INFERRING COMMUNITY STRUCTURE THROUGH MAXIMUM DEGREE-BASED RANDOM WALK WITH RESTART

Hui Xie

School of Information and Communication Engineering
Guangzhou Maritime University
Guangzhou 510725, China

Yongjie Yan[†]

School of Information Engineering, Fujian Business University
Fuzhou 350012, China
yongjieyan@aliyun.com

Received 22 September 2023, accepted 20 February 2024,
published online 27 February 2024

Community structure, a critical topological property of complex networks, has recently received extensive and in-depth attention from researchers. Recognizing the non-uniform degree distribution of nodes within network subgraphs, this paper presents a novel algorithm called MD-RWR (Maximum Degree-based Random Walk with Restart) for community detection in complex networks. The proposed algorithm not only excels at identifying overlapping communities but also enhances the objectivity and accuracy of the results. To evaluate its performance, the algorithm is tested on five real-world networks. The experimental results demonstrate its effectiveness in detecting communities, particularly when dealing with overlapping ones. Furthermore, the algorithm surpasses Walktrap, Infomap, LPA, and LPA-S algorithms in terms of modularity and NMI scores, while exhibiting faster execution time compared to these algorithms.

DOI:10.5506/APhysPolB.55.2-A1

1. Introduction

Many real-world systems in nature, society, or technology are complex networks, which exhibit inherent community structures, where nodes within a group display stronger connections compared to interactions between different groups. A classic example of a complex network is a social network, such as Facebook or Twitter (now also known as X). In these networks, individuals (nodes) are connected by various types of relationships (edges),

[†] Corresponding author.

such as friendships, followings, or interactions. Here is a brief explanation of some basic notions: (1) Nodes: In a social network, nodes represent individual users. Each user is a node in the network. (2) Edges: The connections between nodes represent interactions or relationships. For example, on Facebook, a friendship between two users would be represented by an edge connecting their respective nodes. (3) Community structure: In the context of social networks, community structure refers to the presence of tightly-knit groups of individuals within the network. These groups may represent clusters of friends with strong connections or shared interests. The formation of community structures arises from the interplay among graphs and their internal dynamics, such as resetting on networks with centrality-based criteria [1]. Consequently, the exploration of community detection in networks has emerged as a prominent research domain in recent decades [2–4].

In recent years, numerous community detection methods have been proposed, leveraging random walk and node similarity principles. These approaches employ Markov models to assess node similarity by executing random transitions and assigning nodes with high similarity to the same community [5, 6]. However, in real networks, the compatibility between quality indicators for different types of community structures significantly varies within the same community. This challenges the adaptability of community detection algorithms relying on quality indicators [7]. Conversely, algorithms based on random walk techniques are less influenced by community type, thus exhibiting superior adaptability. Nevertheless, node similarity evaluation through random walk-based community detection algorithms is highly sensitive to the number of iterations in the walking process, frequently necessitating prior knowledge for decision-making [8].

Supervised methods for community detection primarily exploit network structure. For example, Liben-Nowell and Kleinberg [9] conducted experiments on a co-authorship network and identified the Adamic/Adar method as the most effective. Kashima *et al.* [10] proposed a semi-supervised method utilizing link propagation for prediction. Additionally, employing the configuration model as a null model to evaluate network partition quality has proven to be a reliable approach [11].

The Label Propagation Algorithm (LPA), originally introduced by Raghavan *et al.* [12], assigns a node with the most frequent community label within its neighborhood. LPA exhibits nearly linear time complexity and is widely employed for community detection in large-scale complex networks. However, due to various random factors during the label propagation process (*e.g.*, node update order and label selection), the outcomes of community detection display significant instability. In other words, executing the LPA algorithm multiple times on the same network can yield considerably different results. To address this issue, Li *et al.* [13] proposed the LPA-S algorithm, which updates a node's label based on the label of its most similar neighboring node, thereby reducing randomness and enhancing algorithm stability to some extent.

In previous studies, random walk models have been utilized to determine the relevance of nodes in social networks [14, 15]. Understanding the dynamics of communities is crucial as it provides valuable insights into the relationships between nodes in complex networks. However, the identification of underlying communities in a network remains a challenging task [16]. To address this challenge, some approaches have adopted relational learning methods [17, 18]. For instance, Backstrom and Leskovec [19] employed supervised random walk on networks to learn weights on links between community members.

Random walk, based on a Markov model, is a widely used research approach for community detection. The fundamental idea is to release numerous random walkers with an initial distribution and track their distribution function as they diffuse through the network. Extensive research has led to the proposal of several community detection algorithms based on random walk. One such algorithm is the Walktrap algorithm introduced by Pons *et al.* [20], which clusters nodes hierarchically based on the notion that nodes within a community exhibit stronger connections compared to those outside the community. Consequently, a randomly chosen walker is more likely to remain within a community for a longer duration, resulting in a clear hierarchical structure [21]. Although the Walktrap algorithm may not achieve high accuracy, its underlying ideas have significantly influenced subsequent algorithms. Zhang *et al.* [22] proposed a three-stage hierarchical community detection algorithm, PMAC, based on Partial Matrix Approximation Convergence. This algorithm identifies the initial core nodes in the network using a node importance measurement method. The number of random walk steps is then determined based on the convergence of the partial transition matrix of core nodes. Communities are merged around these core nodes, resulting in the final partitioning. Lai *et al.* [23] transformed directed networks into undirected networks by assigning weights to edges based on edge direction information, successfully applying algorithms designed for undirected networks to discover communities in the original directed network. The SCMAG algorithm proposed by Huang *et al.* [24] constructs communities based on node attributes, demonstrating the close connection between node attributes and community structure. The Infomap algorithm is an information theory-based approach that combines random walk ideas to detect community structures [25]. By searching for the shortest encoding of random walk paths, it identifies community structures. Jiao *et al.* [26] comprehensively considered global and local topology structures and proposed a method of calculating node similarity based on this concept, which exhibits stronger adaptability to different types of communities compared to previous algorithms.

These studies propose community detection algorithms that integrate random walk with classical methods and perform well on specific network models. However, most of these algorithms utilize undifferentiated transition probabilities during the node transition step, which fails to fully capture the asymmetry of node degrees in real networks. Additionally, the accuracy of community partitioning is greatly influenced by the number of transition iterations, and more prior knowledge is necessary for decision-making. Given the excellent properties of the node degree metric in simulating stochastic processes through restarted random walks, it can be enhanced and appropriately utilized to evaluate the performance of community detection.

The rest of this paper is organized as follows. Section 2 presents the preliminaries and problem definition. The proposed MD-RWR (Maximum Degree-based Random Walk with Restart) algorithm is described in Section 3. Experimental results are presented in Section 4, followed by the conclusion in Section 5.

2. Preliminaries

In this paper, we propose a clustering method for large undirected graphs based on random walk. The proposed method is founded on the concept that, before a random walk reaches the stationary distribution, the random walker spends more time traversing within clusters than transitioning between clusters. All symbols used throughout this paper are gathered in Table 1.

Table 1. List of symbols.

Symbol	Explanation
G	an undirected graph
V	the nodes of G
E	the edges of G
n	node number, $n = \|V\|$
m	edge number, $m = \|E\|$
A	adjacency matrix
A_{ij}	1 if node i and node j are directly connected, 0 otherwise
$N(i)$	neighborhood of i, namely $N(i) = \{j \mid A_{ij} = 1\}$
$d(i)$	degree of node i, $d(i) = \sum_j A_{ij}$

2.1. RWR on graphs

The basic principle entails traversing the entire network graph, commencing from a specific node or a series of nodes. For each node, the walker will transition to its neighboring nodes with a probability of $1 - \alpha$. Simultaneously, there exists a probability of α for the walker to randomly transition to a neighboring node within the network graph. Here, α denotes the probability of random jumping. Every completed walk yields a probability distribution, indicating the likelihood of accessing each node in the graph. The acquired probability distribution serves as input for subsequent walks, and this iterative process continues until convergence is achieved. In this section, we introduce a novel method for ranking nodes in the network based on their relevance scores, which exhibits similarities to the well-known PageRank [27].

There are two assumptions about undirected graphs:

1. Every node in the graph is connected to itself.

2. The graph is connected, meaning that any node in the graph can be reached from any other node.

The presence of a robust community structure within a network graph implies a dense inter-connectivity among nodes within the community, while exhibiting limited connections with external nodes. Consequently, when the random walker reaches the boundary of a community, the probability of returning to the internal structure of the community surpasses that of venturing outside the community. This observation substantiates two key facts: firstly, the relatively high likelihood of traversing between nodes within the community structure, and secondly, the comparatively low probability of traversing between nodes belonging to distinct communities. Leveraging this principle, we can iteratively conduct independent random walks in complex networks to unveil communities.

The fundamental approach for detecting communities in intricate social networks via repeated random walks entails performing the random walk process multiple times. Each random walk consists of a specified number of steps, and the recorded path at each step represents a set of nodes, potentially indicative of a community based on preliminary analyses. Consequently, by repeating the random walk process t times, t potential communities can be identified. It is important to acknowledge that the quality of these potential communities is influenced by the initial nodes and step length, and there exists a considerable likelihood of overlapping communities. Therefore, further refinement of these potential communities is necessary.

Two methodologies can be employed to terminate the random walk process. The first approach involves determining the fixed number of steps taken during the walk, while the second technique entails establishing a cumulative jump probability threshold. Within a random walk path, the cumulative jump probability is computed as the product of the jump probability from the initial node to each subsequent step leading to the endpoint of the path. Considering the substantial computational complexity of complex networks and the relatively relaxed accuracy requirements during the potential community identification stage, this paper adopts the strategy of setting a predetermined number of steps to conclude the random walk process.

2.2. Distance between nodes and distance between clusters

The probability of starting at node i and transitioning to node j in a random walk of length t is denoted as $P_{i,j}^t$. An appropriate value of t may be chosen, such that it is sufficiently large to capture graph structure information while not being excessively large, thereby allowing $P_{i,j}^t$ to converge to the stationary distribution. This approach enables the establishment of a concept of distance between any two nodes in the graph based on $P_{i,j}^t$. As a result, several desirable properties of this approach emerge:

1. When two nodes, denoted as i and j, belong to the same cluster, the probability $P_{i,j}^t$ should be relatively high.
2. The value of $P_{i,j}^t$ is influenced by the degree of $d(j)$, as the walker is more likely to transition to nodes with higher degrees.
3. If a random walk starts from either node i or node j within the same cluster, it should end in a node k with a similar probability. In other words, for every node k, the probabilities $P_{i,k}^t$ and $P_{j,k}^t$ should be similar.

Consider two clusters, denoted as C_1 and C_2, which are subsets of the node set $C_1, C_2 \subset V$ within the graph G. To establish the distance r_{C_1,C_2} between these clusters, we define

$$r_{C_1,C_2} = \sqrt{\sum_{k=1}^{n} \frac{\left(P_{C_1,k}^t - P_{C_2,k}^t\right)^2}{d(k)}}\,. \tag{1}$$

The probability $P_{C,k}^t$, which denotes the likelihood of initiating a random walk of t steps within cluster C and terminating at node k is

$$P_{C,k}^t = \frac{1}{|C|} \sum_{i \in C} P_{i,k}^t\,. \tag{2}$$

3. Implementation of the MD-RWR algorithm

Walktrap is a clustering algorithm that excels at identifying communities within graphs that contain dense inter-connectivity among specific groups of nodes compared to the rest of the network. Walktrap can recognize such communities even when they are not explicitly labeled or separated in the graph, but like other clustering algorithms, it is vulnerable to noise and outliers, and requires prior specification of the desired number of clusters. Additionally, the time complexity of the Walktrap algorithm is $O(n^3)$, which limits its use for large networks.

In contrast, our proposed MD-RWR algorithm is a graph clustering technique that employs RWR iteratively on the given graph and merges the resulting walks into clusters based on walking on the local maximum degree node. The core concept of MD-RWR is that nearby nodes are likely to be visited by similar random walks. The algorithm begins by constructing a transition matrix for the graph, representing the probability of transitioning from one node to another. Next, it performs random walks on the graph by taking random steps based on the restart probabilities. After generating several walks, the algorithm utilizes a clustering approach to group nodes that are frequently visited together by the same walks.

The fundamental principle of MD-RWR is to estimate the similarity between nodes based on the number of random walks traversing through them. Nodes that are frequently visited together are more likely to belong to the same cluster than those visited together less frequently. To enhance the efficiency of the algorithm, we incorporate the local maximum degree node method by conducting RWR only on the neighboring nodes of the walker, reducing the algorithm's time complexity.

We can define RWR on a graph using the transition matrix P

$$P_{i,j}^{(t+1)} = \alpha \frac{A_{i,j}^{(t)}}{d(i)} + (1-\alpha)q\,, \tag{3}$$

where the degree of node i, denoted as $d(i) = \Sigma_j A_{i,j}$, represents the sum of weights associated with its connections. The restart probability is denoted as α, and q is a row vector $(0,\ldots,1,\ldots,0)$. q is typically used to represent the probability of restarting (or relocating) the current node to a specific node in a complex network. During the random walk, there is a certain probability (denoted by α) of returning to the initial node, and this probability is implemented through q. It should be emphasized that the initial transition matrix P_0 is defined as q. The RWR is a stochastic process in which, at time t, the random walker is located at node i and moves to node j

with probability $P_{i,j}$. Since $P_{i,j}$ is identical for all neighboring states j of state i, the walker moves to a neighbor (or remains in the current state) with equal probability.

3.1. Steps of the algorithm

Step 1. Choosing two clusters to merge.

The distance metric described above can be utilized in a hierarchical clustering algorithm. We begin by defining the initial partition of the graph into n clusters, with each node representing an individual cluster: $P1 = v \mid v \in V$. The algorithm iteratively merges clusters until only one remains, by following these steps for each stage k:

1. Identify the two neighboring clusters C_1 and C_2 that result in the smallest variation $\Delta\sigma(C_1, C_2)$, which will be elaborated in the following section.

2. Combine C_1 and C_2 to form a new cluster, C_3.

3. Update the variations $\Delta\sigma(C_3, C)$ between C_3 and the clusters C that are adjacent to C_3.

4. Create a new partition $\mathcal{P}_{k+1} = (\mathcal{P} \setminus C_1, C_2) \cup C_3$.

By considering the probability vectors $P^t_{C_1,\bullet}$ and $P^t_{C_2,\bullet}$, we can calculate $\Delta\sigma(C_1, C_2)$ with a time complexity of $\mathcal{O}(n)$, as it has been demonstrated that a relationship exists between $\Delta\sigma(C_1, C_2)$ and r_{C_1,C_2} given by

$$\Delta\sigma(C_1, C_2) = \frac{1}{n} \frac{|C_1||C_2|}{|C_1| + |C_2|} r^2_{C_1,C_2} \,. \tag{4}$$

Step 2. Merging the clusters.

This step can be easily performed, as the newly formed cluster C_3 comprises the nodes from both C_1 and C_2

$$C_3 = C_1 \cup C_2 \,. \tag{5}$$

The updated probability vector $P^t_{C_3,\bullet}$ can be computed by utilizing $P^t_{C_1,\bullet}$ and $P^t_{C_2,\bullet}$

$$P^t_{C_3,\bullet} = \frac{C_1 P^t_{C_1,\bullet} + |C_2| P^t_{C_2,\bullet}}{|C_1| + |C_2|} \,. \tag{6}$$

Following that, we can derive the distances between nodes i and j, as well as between communities C_1 and C_2.

Step 3. Updating the distances.

Subsequently, it is necessary to compute the updated variation $\Delta\sigma(C_3, C)$ for each cluster C adjacent to C_3. Two distinct cases need to be considered:

1. If C is adjacent to both C_1 and C_2, the values of $\Delta\sigma(C_1, C)$ and $\Delta\sigma(C_2, C)$ are readily available. In this case, it is feasible to calculate $\Delta\sigma(C_3, C)$ in constant time complexity $\mathcal{O}(1)$ using the following formula:
$$\Delta\sigma(C_3, C) = \frac{(|C_1| + |C|)\Delta\sigma(C_1, C) + (|C_2| + |C|)\Delta\sigma(C_2, C) + |C|\Delta\sigma(C_1, C_2)}{|C_1| + |C_2| + |C|} . \tag{7}$$

2. In instances where C is adjacent to only one of C_1 and C_2, it becomes necessary to calculate $\Delta\sigma(C_3, C)$ using Eq. (8), which can be expressed as
$$\Delta\sigma(C_3, C) = \frac{1}{n}\frac{|C_3||C|}{|C_3| + |C|} r^2_{C_3,C} . \tag{8}$$

It is important to note that C must be adjacent to at least one of C_1 and C_2 since C is now adjacent to C_3, thus ensuring that these two cases encompass all possible scenarios.

Algorithm 1 presents a comprehensive depiction of the MD-RWR algorithm for community detection, providing a detailed description.

Based on the comprehensive analysis conducted above, it has been determined that after n-1n-1n-1 iterations, the system exhibits a significant increase in efficiency, thereby suggesting that the algorithm converges rapidly towards its optimal state. In Step 1 of the MD-RWR, the time complexity is $O(n)$ for selecting the mixing length t, $O(n)$ for calculating P_t, and $O(n \log n)$ for calculating distances from every node to communities. In Step 2, time complexity is $O(|C_1| + |C_2| + d)$, where $O(|C_1|)$ and $O(|C_2|)$ represent the sizes of clusters C_1 and C_2, respectively, where d is the dimensionality of the probability vector. In Step 3, the time complexity for calculating the number of nodes in each community is $O(n \log n)$, while the complexity for calculating the distances between communities is $O(n \log n)$. Additionally, the time complexity for calculating Eq. (8) is also $O(n \log n)$. Thus, the total time complexity of the MD-RWR algorithm is $O(n \log n)$.

Algorithm 1 MD-RWR algorithm.

```
Require: Graph: $G(V, E)$, Start Node: $u$, Restart : $\alpha$.
Ensure: Clustering result vector $C$.
for $i = 1, 2, ..., n$ do
    initialize $C_i$ with $i$;//Every node i is a community.
end for
for $i = 1, 2, ..., n-1$ do
    for $C_1 = 1, 2, ..., m$ do
        for $C_2 = 1, 2, ..., m$ do
            Calculate distance between community $C_1$ and $C_2$ using
            Eq. (1);//$C_1, C_2$ cannot be the same number.
            Calculate the value of $\Delta\sigma$ using Eq. (4);
            $H(C_1)(C_2) \leftarrow \Delta\sigma$;
        end for
    end for
    Find the minimum value in $H$ and Min $\leftarrow H(C_1)(C_2)$;
    for $j$ in $C_1$ and $C_2$ do
        Calculate the value of RWR using Eq. (3);
        Calculate the updated probability vector using Eq. (6);
        $C_3 = C_1 + C_2$;
    end for
    Update vector $C$;
    Update the distances between $C_3$ and other communities using Eq. (7)
    or (8);
end for
```

4. Simulations and analysis

4.1. Evaluation metrics

4.1.1. Modularity function

The modularity Q_c defined by Newman [33] reads

$$Q_c = \frac{1}{2L} \sum_{c \in C} \sum_{uv} \alpha_{uc} \alpha_{vc} \left(A_{uv} - \frac{k_u k_v}{2L} \right), \tag{9}$$

where the term A denotes the adjacent matrix, while $L = \sum_{uv} A_{uv}$ represents the aggregate weight of edges within the network. Furthermore, the sum of edge weights adjacent to node u is denoted as $k_u = \sum_v A_{uv}$. The belonging coefficient α_{uc}, originally introduced by Nicosia *et al.* [34] in 2009, quantifies the degree of affiliation between node u and community c. Its calculation is outlined

$$\alpha_{uc} = \frac{n_{uc}}{\sum\limits_{c \in C} n_{uc}}\,, \tag{10}$$

where n_{uc} represents the count of edges that are adjacent to node u within community c.

4.1.2. Normalized mutual information

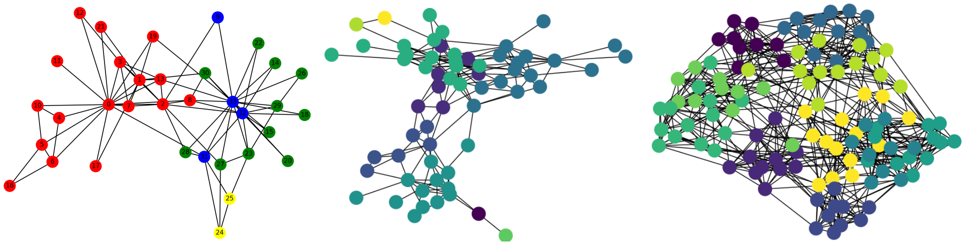
The assessment of network quality entails the utilization of the Normalized Mutual Information (NMI), an extensively employed metric for evaluating the effectiveness of graph clustering algorithms [35]. The NMI is employed as a quantitative measure to compare and contrast clustering outcomes

$$\mathrm{NMI}(X,Y) = \frac{2\sum_{i=1}^{C_A}\sum_{j=1}^{C_B} \frac{n_{ij}}{n}\log\left(\frac{n_{ij}n}{n_i n_j}\right)}{\left(-\sum_{k=1}^{C_A}\frac{n_k}{n}\log\left(\frac{n_k}{n}\right)\right)+\left(-\sum_{k=1}^{C_B}\frac{n_k}{n}\log\left(\frac{n_k}{n}\right)\right)}\,. \tag{11}$$

The symbol C_A is used to denote the standard community partition, while C_B represents the community partition obtained through a specific algorithm. An NMI index value of 1 signifies that the algorithm's community partition perfectly aligns with the standard one. Conversely, if the algorithm's community results are completely contradictory to the standard partition (*e.g.*, all nodes assigned to a single community), the NMI index will be 0.

4.2. Test on real-world networks

The experimental results for the networks including Zachary's Karate Club [28] with 34 nodes and 78 edges, the Dolphin network [29] with 62 nodes and 159 edges, and the Football network [30] with 115 nodes and 613 edges, are illustrated in Fig. 1.



(a) (b) (c)

Fig. 1. Community detection result of three networks. (a) Zachary's Karate Club network; (b) Dolphin network; (c) Football network.

The community detection results obtained through the employment of the MD-RWR algorithm in the networks of Zachary's Karate Club, the Dolphin network, and the Football network are visualized in Fig. 1 (a), Fig. 1 (b), and Fig. 1 (c), correspondingly. The modularity values for the three networks are 0.361, 0.382, and 0.418, respectively. Similarly, the NMI values for the three networks are 0.732, 0.685, and 0.832, respectively.

Based on the results reported in Table 2, it is apparent that Walktrap, Infomap, LPA, LPA-S, and MD-RWR demonstrate consistently high modularity values, indicating their robust performance on the five real-world networks. Nevertheless, in the case of the Enron email network, the modularity values obtained from LPA-S exceed those obtained from MD-RWR. This implies that the MD-RWR algorithm outperforms the other four algorithms, consistently exhibiting higher modularity across the majority of networks.

Table 2. Comparing with other algorithms.

Network	Nodes	Edges	Modularity Q_c				
			Walktrap [20]	Infomap [25]	LPA [12]	LPA-S [13]	MD-RWR
Karate network [28]	34	78	0.308	0.381	0.398	0.413	0.431
Dolphin network [29]	62	159	0.351	0.403	0.412	0.387	0.428
American college football [30]	115	613	0.381	0.395	0.425	0.415	0.453
Enron email [31]	1133	5451	0.402	0.486	0.516	0.536	0.492
PolBlogs [32]	1490	19090	0.482	0.521	0.572	0.586	0.593

The results of five different community detection algorithms (MD-RWR, Walktrap, Infomap, LPA, and LPA-S) as a function of random walk length are shown in Fig. 2, as a line plot where the x-axis represents the random walk length (ranging from 10 to 50) and the y-axis represents modularity.

Here are some conclusions that can be drawn from Fig. 2:

(1) Convergence of algorithms: As the random walk length increases, some algorithms show convergence in their results. This is indicated by the lines becoming flatter or less steep as the random walk length increases. For example, the MD-RWR algorithm's results increase sharply at first but then flatten out, indicating convergence. LPA-S also seems to converge, but at a higher level.

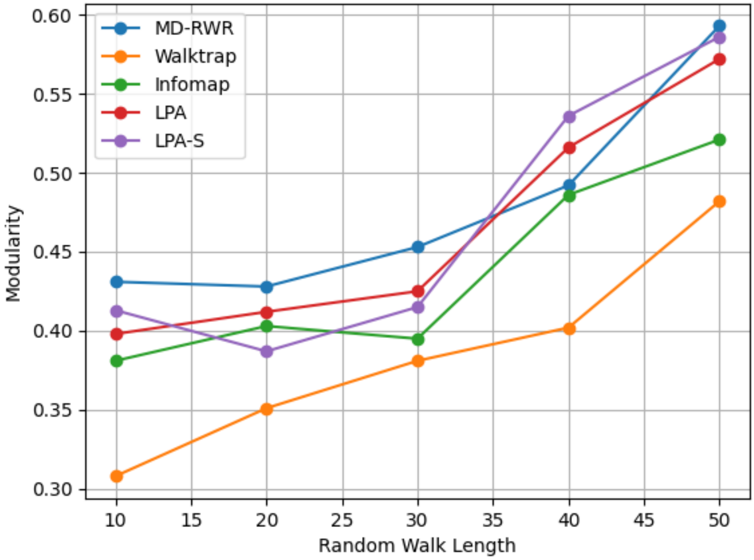


Fig. 2. The convergence of modularity as a function of the random walk length.

(2) Performance differences: Different algorithms perform better or worse depending on the random walk length. For example, MD-RWR performs better than the other algorithms at longer random walk lengths (around 40 and above), with the highest results overall. LPA-S also performs well, especially at longer random walk lengths. Walktrap and Infomap perform similarly, with Walktrap generally outperforming Infomap. LPA seems to lag behind the other algorithms, with consistently lower results.

(3) Optimal random walk length: There may be an optimal random walk length for each algorithm, depending on the evaluation metric used. For MD-RWR, the optimal random walk length may be around 40 or higher. LPA-S also performs well at these longer lengths. Walktrap and Infomap might have an optimal length around 30 to 40, while LPA might not benefit much from increasing the random walk length.

(4) Algorithm stability: Some algorithms, such as MD-RWR and LPA-S, seem more stable (*i.e.*, less sensitive to changes in random walk length) than others. LPA, on the other hand, seems to be more sensitive to changes in random walk length, with more pronounced fluctuations in its results.

5. Conclusion

This paper presents a novel community detection algorithm called MD-RWR, which utilizes the notion of local maximum degree nodes to effectively partition communities in various complex networks. The MD-RWR algorithm assigns transition probabilities to individual nodes and evaluates their importance within communities based on the local maximum degree. In the clustering process, nodes autonomously determine their transition direction using relevant information such as transition probabilities, eliminating the need to predefine the number of communities. Given the widespread existence of overlapping communities in large-scale networks, exploring techniques for assessing node significance in the context of overlapping communities presents a promising avenue for future research.

REFERENCES

[1] K. Zelenkovski *et al.*, «Random Walks on Networks with Centrality-Based Stochastic Resetting», *Entropy* **25**, 293 (2023).

[2] H. Zhou, «Distance, dissimilarity index, and network community structure», *Phys. Rev. E* **67**, 061901 (2003).

[3] S. Fortunato, D. Hric, «Community detection in networks: A user guide», *Phys. Rep.* **659**, 1 (2016).

[4] E. Massaro, F. Bagnoli, «Hierarchical Community Structure in Complex (Social) Networks», *Acta Phys. Pol. B Proc. Suppl.* **7**, 379 (2014).

[5] V. Tejedor, O. Bénichou, R. Voituriez, «Global mean first-passage times of random walks on complex networks», *Phys. Rev. E* **80**, 065104 (2009).

[6] Z. Zhang *et al.*, «Mean first-passage time for random walks on undirected networks», *Eur. Phys. J. B* **84**, 691 (2011).

[7] M. Bilgin, «Novel random models of entity mobility models and performance analysis ofrandom entity mobility models», *Turk. J. Electr. Eng. Comput. Sci.* **28**, 708 (2020).

[8] P. De Meo, E. Ferrara, G. Fiumara, A. Provetti, «Mixing local and global information for community detection in large networks», *J. Comput. Syst. Sci.* **80**, 72 (2014).

[9] D. Liben-Nowell, J. Kleinberg, «The link prediction problem for social networks», *CIKM'03: Proceedings of the* 12th *International Conference on Information and Knowledge Management*, ACM, New York 2003, pp. 556–559.

[10] H. Kashima *et al.*, «Link propagation: A fast semi-supervised learning algorithm for link prediction», *Proceedings of the 2009 SIAM International Conference on Data Mining*, 2009, pp. 1100–1111.

[11] M. Girvan, M.E.J. Newman, «Community structure in social and biological networks», *Proc. Natl. Acad. Sci. USA* **99**, 7821 (2002).

[12] U.N. Raghavan, R. Albert, S. Kumara, «Near linear time algorithm to detect community structures in large-scale networks», *Phys. Rev. E* **76**, 036106 (2007).

[13] W. Li, C. Huang, M. Wang, X. Chen, «Stepping community detection algorithm based on label propagation and similarity», *Physica A* **472**, 145 (2017).

[14] H.H. Tong, C. Faloutsos, J.Y. Pan, «Fast Random Walk with Restart and Its Applications», *Sixth International Conference on Data Mining (ICDM'06)*, Hong Kong, China 2006, pp. 613–622.

[15] P. Sarkar, A.W. Moore, «Fast dynamic reranking in large graphs», *WWW'09: Proceedings of the* 18th *International Conference on World Wide Web*, 2009, pp. 31–40.

[16] M.E.J. Newman, M. Girvan, «Finding and evaluating community structure in networks», *Phys. Rev. E* **69**, 026113 (2004).

[17] B. Taskar, M.F. Wong, P. Abbeel, D. Koller, «Link prediction in relational data», *Proceedings of the* 17th *Annual Conference (NIPS 2003)*, Cambridge, MA: MIT Press 2004, pp. 659–666.

[18] A. Popescul, L.H. Ungar, «Statistical relational learning for link prediction», *Proceedings of the Workshop on Learning Statistical Models from Relational Data at IJCAI'2003*, Acapulco, Mexico 2003, pp. 109–115.

[19] L. Backstrom, J. Leskovec, «Supervised random walks: predicting and recommending links in social networks», *WSDM'11: Proceedings of the* 4th *ACM International Conference on Web Search and Data Mining*, 2011, pp. 635–644.

[20] P. Pons, M. Latapy, «Computing Communities in Large Networks Using Random Walks», *J. Graph Algo. Appl.* **10**, 191 (2006).

[21] S. Fortunato, «Community detection in graphs», *Phys. Rep.* **486**, 75 (2010).

[22] W. Zhang *et al.*, «Hierarchical community detection based on partial matrix convergence using random walks», *Tsinghua Sci. Technol.* **23**, 35 (2018).

[23] D. Lai, H. Lu, C. Nardini, «Finding communities in directed networks by PageRank random walk induced network embedding», *Physica A* **389**, 2443 (2010).

[24] X. Huang, H. Cheng, J.X. Yu, «Dense community detection in multi-valued attributed networks», *Inform. Sci.* **314**, 77 (2015).

[25] M. Rosvall, C.T. Bergstrom, «Maps of random walks on complex networks reveal community structure», *Proc. Natl. Acad. Sci. USA* **105**, 1118 (2008).

[26] Q.J. Jiao, Y. Huang, H.B. Shen, «Community mining with new node similarity by incorporating both global and local topological knowledge in a constrained random walk», *Physica A* **424**, 363 (2015).

[27] M. Bianchini, M. Gori, F. Scarselli, «Inside PageRank», *ACM Trans. Internet Technol.* **5**, 92 (2005).

[28] W.W. Zachary, «An Information Flow Model for Conflict and Fission in Small Groups», *J. Anthropol. Res.* **33**, 452 (1977).

[29] D. Lusseau *et al.*, «The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations», *Behav. Ecol. Sociobiol.* **54**, 396 (2004).

[30] L.A. Adamic, E. Adar, «Friends and neighbors on the Web», *Soc. Networks* **25**, 211 (2003).

[31] M.E.J. Newman, «The Structure and Function of Complex Networks», *SIAM Rev.* **45**, 167 (2003).

[32] L.A. Adamic, N. Glance, «The political blogosphere and the 2004 U.S. election: divided they blog», *LinkKDD'05: Proceedings of the* 3rd *International Workshop on Link Discovery*, 2005, pp. 36–43.

[33] M.E.J. Newman, M. Girvan, «Finding and evaluating community structure in networks», *Phys. Rev. E* **69**, 026113 (2004).

[34] V. Nicosia, G. Mangioni, V. Carchiolo, M. Malgeri, «Extending the definition of modularity to directed graphs with overlapping communities», *J. Stat. Mech.: Theory Exp.* **2009**, P03024 (2009).

[35] L. Danon, A. Díaz-Guilera, J. Duch, A. Arenas, «Comparing community structure identification», *J. Stat. Mech.: Theory Exp..* **2005**, P09008 (2005).