Physica A 506 (2018) 802–815

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Power iteration ranking via hybrid diffusion for vital nodes identification

Tao Wu [a,*], Xingping Xian [b,*], Linfeng Zhong [c,e], Xi Xiong [d], H. Eugene Stanley [e]

[a] School of Cyber Security and Information Law, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[b] College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[c] Department of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
[d] School of Cybersecurity, Chengdu University of Information Technology, Chengdu 610225, China
[e] Center for Polymer Studies and Department of Physics, Boston University, Boston, MA 02215, USA

HIGHLIGHTS

- Combine mass diffusion and heat conduction process for node ranking.
- Design a nonlinear hybrid mechanism for node state updating.
- The proposed PIRank method can capture different structural features.
- PIRank is immune to the localization transition of leading eigenvector.

ARTICLE INFO

Article history:
Received 17 November 2017
Received in revised form 8 March 2018
Available online 4 May 2018

Keywords:
Node ranking
Centrality measure
Physical process
Network disruption

ABSTRACT

One of the most interesting challenges in network science is to understand the relation between network structure and dynamics on it, and many topological properties, including degree distribution, community strength and clustering coefficient, have been proposed in the last decade. Prominent in this context is the centrality measures, which aim at quantifying the relative importance of individual nodes in the overall topology with regard to network organization and function. However, most of the previous centrality measures have been proposed based on different concepts and each of them focuses on a specific structural feature of networks. Thus, the straightforward and standard methods may lead to some bias against node importance measure. In this paper, we introduce two physical processes with potential complementarity between them. Then we propose to combine them as an elegant integration with the classic eigenvector centrality framework to improve the accuracy of node ranking. To test the produced power iteration ranking (PIRank) algorithm, we apply it to the selection of attack targets in network optimal attack problem. Extensive experimental results on synthetic networks and real-world networks suggest that the proposed centrality performs better than other well-known measures. Moreover, comparing with the eigenvector centrality, the PIRank algorithm can achieve about thirty percent performance improvement while keeping similar running time. Our experiment on random networks also shows that PIRank algorithm can avoid the localization phenomenon of eigenvector centrality, in particular for the networks with high-degree hubs.

[*] Corresponding authors.
E-mail addresses: wutaoadeny@gmail.com (T. Wu), xxp0213@gmail.com (X. Xian).

https://doi.org/10.1016/j.physa.2018.04.012
0378-4371/© 2018 Elsevier B.V. All rights reserved.

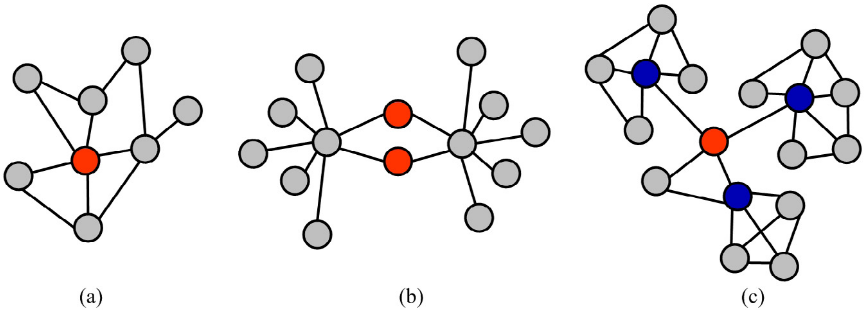


Fig. 1. Illustration of vital nodes with different structural features. (a) Local dominant node. (b) Intermediary nodes. (c) Network including vital nodes with complex structural features. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

1. Introduction

Networked data have become ubiquitous, and networks describing physical systems, protein interactions, computer communications, and people relationships are all becoming increasingly important in our day-to-day life. Complex network analysis has proven to be a successful tool for modeling and mining enormous networked data. Because of the heterogeneous of complex networks, some network nodes are more important to network function than others. Measuring the relative importance of individual nodes is important in both theoretical research and practical application. For example, identifying and protecting the crucial elements of the Internet so that the functioning of the system can be maintained, vaccinating influential individuals in contact networks so that the spread of an epidemic can be suppressed, and identifying and removing key vertices in a molecular network so that the bacteria can be eliminated. This context raises a fundamental question: given the data, how should one construct the node centrality measure such that it can capture structural features effectively?

In the last decade, node ranking problem has been particularly proposed to measure the importance of nodes within a given network [1], and a variety of centrality measures have been suggested based on different interpretations, including neighbor-based local centralities (degree centrality, local centrality [2], and collective influence [3]), location-based global centralities (closeness centrality [4], betweens centrality [5], and *k*-shell centrality [6,7]), and path counting centralities (eigenvector centrality [8], Katz's centrality [9], PageRank [10], and RA centrality [11,12]). Because of the variety of networked systems, the vital nodes in every task are concretized from different perspective. For example, the vital nodes in influence maximization problem are the influential sources that spread information [13], and the vital nodes in network disruption problem are the intermediate nodes that maintain network integrity [14]. Specifically, influence maximization problem has attracted attention recently [15–17], and many single node ranking methods [18–20] and multiple spreaders identification methods [21–23] been proposed for the problem. Meanwhile, researchers begin to be concerned about node ranking method for temporal networks [24]. Therefore, vital nodes may mean different things in various applications, and there is no general consensus on the definition. This paper focus on the fundamental problem, i.e. vital nodes identification based on network structure analysis, and explore the method to optimize the ranking accuracy. In face of such a problem, most of the existing centrality measures focus on a specific structural feature and have limits in node ranking. For example, the degree correlated local centralities (degree centrality, local centrality), global centralities (closeness centrality, betweens centrality) and random walk based centralities (PageRank, LeaderRank [25]) are all towards to high degree nodes. However, researchers have unveiled that low degree nodes always be very important and meaningful in many complex systems [26–28]. Therefore, to identify vital nodes accurately, centrality measures should be optimized to capture network nodes' structural information as comprehensively as possible. To illustrate this idea, Fig. 1 gives an example of vital nodes with different structural features. In detail, the local dominant node with red color in Fig. 1(a) preserves connections with most of the network nodes, the intermediary nodes with red color in Fig. 1(b) maintain the network's integrality and have more control on communications between network components, and in Fig. 1(c) the nodes with blue color have features of local dominant node and the nodes with red color have features of local dominant node and intermediary node simultaneously. It is easy to see that the nodes with colors play critical role in network structure and accurate node ranking should consider all the strucutral features.

As representative of the class of spectral centralities, eigenvector centrality measures the importance of a node based on the influence of its neighbors. High-influence neighbors contribute more to central node's influence than low-influence ones, and a node is influential if it has many influential neighbors. Eigenvector centrality calculates based only on local information in each step but can utilize global network information through successive iteration. It assigns a relative importance score v_i for node i that is proportional to the sum of the scores of the neighbors of node i. Mathematically, this can be written as $v_i = \lambda^{-1}\sum_j A_{ij}v_j$, where λ is a constant of proportionality and A_{ij} is an element of the adjacency matrix $\mathbf{A}$ of a network having value one if there is an edge between node i and j and zero otherwise. In the matrix form, we have $\mathbf{Av} = \lambda\mathbf{v}$, which means that the vector $\mathbf{v}$ of centralities v_i is an eigenvector of the adjacency matrix $\mathbf{A}$. Because centralities are all nonnegative, the Perron–Frobenius theorem [29] guarantees that the vector $\mathbf{v}$ of centralities v_i must be the leading eigenvector of non-negative real square matrix $\mathbf{A}$. Meanwhile, physical diffusion process mass diffusion (MD) and heat conduction (HC) have been proposed to design diffusion-based recommendation algorithms in user–object bipartite networks recently [30–33]. The MD process works by equally distributing the resource to the nearest neighbors, while the HC process redistributes the resource via a nearest-neighbor averaging process [32]. The standard MD process and HC processes are in fact random walk processes on bipartite network, in which MD process always achieve high accuracy but low diversity while HC process has high diversity but low accuracy.

In this paper, we aim to address the vital nodes identification problem by proposing a sophisticated and efficient centrality to capture structural information comprehensively. Inspired by the application of the above physical processes in the field of recommendation [31], we melt the MD and HC processes into the framework of eigenvector centrality, and propose a power iteration ranking (PIRank) algorithm. The general idea behind this is that the physical processes have similar iterative updating mechanism with the eigenvector centrality, and the hybrid of the physical processes with different preference has a potential to identify vital nodes accurately. To integrate the MD and HC processes, a nonlinear hybridization mechanism is defined in PIRank. By analyzing the characteristics of top vital nodes and the node ranking performance under varying hybrid parameter settings, we explore the rationality of the proposed nonlinear hybridization mechanism. To evaluate the performance of the proposed PIRank algorithm, we apply it in synthetic networks and real-world networks, experimental results show that our method outperforms other state-of-the-art methods in network optimal attack problem. In addition to this, our experiment shows that PIRank algorithm can avoid the localization phenomenon of eigenvector centrality in networks with high-degree hub nodes. Except the proposed algorithm, our work provides a optimizing paradigm for the centrality measures akin to iterative updating process, and demonstrates that they may can be improved further by consider more structural features.

The remainder of this paper is organized as follows. Section 2 of this paper introduces the preliminaries of our work, including standard centralities, and standard MD and HC processes. Section 3 describes our hybrid updating-mechanism-based PIRank algorithm for node ranking. Section 4 presents an empirical evaluation of the proposed algorithm. Section 5 presents a discussion and our conclusions.

2. Preliminaries

2.1. Standard centrality measures

Let $A = \{a_{ij}\} \in R^{N,N}$ as the adjacency matrix of network G where $a_{ij} = 1$ if node i is connected with node j and $a_{ij} = 0$ otherwise. We also use notion $\Gamma(i)$ to denote the set of neighbors of node i, $\Gamma(i) = \{j \in V : a_{ij} = 1\}$. Here we provide the basic definitions of the centrality measures that will be used in the rest of the paper. For more details, we refer readers to the literature cited.

Degree centrality. Degree measures the network scope on which a node can have direct impact. The degree centrality of node i can be calculated as

$$C_{DC}(i) = \sum_{j=1}^{N} a_{ij} = |\Gamma(i)| \tag{1}$$

The computational complexity of degree centrality is $O(N)$.

Closeness centrality. The closeness centrality [34] of node i is defined as the sum of the length of the shortest paths between the node and all other nodes in the network. The more central a node is, the closer it is to all other nodes. To compare the nodes of networks of different sizes, the sum of the length of the shortest paths is usually normalized by the size of the network, and represents the average length of the shortest paths. Thus, the closeness centrality of node i is given by

$$C_{CC}(i) = \frac{N-1}{\sum_{j=1}^{N} d_{ij}} \tag{2}$$

where d_{ij} is the length of the shortest paths between the node i and j. The computational complexity of closeness centrality is $O(N^3)$.

K-shell centrality. k-shell centrality [35] is obtained by employing k-shell decomposition on the network. The k-shell decomposition removes all nodes with degree equals to 1 firstly. This causes new nodes with degree $k \leq 1$ to appear. The nodes are also removed and the process is continued until all remaining nodes are of degree $k > 1$. All the removed nodes and the links between them form the k-shell with index $k_s = 1$. In a similar fashion, we continue the process until all higher-layer shells are identified and subsequently with higher values of k_s. The computational complexity of k-shell centrality is $O(E)$.

Eigenvector centrality. Eigenvector centrality [8] computes the centrality of a node based on the centrality of its neighbors. The eigenvector centrality of node i is defined as

$$C_{EC}(i) = \frac{1}{\lambda} \sum_{j \in V} a_{ij} C_{EC}(j) \tag{3}$$

where λ is a constant. In the matrix form, $\mathbf{Av} = \lambda \mathbf{v}$, the vector of centralities is the leading eigenvector of the adjacency matrix. The computational complexity of eigenvector centrality is $O(N^2)$.

Collective Influence centrality. CI centrality [36] considers the degree of the target node and its neighbors that are a certain number of steps away from it. The CI centrality of node i is defined as

$$CI_{\ell}(i) = (k_i - 1) \sum_{j \in \partial Ball(i,\ell)} (k_j - 1) \tag{4}$$

where k_i is the degree of node i, and $\partial Ball(i, \ell)$ denotes the set of the nodes that are at distance ℓ to node i. The computational complexity of CI centrality is $O(N\langle k\rangle^{\ell})$.

Enhanced Collective Influence centrality (ECI). ECI centrality [14] is the enhancement of CI centrality by considering more local structure information. The ECI centrality of node i is defined as:

$$ECI_{\ell}(i) = \frac{H(i)}{1 + \sum_{x,y \in \partial Ball(i,\mu), \mu \le \ell} \frac{|\Gamma(x) \bigcap \Gamma(y)|}{|\Gamma(x) \bigcup \Gamma(y)|}} \cdot CI_{\ell}(i) \tag{5}$$

where $\frac{|\Gamma(x) \bigcap \Gamma(y)|}{|\Gamma(x) \bigcup \Gamma(y)|}$ is the connection intensity of node x and y, $H(i)$ is the normalized information entropy to characterize the neighbor diversity of node i.

2.2. Standard MD and HC processes

An undirected network system can be represented as $G(V, E)$, where V is the set of nodes, E the set of edges, r_i the resource of node i, $i \in V$, and Γ_i the set of its neighbors. Standard MD process works by assigning nodes an initial level of resource denoted by $r_i = r_i(0)$. At each iteration each node's resource is then redistributed evenly to its neighbors. In this redistribution procedure, the updated resource of each node is equal to the sum of contributions from its neighbors,

$$r_i(t+1) = \sum_{j \in \Gamma_i} \frac{r_j(t)}{d_j}. \tag{6}$$

where d_j is the degree of node j, $r_j(t)$ is node j's resource level at time t. Through the iterations of the redistribution procedure, the resources of all network nodes tend to be constant. The process converges and reaches equilibrium when the resources no longer change. The resource preserved by a node is proportional to the probability that random walkers released at the other nodes happen to arrive at this node in iterative updating. The ranking list of network nodes is generated by ranking all nodes in decreasing order according to their final resource.

Similar to MD process, HC process also redistributes resources in a manner akin to random-walk process. Network nodes have different initial temperature, and heat transfers from high temperature nodes to low temperature neighbors. The difference is that HC process updates nodes' states via a nearest-neighbor averaging procedure in which the sum of the temperatures of their neighbors is divided by their degree, while MD process works by equally distributing the resource to the nearest neighbors. Mathematically,

$$r_i(t+1) = \frac{1}{d_i} \sum_{j \in \Gamma_i} r_j(t). \tag{7}$$

Through the iterations of the averaging procedure, the gaps between high temperature nodes and low temperature nodes are narrowed. Note that the total heat in standard HC process is variable, and the total amount of resources in standard MD process keeps constant. For MD process, the resource will be evenly distributed to its neighbors. On the contrary, in HC process the resource is redistributed via an averaging procedure, with nodes receiving a level of resource being equal to the mean amount possessed by their neighboring nodes (see Fig. 2).

3. Power iteration node ranking (PIRank) algorithm

According to the definition in Section 2.2, with standard MD process, the resource of every node is not solely determined by neighbor nodes' resource level, but also determined by neighbor nodes' number and their degree. The contribution of high degree neighbors to the central node may be little even through they have very high resource level. Therefore, MD process prefers local dominant nodes with low degree neighbors. With standard HC process, the resource account of every node is the average value of neighbor nodes' resource level and is unrelated to neighbor nodes' number and degree. So standard HC process prefers intermediary nodes bridging resource-rich neighbors. Thus, there is a potential complementarity between standard MD and HC processes and combining them may enable us to identify vital nodes with different structural features.

In addition, as the preferences of standard MD and HC processes to nodes with different degree, in order to regulate the preferences' influence on the nodes' steady state, iterative updating operations should be normalized by a function of node degree. Take Karate network [37] as an example, we apply HC process with randomized initialization on it to investigate the problem empirically. And to the iterative updating operations of the processes, we make a preliminary attempt through dividing nodes' state by nodes' degree and the square root of nodes' degree respectively, and the results are shown in Fig. 3. Comparing the ranking result in Fig. 3(a) and the network topology in Fig. 3(c), we can find that the ranking result is not consistence with the ground truth. The reason behind the result is that important nodes temperature may become very low

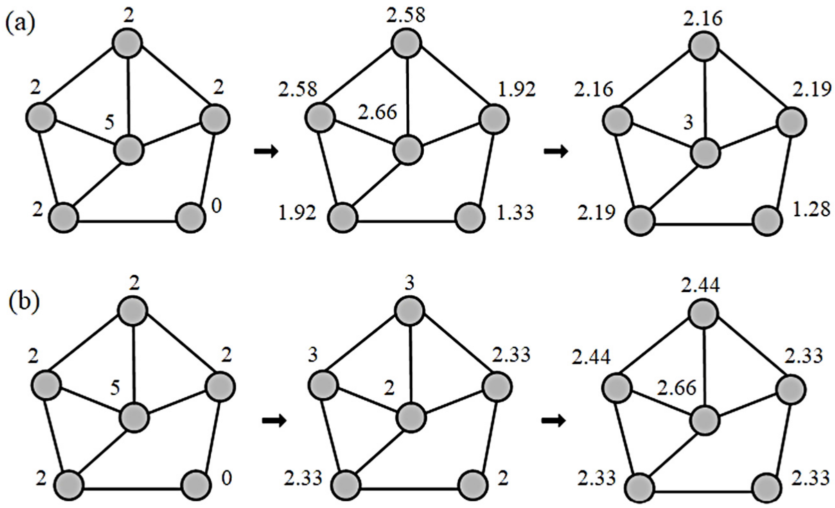


Fig. 2. Illustrating the updating mechanism of standard mass diffusion and heat conduction process. (a) The balanced redistribution procedure of standard MD process. (b) The averaging procedure of standard HC process. The network has $N = 6$ nodes and $M = 9$ links. The networks in the left panel of (a) and (b) are initialized randomly, the networks in the middle panel of (a) and (b) are the results after one iterative updating, and the networks in the right panel of (a) and (b) are the results after two iterative updating. The final results of the iterative updating can be used for node ranking.

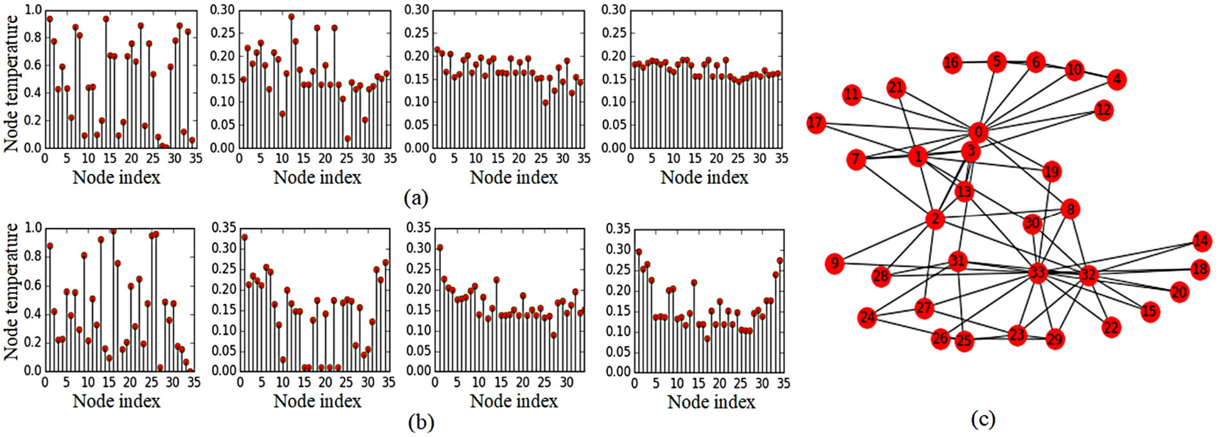


Fig. 3. Illustration of the iterative updating of HC process on Karate network. (a) The iterative updating of HC process from step one to five with nodes' degree normalization. (b) The iterative updating of HC process from step one to five with the square root of nodes' degree normalization. (c) Topology of Karate club network.

because of their high degree, even if they get lots of heat, while trivial nodes may obtain high temperature because of their small degree and thus be put in the top positions in ranking list. Rather than Fig. 3(a), the result in Fig. 3(b) instead is close to the ground truth about Karate network. Therefore, the HC process based node ranking can be improved by controlling the influence of node degree, and a adjustable regularization mechanism is necessary for accurate node ranking. To the MD process, there is similar results with HC process.

3.1. Hybrid updating mechanism

Because of the preferences about vital node and the role of node degree in the updating rule of standard MD and HC processes, now the goal is to construct a hybrid updating mechanism and regulate how node degree affects node importance. Typically, the hybrid combination of the physical processes should provide a smooth transition between them. And by tuning the hybridization parameter appropriately, the hybrid updating mechanism gets a good performance for the identification of vital nodes with specified structural feature. Define $N = |V|$, $M = |E|$ and adjacent matrix $\mathbf{A} = \{a_{ij}\} \in R^{N,N}$ in which $a_{ij} = 1 (i \neq j)$ if node i and j are connected, and $a_{ij} = 0$ otherwise, the iterative updating operation on network nodes can be formulated as:

$$\mathbf{R}(t+1)=\mathbf{W}\cdot\mathbf{R}(t)=\begin{bmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{bmatrix}\begin{bmatrix} r_1(t) \\ \vdots \\ r_n(t) \end{bmatrix}, \tag{8}$$

where $\mathbf{R}(t)$ is a $\mathbf{N}$-dimensional vector containing the ranking scores held by all nodes at step t, $\mathbf{W}$ is the transition matrix, and $\mathbf{R}(t+1)$ is the scores held by all nodes at step $t+1$. This formulation defines the updating mechanism of eigenvector centrality when $\mathbf{W}$ equals the adjacent matrix $\mathbf{A}$. As MD process evenly distributes node's resource to its neighbors, the element w_{ij}^M of the transition matrix $\mathbf{W}^M$ corresponding to MD process can be defined as

$$w_{ij}^M=\frac{a_{ij}}{d_j}, \tag{9}$$

where d_j is the degree of node j. HC process updates node's state by a nearest-neighbor averaging process, thus the element w_{ij}^H of the transition matrix $\mathbf{W}^H$ corresponding to HC process can be defined as

$$w_{ij}^H=\frac{a_{ij}}{d_i}, \tag{10}$$

To concretize the hybrid updating mechanism and provide a smooth transition between the two physical processes, we combine MD and HC processes through a nonlinear hybridization and incorporate hybridization parameter λ into transition matrix normalization. Mathematically, the transition matrix $\mathbf{W}^{M+H}$ of the hybrid updating mechanism is defined as

$$w_{ij}^{M+H}=\frac{a_{ij}}{d_i^{\lambda}\cdot d_j^{1-\lambda}}, \tag{11}$$

Clearly, it will degenerate to MD algorithm when $\lambda=0$ and degenerate to HC algorithm when $\lambda=1$. Based on transition matrix $\mathbf{W}^{M+H}$, the iterative updating operation in Eq. (3) can be reformulated in matrix formulation as

$$\mathbf{R}(t+1)=\mathbf{W}^{M+H}\cdot\mathbf{R}(t) \tag{12}$$

where $\mathbf{W}^{M+H}=\{w_{ij}^{M+H}\}\in R^{N,N}$. Then the total resource node i received will be:

$$r_i(t+1)=\sum_{j=1}^{N}\frac{a_{ij}}{d_i^{\lambda}\cdot d_j^{1-\lambda}}\cdot r_j(t)=\sum_{j\in\Gamma(i)}\frac{r_j(t)}{d_i^{\lambda}\cdot d_j^{1-\lambda}}, \tag{13}$$

where Γ_i is the set of neighbors of node i. Thus we can find that the resource account of node i is the sum of the resource of its neighbors weighted by the inverse of the degree nonlinear hybridization of node i and its neighbors, and the resource redistributed by a node to its neighbors is proportional to the probability $p_{ij}=1/d_i^{\lambda}\cdot d_j^{1-\lambda}$ when there exist an edge between them. Based on probability matrix $\mathbf{P}=\{p_{ij}\}\in R^{N,N}$, the updating matrix $\mathbf{W}^{M+H}$ can be rewritten as $\mathbf{W}^{M+H}=\mathbf{A}\odot\mathbf{P}$, and the iterative hybrid updating operation can be reformulated further as $\mathbf{R}(t+1)=\mathbf{A}\odot\mathbf{P}\cdot\mathbf{R}(t)$.

3.2. PIRank algorithm

It is noteworthy that the proposed hybrid updating mechanism provides the opportunity that the resource is prevented to be centralized on a few high degree nodes and the low degree nodes become more significant than high degree nodes. Thus, the hybrid updating mechanism based centrality measure can provide diversity while pursuing accuracy about vital network nodes. Moreover, we can find that the eigenvector centrality is a special form of iterative updating algorithm, thus the proposed hybrid updating mechanism can utilize its framework and solution method. Therefore, we propose a power iteration node ranking (PIRank) algorithm based on classic eigenvector centrality. As power iteration is the most basic method of computing eigenvalue and eigenvector [38], here we adopt power iteration for problem solving. Meanwhile, the same way for PIRank solving with eigenvector centrality would better reveal the principle causes behind the performance difference between them.

In PIRank algorithm, the nodes are randomly initialized, the hybrid updating mechanism is used to update nodes' ranking score, and the steady states of network nodes are used for node ranking. The Pseudo code of PIRank algorithm is given in Algorithm 1, and there are seven steps in the algorithm, where the network nodes are initialized with randomized scores in line 1. In line 2, the influence strength between arbitrary node pairs is calculated based on adjacency relation and node degree, and constructing updating matrix using the influence strength. In line 3, updating network nodes' ranking score based on power iteration method. In line 4, calculating the difference between successive ranking score vectors. In line 5, updating the current value of the counter. In line 6, determining whether the algorithm converges based on the differences between successive score vectors. Finally, the returned steady ranking scores are used for vital nodes identification.

An effective node ranking algorithm only need to run updating rule to get correct ranking list rather than final convergence. We thus set the stopping point of PIRank at $\delta^{t+1}\approx\delta^{t}$ and not at $\mathbf{R}(t+1)\approx\mathbf{R}(t)$. The algorithm converges

Algorithm 1 PIRank algorithm

```
Input: Network $G = (V, E)$, and adjacent matrix $A$.
Output: Nodes' ranking score.
Initialize. Choose an initial vector $\mathbf{R}(0)$ that contains the centrality scores of all nodes, and $t \leftarrow 0$.
Determine the updating matrix $\mathbf{W} = \{w_{ij}\} \in R^{N,N}$ in which $w_{ij} = \frac{a_{ij}}{d_i^{\lambda} \cdot d_j^{1-\lambda}}$, i.e., $\mathbf{W} = \mathbf{A} \odot \mathbf{P}$.
Update. $\mathbf{R}(t+1) \leftarrow \frac{\mathbf{W} \cdot \mathbf{R}(t)}{\|\mathbf{W} \cdot \mathbf{R}(t)\|_2}$.
Calculate the error. $\delta^{t+1} \leftarrow |\mathbf{R}(t+1) - \mathbf{R}(t)|$.
Increment $t$. $t + 1 \leftarrow t$.
Repeat 3, 4, and 5 until $\left|\delta^{t+1} - \delta^{t}\right|$ approach zero.
Return the ranking score vector $\mathbf{R}(t+1)$.
```

when the dominant and the subdominant eigenvalues of **W**, respectively denoted by λ' and λ'', differ in absolute value and $\left|\lambda'\right| > \left|\lambda''\right|$. The rate of convergence is proportional to $\left|\lambda'/\lambda''\right|$. When the subdominant eigenvalue is smaller than the dominant eigenvalue, there is rapid convergence. In practice, convergence usually occurs in fewer than ten iterations, and always more than 90% nodes reach their steady state after five iterations. The power iteration solution requires that the transition matrix **W** be symmetric, but it always converges even when there are poorly conditioned matrices.

The total computational time includes three parts as follows: the time of initializing ranking scores, the time of calculating updating matrix, and the time of updating the ranking scores. For the first part, the time of initializing ranking scores is $O(n)$. For the second part, the computational complexity is $O(\langle k \rangle n) = O(m)$ where $\langle k \rangle$ is the average degree of network and m is the number of edges. For the third part, the computational complexity mainly comes from matrix multiplication, and the complexity of every updating operation is $O(n^3)$ in native solving. However, because of real-world networks always are large scale and sparse, the updating operation can be realized along the interactions between nodes in the actual coding process. Thus, the complexity of updating operation is $O(\langle k \rangle n) = O(m)$. Because of the iteration number before convergence is small, the computational complexity of this step can be approximated as $O(m)$. So, the total computational complexity of this algorithm is $O(n) + O(m)$.

4. Models and materials

Optimal targeted attack model. As the largest connected part of networks is the basis of network function, the optimal targeted attack problem studies how to select nodes and remove them continuously to disrupt network structure as thoroughly as possible. Consider a network $G = (V, E)$, with a set of $N = |V|$ nodes tied together by $M = |E|$ links. Let G_q is the network that results from removing a fraction q of the nodes based on a specified centrality measure. The key quantity that we will study here is the relative size of the giant component G_q^c of G_q to the initial network G, which is denoted by $G(q)$. Thus, the optimal targeted attack problem is finding the minimum number of nodes q to be removed such that $G(q)$ is minimized. Moreover, the number of components $C(q)$ of network G_q is also an effective metric. To quantify the performance of centralities in optimal attack targets identification, this paper removes nodes sequentially based on different centralities and compare the resulted $G(q)$ and $C(q)$. To decrease complexity, a tiny fraction f of nodes (instead of a single node) are removed in each step.

Data Description. The synthetic networks include networks generated by the Erdos–Renyi random graph model [39], the Barabasi–Albert scale-free network model [40] and the Lancichinetti–Fortunato–Radicchi (LFR) network model [41]. The four representative real-world networks include: (1) Erdos network [42]. The network is a scientific collaboration network in which each node represents a scientist and each edge represents the cooperative connection between each pair of scientists. (2) Polblogs network [43]. This network represents the interactions between political blogs during the 2004 US presidential election. (3) Protein network [44]. This network is an undirected network of protein interactions contained in yeast. A node represents a protein and an edge represents a metabolic interaction between two proteins. (4) Router network [45]. This network represents the Internet at the router level, in which each node is a router and each edge a connection between two routers. The all networks are turned into undirected and simple connected networks for performance evaluation. The statistical properties of the networks, including the number of nodes N, the number of edges M, the average degree $\langle k \rangle$, the maximum node degree $k_{\max}$, the clustering coefficient C, the assortative coefficient r and the degree heterogeneity $H = \frac{\langle k^2 \rangle}{\langle k \rangle^2}$ [46] are summarized in Table 1.

It needs to be emphasized that the PIRank method is not proposed to solve certain specific problem. To test the proposed PIRank algorithm, here we take the optimal network targeted attack problem as an example. We apply PIRank method to identify the target nodes and remove them to disrupt network structure [47–49]. Especially, for any network under consideration, centrality measures are calculated for all nodes to determine nodes importance, and the nodes are removed in order of the centrality measure, from highest to lowest. Then the effect of removing a given number of the nodes on the giant component's size is analyzed to evaluate the centrality. Moreover, as outlier nodes have no effect on networks' robustness, all networks are preprocessed by deleting their outlier nodes. To evaluate the performance of PIRank method in the above problems, we choose five classic centrality measures as baselines, including degree centrality, closeness centrality, k-shell

Table 1
The basic topological features of the networks studied in this work. Structural properties include node number N, edge number M, average degree $\langle k\rangle$, maximum node degree $k_{\max}$, clustering coefficient C, assortative coefficient r and degree heterogeneity $H = \frac{\langle k^2\rangle}{\langle k\rangle^2}$.

Networks	N	M	$\langle k\rangle$	k_{max}	C	r	H
ER	1000	3000	6.098	18	0.006	−0.001	1.155
SF	400	1975	9.875	72	0.081	−0.056	1.766
LFR_1	1000	2782	5.564	49	0.413	−0.275	2.475
LFR_2	1000	4769	9.538	50	0.344	−0.238	1.777
LFR_3	1000	5109	10.218	50	0.108	−0.165	1.772
Erdos	474	1639	8.036	51	0.357	0.185	2.021
Polbblogs	469	2106	8.981	125	0.331	−0.219	2.610
Protein	1511	5873	7.773	61	0.212	−0.073	2.106
Routers	2113	6632	8.493	107	0.407	−0.006	2.646

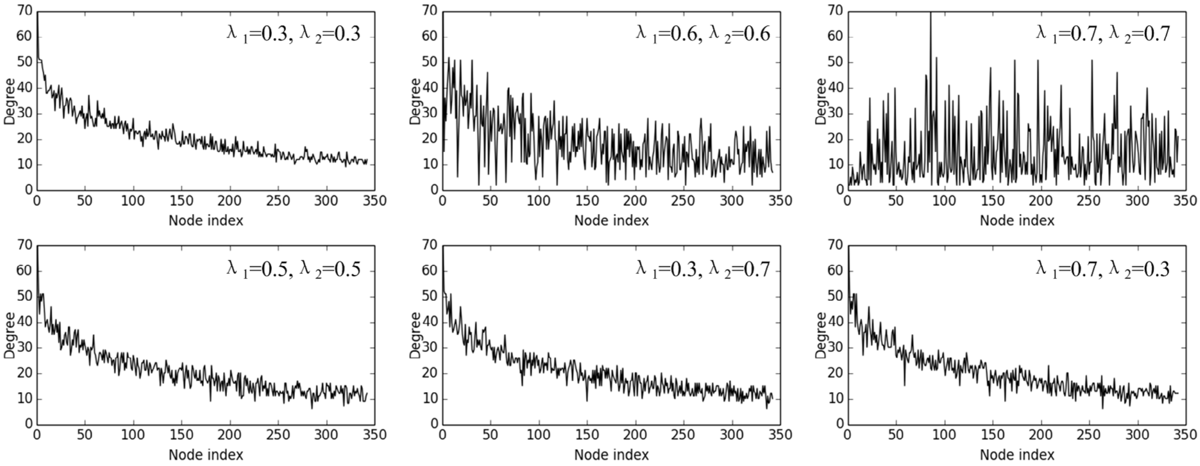


Fig. 4. Degree distribution of the top 350 vital nodes of Protein network resulted from PIRank algorithm with varied hybridization parameters.

centrality, ECI centrality and Eigenvector centrality. As ECI centrality is the enhancement of CI centrality and its advantage has already investigated in Ref. [14], CI centrality is not selected in this paper. For more details of these centrality measures, we refer the reader to the literature cited.

5. Experimental results

5.1. Parameter analysis

In this subsection, we prove the rationality of the proposed nonlinear hybrid mechanism by numerical analysis. Toward this end, the nonlinear hybridization mechanism defined by Eq. (11) is reformulated as $w_{ij}^{M+H} = \frac{a_{ij}}{d_i^{\lambda 1}\cdot d_j^{\lambda 2}}$, where λ_1 and λ_2 are the hybridization parameters. Firstly, two sets of experiments are performed with restricted condition λ_1+λ_2=1 and λ_1+$\lambda_2 \neq 1$ respectively, in which we investigate the degree distribution of the vital nodes with varying hybridization parameters λ_1 and λ_2. The results are shown in Fig. 4, and we can find that the ranking positions of network nodes are strongly and positively correlated with their degree when parameters λ_1 and λ_2 are low, e.g., $\lambda_1 = 0.3$ and $\lambda_2 = 0.3$. With the increase of their values, the node ranking position becomes independent of node degree gradually. In particular, when $\lambda_1 = 0.7$ and $\lambda_2 = 0.7$, they are completely unrelated. According to the definition of vital nodes [50], the above two cases, i.e. the strong correlation and the independence between node importance and node degree, are all counterintuitive. By contrast, from the results of experiments under $\lambda_1 + \lambda_2 = 1.0$, we find that there is a balance between the above two cases. This intermediate state between the two cases is consistent with the general cognition about vital nodes. Therefore, although the method for parameter value selection is not yet clear, the restricted condition λ_1+λ_2=1 of the nonlinear hybridization mechanism defined in Eq. (11) is rational.

As PIRank method is proposed based on eigenvector centrality, here we verify PIRank method's validity by quantifying the improvement of over eigenvector centrality. We apply the revised PIRank algorithm with updating matrix $w_{ij}^{M+H} = \frac{a_{ij}}{d_i^{\lambda 1}\cdot d_j^{\lambda 2}}$ into the optimal targeted attack problem, where λ_1 and λ_2 are the hybridization parameters. We calculate the sum R of the giant connected components' size G_q^c under varying number of attacked nodes, $R = \sum_q G_q^c$, then we measure the performance

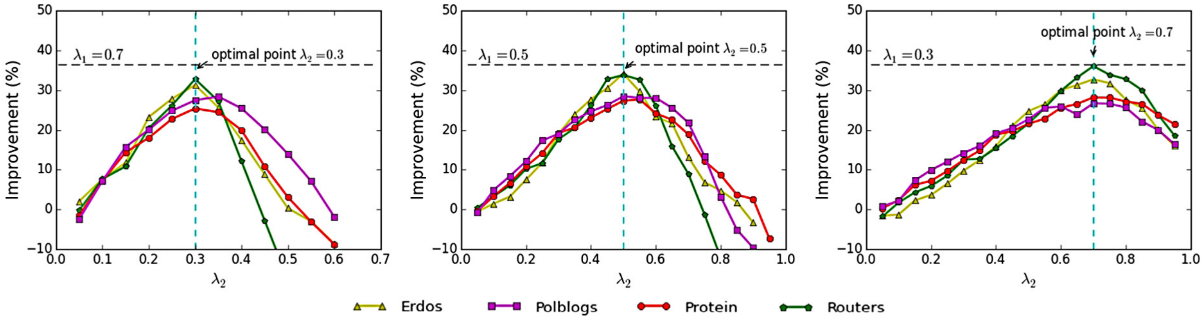


Fig. 5. (Color online) The improvement of PIRank over Eigenvector centrality for network attack under varied hybrid parameter.

improvement by $Imp = \frac{R_{PIRank} - R_{Eigen}}{R_{Eigen}}$, where R_{PIRank} and R_{Eigen} are the total size of the giant components under PIRank and eigenvector centrality guided attack process respectively. With varied hybridization parameter λ_1 and λ_2, the performance improvement of PIRank method over eigenvector centrality in the four real-world networks are shown in Fig. 5. From Fig. 5, we have three main findings. First, for any values of parameter λ_1, the optimum value of the improvement exists in all networks as long as the sum of parameter λ_1 and λ_2 is equal to 1. That is, the optimal values of parameter λ_2 approaches 0.3, 0.5 and 0.7 respectively when parameter λ_1 is equal to 0.7, 0.5, 0.3. Thus, the rationality of the restricted condition $\lambda_1+\lambda_2=1$ is proved once again. Second, we can find from Fig. 5 that the performance improvement of PIRank is greater than 30% in all the cases when the sum of parameter λ_1 and λ_2 is equal to 1. The results imply that the proposed PIRank method is effective and the nonlinear hybridization mechanism is reasonable. Third, we can find from Fig. 5 that the performance improvement of PIRank increases with the rise of hybridization parameter λ_2, which indicates that more satisfactory performance may be got if MD process is predominant in the nonlinear hybridization mechanism.

To further analyze the influence of hybridization parameter λ on PIRank's performance, we apply PIRank method with different parameter values for network optimal attack problem, in which parameter λ corresponds to parameter λ_1 in the last paragraph. The results are shown in Fig. 6, in which the size of the giant connected component $G(q)$ declines with network nodes being removed successively. We can find from Fig. 6 that the attack schemes guided by PIRank method with different parameter values lead to different disruption patterns in every network, but the performance of PIRank method for network attack problem generally deteriorates with the increase of parameter λ in all the networks. In all networks, the attack scheme guided by PIRank method with parameter $\lambda = 1.0$, i.e. standard HC process, has the worst performance in all the cases, followed by the cases with $\lambda = 0.8$ and $\lambda = 0.6$. Moreover, we find that the best performance are always arise in the cases when $\lambda = 0.0$, $\lambda = 0.2$ and $\lambda = 0.4$, and the difference between them is very small in most of the networks. This phenomena indicates that the combination of the MD and HC process is necessary for node ranking, and the good performance of PIRank algorithm comes from MD process' dominant position in the nonlinear hybridization mechanism.

5.2. Performance evaluation

According to the analysis about hybridization parameter in the last subsection, we fix $\lambda = 0.3$ in the forthcoming experiments. We first evaluate PIRank method's performance in synthetic networks, including Erdos–Renyi network, Scale-Free network and Lancichinetti–Fortunato–Radicchi (LFR) network. The Erdos–Renyi random graph model generates a ER network of N nodes and $M = (\langle k \rangle/2)N$ links by first selecting M different node pairs uniformly at random from the whole set of $N(N-1)/2$ candidate pairs and then adding a link between the chosen two nodes. Each node in the network has $\langle k \rangle$ attached links on average. In the first panel of Fig. 7, we report the size $G(q)$ of the giant connected component G_q^c of the ER network under varied number q of nodes being removed in decreasing rank order with respect to different centrality measures. From Fig. 7, it can be seen that PIRank centrality and ECI centrality outperform the degree centrality, closeness centrality, k-shell centrality and eigenvector centrality in the selection of the attacked nodes, and PIRank centrality has comparable performance with ECI centrality. We notice from the first panel of Fig. 7 that, during the attack processes guided by degree centrality, closeness centrality, k-shell centrality and eigenvector centrality, the size of the giant connected component decreases gradually and smoothly. In contrast, if the attacked nodes are selected based on PIRank centrality, although the giant connected component initially shrinks slowly, the giant component's size decreases sharply after about 400 nodes are removed. The result means that a smaller number of removed nodes are needed to break down the network in PIRank-guided attack scheme than that of the others, and PIRank method is more effective than them.

In order to further demonstrate the advantages of PIRank method, we compare the components number $C(q)$ of the ER network under varied number q of nodes being removed guided by the different centrality measures. In the first panel of Fig. 8, we can observe that the component number $C(q)$ of the ER network increases with the network nodes being removed, then $C(q)$ reaches maximum at the peak of the curves where the network is in a very fragile state. Then $C(q)$ slows down gradually with more nodes being removed until the giant component of the network disappears. Comparing the peaks of the

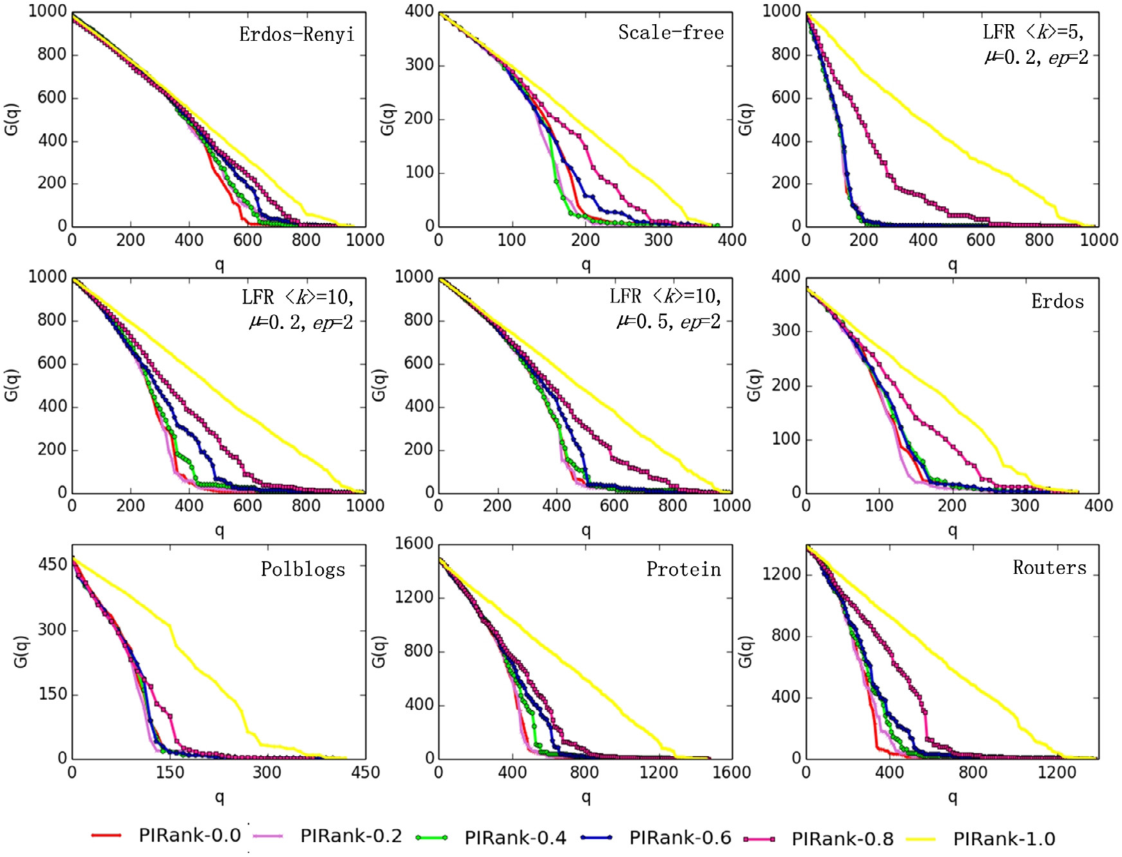


Fig. 6. (Color online) Sensitive analysis of PIRank to hybrid parameter λ. λ regulates the normalization corresponding to HC's averaging procedure, and $1 - \lambda$ regulates the normalization corresponding to MD's redistribution procedure.

curves, it can be seen that the attack schemes guided by PIRank centrality, ECI centrality and degree centrality disrupt the ER network faster and more thoroughly than that guided by closeness centrality, k-shell centrality and eigenvector centrality.

We then examine Scale-Free (SF) network, which is generated starting with a set of connected nodes. After that, new nodes with $m = 5$ edges are included in the network. The probability of the new node i to connect with a node j in the network is proportional to the degree of j, i.e., $p(i, j) = k_j / \sum_{\mu \in V} k_\mu$. The networks generated by this model present a power-law degree distribution, $p(k) = k^{-\gamma}$, where $\gamma = 3$. There are many highly connected (hub) nodes in SF network, and the degrees of them greatly exceed the mean node degree. To SF network, we find from Fig. 7 that PIRank-guided attack scheme is more efficient than the other attack schemes. In order to break down the SF network, the PIRank method only needs to delete about 200 nodes, while the closeness centrality, k-shell centrality and eigenvector centrality would need to delete about 300 nodes. In Fig. 8, we present SF network's component number $C(q)$ under varied number of nodes being removed. With the number of attacked nodes increases, the component number $C(q)$ corresponding to PIRank centrality grow much faster and get much greater values than the other centralities. As Fig. 8 shows the network attack process before networks' giant connected component disappearing, we can find that the giant connected component of the SF network under the PIRank guided attack scheme disappears much earlier than that of the others. So, PIRank centrality is more effective than the others for SF network disruption.

Community structure is one of the most important structural properties of complex networks. Here we use LFR benchmark to generate synthetic networks with build-in communities, with node number $N = 1000$, maximum degree $k_{max} = 50$ and the exponent for the degree distribution $ep = 2$. Then, we vary the average degree $\langle k \rangle$ and the mixing parameter μ to evaluate the performance of PIRank in these networks. We notice from Fig. 7 that, the network with smaller average node degree, i.e. $\langle k \rangle = 5$, is more sensitive to the network attack process and being disrupted more earlier than the other networks. The networks with $\langle k \rangle = 10$ are generally more robust to network attack. The size of the giant component under closeness centrality, k-shell centrality and eigenvector centrality guided attack schemes decreases slowly and almost linearly, and the number of the nodes that need to be removed to shrink the giant component sharply are all greater than 400. In the optimal targeted attack process of these networks, PIRank still is the best method among the centralities for attacked nodes selection. From the perspective of component number $C(q)$, with the average node degree $\langle k \rangle$ and the mixing

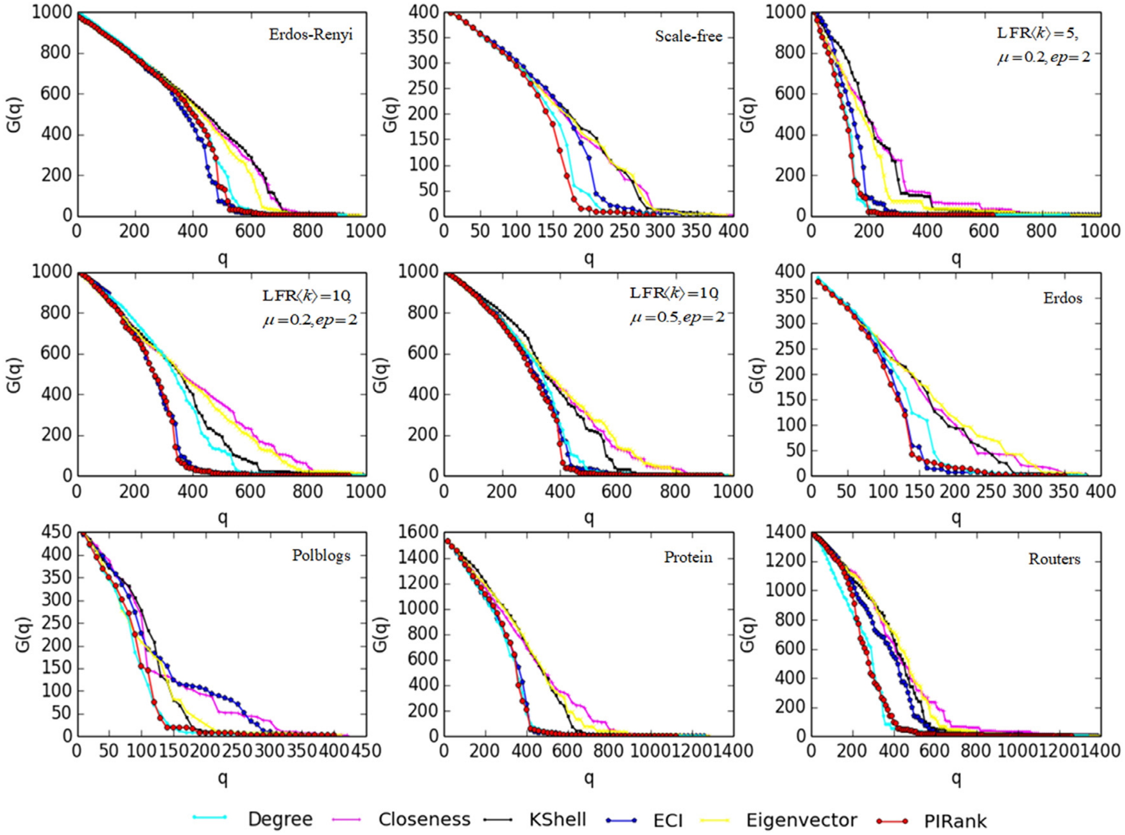


Fig. 7. (Color online) Giant component's size curves under varied number of removed nodes.

parameter μ increasing, the community networks becomes more robust with respective of maximum component number and its position, as shown in Fig. 8. According to the results in Fig. 8, we can conclude that PIRank-guided attack scheme has excellent performance in the community networks. In general, PIRank centrality obtains quantitatively the best results over all community networks for the network attack problem.

Next we evaluate the performance of PIRank method with baseline measures in four real-world networks, including Erdos network, Polblogs network, Protein network and Router network. The last four panels of Fig. 7 show the dynamics of the giant component's size $G(q)$ with varied number q of nodes being removed. The removal order of nodes is determined based on different centrality measures. It can be seen from the results that all the real-world networks exhibit similar disruption patterns with the synthetic networks. That is, the size of the networks' giant component all decrease gradually with the number of the removed nodes increases. After a specific point of the number of the nodes being removed, the giant connected component shrinks sharply with the deletion of a little additional nodes. According to Fig. 7, we can find that the size of the giant component $G(q)$ corresponding to PIRank method generally declines more rapidly than that of the other methods with the number of the removed nodes increases. Moreover, we compare the real-world networks' component number $C(q)$ resulted by the centralities, as shown in the last four panels of Fig. 8. The results show that the number of the components of the networks increase initially with more nodes being removed, and then they decrease after reaching the maximum value. The differences between the component number curves guided by different centralities are the height and the position of the peaks. To a specific network, the higher and the earlier of the curve peaks, the better the performance of the centrality measure. In this sense, PIRank method obtains admirable performance from the perspective of component number in the real-world networks. It is interesting to note that the points where the number of the components achieves the maximum value in Fig. 8 are consistent with the number of the removed nodes which lead the giant connected components disappear sharply in Fig. 7. After that, more nodes being removed just reduces the number of the components but has less influence on the size of the giant components of the networks. Generally speaking, the PIRank method obtains quantitatively satisfactory results in all these disruption cases, and PIRank-guided attack scheme allows attackers to disrupt networks more quickly and completely by deleting a smaller number of nodes than that of the others.

To further demonstrate the advantages of PIRank method over other centralities for the network optimal attack problem, we compare the number of the removed nodes when the component's number is maximized and the giant component disappears in every network attack process, as shown in Table 2. We can find that the number of the removed nodes needed

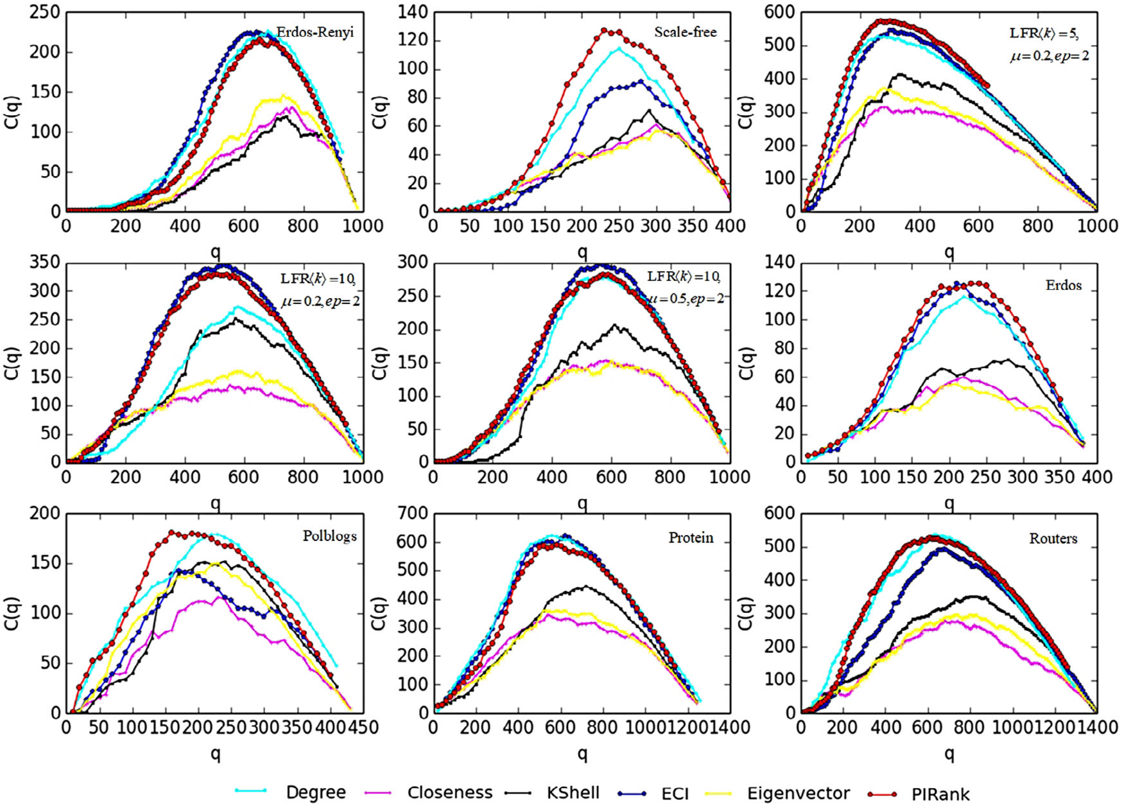


Fig. 8. (Color online) Component number curves under varied number of removed nodes.

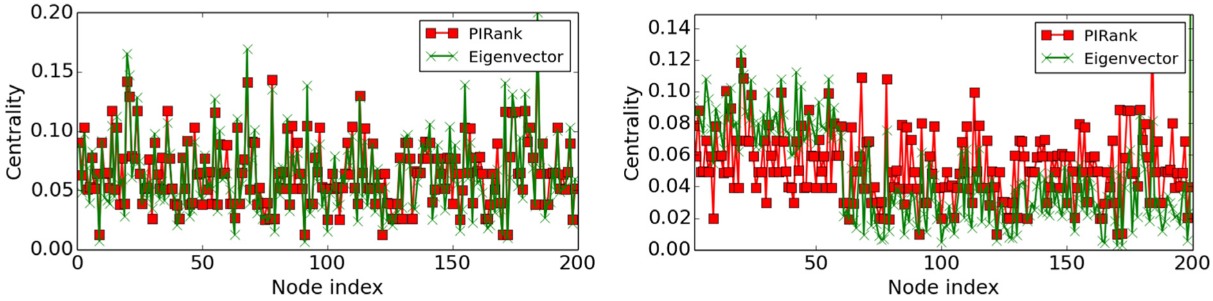
Table 2
The number of the removed nodes when the component's number is maximized and the giant component disappears. (In the network optimal attack problem, the targeted nodes are removed successively until the size of the network's giant connected component is reduced to one. The maximum values of the removed node number corresponding to component number peak and giant component disappearing are highlighted in bold.)

Algorithm	ER	SF	LFR1	LFR2	LFR3	Erdos	Polblogs	Protein	Routers
Degree	651/922	243/344	265/889	570/992	**563**/**971**	213/375	242/451	654/1506	624/1286
Closeness	747/969	286/393	254/997	538/989	573/990	203/380	239/467	**621**/**1481**	693/1390
K-shell	726/967	277/378	316/994	560/978	597/972	267/374	246/451	841/1486	814/1353
ECI	**596**/**920**	256/361	293/996	520/983	542/957	199/379	177/434	718/1458	645/1384
Eigenvector	717/972	294/390	263/995	558/996	588/989	**187**/**379**	241/466	665/1484	813/1386
PIRank	**641**/**882**	**218**/**289**	**291**/**626**	**492**/**933**	**577**/**954**	**223**/**341**	**168**/**437**	**671**/**1362**	**601**/**1255**

Table 3
Running time of centrality measures in all networks when the components number is maximized and the giant component disappears.

Algorithm	ER	SF	LFR1	LFR2	LFR3	Erdos	Polblogs	Protein	Routers
Degree	0.80/1.13	0.14/0.20	0.46/1.55	0.80/1.39	0.79/1.36	0.12/0.21	0.18/0.33	1.41/3.25	1.22/2.87
Closeness	2.53/3.54	0.35/0.63	0.33/0.41	3.26/5.03	3.10/4.96	0.28/1.04	0.50/0.99	5.57/11.7	4.46/10.2
K-shell	0.01/0.13	0.01/0.24	0.02/1.88	0.02/2.12	0.01/1.65	0.01/0.26	0.01/0.51	0.02/3.96	0.02/3.32
ECI	22.1/22.8	99.9/11.0	46.4/59.9	138/143	410/514	23.8/37.1	61.3/90.3	23.8/420	38.4/569
Eigenvector	0.95/2.38	0.25/0.42	2.48/2.99	1.40/2.82	0.92/2.88	0.18/0.48	0.28/0.92	2.20/6.39	1.62/5.36
PIRank	1.19/2.65	0.54/0.70	3.18/3.91	1.61/3.13	1.26/5.04	0.33/0.73	0.93/1.02	2.38/10.8	1.93/9.42

for the PIRank-guided network attack process is minimal in most of the networks. Thus, PIRank method is more effective for the network optimal attack problem than the other methods. Moreover, we also demonstrate the efficiency of PIRank method from the perspective of running time. The running time of the centrality measures on all the networks are shown in Table 3. Among all the measures, K-shell has the smallest computation time, but its performance is unsatisfactory for targeted nodes selection. ECI centrality takes a great deal of time although it performs better than closeness and K-shell centrality. PIRank method has the same amount of running time as closeness and eigenvector centrality in all the networks while has the best performance for the network optimal attack problem.



(a) Performance of centralities in random network.

(b) Performance of centralities in random network with a single hub node.

Fig. 9. (Color online) Localization and centrality in networks .

5.3. *Localization and centrality measures*

Although PIRank method has been proved superior to the classic eigenvector centrality, the backside causes are yet fully to be understood. Besides PIRank's ability of capturing different structural features, avoiding localization transition may be another explanation. Eigenvector centrality has been found to be defective, and the leading eigenvector of the adjacency matrix can undergo a localization transition in which most of the weight of the vector concentrates around one or a few nodes in the network and vanishing weight is assigned to the average node [51]. The work also shows that the fundamental cause of the localization phenomenon is the presence of "hubs" within networks, nodes of unusually high degree, which are a common occurrence in many real-world networks. We argue that the answer of the superiority of PIRank method over eigenvector centrality lies partly in the ability that avoids localization transition and gives useful results in regimes where the eigenvector centrality fails.

In order to prove the argument, here we consider the following simple undirected network model consisting of a random graph plus a single hub node. In a network of n nodes, $n-1$ of them form a random graph in which every pair of nodes is connected with probability $c/(n-2)$, where c is the average degree. The nth node is the hub and is connected to every other node with probability $d/(n-1)$, where d is the expected degree of the hub. According to the work [51], when $d > c(c+1)$, the hub eigenvalue becomes the leading eigenvalue and a non-vanishing fraction of the eigenvector centrality falls on the hub node and its neighbors.

We apply PIRank method and eigenvector centrality on the generated random network with $n = 200$, $c = 5$ and $d = 60$, and the resulted centrality of the network nodes are shown in Fig. 9. We can find from Fig. 9 that PIRank method and eigenvector centrality have similar performance in the random graph without the hub node while eigenvector centrality deteriorate clearly in the random graph with a hub node. From Fig. 9(b), we notice that there is a localization phenomenon arising in eigenvector centrality and the eigenvector values corresponding to the hub node with index 201 and its neighbors, i.e. the nodes with index from 1 to 60, are collectively higher than that of the other nodes. In contrast, the PIRank method is not obviously affected by the change of node degree distribution.

6. Conclusions and discussion

Node ranking in complex networks is an important and challenging problem. In this paper, we introduced physical processes for structural features capturing to improve the performance of node ranking. We showed that although the proposed PIRank method adopts the same iterative framework with the basal eigenvector centrality, the nonlinear hybrid updating mechanism defined based on the physical processes can help to improve the performance remarkably. Meanwhile, the PIRank method was proved to outperform the other centralities and renders a satisfactory performance in all experiment networks. We also prove the rationality of the proposed nonlinear hybrid updating mechanism of PIRank method by parameter analysis. We found with surprise that the redistribution mechanism stemmed from mass diffusion process plays a major role while the average procedure motivated by mass heat conduction process is complementary when PIRank gets good performance. We believe that the effect of structural feature integration may exist in many backgrounds [52–54], the hidden information revealed by structural feature integration can help to improve the accuracy of network structure mining algorithms.

Although structural features integration is useful for the optimization of node ranking, for real applications, how to identify complementary structural features and design the corresponding mechanisms is still an open and challenging problem. In addition, because real-world networks are always dynamic and multiple layers, it would be a worthwhile topic for future research to probe the ways of applying the proposed method in temporal and multiplex networks.

Acknowledgments

This work was supported by Talent Introduction Project of Chongqing University of Posts and Telecommunications, China (Grant No. A2017-131). We thank the anonymous reviewers for their thorough review and highly appreciate their useful comments and suggestions.

References

[1] G. Lawyer, Understanding the influence of all nodes in a network, Sci. Rep. 5 (2015) 8665.
[2] D. Chen, L.L., M.S. Shang, Y.C. Zhang, T. Zhou, Identifying influential nodes in complex networks, Physica A 391 (4) (2012) 1777–1787.
[3] F. Morone, H.A. Makse, Influence maximization in complex networks through optimal percolation, Nature 524 (7563) (2015) 65.
[4] G. Sabidussi, The centrality of a graph, Psychometrika 31 (1966) 4.
[5] L.C. Freeman, A set of measures of centrality based on betweeness, Soc. Netw. 1 (1979) 3.
[6] M. Kitsak, L.K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H.E. Stanley, H.A. Makse, Identifying influential spreaders in complex networks, Nat. Phys. 6 (2010) 11.
[7] Y. Liu, M. Tang, T. Zhou, Y. Do, Core-like groups result in invalidation of identifying super-spreader by k-shell decomposition, Sci. Rep. 5 (2015) 9602.
[8] B. Ruhnau, Eigenvector-centrality-a node-centrality?, Soc. Netw. 22 (2000) 4.
[9] L. Katz, A new status index derived from sociometric analysis, Psychometrika 18 (1953) 1.
[10] S. Brin, L. Page, The anatomy of a large scale hypertextual web search engine, Netw. Isdn Sys. 30 (1998) 1.
[11] L.F. Zhong, J.G. Liu, M.S. Shang, Iterative resource allocation based on propagation feature of node for identifying the influential nodes, Phys. Lett. A 379 (38) (2015) 2272–2276.
[12] Z.M. Ren, A. Zeng, D.B. Chen, et al., Iterative resource allocation for ranking spreaders in complex networks, Europhys. Lett. 106 (4) (2014) 48005.
[13] W. Wang, Q.H. Liu, S.M. Cai, M. Tang, L.A. Braunstein, H. Eugene Stanley, Suppressing disease spreading by using information diffusion on multiplex networks, Sci. Rep. 6 (2016) 29259.
[14] T. Wu, L.T. Chen, L.F. Zhong, X.P. Xian, Enhanced collective influence: A paradigm to optimize network disruption, Physica A 472 (2017) 43–52.
[15] Z.K. Zhang, C. Liu, X.X. Zhan, et al., Dynamics of information diffusion and its applications on complex networks, Phys. Rep. 651 (2016) 1–34.
[16] C. Liu, X.X. Zhan, Z.K. Zhang, et al., How events determine spreading patterns: information transmission via internal and external influences on social networks, New J. Phys. 17 (11) (2015) 113045.
[17] C. Liu, Z.K. Zhang, Information spreading on dynamic social networks, Commun. Nonlinear Sci. Numer. Simul. 19 (4) (2014) 896–904.
[18] M. Sikic, A. Lancic, N. Antulov-Fantulin, et al., Epidemic centrality is there an underestimated epidemic impact of network peripheral nodes, Eur. Phys. J. B 86 (10) (2013) 440.
[19] J.G. Liu, J.H. Lin, Q. Guo, et al., Locating influential nodes via dynamics-sensitive centrality, Sci. Rep. 6 (3) (2015) 032812.
[20] P. Bonacich, P. Lloyd, Eigenvector-like measures of centrality for asymmetric relations, Social Networks 23 (3) (2001) 191–201.
[21] X.Y. Zhao, B. Huang, M. Tang, et al., Identifying effective multiple spreaders by coloring complex networks, Europhys. Lett. 108 (6) (2014) 68005.
[22] J.G. Liu, Z.Y. Wang, Q. Guo, et al., Identifying multiple influential spreaders via local structural similarity, Europhys. Lett. 119 (1) (2017) 18001.
[23] Z.L. Hu, J.G. Liu, G.Y. Yang, et al., Effects of the distance among multiple spreaders on the spreading, Europhys. Lett. 106 (1) (2014) 18002.
[24] H. Kim, R. Anderson, Temporal node centrality in complex networks, Phys. Rev. E 85 (2) (2012) 026107.
[25] L.L., Y.C. Zhang, C.H. Yeung, T. Zhou, et al., Leaders in social networks, the delicious case, PLoS One 6 (2011) e21202.
[26] W. Hwang, Y.R. Cho, A. Zhang, M. Ramanathan, Bridging centrality: Identifying bridging nodes in scale free networks, in: KDD'06, pp. 4292–4290.
[27] T.W. Valente, K. Fujimoto, Bridging: Locating critical connectors in a network, Soc. Netw. 32 (2010) 3.
[28] T.C. Lou, J. Tang, Mining structural hole spanners through information diffusion in social networks, in: WWW'13, pp. 825–836.
[29] S. Wasserman, K. Faust, Social network analysis: social network analysis in the social and behavior sciences, in: Social Network Analysis, Cambridge University Press, Cambridge, 1994.
[30] Y.C. Zhang, M. Blattner, Y.K. Yu, Heat conduction process on community networks as a recommendation model, Phys. Rev. Lett. 99 (15) (2007) 154301.
[31] T. Zhou, Z. Kuscsik, J.G. Liu, M. Medo, J.R. Wakeling, Y.C. Zhang, Solving the apparent diversity-accuracy dilemma of recommender systems, Proc. Natl. Acad. Sci. USA 107 (2010) 10.
[32] J.G. Liu, T. Zhou, Q. Guo, Information filtering via biased heat conduction, Phys. Rev. E 84 (2011) 037101.
[33] Z.K. Zhang, L. Yu, K. Fang, Z.Q. You, C. Liu, H. Liu, X.Y. Yan, Website-oriented recommendation based on heat spreading and tag-aware collaborative filtering, Physica A 399 (2014) 4.
[34] L.C. Freeman, Centrality in social networks conceptual clarification, Soc. Netw. 1 (3) (1979) 215–239.
[35] M. Kitsak, L.K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H.E. Stanley, H.A. Makse, Identifiation of inflential spreaders in complex networks, Nat. Phys. 6 (11) (2010) 888–893.
[36] F. Morone, H.A. Makse, Influence maximization in complex networks through optimal percolation, Nature 524 (2015) 65–68.
[37] W. Zachary, An information flow model for conflict and fission in small groups, J. Anthropol. Res. 33 (1977) 452–473.
[38] F. Lin, W.W. Cohen, Power iteration clustering, in: ICML'10, pp. 655–662.
[39] P. Erdos, A. Renyi, On random graphs, Publ. Math. 6 (1959).
[40] A.L. Barabsi, R. Albert, Emergence of scaling in random networks, Science 286 (1999) 5439.
[41] A. Lancichinetti, S. Fortunato, F. Radicchi, Benchmark graphs for testing community detection algorithms, Phys. Rev. E 78 (2008) 046110.
[42] A.L. Barabsi, H. Jeong, Z. Nda, E. Ravasz, A. Schubert, T. Vicsek, Deterministic scale-free networks, Physica A 311 (2002) 3.
[43] L.A. Adamic, N. Glance, The polotical blogosphere and the 2004 U.S. election: divided they blog, in: WWW'05, pp. 36-43.
[44] D. Bu, Y. Zhao, L. Cai, H. Xue, X. Zhu, H. Lu, J. Zhang, S. Sun, L. Ling, N. Zhang, G. Li, R. Chen, Topological structure analysis of the protein-protein interation network in budding yeast, Nucl. Acids Res. 31 (2003) 9.
[45] B.C. Zhang, R. Liu, D. Massey, L.X. Zhang, Collecting the Internet AS-level topology, SIGCOMM Comp. Comm. Rev. 31 (2005) 35.
[46] H.B. Hu, X.F. Wang, Unified index to quantifying heterogeneity of complex networks, Physica A 14 (2008) 387.
[47] S. Mugisha, H.J. Zhou, Indentifying optimal targets of network attack by belief propagation, Phys. Rev. E 94 (2016) 012305.
[48] I. Swami, K. Timothy, S. Bala, Z. Wang, Attack robustness and centrality of complex networks, PLoS One 8 (2013) e59613.
[49] R. Albert, H. Jeong, A.L. Barabasi, Error and attack tolerance of complex networks, Nature 406 (2000) 6794.
[50] L.L., D. Chen, X.L. Ren, Q.M. Zhang, Y.C. Zhang, T. Zhou, Vital nodes identification in complex networks, Phys. Rep. 650 (2016).
[51] T. Martin, X. Zhang, M.E. Newman, Localization and centrality in networks, Phys. Rev. E 90 (2014) 052808.
[52] L. Duan, S. Ma, C. Aggarwal, et al., An ensemble approach to link prediction, IEEE Trans. Knowl. Data Eng. 29 (11) (2017) 2402–2416.
[53] L. Yu, J. Huang, G. Zhou, et al., TIIREC: A tensor approach for tag-driven item recommendation with sparse user generated content, Inform. Sci. 411 (2017) 122–135.
[54] L. Yu, C. Liu, Z.K. Zhang, Multi-linear interactive matrix factorization, Knowl.-Based Syst. 85 (2015) 307–315.