Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **9**, Number **2**, May **2024**

Node and Graph Similarity Based on Personalized Random Walks with Restart Process

Hao-Lai Li

EFORT Intelligent Equipment Co., Ltd.,
Shanghai, 201600, P. R. China
lihaolai@efort.com.cn

Xiao Chen

School of Aeronautics and Astronautics
Zhejiang University
Hangzhou, 310027, China
nampl@zju.edu.cn

Zhe-Ming Lu*

Center for Generic Aerospace Technology
Huanjiang Lab
Zhuji, 311816, China
zheminglu@zju.edu.cn

Jia-Lin Cui

School of Information Science and Engineering
NingboTech University
Ningbo, 315100, P. R. China
cuijl_jx@163.com

*Corresponding author: Zhe-Ming Lu

Received June 3 2023, revised September 17, 2023, accepted December 28, 2023.

Abstract. *Identifying and quantifying node or graph similarity of different graphs is a challenging important task for practical use. Current methods of node comparison are only able to extract very limited local information or are computationally very demanding. In this paper, we present an efficient and intuitive measure for node comparison in different networks based on the Personalized Random Walks with Restart (RWR) process. We find that it can effectively and efficiently quantify topological differences of a node in different networks and is sensitive to changes of mesoscopic structure, such as merge/split of clusters or remove of critical links that connect or disconnect connected components. We then extend this method to measure graph similarity and show its superiority over the state-of-the-art method.*
Keywords: Node similarity, Graph similarity, Random walks.

1. **Introduction.** Using a network representation is a good way to model the relationship among a group of entities. In a network, nodes denote entities, while edges represent some relationship between each pair of entities. In many applications, there are multiple methods for defining edges, which can be collectively analyzed to gain a more comprehensive understanding of the data. These data and corresponding networks have been proven

to be useful in many situations, such as comparing genetics and protein protein interactions in cells, understanding potential relationships and community structures between social networks, and analyzing time networks [1]. Considering the inherent diversity of cross domain network data and the latest theoretical developments in processing these types of data, it is necessary to develop appropriate tools that can utilize information from all network layers.

A problem that comes up often is the following: how much do nodes of different networks differ in terms of topological structure and how to calculate the similarity value between two networks? Measure this similarity is an important task making sense: study the relationship of neurons in chemical and electric networks; exceptional changes in the network traffic may result in a computer attack. In addition, network similarity can provide insight into behavioral patterns: is Facebook's message graph similar to Facebook's wall to wall graph? Tracking the changes of networks over time, discovering anomalies, and detecting events is a highly focused research direction [2].

There are two main kinds of node similarity in a single network: local structure based and global structure based. The similarity measurement based on local structure uses local structure to determine the similarity score between two nodes. The similarity score calculated through this type of measurement is usually proportional to the number of mutual neighbors between two target nodes, and can be written as $S_{ij} = (|N(i) \cap N(j)|)/C$, where $N(i)$ is the set of neighbors of node i, $|\cdot|$ returns the number of elements of a set, and C is a normalizing term whose value is determined by the specified similarity measure. For example, by Jaccard similarity the value of C is $|N(i) \cup N(j)|$, by cosine similarity the value of C is $|N(i)||N(j)|$, by topology overlapping the value of C is $\min(|N(i)|, |N(j)|)$. See Ref. [3] for detail. The measurement based on global structure considers the structure of the entire network to determine the similarity score between node pairs. Katz similarity [4] is based on the total number of paths between two nodes, where longer paths are assigned lower weights.

Another way to look at the similarity of nodes in a network is to measure the approximation between nodes. PageRank, Personalized Random Walks with Restarts 0, SimRank 0, and Belief Propagation 0 are some of the most successful techniques. The SimRank algorithm defines the similarity score between two nodes in a recursive manner: two objects are similar if they are referenced by similar objects. These schemes have been successfully adopted in a lot of tasks including ranking, classification, information retrieval [18], malware and fraud detection [19], and recommendation systems [20]. However, this kind of approximation, which is a proxy to measure the tendency of link formation, is usually asymmetric, not normalized and inconvenient for comparison. There are also some other ideas to consider the similarity problem [21-23].

To the best of our knowledge, there is still not direct approach to measure the similarity or proximity of nodes in different networks. The main difficulty lies in the lack of explicit connections among networks, which disables most of the similarity measures used in a single network. A more or less similar problem is anomalous node detection in dynamic networks. The major task is to detect events in time when nodes deviate from their normal behavior, or to identify the particular parts of the network that are responsible for the change. The main idea behind these methods is extracting and comparing various features of the local structure of a node, such as in-degree, out-degree, number of neighbors, or eigenvector, between two layers. According to different definitions of the local structure, some methods compare the direct neighbors of the node 0, and others compare the local cluster 0. Unfortunately, this problem is difficult to be transformed to ours because most methods only concern which parts of a network induce anomalies and do not give similarity scores.

Another problem similar to ours is the measure of graph similarity with known node correspondence. This kind of problems mainly concern how much two graphs or networks differ in terms of connectivity and some can be considered measuring nodal similarity first and then averaging them to get graph similarity score. One of the most remarkable methods is DELTACON algorithm 0 which computes the pairwise node affinities in a single network and treats the difference of nodal affinities of two networks as the graph similarity. We compare our method with DELTACON and show results in Section 4.

In this paper, we focus on two main issues: How to measure the similarity of a node among different networks? How to measure the graph similarity? Our main contributions are:

(1) Node similarity measure algorithm: we propose a novel method for measuring critical structural differences of a node between two networks, and show that it is: (a) sensitive to important topological structural changes; (b) intuitive, giving similarity scores that agree with common sense and can be easily explained.

(2) Graph similarity measure algorithm: based on node similarity, we propose a graph similarity measure and conduct experiments on extensive synthetic networks to show that it achieves many good properties and overcomes some critical disadvantages of the state-of-the-art method.

The rest of the paper is organized as follows. Section 2 explains our proposed method in detail. Section 3 discusses the experimental result of our proposed method. Finally, Section 4 concludes the whole paper.

2. **Proposed Scheme.**

2.1. **Desired Properties.**

How can we evaluate the similarity in connectivity between two nodes in various graphs, or more formally how can we tackle the following problem?

Given: (a) two graphs, $G_1(V; E_1)$ and $G_2(V; E_2)$ with the same node set V, and different edge sets E_1 and E_2, (b) the node correspondence, and (c) a node $v_i^{(1)}$ in G_1 and its counterpart $v_i^{(2)}$ in G_2.

Find: a similarity score $0 \leq sim(v_i^{(1)}, v_i^{(2)}) \leq 1$ between the two nodes. Similarity score of value 0 means that two nodes have totally different local structures, while 1 means identical local structures.

Then we want the similarity measure to obey the following properties:

A1. Identity property: $sim(v_i^{(1)}, v_i^{(2)}) = 1$, when $G_1 = G_2$.

A2. Symmetric property: $sim(v_i^{(1)}, v_i^{(2)}) = sim(v_i^{(2)}, v_i^{(1)})$.

A3. Null property: $sim(v_i^{(1)}, v_i^{(2)}) = 0$ when G_1 and G_2 do not have any overlapped edges.

P1. [**Distance Awareness 1**] The farther a change happens from a node, the less impact it has on the node.

P2. [**Distance Awareness 2**] The farther we move a node from its original position, the less similarity the node has to its initial state.

P3. [**Cluster Awareness**] Structural changes such as clusters split around should be much more importance than changes maintaining the connectivity or affiliation to clusters.

2.2. **Local Region Definition and Local Structural Feature Extraction.**

Measuring the proximity or similarity between nodes in a graph becomes the base for various data mining tasks. Among many methods computing nodal proximity, random walk with restart (RWR) is very popular for its good intuition and ability to consider the global network structure. RWR measures proximity between a given seed node s and the other nodes. It assumes a random walker walking to its neighbors by the given rule. The walker

first starts at a seed node s and then at each walk step either restarts at s with probability c or moves to a neighboring node along an edge with probability $1 - c$. The probability of the next edge that the walker passes through is proportional to its weight. Finally, the stationary probability of the walker being at all the nodes of a graph corresponds to the RWR score and is denoted by a vector $\mathbf{r}$. The $i-$th entry of $\mathbf{r}$ corresponds to RWR score of the node i. The vector $\mathbf{r}$ satisfies the following equations:

$$\mathbf{r} = (1 - c)(\mathbf{D}^{-1}\mathbf{A})^{\mathrm{T}}\mathbf{r} + c\mathbf{q} \tag{1}$$

or more compact form:

$$[\mathbf{I} - (1 - c)(\mathbf{A}\mathbf{D}^{-1})]\mathbf{r} = c\mathbf{q} \tag{2}$$

Where $\mathbf{D}$ is the diagonal matrix of nodal degrees and $\mathbf{q}$ is the starting vector in which the index of the seed node is set to 1 and others to 0. In the rest of the work, we always let $c = 0.15$, following the convention of PageRank algorithm.

The reasons we choose RWR to measure nodal similarity are: (a) it satisfies all the desired properties A1-A3 and P1-P3, (b) the RWR score, indicating the stationary probability distribution of a random walker's position, is a naturally normalized feature, which makes it very suitable to be a similarity measure and (c) unlike most methods considering only the 1- or 2-step neighbors as the local region, RWR takes advantage of the whole structure information of a graph. It is convenient to assume that RWR defines a "soft" local region and meanwhile gives features reflecting or summarizing the topological structure that we are interested in.

2.3. **Similarity Definition.** After we have obtained the RWR score of a node in two networks, the next step is to compute the similarity between features. Because the RWR score is essentially a probability distribution, we use the Hellinger distance [14] which quantifies the distance between two probability distributions in statistics to measure the similarity between two nodes. For two discrete probability distributions P and Q, their Hellinger distance is defined as

$$H(P, Q) = \frac{\sqrt{2}}{2} \parallel \sqrt{P} - \sqrt{Q} \parallel_2 \tag{3}$$

As can be seen, the distance of two identical probability distributions is zero and the Hellinger distance $H(P, Q)$ satisfies the desired property,

$$0 \leq H(P, Q) \leq 1 \tag{4}$$

We then defined the similarity of two graphs as $1 - H(P, Q)$. It is notable that a random walker restarts its trip at the seed node with probability c in the original configuration of RWR algorithm, which always makes the stationary probability, i.e. the RWR score, of the seed node larger than c. As a result, two graphs without any overlapped edge may have Hellinger distance less than 1, namely, non-zero similarity score. For example, if the stationary probability of two graphs without any overlapped edge are $S_1 = [0.3, 0.3, 0.4, 0, 0]$ and $S_2 = [0.2, 0, 0, 0.4, 0.4]$ respectively and the first element of S_1 and S_2 are stationary probability of the shared seed node, we then find the Hellinger distance $H(S1, S2) \approx 0.9$. This means the similarity of the two not overlapped graphs is about 0.1. To avoid this counterintuitive case, we normalize the RWR score vector $\mathbf{r}$ to be $\mathbf{R}$ by the following equation:

$$\mathbf{R} = \frac{\mathbf{r} - [0, ..., r_{seed}, ...]}{1 - r_{seed}} \tag{5}$$

Formally, our algorithm is described as follows:

Algorithm of RWR Node Similarity Computation
INPUT: two networks $G_1(V; E_1), G_2(V; E_2)$ and a target node v.
Calculate the RWR score vector of the node v in G_1 and G_2: r_1 and r_2.
Calculate the normalized RWR score vector R_1, R_2
OUTPUT: $sim(v_1, v_2) = 1 - H(P, Q)$ //Hellinger distance

Our method can be generalized to measuring similarity between graphs easily. For example, we may use averaged nodal similarity as graph similarity. We will show in next section that our graph similarity has many desired properties and produces more reasonable and robust result than the state-of-the-art method.

3. Simulation and Results.

3.1. **Node Similarity.** In this section, we conduct several experiments on important topologies (circles, paths, barbell and lollipops graphs) and synthetic benchmark networks to find if the RWR node similarity satisfies properties P1 to P3. We compare our method to four effective similarity or dissimilarity measures.

(1) Vertex/Edge Overlap (VEO) that is defined as follows:

$$sim_{VEO} = 2\frac{|V_1 \cap V_2| + |E_1 \cap E_2|}{|V_1| + |V_2| + |E_1| + |E_2|} \tag{6}$$

(2) Graph Edit Distance (GED) that is defined as follows:

$$sim_{GEO} = |V_1| + |V_2| - 2|V_1 \cap V_2| + |E_1| + |E_2| - 2|E_1 \cap E_2| \tag{7}$$

(3) Jaccard Similarity (JCD) that is defined as follows:

$$sim_{JCD} = \frac{|V_1 \cap V_2|}{|V_1 \cup V_2|} \tag{8}$$

(4) HOAD (HOrizontal Anomaly Detection) [18]: This algorithm also uses the random-walk-like method to extract global topological information for nodal similarity. Calculation flow is shown in Figure 1, where $\mathbf{A}$ and $\mathbf{W}$ are adjacent matrices of two graphs to be compared, and $\mathbf{M}$ is identity Matrix. $\mathbf{L}$ is the Laplacian Matrix and matrix $\mathbf{H}$'s column is composed of the top k eigenvectors of $\mathbf{L}$. The similarity is defined as

$$sim_{HOAD} = 1 - \frac{\mathbf{u}_i \cdot \mathbf{v}_i}{\| \mathbf{u}_i \| \| \mathbf{v}_i \|} \tag{9}$$

Because methods 1, 2 and 3 are based on local structures, to calculate the node similarity, we compare the egonet (the neighbor) of target nodes between two network layers.

P1. Distance Awareness 1: "The farther a structural change happens from a node, the less impact the change has on the node". In the cycle graph (Figure 2(a)), we can see that impacts of the edge-deletion on nodes' similarity decrease with the distance from the change increase.

P2. Distance Awareness 2: "The farther we move a node from its original position in a graph, the less similarity the node has to its initial state". In the path graph (Figure 3), we disconnect the node 0 first, and then link it with node 1, node 2, node 3 and node 4 respectively. When the node 0 deviates farther from its original position, its nodal similarity decreases monotonically from 0.74 to 0.5.

P3. Structure Awareness: "Important structural changes such as cluster split/merge around the node should be penalized more than other changes maintaining the connectivity or affiliation to clusters". From the barbell graphs as shown in Figure 2(b), where the graph in the first row is the original graph and the second and the third graph is compared with the original graph respectively, we can see that the change of connectivity

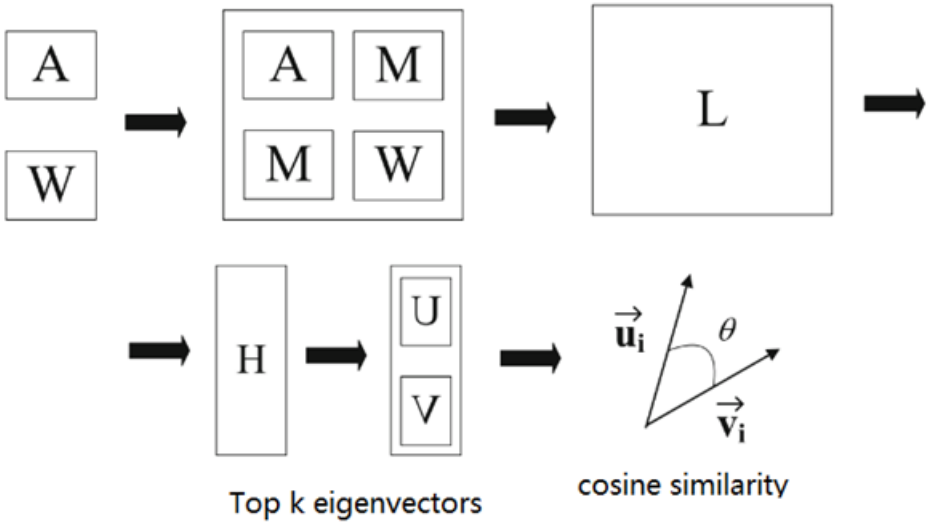


FIGURE 1. Calculation flow chart of the HOAD algorithm.

of a local region around a node has much more impacts on nodal similarity than a trivial change.

To test the sensitivity of our method to non-trivial structure changes such as cluster split, we use our method to find the split clusters in a dynamical network that evolves as time goes by. We generate two network snapshots with a dynamical community network generator 0. The first snapshot contains 1000 nodes constituting about 50 communities. In the second snapshot, 5 (about 10%) of the communities in the first snapshot split into two smaller clusters and meanwhile, links inside all communities are randomly rewired in its own community for simulating internal activities of community members. We then randomly select a seed node from each community in the first snapshot and measure its nodal similarity score as the indicator of the community split (an anomaly event). In other words, the lower the similarity score is, the more likely to split the community is. We use ROC analysis, one of the most widely used evaluation approaches in anomaly detection, which represents the trade-off between the detection rate and false alarm rate. A good detection method would produce an ROC curve as close to the left-top corner as possible, and thus the area under the ROC curve (AUC) in the range [0,1] is a good evaluation metric. The higher the AUC value is, the better the algorithm performs.

The experimental results are shown in Figure 4, where the first graph is the original one, node 1 is moved from its original position farther and farther. As can be seen, the RWR method always has higher AUC score than the baseline methods when the community overlap rate varies. Our method satisfies the structural awareness property and performs well even in high noisy circumstance. It also provides an effective and efficient method to detect the split or merge of communities before any community detection operation.

In a word, the baseline methods, VEO, JCD and GED, do not have the desirable properties, and despite their straightforwardness and fast computation speed, they fail to discern various changes in the graphs.

3.2. **Graph Similarity.** Graph similarity computation is an important task in many real-world applications such as brain connectivity comparison, social network analysis and anomaly detection. Our method is natural to be extended to measure graph similarity. Recently, the DELTACON 0 graph similarity method using the so-called Fast Belief

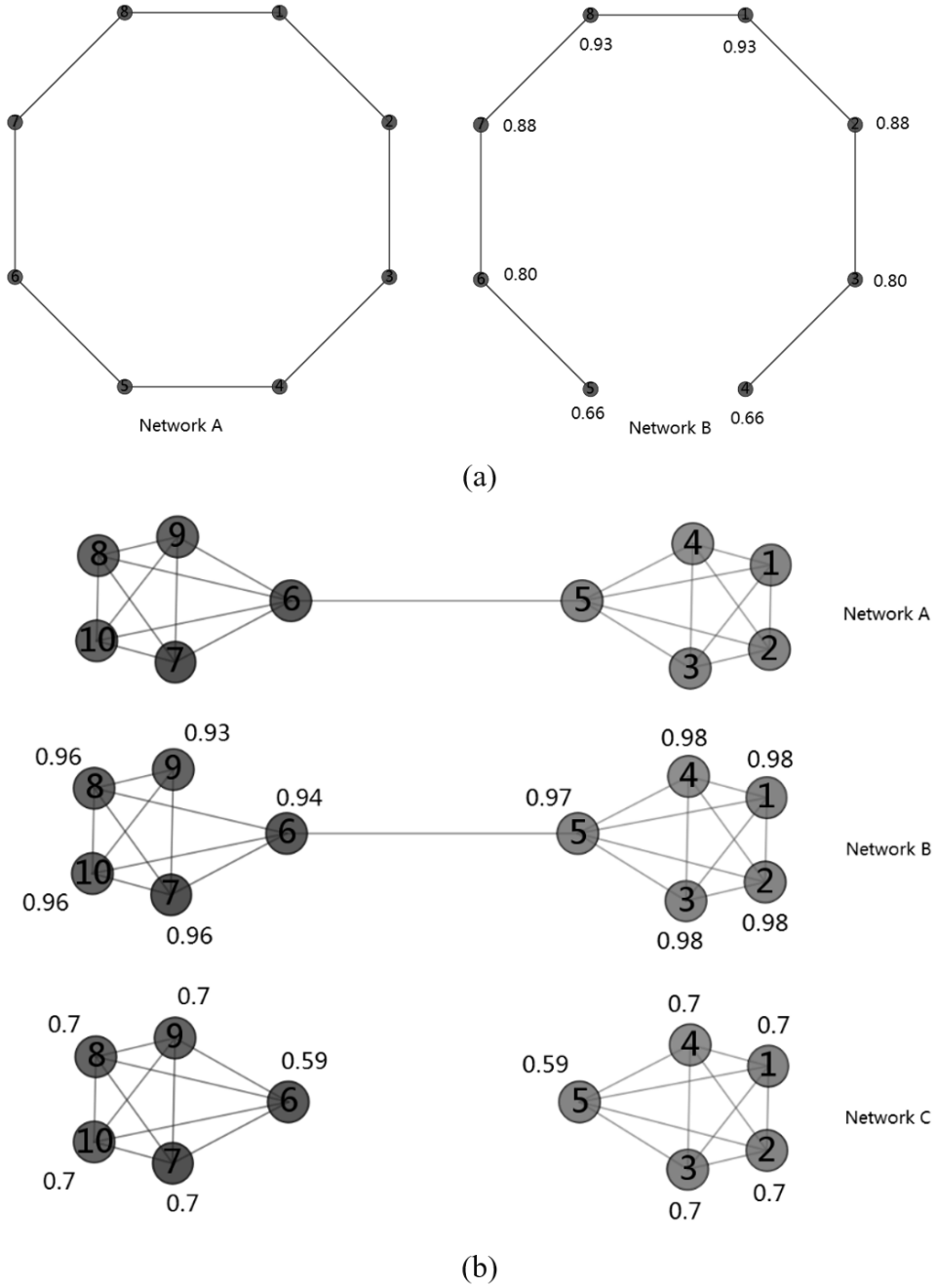


FIGURE 2. Nodal similarity of some graphs. (a) Nodal similarity of two cycle graphs,(b) Nodal similarity of barbell graphs.

Propagation algorithm has obtained many good results. Its stationary equation is:

$$[\mathbf{I} + \varepsilon^2 \mathbf{D} - \varepsilon \mathbf{A}]\mathbf{r} = \mathbf{q} \tag{10}$$

where ε is a coefficient, $\mathbf{D}$ is diagonal degree Matrix, $\mathbf{A}$ is adjacent Matrix and $\mathbf{q}$ is initial position vector.

The graph distance d and similarity is consequently defined as:

$$d = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n}(\sqrt{r_{1,ij}} - \sqrt{r_{2,ij}})^2} \tag{11}$$

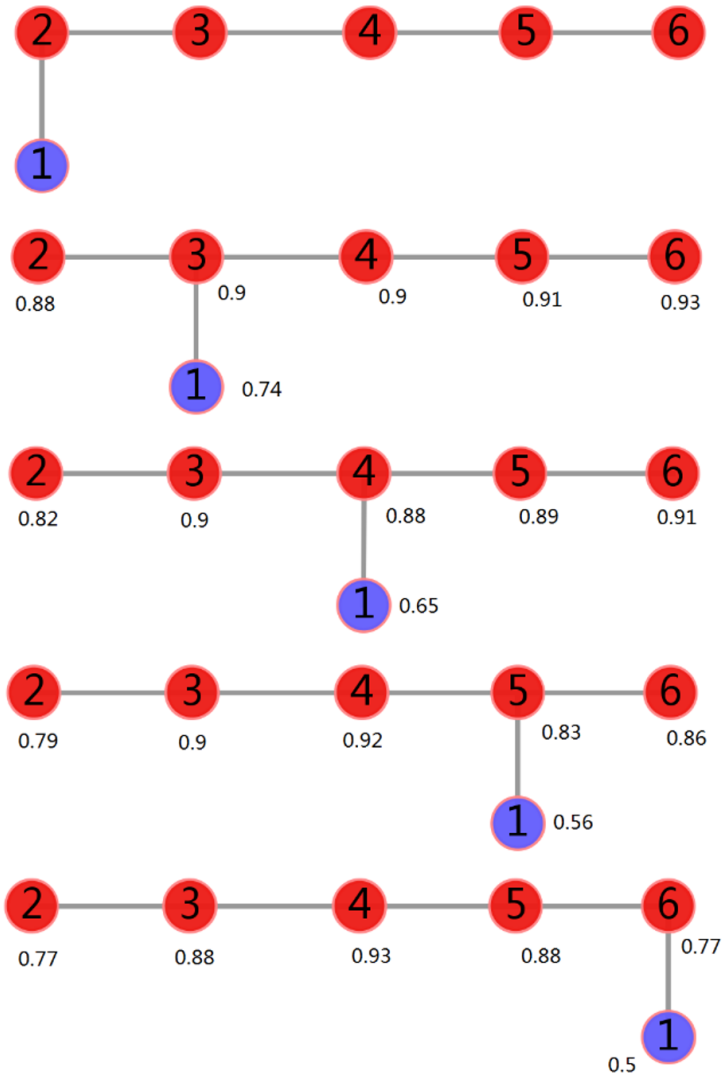


FIGURE 3. Node similarity of a path graph.

$$sim_{DELTACON} = \frac{1}{d+1} \tag{12}$$

The intuition behind the equation is that it imitates a belief propagation process in a network. It takes into account not only direct neighbors, but also 2-, 3- and k-step-away neighbors, with decreasing weight. Although DELTACON satisfies many desired properties, we find two critical problems of DELTACON that prevent its practical use. The first problem is that DELTACON will produce a non-zero or not small enough value even if two graph have no common edges. We find the reason is the same as the RWR algorithm's we mentioned above. The seed node always accounts for some affinity score even if two graphs are totally different, which in a distance d not large enough in equation 9. The similarity score of two graphs in Figure 5(a) is shown in Figure 5(b). The author defined parameter is marked by the red point. We can see that the similarity values are always far away from zero, which does not make any sense. The second problem is that the parameter $\varepsilon = 1/(1+\max(d_{ij}))$ requires the max degree of nodes in a network which could be as large as tens of millions in some social network platforms. Many practical networks have power-law degree distribution, which means a small proportion of nodes have huge

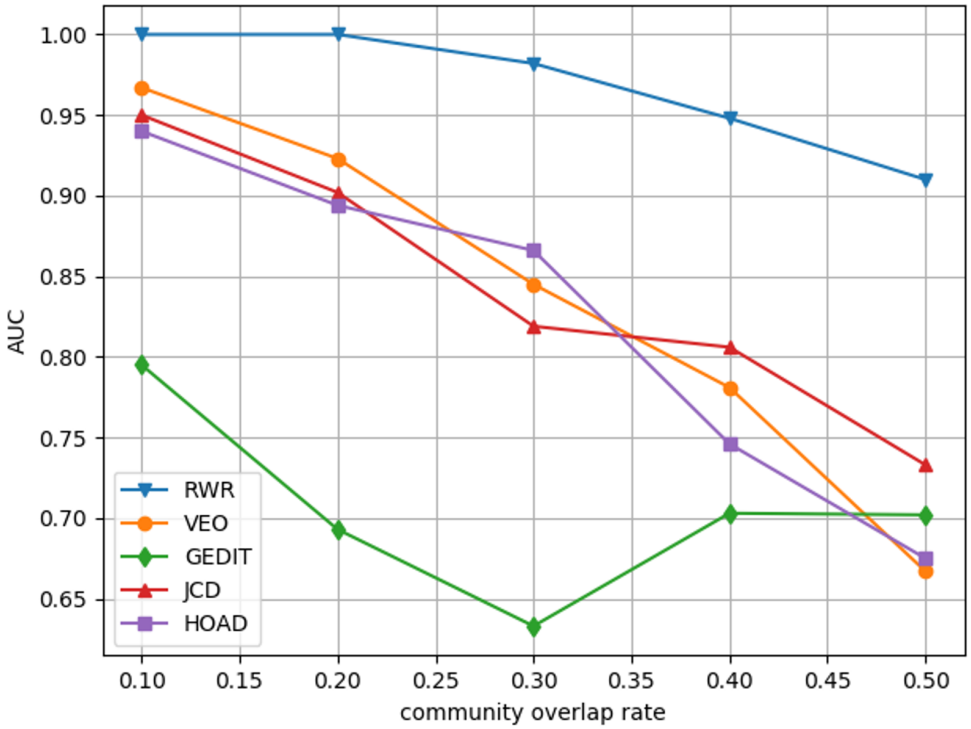


FIGURE 4. Prediction of community split with different methods.

degree and the rest only have small degree. The great unbalance of degree would reduce serious numerical problem in practical and things would become worse when we consider edges' weight.

Although the algorithm has above drawback, it has the desired properties proposed by the author:

P'1. [**Edge Importance**] Creating changes to disconnected components should be punished more severely than maintaining changes to connection properties.

P'2. [**Weight Awareness**] In a weighted graph, the greater the weight of the removed edges, the greater the impact on similarity measurement.

P'3. [**Edge-"Submodularity"**] In graphs with few edges, specific changes are more important than in graphs with higher density but equal size.

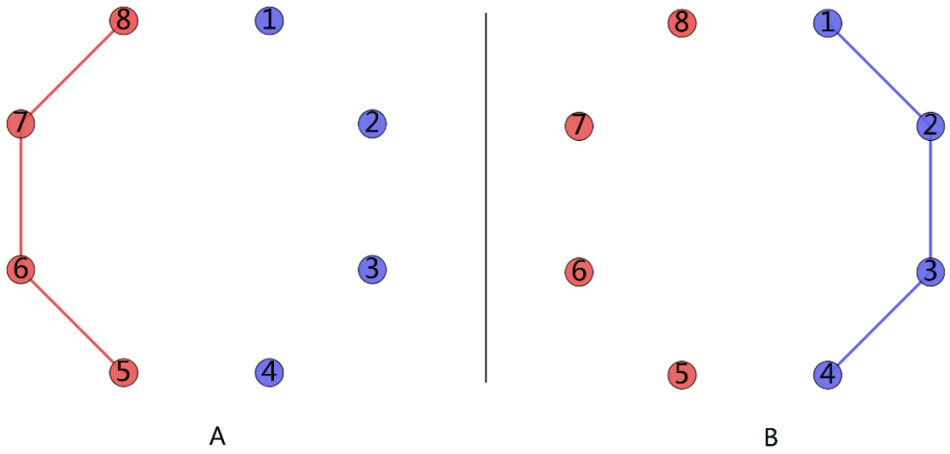
To show that our RWR graph similarity also has properties P'1, P'2 and P'3, we follow the author's test on the same networks as shown in Figure 6. Here we only consider some limited amount of classical networks for two reasons: 1. Many practical networks are composed of such small modular networks. 2. Real networks topology is too complex to make solid mathematical analysis.

The results for the first three properties are presented in Table 1, Table 2 and Table 3 respectively. We compare the graphs (A, B) and (A, C) and report the difference between the pairwise similarities of our proposed methods. Table entries that are positive mean that the corresponding method satisfies the corresponding property.

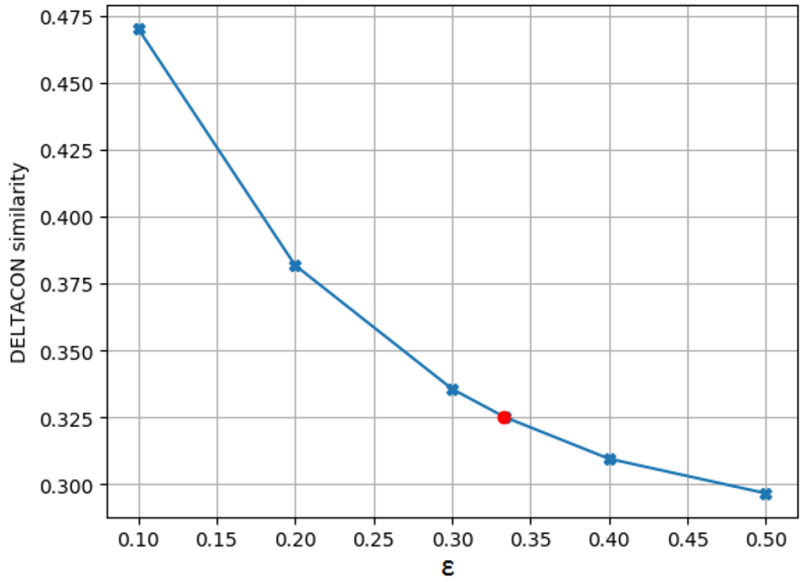
The graph distance measure λ-distance is defined as:

$$d_\lambda(G_1, G_2) = \sqrt{\sum_{i=1}^{k} (\lambda_{1i} - \lambda_{2i})^2} \tag{13}$$

where λ_{1i} and λ_{2i} are eigenvalues of G_1 and G_2 respectively, and k is the node number.



(a)



(b)

FIGURE 5. Drawback of the DELTACON algorithm.(a) Two graphs with no overlapped edge,(b) Graph similarity versus the parameter ε of DELTACON algorithm.

TABLE 1. "Edge Importance" (P'1)

Graphs			RWR	DELTACON	VEO	GEDIT	λ-distance
A	B	C	$\Delta s = sim(A,B) - sim(A,C)$				
B10	mB10	mmB10	0.291	0.07	0	0	0.21
L10	mL10	mmL10	0.293	0.04	0	0	-0.30
WhB12	mWhB12	mmWhB12	0.078	0.03	0	0	0.22
WhB12	m2WhB12	mm2WhB12	0.344	0.07	0	0	0.59

Generally speaking, our method is more sensitive and reasonable. In Table 1, "Edge Importance", the similarity differences Δs are always larger than DELTACON's, which strengthens the importance of the bridge edges. Furthermore, the first and forth items

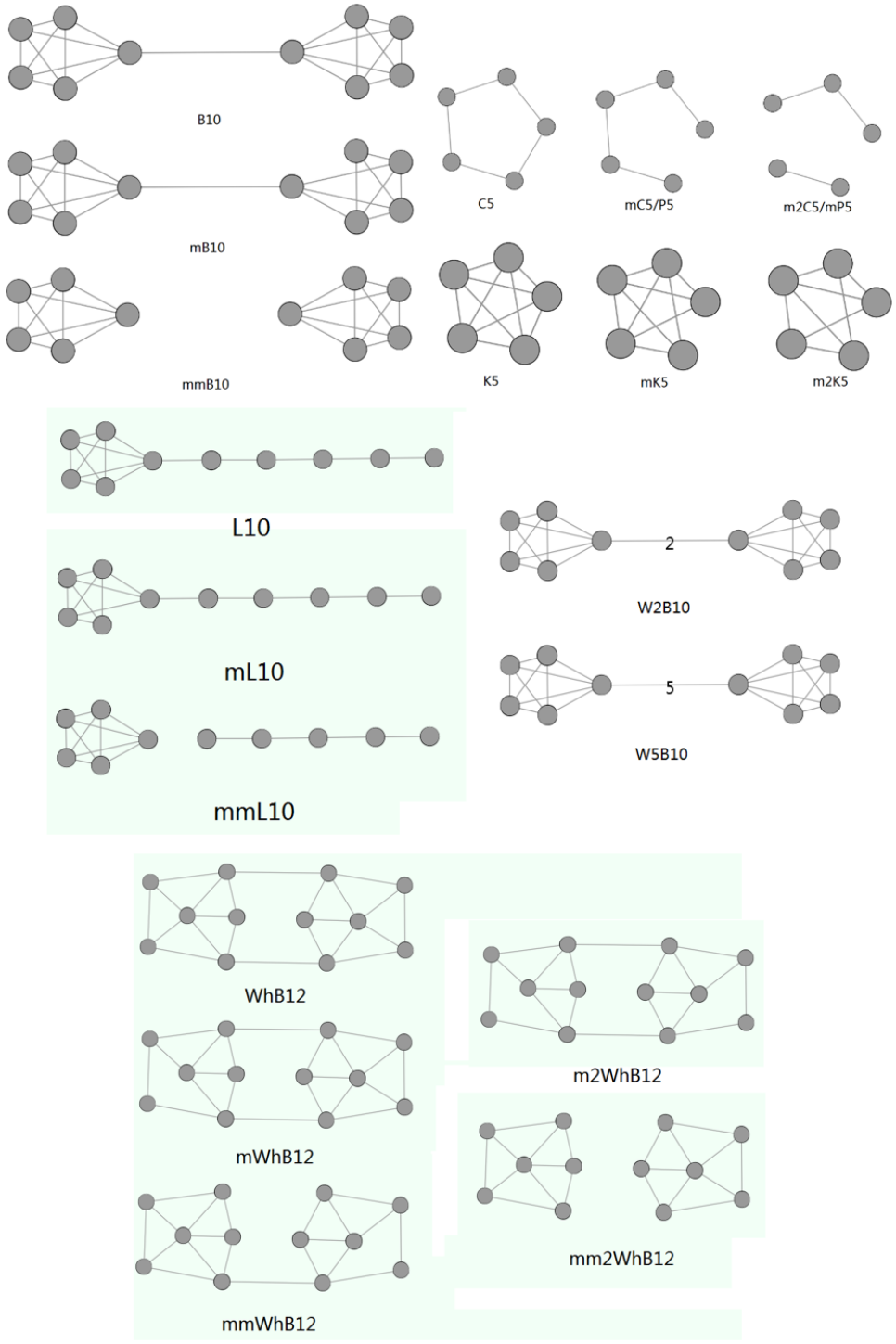


FIGURE 6. Small synthetic graphs.

of Δs is the same 0.07 in DELTACON, which means DELTACON can tell the similarity difference of the two groups of graphs, namely (B10, mB10, mmB10) and (WhB12, m2WhB12, mm2WhB12). However, it is intuitively reasonable to see the similarity difference of (WhB12, m2WhB12, mm2WhB12) should be larger than the similarity difference of (B10, mB10, mmB10) for (WhB12, m2WhB12) is more similar than (B10, mB10) and (WhB12, mm2WhB12) is less similar than (B10, mmB10), which definitely leads to a larger Δs.

Table 2. "Weight Awareness" (P'2)

Graphs			RWR	DELTACON	VEO	GEDIT	λ-distance
A	B	C	$\Delta s = sim(A,B) - sim(A,C)$				
B10	mB10	B10	0.069	0.09	-0.02	-1	3.67
mmB10	B10	mmB10	0.095	0.10	0	0	4.57
B10	mB10	w5B10	0.02	0.06	-0.02	-1	2.55
w5B10	w2B10	w5B10	0.227	0.10	0.02	1	2.23
w5B10	w2B10	w5B10	0.048	0.03	0	0	1.12

Table 3. "Edge-Submodularity" (P'3)

Graphs				RWR	DELTACON	VEO	GEDIT	λ-distance
A	B	C	D	$\Delta s = sim(A,B) - sim(A,C)$				
K5	mK5	C5	mC5	0.152	0.03	0.02	0	-0.24
C5	mC5	P5	mP5	0.178	0.03	0.01	0	-0.55

In Table 2 "Weight Awareness", we can find the same result for groups (mmB10, B10, mmB10, w5B10) and (w5B10, w2B10, w5B10, mmB10).

$$sim(w5B10, w2B10) - sim(w5B10, mmB10) = 0.1 \tag{14}$$

$$sim(mmB10, B10) - sim(mmB10, w5B10) = 0.1 \tag{15}$$

The Δs of the two group by DELTACON is the same as 0.10 but it is easy to find that (w5B10, w2B10) is more similar than (mmB10, B10),

$$sim(w5B10, w2B10 > sim(mmB10, B10) \tag{16}$$

which should cause a larger Δs. In the contrast, the result of our algorithm is more reasonable in all situations.

$$\Delta s = sim(A,B) - sim(A,C) \tag{17}$$

In Table 3 "Edge-Submodularity", following the same thought, we find the similarity difference Δs of (C5, mC5, P5, mP5) should be larger than the similarity difference of (K5, mK5, C5, mC5) because the similarity of (P5, mP5) should punished much more for P5 is essentially divided into two sub-graphs.

4. **Discussion and Conclusions.** In this paper, we aim to investigate node and graph similarities in the case that the correspondence between each pair of nodes is known. We provide a new scheme based on random walk to compare topological similarity of nodes in two different networks. Moreover, the graph similarity induced by our nodal similarity is shown to have many advantages over the state-of-the-art algorithm and meanwhile avoids its main drawbacks.

As can be seen, the key idea behind our method and DELTACON is same: both considering the global structure and focusing on local structure information. The local topological information is obtained by the random walker in RWR algorithm and at the same time, we need not explicitly define the meaning of 'local'. Nodes are naturally embedded into a Euclidean space by a random walker with stationary probability distribution and meanwhile keep their topological affinity in the original network. An interesting future work includes find a generalized framework for embedding nodes and graphs.

Acknowledgment. This work is partially supported by the specialized research projects of Huanjiang Laboratory, and Ningbo Science and Technology Innovation 2025 major projects(2020Z106 and 2023Z040). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] R. S. Caceres, T. Y. Berger-Wolf, and R. Grossman, "Temporal scale of processes in dynamic networks," *IEEE 11th International Conference on Data Mining Workshops*, Vancouver, BC, Canada, pp. 925—932, December 2011.

[2] Y. Wang, S. Parthasarathy, and S. Tatikonda, "Locality sensitive outlier detection: a ranking driven approach," *IEEE 27th International Conference on Data Engineering*, May 2011.

[3] Y. Dong, Q. Ke, B. Wang, and B. Wu, "Link prediction based on local information," *IEEE International Conference on Advances in Social Networks Analysis and Mining*, pp.382-386, July 2011.

[4] L. Katz, "A new status index derived from sociometric analysis," *Psychometrika*, vol. 8, pp. 39–43, 1953.

[5] S. Brin, and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1-7, pp.107–117, April 1998.

[6] Q. Zhao, and H. Lu, "A PCA-based watermarking scheme for tamper-proof of web pages," *Pattern Recognition*, vol. 38, no. 8, pp. 1321–1323, 2005. H. Tong, H. Qu, and H. Jamjoom, "Measuring proximity on graphs with side information," *Proceedings of the 8th IEEE International Conference on Data Mining*, Pisa, Italy, December 2008.

[7] G. Jeh, and J. Widom, "SimRank: a measure of structural-context similarity," *8th ACM SIGKDD International Conference*, Edmonton, Alberta, Canada, pp. 538–543, July 2002.

[8] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," *Morgan Kaufmann Publishers Inc.*, 2003.

[9] D. H. Chau, C. Nachenberg, J. Wilhelm, A. Wright, and C. Faloutsos, "Polonium: tera-scale graph mining and inference for malware detection",*SIAM International Conference on Data Mining*, pp. 131–142, April 2011.

[10] H. N. Kim, and A. Saddik, "Personalized pagerank vectors for tag recommendations: inside folkrank," *Proceedings of the fifth ACM conference on Recommender Systems*, pp. 45–52, 2011.

[11] W. Yu, C. Aggarwal, S. Ma, and H. Wang, "On anomalous hotspot discovery in graph streams," *Proceedings of the IEEE 13th International Conference on Data Mining*, Vol. 1, pp. 1271–1276, 2013.

[12] T. Ji, D. Yang, and J. Gao, "Incremental local evolutionary outlier detection for dynamic social networks," *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, Berlin, Heidelberg, 2013.

[13] D. Koutra, J. T. Vogelstein, and C. Faloutsos, "DELTACON: a principled massive-graph similarity function," *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 162–170, 2013.

[14] E. Hellinger, "Neue begründung der theorie quadratischer formen von unendlichvielen veränderlichen," *Journal Für Die Reine Und Angewandte Mathematik (in German)*, vol. 136, pp. 210–271, 1909.

[15] P. Papadimitriou, A. Dasdan, and H. G. Molina, "Web graph similarity for anomaly detection," *Journal of Internet Services and Applications*, vol.1, no.1, pp.19–30, 2010.

[16] D. Greene, D. Doyle, and P. Cunningham, "Tracking the evolution of communities in dynamic social networks," *IEEE International Conference on Advances in Social Networks Analysis and Mining*, Odense, Denmark, September 2010.

[17] J. Jung, L. Sael, K. Shin, and U. Kang, "Random walk with restart on large graphs using block elimination," *ACM Transactions on Database Systems*, vol. 41, no. 2, Article Number 12, May 2016.

[18] J. Gao, N. Du, W. Fan, D. Turaga, S. Parthasarathy, and J. Han, "A multi-graph spectral framework for mining multi-source anomalies," *Graph Embedding for Pattern Analysis*, Springer, pp. 205–227, 2013.

[19] J. W. Qu, X. Q. Lyu, C. C. Zhang, P. H. Sun, B. Wang, and Z.Tang,"PharmKi: a retrieval system of chemical structural formula based on graph similarity," *IEEE Conference on Multimedia Information Processing and Retrieval*, Vol.1, pp. 45–50, 2018.

[20] G. S. Bopcheab, and B. M. Mehtrea, "Graph similarity metrics for assessing temporal changes in attack surface of dynamic networks," *Computers and Security*, vol. 64, pp. 16–43, January 2017.

[21] F. Zhang, T. -Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, L. Liu, "Application of quantum genetic optimization of LVQ neural network in smart city traffic network prediction," *IEEE Access*, vol. 8, pp. 104555-104564, 2020.

[22] L. Kang, R.-S. Chen, Y. -C. Chen, C. -C. Wang, X. Li, T. -Y. Wu, "Using cache optimization method to reduce network traffic in communication systems based on cloud computing," *IEEE Access*, vol. 7, pp. 124397-124409, 2019.

[23] D. Zhang, W. Zhang, R. Yang, M. Guo, C. -M. Chen, "A distributed computation of the shortest path in large-scale road network," *Journal of Ambient Intelligence and Humanized Computing*, 2019. Available: https://doi.org/10.1007/s12652-019-01615-5.