

Identifying important nodes of hypergraph: An improved PageRank algorithm

Yu-Hao Piao(朴宇豪), Jun-Yi Wang(王俊义), and Ke-Zan Li(李科赞)

Citation: Chin. Phys. B, 2025, 34 (**4**): 048902. DOI: 10.1088/1674-1056/adb269
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Node ranking based on graph curvature and PageRank

Hongbo Qu(曲鸿博), Yu-Rong Song(宋玉蓉), Ruqi Li(李汝琦), Min Li(李敏), and Guo-Ping Jiang(蒋国平)
Chin. Phys. B, 2025, 34 (**2**): 028901. DOI: 10.1088/1674-1056/ad9a9b

Accurate prediction of essential proteins using ensemble machine learning

Dezhi Lu(鲁德志), Hao Wu(吴淏), Yutong Hou(侯俞彤), Yuncheng Wu(吴云成), Yuanyuan Liu(刘媛媛), and Jinwu Wang(王金武)
Chin. Phys. B, 2025, 34 (**1**): 018901. DOI: 10.1088/1674-1056/ad8db2

Important edge identification in complex networks based on local and global features

Jia-Hui Song(宋家辉)
Chin. Phys. B, 2023, 32 (**9**): 098901. DOI: 10.1088/1674-1056/aca6d8

LCH: A local clustering H-index centrality measure for identifying and ranking influential nodes in complex networks

Gui-Qiong Xu(徐桂琼), Lei Meng(孟蕾), Deng-Qin Tu(涂登琴), and Ping-Le Yang(杨平乐)
Chin. Phys. B, 2021, 30 (**8**): 088901. DOI: 10.1088/1674-1056/abea86

Influential nodes identification in complex networks based on global and local information

Yuan-Zhi Yang(杨远志), Min Hu(胡敏), Tai-Yu Huang(黄泰愚)
Chin. Phys. B, 2020, 29 (**8**): 088903. DOI: 10.1088/1674-1056/ab969f

---

COMPUTATIONAL PROGRAMS FOR PHYSICS

Identifying important nodes of hypergraph: An improved PageRank algorithm

Yu-Hao Piao(朴宇豪)[1], Jun-Yi Wang(王俊义)[2], and Ke-Zan Li(李科赞)[1,3,†]

[1] *School of Mathematics and Computing Science, Guilin University of Electronic Technology, Guilin* 541004, *China*
[2] *School of Information and Communication, Guilin University of Electronic Technology, Guilin* 541004, *China*
[3] *Guangxi Wireless Broadband Communication and Signal Processing Key Laboratory, Guilin University of Electronic Technology, Guilin* 541004, *China*

(Received 6 December 2024; revised manuscript received 12 January 2025; accepted manuscript online 5 February 2025)

Hypergraphs can accurately capture complex higher-order relationships, but it is challenging to identify their important nodes. In this paper, an improved PageRank (ImPageRank) algorithm is designed to identify important nodes in a directed hypergraph. The algorithm introduces the Jaccard similarity of directed hypergraphs. By comparing the numbers of common neighbors between nodes with the total number of their neighbors, the Jaccard similarity measure takes into account the similarity between nodes that are not directly connected, and can reflect the potential correlation between nodes. An improved susceptible–infected (SI) model in directed hypergraph is proposed, which considers nonlinear propagation mode and more realistic propagation mechanism. In addition, some important node evaluation methods are transferred from undirected hypergraphs and applied to directed hypergraphs. Finally, the ImPageRank algorithm is used to evaluate the performance of the SI model, network robustness and monotonicity. Simulations of real networks demonstrate the excellent performance of the proposed algorithm and provide a powerful framework for identifying important nodes in directed hypergraphs.

Keywords: hypergraph, important node, PageRank, susceptible–infected (SI) model, centrality index

PACS: 89.75.–k

DOI: 10.1088/1674-1056/adb269 **CSTR:** 32038.14.CPB.adb269

1. Introduction

In a network, identifying important nodes is crucial. This issue, also known as the centrality problem, involves assigning scores to nodes that quantify their importance. In the past few decades, research on identifying important nodes has been widely used in various domains, such as disease transmission,[1–3] political influence,[4] rumor control,[5,6] advertising,[7] and traffic management.[8] These studies have focused on ordinary graphs that represent relationships between individuals.[9–11] Many effective algorithms have been proposed to identify the important nodes in ordinary graphs.[12–17] However, interactions involving multiple individuals are more commonly observed in real-world systems, such as chatting groups, protein complexes, and so on.[18,19] Although ordinary networks are simple, they fail to capture the synergistic effects among multiple nodes. In contrast, a hypergraph that allows a hyperedge to connect multiple nodes can effectively overcome the limitations of representing a complex system with a conventional network. Currently, there is limited research on the identification of important nodes in hypergraphs.[20–23] The above research mainly focuses on enhancing algorithms in ordinary graphs and applying them in hypergraphs. The above improvements have yielded some results, but there are still many algorithms that are not applicable to hypergraphs, such as the PageRank algorithm.

Currently, research on the PageRank algorithm primarily focuses on ordinary graphs. Kamvar *et al.*[24] have proposed an adaptive PageRank algorithm to expedite the computation of PageRank. In this algorithm, the PageRank of pages that have already converged is not recalculated at each iteration after convergence. Zheng *et al.*[25] have proposed an enhanced PageRank algorithm. This algorithm clusters all web pages by analyzing search rows and utilizes the information entropy obtained from clustering to adjust the weight of the PageRank algorithm. This adjustment resolves the issue of the PageRank algorithm showing a strong correlation between continuously visited web pages. Hasan *et al.*[26] have proposed a hybrid PageRank algorithm based on an optimized normalization technique and a content-based method. This algorithm reduces the number of iterations needed to calculate page ranking, enhancing efficiency. Rani *et al.*[27] have proposed a weighted PageRank algorithm (WPR), takes into account the importance of both inlinks and outlinks of the pages and distributes rank scores based on the popularity of the pages. Haveliwala *et al.*[28] have proposed a topic-sensitive PageRank algorithm, which utilizes precomputed biased PageRank vectors to generate query-specific importance scores for pages at query time. Tran *et al.*[29] proposed the Pagerank algorithm in hypergraph

†Corresponding author. E-mail: lkzzr@guet.edu.cn

© 2025 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

for the first time, but it lacks comparison and improvement with other methods.

However, there are still some shortcomings in the previous work. The PageRank algorithm in hypergraphs has been proposed, but it only considers the relationships between directly linked nodes. It does not account for the relationships between indirectly adjacent nodes, and as a result, it cannot fully explore the structural features of a hypergraph. Moreover, due to the lack of evaluation indicators in directed hypergraphs, the superiority of the algorithm has not been reflected. To solve this problem, in this paper, an ImPageRank algorithm is proposed to evaluate the importance of nodes in hypergraphs. By introducing the Jaccard similarity in the directed hypergraph, the algorithm considers the similarity of unconnected nodes. By transferring the centrality index from the undirected hypergraph to the directed hypergraph, compared with other centrality methods, the algorithm shows advantages in robustness, accuracy and monotonicity. The main contributions of this paper are as follows.

1. An ImPageRank algorithm is proposed, incorporating a weighted Jaccard similarity matrix to enhance node similarity and propagation. The transition probability matrix integrates the original transition matrix with the similarity matrix to optimize node correlations.

2. An improved susceptible–infected (SI) dynamic model is proposed, incorporating head–tail and intra-node propagation to simulate complex interactions. The model enhances the influence of network topology, aligning propagation probability with real-world infection mechanisms and group behavior.

3. Centrality measures for directed weighted hypergraphs are proposed, including hypergraphs degree centrality, hypergraphs closeness centrality, hypergraphs betweenness centrality, and hypergraphs vector centrality, providing a theoretical framework for analyzing and evaluating the importance of nodes in hypergraphs.

The structure of this paper is as follows. Section 2 provides the necessary background knowledge and introduces the index of centrality in directed hypergraph. Section 3 focuses on the implementation of the core algorithm, the ImPageRank algorithm. Section 4 introduces the improved SI algorithm. Section 5 covers data sources and experimental results. Finally, Section 6 concludes the paper.

2. Preliminaries

2.1. Definition of directed hypergraph[30]

We denote a hypergraph with N nodes and M directed hyperedges as $H = (V, E)$, where $V = \{v_1, v_2, \ldots, v_N\}$ is set of nodes and $E = \{e_1, e_2, \ldots, e_M\}$ is the set of hyperedges. For a hyperedge $e = (e^{\text{Tail}}; e^{\text{Head}})$, e^{Tail} is called the set of hyperedge tails and e^{Head} is called the set of hyperedge heads. Please note that $e^{\text{Head}} \subset V, e^{\text{Tail}} \subset V$, $e^{\text{Tail}} \neq \emptyset$, $e^{\text{Head}} \neq \emptyset$, $e^{\text{Tail}} \cap e^{\text{Head}} = \emptyset$.

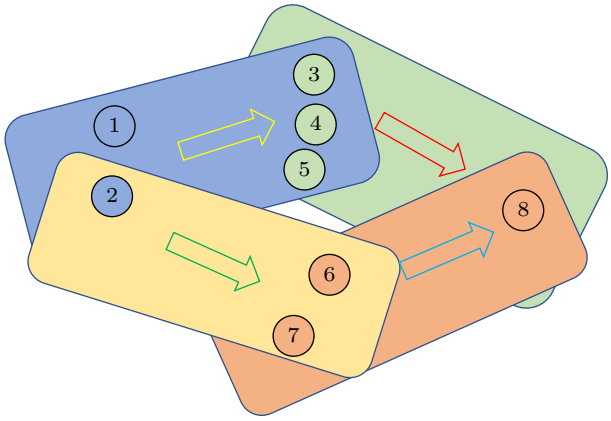


Fig. 1. There are four hyperedges $e_1 = (3,4,5;1,2)$, $e_2 = (6,7;2)$, $e_3 = (8;3,4,5)$, $e_4 = (8;6,7)$.

It can be seen from Fig. 1 that node 8 has the highest importance, because all nodes point to it (that is, the degree is the largest). However, the importance ranking of nodes 3,4,5 and 6,7 is difficult to define, because it involves the weight of the hyperedge and the allocation of node 2 to the hyperedges e_1 and e_2. One possible importance ranking without considering the weight of the hyperedge is that nodes 3,4,5 have a greater importance ranking than nodes 6,7, because nodes 3,4,5,6,7 are contributed by node 2, but nodes 3,4,5 are also contributed by node 1. However, real hypergraphs are much more complex than this, so a better way to measure the importance of nodes is needed.

2.2. Incidence matrices for directed hypergraph[29]

The directed hypergraph $H = (V, E)$ can be represented by two incidence matrices H^{Tail} and H^{Head}. These two incidence matrices H^{Tail} and H^{Head} can be defined as follows:

$$h^{\text{Tail}}(v, e^{\text{Tail}}) = \begin{cases} 1, & \text{if } v \in e^{\text{Tail}}, \\ 0, & \text{otherwise;} \end{cases}$$

$$h^{\text{Head}}(v, e^{\text{Head}}) = \begin{cases} 1, & \text{if } v \in e^{\text{Head}}, \\ 0, & \text{otherwise.} \end{cases}$$

Let $w(e)$ be the weight of the hyperedge e. Let W be the diagonal matrix containing the weights of hyperedges in its diagonal entries. From the above definitions, we can define the tail and head degrees of the nodes v and the tail and head degrees of the hyperedge e as follows:

$$d^{\text{Tail}}(v) = \sum_{e \in E} w(e) h^{\text{Tail}}(v, e^{\text{Tail}}),$$

$$d^{\text{Head}}(v) = \sum_{e \in E} w(e) h^{\text{Head}}(v, e^{\text{Head}}),$$

$$d^{\text{Tail}}(e) = \sum_{v \in V} h^{\text{Tail}}(v, e^{\text{Tail}}),$$

$$d^{\text{Head}}(e) = \sum_{v \in V} h^{\text{Head}}(v, e^{\text{Head}}).$$

Let $D_v^{\text{Tail}}, D_v^{\text{Head}}, D_e^{\text{Tail}}$, and D_e^{Head} be four diagonal matrices containing the tail and head degrees of nodes and the tail and

head degrees of hyperedges in their diagonal entries, respectively. Please note that D_v^{Head}, $D_v^{\text{Tail}} \in R^{|V|\times|V|}$ matrices and D_e^{Head}, $D_e^{\text{Tail}} \in R^{|E|\times|E|}$ matrices.

2.3. Pearson correlation coefficient[31]

The Pearson correlation coefficient is a statistical measure employed to assess the strength and direction of the linear relationship between two variables. It quantifies the extent of linear association between two variables, yielding a value ranging from -1 to 1. A coefficient of -1 indicates a perfect negative correlation, 1 denotes a perfect positive correlation, and 0 signifies a lack of a linear relationship. The following presents the definition of the Pearson correlation coefficient:

$$r = \frac{\sum (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum (X_i - \bar{X})^2 \cdot \sum (Y_i - \bar{Y})^2}}, \tag{1}$$

where X_i and Y_j represent the observed values of the two variables respectively, and $\overline{X}$ and $\overline{Y}$ denote the mean values of these variables. The Pearson correlation coefficient can be obtained by calculating the covariance and standard deviation of the sample. This process yields the standardized form of the relationship between the two variables.

2.4. Evaluation index in hypergraphs

Note that here our definitions are all for directed weighted hypergraphs.

Hypergraphs degree centrality (HDC) The formula for hypergraphs degree centrality can be expressed in terms of the out-degree of a node within the hypergraph and the weights of the associated hyperedges. For a given node, its hyperdegree centrality is calculated as the sum of the weights of all hyperedges for which it serves as a tail node. The definition is as follows:

$$\text{HDC}(v) = \sum_{e \in E_v} w(e), \tag{2}$$

where E_v denotes the set of hyperedges containing nodes v as the tail nodes and $w(e)$ is the hyperedge weight of e.

Hypergraph closeness centrality (HCC) The closeness centrality measures the reciprocal of the shortest path distance from a node to all other nodes and is often used to reflect the or of a node in a network. We first define the distance on a weighted directed graph as being measured by the inverse of the weights on the edges. The definition is as follows:

$$d(x,y) = \frac{1}{w(x,y)}, \tag{3}$$

where $d(x,y)$ is the distance from node x to y, and $w(x,y)$ is the weight from node x to y, and the distance is defined as the reciprocal of the weight. The distance is defined as the inverse of the weight because in many network propagation or transmission problems, the distance is often used to represent the 'time' or 'cost' of information. Therefore, if the weight $w(x,y)$ is higher, then the time/cost required is lower, i.e., the distance $d(x,y)$ should be smaller. Then we give the definition of hypergraphs closeness centrality:

$$\text{HCC}(v) = \frac{n-1}{\sum_{u \neq v} d(v,u)}, \tag{4}$$

where n is the number of nodes in the hypergraph, and $d(v,u)$ is the shortest path distance from node v to node u. The Dijkstra algorithm is generally used to find the shortest path.

Hypergraph betweenness centrality (HBC) Hypergraphs betweenness centrality can be defined as the degree to which a node v contributes to the shortest path between all pairs of nodes in the hypergraph. Specifically, the intermediary centrality of node v can be calculated by the following formula:

$$\text{HBC}(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}}, \tag{5}$$

where σ_{st} denotes the number of hyperedges connecting node s and node t, and $\sigma_{st}(v)$ represents the number of hyperedges passing through node v in the shortest path connecting node s and node t.

Hypergraph vector centrality (HVC) Hypergraph vector centrality can be measured by the neighborhood matrix of a node. Specifically, hypergraph vector centrality can be computed from the relative importance of nodes in the hypergraph. For a given hypergraph vector v, its hypergraph vector centrality is defined in the following form:

$$\text{HVC}(v) = \frac{(A \cdot x)_v}{\|A \cdot x\|_1}, \tag{6}$$

where the adjacency matrix A of the hypergraph can be computed by the formula in Subsection 2.2:

$$A = H^{\text{Head}} \cdot (H^{\text{Tail}})^{\text{T}}. \tag{7}$$

To solve for the eigenvector x in Eq. (6). the iterative update is performed using the power iteration method

$$x_{k+1} = \frac{A \cdot x_k}{\|A \cdot x_k\|_2}, \tag{8}$$

where x_k is the feature vector of the k-th iteration. The $\|A \cdot x_k\|_2$ is the L_2 paradigm of the $A \cdot x_k$ vector. The iterative process continues until convergence, which is usually determined by examining the L_2 parameters of the feature vector changes

$$\|x_{k+1} - x_k\|_2 < \varepsilon, \tag{9}$$

where ε is a small tolerance threshold (taken as 10^{-6} in this paper).

3. ImPageRank algorithm

We know that the transition probability of the random walk on directed hypergraph can be defined as follows:[34]

$$p(u,v)=\sum_{e\in E} w(e)\frac{h^{\mathrm{Tail}}(u,e^{\mathrm{Tail}})}{d^{\mathrm{Tail}}(u)}\frac{h^{\mathrm{Head}}(v,e^{\mathrm{Head}})}{d^{\mathrm{Head}}(e)}. \tag{10}$$

We know that the transition probability matrix P of the random walk on the directed hypergraph can be defined as follows:[29]

$$P=(D_v^{\mathrm{Tail}})^{-1}H^{\mathrm{Tail}}W(D_e^{\mathrm{Head}})^{-1}(H^{\mathrm{Head}})^{\mathrm{T}}. \tag{11}$$

However, the above methods do not take into account the similarity relationships between nodes that are not directly connected when constructing the transition probability matrix. In this paper, we enhance the similarity and propagation relationships between nodes in the network by introducing a weighted Jaccard similarity matrix in the hypergraph. The improved transition probability of the random walk p_i is defined as follows:

$$p_i=\alpha\cdot p+(1-\alpha)\cdot s^{\mathrm{norm}}. \tag{12}$$

The improved transition probability matrix P_i is defined as follows:

$$P_i=\alpha\cdot P+(1-\alpha)\cdot S^{\mathrm{norm}}, \tag{13}$$

where P is the original transition probability matrix and S^{norm} is the normalized weighted Jaccard similarity matrix, which represents the similarity between nodes based on the hypergraph structure. Here α is a tuning parameter, notably $\alpha\in[0,1]$. The weighted Jaccard similarity matrix S is defined as follows:

$$S(i,j)=\frac{\sum_{e\in E_i\cap E_j}\min(w_e(i),w_e(j))}{\sum_{e\in E_i\cup E_j}\max(w_e(i),w_e(j))}. \tag{14}$$

In order to make the similarity distribution of each node conform to the probability distribution, the similarity matrix is normalized to ensure that the sum of each row is 1,

$$S(i,j)^{\mathrm{norm}}=\frac{S(i,j)}{\sum_j S(i,j)}. \tag{15}$$

We have that[29]

$$\begin{aligned}\sum_{v\in V}p(u,v)&=\sum_{v\in V}\sum_{e\in E}w(e)\frac{h^{\mathrm{Tail}}(u,e^{\mathrm{Tail}})}{d^{\mathrm{Tail}}(u)}\frac{h^{\mathrm{Head}}(v,e^{\mathrm{Head}})}{d^{\mathrm{Head}}(e)}\\&=\sum_{e\in E}w(e)\frac{h^{\mathrm{Tail}}(u,e^{\mathrm{Tail}})}{d^{\mathrm{Tail}}(u)}\frac{1}{d^{\mathrm{Head}}(e)}\sum_{v\in V}h^{\mathrm{Head}}(v,e^{\mathrm{Head}})\\&=\sum_{e\in E}w(e)\frac{h^{\mathrm{Tail}}(u,e^{\mathrm{Tail}})}{d^{\mathrm{Tail}}(u)}\\&=\frac{1}{d^{\mathrm{Tail}}(u)}\sum_{e\in E}w(e)h^{\mathrm{Tail}}(u,e^{\mathrm{Tail}})=1.\end{aligned} \tag{16}$$

Next, we need to prove that the row sum of P_i is 1,

$$\begin{aligned}\sum_{v\in V}p_i(u,v)&=\sum_{v\in V}(\alpha\cdot p(u,v)+(1-\alpha)\cdot s(u,v)^{\mathrm{norm}})\\&=\alpha\sum_{v\in V}p(u,v)+(1-\alpha)\sum_{v\in V}s(u,v)^{\mathrm{norm}}\\&=\alpha\cdot 1+(1-\alpha)\cdot 1=1.\end{aligned} \tag{17}$$

Finally the power iteration method is used to compute the Pagerank vector and the iteration process is as follows:

$$\pi^{(t+1)}=P_i\cdot\pi^{(t)}, \tag{18}$$

where $\pi^{(t)}$ is the number of PageRank vector for the t-th iteration until convergence when the iteration is stopped. Furthermore, we know that the above equation is easily solved by the Power method.

Since in the traditional PageRank algorithm for hypergraphs, only directly connected nodes contribute, the introduction of a normalized weighted Jaccard similarity matrix can be seen as a way to capture the extended proximity between nodes. Even if they are not directly connected, they can still influence each other as long as they are indirectly connected through common neighbors. This effect is particularly noticeable when some highly important nodes are neighbors. This can be seen as a form of soft connectivity, where the algorithm is able to sense relationships beyond the direct connections in the hypergraph. By introducing such a matrix, the algorithm is better able to explore the internal structure of the hypergraph, thus improving accuracy and discrimination.

4. SI extension dynamics in improved directed hypergraphs

In the real world, infections through pairwise interactions such as disease and information transmission are common and are referred to as simple infections. However, complex infections involve more intricate mechanisms of influence, where individual decisions may be shaped by group dynamics, leading to group effects. Understanding these dynamics is crucial, as individual behavior is largely influenced by group behavior. For instance, in the process of information dissemination, communication often starts with a specific information disseminator and a relevant social group, becoming more effective as group awareness increases. Specifically, the dissemination process can be divided into two parts.

(i) Head node infects tail node

For the tail node in each hyperedge, if there is an uninfected node in the tail node and an infected node in the head node, the propagation probability is calculated based on the structure of the hyperedge (number of nodes, weights) with the following infection rates:

$$\beta_m'=\beta_m\cdot\exp\left(-\frac{N_{mt}}{w_m\cdot N_{mh}}\right), \tag{19}$$

where β'_m is the adjusted contagion probability of the m-th hyperedge, β_m is the initial m-th hyperedge contagion probability, $N_{m\mathrm{t}}$ is the number of tail nodes of the m-th hyperedge, $N_{m\mathrm{h}}$ is the number of head nodes of the m-th hyperedge, and w_m is the weight of the m-th hyperedge. This means that as the number of tail nodes increases, the probability of propagation decreases because propagation becomes more challenging. Conversely, a larger hyperedge weight results in a smaller decrease in the probability of contagion; that is, the stronger the influence of the hyperedge, the easier the propagation. In contrast, an increase in the number of head nodes enhances the probability of contagion, as more head nodes signify a stronger source of propagation. This adjustment enables the contagion probability to more accurately reflect the characteristics of the network topology and the hyperedge structure.

(ii) Head nodes infect head nodes, tail nodes infect tail nodes

The propagation probability among tail nodes is influenced by the number of infected tail nodes within the current hyperedge and the total number of tail nodes. The specific transmission probabilities are outlined as follows:

$$\beta_m^{\mathrm{tail}} = \beta_m \cdot \frac{I_{m\mathrm{t}}}{N_{m\mathrm{t}}}, \tag{20}$$

where β_m^{tail} is the propagation probability between the tail nodes of the m-th hyperedge, β_m is the base propagation probability of the m-th hyperedge, $I_{m\mathrm{t}}$ is the number of tail nodes currently infected in the m-th hyperedge, and $N_{m\mathrm{t}}$ is the total number of all tail nodes of the m-th hyperedge. In other words, the propagation probability among tail nodes is proportional to the ratio of infected tail nodes. Consequently, as the number of tail nodes increases, the propagation probability decreases, which aligns with the actual dynamics of the infection mechanism.

The probability of transmission between head nodes is influenced by the number of infected head nodes within the current hyperedge and the total number of head nodes. The specific transmission probabilities are outlined as follows:

$$\beta_m^{\mathrm{head}} = \beta_m \cdot \frac{I_{m\mathrm{h}}}{N_{m\mathrm{h}}}, \tag{21}$$

where β_m^{head} is the propagation probability between the head nodes of the m-th hyperedge, β_m is the base propagation probability of the m-th hyperedge, $I_{m\mathrm{h}}$ is the number of head nodes currently infected in the m-th hyperedge, and $N_{m\mathrm{h}}$ is the total number of all head nodes of the m-th hyperedge.

5. Method evaluation

5.1. Dataset

In this section, we present the datasets used in the subsequent experiments. Note that these datasets are empirical data from multiple domains, and due to the lack of directed hypergraph datasets, in this paper, we use the method[33] to transform directed ordinary graphs into directed hypergraphs (Specifically, the edge of all tail nodes with the same head node is fused into a hyperedge first, and there is only one head node in this hyperedge, while multiple tail nodes are allowed. Finally, the head node with the same tail node is fused into the hyperedge, that is, the construction of the hyper edge is completed), in this method, although the form of the edge changes from a binary relationship to a hyperedge, the connectivity of each node does not change. The relationship between the head node and the tail node is preserved in the hyperedge. However, since the abstraction of hyperedges may obscure some specific binary relationships (for example, the direct effect of a specific head node on a specific tail node), this may affect the interpretation of some microscopic analyses. However, for studies that focus on global topological properties and overall relationships (such as importance), this effect is acceptable and, to some extent, improves the efficiency of analysis and clarity of results. Each of the constructed datasets has different topological properties, as shown in Table 1.

Table 1. Topological characterization of network data.

Dataset	$V(B)$	$E(B)$	$V(L)$	$E(L)$	$\overline{D}$	Δr_j	E	C
email-Eu-core	1005	866	854	865	29.94	334	0.3219	0.3905
email-Eu-core-temporal	986	816	803	814	337.05	333	0.32076	0.3396
bn-fly-drosophila_medulla_1	1781	1340	845	1311	18.88	127	0.32197	0.2796
CollegeMsg	1899	1318	1313	1314	31.51	237	0.22689	0.0711
bio-celegans-dir	453	349	309	349	10.14	146	0.0853	0.4104
BA-1_10_60-L5	804	477	490	477	57.7	673	0.09891	0.0102

The email-Eu-core dataset was generated using email data from a large European study, where members represent nodes and the relationships formed by sending messages represent edges. The email-Eu-core-temporal dataset was also derived from the same study but is structured as a temporal network. The bn-fly-drosophila_medulla_1 dataset contains data from the Drosophila brain network, where nodes represent neural elements and edges represent fiber bundles. The CollegeMsg dataset consists of private messages exchanged on the UC Irvine online social network, with nodes denoting users

and edges representing private messages. The bio-celegans-dir dataset is derived from a biological network, where nodes represent substrates and edges represent metabolic reactions. Finally, the BA-1_10_60-l5 dataset originates from a labeling network, where nodes represent product numbers and edges represent labeling paths.

In order to avoid the zero degree of the tail node or the degree of the head node which leads to the invertibility of the matrix of Eq. (11), these nodes as well as the hyperedges are eliminated. Here $V(B)$ and $E(B)$ represent the number of nodes and hyperedges of the directed hypergraph before elimination, and $V(L)$ and $E(L)$ represent the number of nodes and hyperedges of the directed hypergraph after elimination. $\overline{D}$ represents the average hyperdegree of each node in the hypergraph, Δr_j represents the maximum number of bases of hyperedges, E represents the efficiency of the hypergraph, and C represents the global clustering coefficient of the hypergraph.

5.2. Node diffusion influence quantization

(i) Improved SI model evaluation To quantify the impact of node diffusion, at early times, decision makers typically prioritize the highest ranked nodes. To meet this requirement, we measured the diffusion effect of the top 0.01 ranked nodes using each centroid metric and plotted the average diffusion curves of different hypergraphs and algorithms for $T = 5$ steps. The choice of $T = 5$ aims to strike a balance between computational efficiency and algorithmic feasibility. Using smaller time steps gives faster results while still providing insight into the overall behavior and performance of the algorithm.

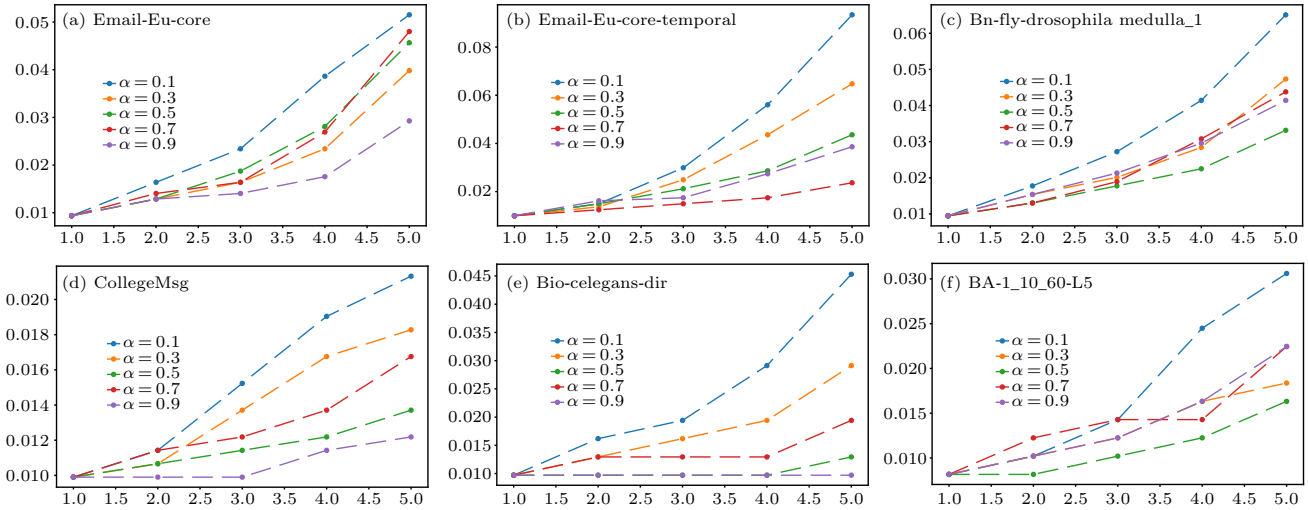


Fig. 2. On six real datasets, we choose the parameter α in Eqs. (12) and (13) to be set to 0.1,0.3,0.5,0.7,0.9, respectively. The initial contagion rate β_i (β_i represents the infection rate of the hyperedge i where the initially infected node resides) is set to 1×10^{-3}, because in many practical situations, the initial infection rate tends to be low. This low infection rate reflects the fact that only a few nodes in the network are initially infected, and the propagation gradually spreads to see what value of the parameter is set to give the best performance of the model. The final result is the average of 100 simulations.

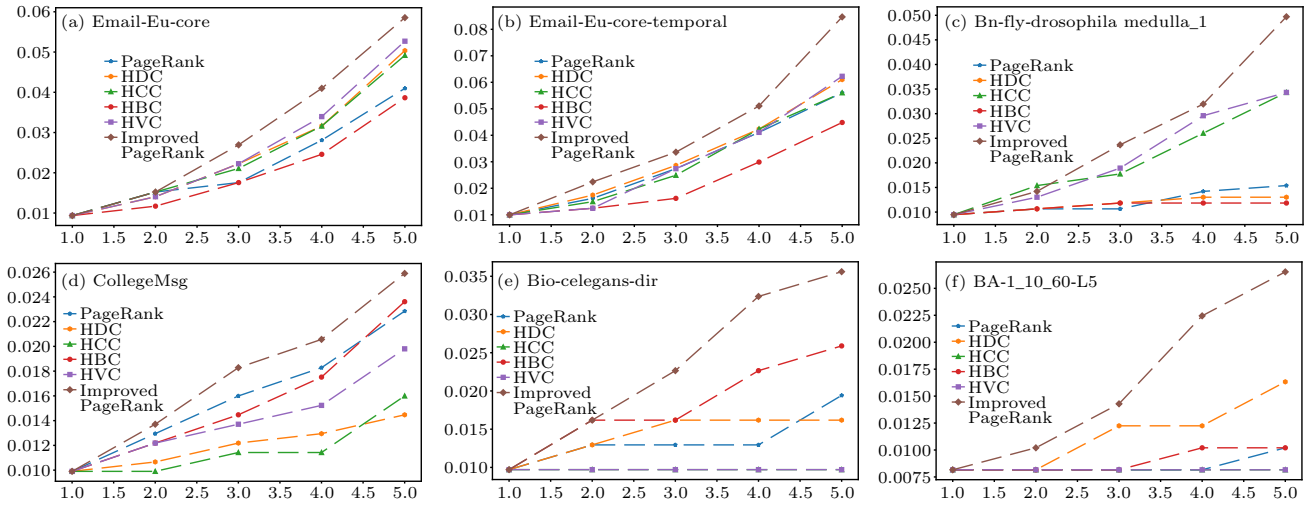


Fig. 3. On six real datasets, we choose the parameter $\alpha = 0.1$. The initial contagion rate β_i (β_i represents the infection rate of the hyperedge i where the initially infected node resides) is set to 1×10^{-3}. Comparison of the performance of the algorithms HDC, HCC, HBC, HVC, Pagerank and ImPageRank. The final result is the average of 100 simulations.

Observation of the baseline shows that in most hypergraphs, for example, email-Eu-core, CollegeMsg, bio-celegans-dir, it performs well at moments when α is 0.1, and even keeps the infected node leading for the first five time steps, for the other hypergraphs BA-1_10_60-L5, bn-fly-drosophila_medulla_1, although the infected node is not always leading in the first five time steps when α is 0.1, it is also better compared to the other methods. This may be due to the fact that smaller α values make the propagation process smoother. By incorporating node similarity, the propagation process not only relies on direct neighborhoods, but also takes into account potential connections between nodes, enhancing the flexibility and robustness of information flow. This smoother propagation effect may be superior in real networks.

We then measured the diffusion effect of the top 0.01 nodes using different centroid metrics and plotted the average diffusion curves for different hypergraphs and algorithms for $T = 5$ steps.

The observation of the baseline across multiple hypergraphs provides valuable insights into the performance of the ImPageRank algorithm compared to other algorithms. In networks such as email-Eu-core, bn-fly-drosophila_medulla_1, and CollegeMsg, the ImPageRank algorithm demonstrates superior performance, particularly in the early stages of the infection process. It rapidly propagates the infection to a large number of nodes, indicating that the incorporation of the weighted Jaccard similarity matrix effectively accelerates the information spread and enhances the model's ability to identify and infect connected nodes more quickly. However, in the case of the hypergraph BA-1_10_60-L5, the infection dynamics are somewhat different. While the ImPageRank algorithm shows slower initial infection during the first two time steps, it significantly outperforms the other algorithms in the later stages. This delayed initial infection followed by a rapid infection phase can be attributed to the model's ability to utilize the structural relationships and similarity between nodes, which accelerates the infection as the process progresses. This suggests that the ImPageRank algorithm may be better suited for capturing long-range dependencies and latent connections that are not immediately apparent, leading to faster and more widespread infection as time progresses. In summary, the ImPageRank algorithm shows excellent performance on various hypergraph datasets.

(ii) Monotonicity evaluation

An effective method for identifying important nodes should not only ensure the accuracy of the identification results, but also emphasize the distinguishability of the results. Therefore, we introduce the monotonicity index,[32] which is defined as follows:

$$M(R) = \left(1 - \sum_{r \in R} \frac{N_r(N_r - 1)}{N(N-1)}\right)^2, \tag{22}$$

where R denotes the node importance ranking table obtained by the node identification method, N is the total number of nodes, and N_r is the number of nodes with the same importance level r. $M(R) \in [0,1]$, the closer the value is to 1, the higher the discriminability of the node importance, and vice versa.

Table 2. Monotonicity values of identification results for different importance ranking methods.

Dataset	HDC	HCC	HBC	HVC	PageRank	ImPageRank
email-Eu-core	0.1240247047	0.9858977264	0.6452976316	0.9936830764	0.9936830764	0.9989972981
email-Eu-core-temporal	0.1978081648	0.9969102811	0.7113344797	0.9997383771	0.9997383771	0.9999881073
bn-fly-drosophila_medulla_1	0.6684082938	0.9170543267	0.9968669891	0.9990652559	0.9991131808	1.00000000
CollegeMsg	0.3100725129	0.989087885	0.91264049	0.997168383	0.9973244826	0.9999833708
bio-celegans-dir	0.3947154657	0.7258043793	0.9509052012	0.9623760541	0.9658708609	0.9992999446
BA-1_10_60-L5	0.5092995135	0.9883472389	0.805600687	0.855400632	0.8265366892	1.00000000

For the ImPageRank algorithm, $\alpha = 0.1$ is chosen here. The ImPageRank method consistently outperforms all other evaluation algorithms on multiple hypergraph datasets. It achieves near-perfect results, with values close to 1 in most cases, demonstrating its superior ability to efficiently rank node importance. Specifically, ImPageRank shows excellent performance on datasets such as bn-fly-drosophila_medulla_1, CollegeMsg, and BA-1_10_60-L5, achieving a perfect score of 1, demonstrating its high discriminative power and accuracy in capturing node importance. Compared to traditional PageRank, ImPageRank incorporates additional structural information through a weighted Jaccard similarity matrix, which enhances the ranking. Overall, the performance of ImPageRank is significantly improved, making it a more reliable and efficient method for node identification and ranking based on hypergraph models.

(iii) Correlation

The Pearson correlation coefficient was introduced to further investigate the correlation between the identification results of different methods. In the natural sciences, the Pearson correlation coefficient is commonly used to measure the correlation between two variables, which ranges between -1 and 1. Figure 4 shows the Pearson correlation results of six different significant node identification methods on six real

hypergraphs. After sorting the important nodes of the network by using six algorithms, such as PageRank, ImPageRank, HDC, HBC, HCC and HVC, Pearson correlation analysis is carried out on the six data groups obtained, so as to analyze the strength of the relationship between different algorithms. Because the correlation of the ranking results of important nodes can reveal the consistency and difference of algorithms in the evaluation of node importance, it is helpful to understand the internal relationship between different algorithms.

As can be seen in Fig. 4, for the hypergraphs of the six real networks, the ImPageRank and PageRank algorithms have a high similarity compared to several other methods, which indicates that the ImPageRank algorithm, as an improved version of PageRank, retains the original features in its mechanism. In the cases of email-Eu-core, bn-fly-drosophila_medulla_1, CollegeMsg datasets, ImPageRank has higher similarity to algorithms HBC and HCC compared to PageRank algorithm, because ImPageRank optimizes the propagation path between nodes by introducing the weights of hyper edges and similarity information, which increases the propagation speed and improves the coverage. This design allows ImPageRank to better consider indirect connections and similarities between nodes, thus presenting a propagation pattern in network propagation more similar to the HBC and HCC algorithms than the PageRank algorithm.

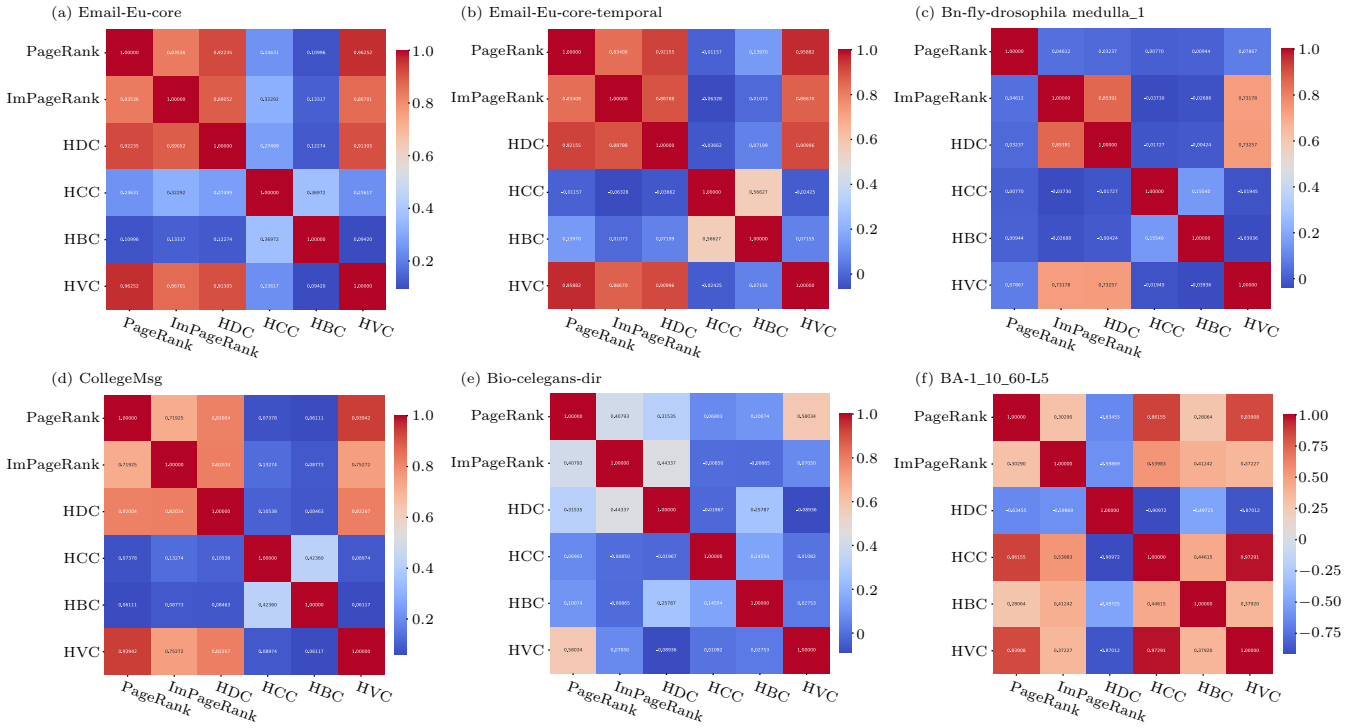


Fig. 4. Correlation matrix identification method for important nodes in hypergraphs. The colors of the matrix elements correspond to Pearson phase relationship values between different methods, ranging from bright red to dark blue.

(iv) Robustness

Network robustness is a fundamental measure of the effectiveness of critical node identification methods.[32] It refers to the ability of a network to maintain its functional and structural integrity when perturbed or attacked. Specifically, network robustness measures the degree to which a network can still function and transmit information effectively when nodes, edges or other critical elements of the network are removed or damaged, here the maximum component is utilized to measure network robustness, i.e., as the more important the node is removed, the smaller the number of the maximum component will be.

As can be seen from the data in Table 3, the ImPageRank algorithm exhibits significant differences in network robustness after node deletion, especially in the case of a high percentage of node deletions. Specifically, on multiple datasets, the ImPageRank screened significant nodes show greater impact when the network is disrupted. For example, on the email-Eu-core dataset, after removing 50% of the nodes, ImPageRank has 550 remaining nodes compared to 556 for PageRank, and although the difference is not significant, ImPageRank removes more significant nodes, resulting in a more fragile structure of the network. On the bn-fly-drosophila_medulla_1 and CollegeMsg datasets, ImPageRank also shows higher robustness after node deletion, especially when 50% of the nodes are deleted, the number of remaining nodes in ImPageRank is lower than that of PageRank, suggesting that the deleted nodes are more important in maintaining the network's functionality and more critical to its structural integrity, which further suggests that the ImPageRank algorithm, by identifying and retaining critical nodes in the network, makes the network better to withstand damage in the event of an attack.

Table 3. The maximum components after removing the top 10%, 30%, and 50% nodes in a hypergraph using different methods.

email-Eu-core				email-Eu-core-temporal			
Algorithms	Remove 10%	Remove 30%	Remove 50%	Algorithms	Remove 10%	Remove 30%	Remove 50%
PageRank	902	725	556	PageRank	909	750	589
ImPageRank	901	726	550	ImPageRank	906	747	587
HDC	901	723	552	HDC	906	748	590
HBC	901	730	552	HBC	906	748	590
HCC	901	724	551	HCC	908	749	589
HVC	901	726	555	HVC	906	747	589
bn-fly-drosophila_medulla_1				BA-1_10_60-L5			
Algorithms	Remove 10%	Remove 30%	Remove 50%	Algorithms	Remove 10%	Remove 30%	Remove 50%
PageRank	1680	1492	1310	PageRank	756	658	560
ImPageRank	1672	1486	1303	ImPageRank	755	657	559
HDC	1679	1486	1305	HDC	756	658	560
HBC	1679	1495	1308	HBC	756	658	560
HCC	1678	1489	1307	HCC	756	658	560
HVC	1675	1491	1310	HVC	756	658	560
CollegeMsg				bio-celegans-dir			
Algorithms	Remove 10%	Remove 30%	Remove 50%	Algorithms	Remove 10%	Remove 30%	Remove 50%
PageRank	1764	1497	1231	PageRank	424	362	297
ImPageRank	1763	1498	1228	ImPageRank	423	361	296
HDC	1762	1497	1233	HDC	421	361	299
HBC	1763	1500	1232	HBC	423	361	297
HCC	1762	1498	1230	HCC	424	358	294
HVC	1762	1497	1230	HVC	423	360	295

6. Conclusion

In this study, we propose an ImPageRank algorithm for directed hypergraphs, which incorporates a weighted Jaccard similarity matrix that takes into account the relationship between the similarities of disconnected nodes and enhances the ability to identify important nodes. And the SI model of directed hypergraph is proposed, which not only proposes the nonlinear propagation relationship of head nodes infecting tail nodes, but also proposes the contagion relationship of head nodes infecting head nodes and tail nodes infecting tail nodes, which is more in line with real-world contagion situations. And some evaluation metrics in directed hypergraphs are proposed: HDC, HBC, HCC, HVC. Experimental results on multiple real-world datasets show that ImPageRank significantly improves node identification by considering direct and indirect node relationships. Our approach outperforms traditional algorithms such as PageRank, HDC, HBC , HCC and HVC, especially in distinguishing node importance.

Our results have important practical implications, in social networks, ImPageRank can help identify influential individuals or communities; in biological networks, it can reveal key genes or proteins in disease transmission; in recommender systems, ImPageRank can help improve the accuracy of personalized recommendations by considering direct and indirect node relationships. Future research can explore the optimization of the computational process through parallel computing or approximation algorithms to improve its application on large-scale datasets.

7. Research outlook

While the ImPageRank algorithm has demonstrated promising results in enhancing node importance identification and network robustness in hypergraphs, several avenues for future research remain. First, extending the method to handle dynamic hypergraphs, where the structure of the network evolves over time, could further improve its applicability to real-world systems, such as social media platforms and biological networks, where node relationships frequently change. Secondly, ImPageRank still needs to invert the matrix, for which some nodes and hyperedges need to be deleted to satisfy the condition of matrix invertibility, which will destroy the topological features of the hypergraph. One feasible idea is to use pseudo-inversions to handle matrix inversions by performing a singular value decomposition (SVD) of the original matrix, which makes it possible to ensure the stability of the solution process even when dealing with matrices that are not full rank.

Program availability

The source code is openly available in the Science Data Bank at https://doi.org/10.57760/sciencedb.j00113.00151.

Acknowledgments

Project supported by the National Natural Science Foundation of China (Grant No. 62166010) and the Guangxi Natural Science Foundation (Grant No. 2023GXNSFAA026087).

References

[1] Bell D C, Atkinson J S and Carlson J W 1999 *Soc. Networks* **21** 1
[2] Zeng Q, Liu Y, Tang M and Gong J 2021 *Knowl. Based. Syst.* **229** 107365
[3] Liu C, Ma Y, Zhao J, Nussinov R, Zhang Y C, Cheng F and Zhang Z K 2020 *Phys. Rep.* **846** 1
[4] Aghdam S M and Navimipour N J 2016 *Karbala International Journal of Modern Science* **2** 88
[5] Shah D and Zaman T 2011 *IEEE. T. Inform. Theory* **57** 5163
[6] Askarizadeh M and Ladani B T 2021 *Eng. Appl. Artif. Intel* **100** 104198
[7] Mochalova A and Nanopoulos A 2014 *Electron. Commer. R. A* **13** 283
[8] Du Y, Gao C, Hu Y, Mahadevan S and Deng Y 2014 *Phys. A* **399** 57
[9] Strogatz S H 2001 *Nature* **410** 268
[10] Hofmann S G, Curtiss J and McNally R J 2016 *Psychol. Sci* **11** 597
[11] Gao Z K, Small M and Kurths J 2017 *Epl-Europhys. Lett.* **116** 50001
[12] Ruan Y, Tang J, Hu Y, Wang H and Bai L 2020 *IEEE Access* **8** 28947
[13] Jiang L, Jing Y, Hu S, Ge B and Xiao W 2018 *Appl. Sci.* **8** 1914
[14] Xu S and Wang P 2017 *Phys. A* **469** 654
[15] Hu F and Liu Y 2015 *Mod. Phys. Lett. B* **29** 1450268
[16] Wang T, Chen S, Wang X and Wang J 2020 *Physica. A* **551** 124137
[17] Tang P, Song C, Ding W, Ma J, Dong J and Huang L 2020 *Entropy* **22** 364
[18] Van Dolen W M, Dabholkar P A and De Ruyter K 2007 *J. Retail* **83** 339
[19] Lung R I, Gaskó N and Suciu M A 2018 *Scientometrics* **117** 1361
[20] Rafferty J, Watkins A, Lyons J, Lyons R A, Akbari A, Peek N, Jalali-Najafabadi F, Dhafari T B, Pate A and Martin G P 2021 *J. Biomed Inform.* **122** 103916
[21] Hu F, Tian K and Zhang Z K 2023 *Entropy* **25** 1263
[22] Képes T Z 2023 *PeerJ Comput. Sci.* **9** e1351
[23] Benson A R 2019 *SIAM Journal on Mathematics of Data Science* **1** 293
[24] Kamvar S, Haveliwala T and Golub G 2004 *Linear Algebra Appl.* **386** 51
[25] Zheng W, Mo S, Duan P and Jin X 2017 *3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE)* **p. 615-618**
[26] Hasan F, Ze K K, Razali R, Buhari A and Tadiwa E 2020 *Science Proceedings Series* **2** 17
[27] Rani R and Jain V 2013 *Int. J. Sci. Res. Pub.* **3** 1
[28] Haveliwala T H 2002 *Proceedings of the 11th international conference on World Wide Web* pp. 517–526
[29] Tran L, Quan T and Mai A 2019 arXiv:1909.01132 [cs.SI]
[30] Li K, Lin Y and Wang J 2024 *Chaos Solitons Fractals* **184** 115000
[31] Cohen I, Huang Y, Chen J, Benesty J, Benesty J, Chen J, Huang Y and Cohen I 2009 *Noise Reduction in Speech Processing* **1**
[32] Hu F, Tian K and Zhang Z 2019 *Entropy* **25** 1263
[33] Luo X, Peng J and Liang J 2022 *IET Intelligent Transport Systems* **16** 85
[34] Ducournau A and Bretto A 2014 *Computer Vision and Image Understanding* **120** 91
[35] Xie X, Zhan X, Zhang Z and Liu C 2023 *Chaos* **33** 1
[36] Al-Ghour S and Al-Saadi H 2024 *Aims. Math.* **9** 1562
[37] Battiston F, Cencetti G, Iacopini I, Latora V, Lucas M, Patania A, Young J G and Petri G 2020 *Phys. Rep.* **874** 1
[38] Aksoy S G, Joslyn C, Marrero C O, Praggastis B and Purvine E 2020 *EPJ. Data. Sci.* **9** 16
[39] Dode A and Hasani S 2017 *IOSR J. Comput.* **19** 01
[40] Huling J D, Greifer N and Chen G 2024 *J. Am. Stat. Assoc* **119** 1657
[41] Ren X L, Gleinig N, Helbing D and Antulov-Fantulin N 2019 *Proc. Natl. Acad. Sci. USA* **116** 6554
[42] Lu P, Lan J, Tang J, Zhang L, Song S and Zhu H 2023 *Chin. Phys. B* **33** 018901
[43] Lu P, Zhong Y and Yang P 2023 *Chin. Phys. B* **33** 018903
[44] Yan Z, Xia Y, Guo L, Zhu L, Liang Y and Tu H 2023 *Chin. Phys. B* **32** 068901
[45] Ye Q, Yan G, Chang W and Luo H 2023 *Eur. Phys. J. B* **96** 15
[46] Ullah A, Wang B, Sheng J, Long J, Khan N and Sun Z 2021 *Expert Syst. Appl* **186** 115778
[47] Da Ai *et al.* 2023 *Chin. Phys. B* **32** 118902
[48] Song J H 2023 *Chin. Phys. B* **32** 098901