer Science and Physics of Community Detection," *Landscapes, Phase Transitions, and Hardness*, Bull. EATCS 121, 2017.

[5] B. MacQueen, "Some Methods for classification and Analysis of Multi- variate Observations," In *Proceedings of 5th Berkeley Symposium on Math- ematical Statistics and Probability*, vol. 1. University of California Press. pp. 281-297 ,1967.

[6] P. Pons and M. Latapy, "Computing communities in large networks using random walks," *Journal of Graph Algorithms and Applications*, volume 10, no. 2, 2006, pp 191-218, 2006. DOI: 10.7155/jgaa.00124

[7] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte and Etienne Lefebvre1, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics, Theory and Experiment*, 2008. DOI: 10.1088/1742-5468/2008/10/P10008

[8] Raghavan, U. N., Albert, R., and Kumara S, "Nearlinear time algorithm to detect community structures in large-scale networks," *Physical Review E*, 76(3) :036106, 2007. DOI: 10.1103/PhysRevE.76.036106

[9] W. Li, J. Wang and J. Cai, "New label propagation algorithms based on the law of universal gravitation for community detection," *Physica A*, 2023.

[10] M. E. J. Newman, "Spectral methods for network community detection and graph partitioning," *Phys. Rev. E*, vol. 88, p. 042822, October 2013. Rev. Lett. 100 (11), 118703, 2008. DOI: 10.1103/PhysRevE.88.042822

[11] Zhe Wang, Yingbin Liang and Pengsheng Ji, "Spectral Algorithms for Community Detection in Directed Networks," *Journal of Machine Learning Research 21,* 1-45, 2020.

[12] D. J. Watts and Steven Strogatz, "Collective dynamics of 'small-world' networks," *Nature*. 393 (6684): 440-442, 1998. DOI: 10.1038/30918

[13] Thibaud Trolliet *et al.*, "Interest clustering coefficient: a new metric for directed networks like Twitter," *Journal of Complex Networks*, 1-30, 2021. DOI: 10.1093/comnet/cnab030

[14] Hossein Hajibabaei, Vahid Seydi and Abbas Koochari, "Community detection in weighted networks using probabilistic generative model," *Journal of Intelligent Information Systems*, vol. 60, pp 119-136 , 2023. DOI: 10.1007/s10844-022-00740-6

[15] L. Lov´asz, "Random walks on graphs: a survey", In *Bolyai society mathematical studies*, Gergely Ambrus, Spinger, 1996.

[16] M. S. Aldenderfer and R. K. Blashfleld, "Cluster Analysis," *Sage University Paper Series on Quantitative Applications in the Social Sciences,* Sage, Beverly Hills, 07-0441984.

[17] M. E. J. Newman, "Modularity and community structure in networks," In *Proceedings of the National Academy of Sciences of the United States of America*, 103 (23), 8577-8696, 2006. DOI: 10.1073/pnas.0601602103

[18] R. Guimer`a, L. Danon, A. D´ıaz-Guilera, F. Giralt, and A. Arenas, "Self-similar community structure in a network of human interactions," *Phys. Rev. E*, 68(6):065103, 2003. DOI: 10.1103/PhysRevE.68.065103

[19] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, 75-174, 2010. DOI: 10.1016/j.physrep.2009.11.002

TÓM TẮT

TÌM KIẾM CỘNG ĐỒNG MẠNG DỰA TRÊN CẢI TIẾN TOẠ ĐỘ CỦA ĐỈNH

Lại Văn Trung, Nguyễn Thị Thanh Giang
Trường Đại học Công nghệ thông tin và truyền thông, Đại học Thái Nguyên, Việt Nam
Ngày nhận bài 05/02/2024, ngày nhận đăng 17/4/2024

Trong những năm gần đây, với sự phát triển mạnh mẽ của công nghệ thông tin, việc phát hiện cộng đồng trong mạng thực lớn là vấn đề rất quan trọng, được nhiều nhà khoa học quan tâm nghiên cứu. Việc phát hiện cộng đồng trong các mạng thực lớn với hàng triệu nút thường khó khăn. Để giải quyết vấn đề này, nhiều thuật toán tìm kiếm cộng đồng mạng đã được đề xuất với nhiều cách tiếp cận khác nhau. Một trong những cách tiếp cận là tọa độ các đỉnh của đồ thị và xây dựng khoảng cách hợp lý giữa các đỉnh đó. Chúng tôi quan sát thấy rằng các đỉnh trong cùng một cộng đồng có xác suất đi đến các đỉnh khác là gần như nhau thông qua bước đi ngẫu nhiên. Dựa trên nguyên tắc này, chúng tôi đề xuất một cách toạ độ hoá các đỉnh và xây dựng khoảng cách giữa các đỉnh trong đồ thị làm giảm độ phức tạp tính toán so với các kỹ thuật hiện có. Cách tiếp cận này liên quan đến việc biểu diễn các đỉnh dưới dạng vectơ và sử dụng thuật toán K-means++ để phát hiện cộng đồng, được đánh giá tính hiệu quả qua một số kết quả thực nghiệm được trình bày.

Từ khóa: Phát hiện cộng đồng; bước đi ngẫu nhiên; toạ độ; khoảng cách; modularity.