Assessing Percolation Threshold Based on High-Order Non-Backtracking Matrices

Yuan Lin†, Wei Chen‡, and Zhongzhi Zhang†‡
†School of Computer Science, Fudan University, Shanghai, China
‡Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai, China
§Microsoft Research, Beijing, China

yuanlin12@fudan.edu.cn, weic@microsoft.com, zhangzz@fudan.edu.cn

ABSTRACT
Percolation threshold of a network is the critical value such that when nodes or edges are randomly selected with probability below the value, the network is fragmented but when the probability is above the value, a giant component connecting a large portion of the network would emerge. Assessing the percolation threshold of networks has wide applications in network reliability, information spread, epidemic control, etc. The theoretical approach so far to assess the percolation threshold is mainly based on spectral radius of adjacency matrix or non-backtracking matrix, which is limited to dense graphs or locally treelike graphs, and is less effective for sparse networks with non-negligible amount of triangles and loops. In this paper, we study high-order non-backtracking matrices and their application to assessing percolation threshold. We first define high-order non-backtracking matrices and study the properties of their spectral radii. Then we focus on the 2nd-order non-backtracking matrix and demonstrate analytically that the reciprocal of its spectral radius gives a tighter lower bound than those of adjacency and standard non-backtracking matrices. We further build a smaller size matrix with the same largest eigenvalue as the 2nd-order non-backtracking matrix to improve computation efficiency. Finally, we use both synthetic networks and 42 real networks to illustrate that the use of the 2nd-order non-backtracking matrix does give better lower bound for assessing percolation threshold than adjacency and standard non-backtracking matrices.

CCS Concepts
•Information systems → Social networks;

Keywords
percolation theory; percolation threshold; non-backtracking matrix; high-order non-backtracking matrix; information and influence diffusion

1. INTRODUCTION
Percolation theory is a powerful statistical physics tool to describe information and viral spreading in social environment [29, 16, 14], robustness and fragility of infrastructural or technological networks [5, 11, 8], among other things, and thus its impact spreads well beyond statistical physics and reaches computer science, network science and related areas. Percolation is a random process independently occupying sites (a.k.a. nodes) or bonds (a.k.a. edges) in a network with probability $p$ (with probability $1 - p$ the site or bond is removed). In particular, the bond percolation can be identified to a special case of independent cascade model [23, 16]. In this paper, our discussion focuses on bond percolation, although similar approach applies to site percolation as well. As probability $p$ increases from 0 to 1, the network is expected to experience a phase transition from a large number of small connected components to the emergence of a giant connected component, with the size proportional to the size of the network. The value $p$ at this transition point is referred to as the percolation threshold.

The percolation threshold can be used to assess the spreading power of a network from a topological point of view. For example, in studying epidemics in a social network, a small percolation threshold means that a virus is easy to spread in the network and infects a large portion of the network. Thus, many applications rely on the concept of percolation threshold, such as finding influential nodes in a social network [21, 18], facilitating or curbing propagations [7, 6], and determining transmission rates in wireless networking [10, 1]. Therefore, it is crucial to have an accurate understanding of the percolation threshold in networks.

However, since the exact percolation threshold for a general network is analytically difficult to obtain, these applications rely on some theoretical estimates of the actual percolation threshold. A commonly used theoretical estimate is the reciprocal of the largest eigenvalue $\lambda_A$ (same as the spectral radius) of the network’s adjacency matrix $A$ (e.g. in [28, 3, 7, 6]). In particular, Bollobás et al. show that this estimate is accurate for dense networks [3]. However, in sparse networks, it could be far off. For example, in a ring network (Fig. 1a), the real percolation threshold is 1, but its adjacency matrix has the largest eigenvalue 2, predicting the percolation threshold is 0.5.

Recently, the reciprocal of the largest eigenvalue of the network’s non-backtracking matrix [15, 12] is introduced as a better theoretical estimate of the percolation threshold. We defer its technical definition to Section 2, and give an intuitive explanation on the issue of the adjacency matrix

©2017 International World Wide Web Conference Committee (IW3C2), published under Creative Commons CC BY 4.0 License.
ACM 978-1-4503-4913-0/17/04.
http://dx.doi.org/10.1145/3038912.3052610
that is addressed by the non-backtracking matrix. Let \( \pi \) be a vector, with \( \pi_i \) representing the probability that node \( i \) is in the giant connected component, when every edge has a probability of \( p \) to be occupied. Let \( A \) be the adjacency matrix of the graph, and \( N_i \) be the set of neighbors of \( i \). The \( i \)-th entry of \( pA\pi \) satisfies \( [pA\pi]_i = \sum_{j \in N_i} p\pi_j \), which approximately represents that \( i \) could connect to the giant component through one of its neighbors \( j \) (with probability \( p\pi_j \)). Thus it should be the same as \( \pi_i \), and in matrix form, we have \( pA\pi = \pi \). This suggests that \( 1/p \) should be at most \( \lambda_A \), i.e. \( 1/\lambda_A \) is a lower bound of the percolation threshold. However, for every edge \((i,j)\) in the network, the above approximation considers both that \( i \) may rely on \( j \) to connect to the giant component and \( j \) may rely on \( i \), but the two cases should not jointly occur. Therefore, the estimate by \( 1/\lambda_A \) inflates the probability that a giant component emerges and underestimates the percolation threshold. The non-backtracking matrix addresses this issue by disallowing such directed circular dependency between a pair of nodes on an edge.

However, non-backtracking matrix only works well in locally treelike graphs. In non-treelike graphs, it provides a lower bound, but not as close as the true percolation threshold. As suggested by the famous small-world network study \cite{39}, a significant amount of triangles exist in many real networks. The non-backtracking matrix does not eliminate the circular dependency through triangles or other local structures, and thus it may still underestimate the percolation threshold. For example, in the triangle ring network in Fig. 1b, the real percolation threshold is 1, but the theoretical estimates by the adjacency and non-backtracking matrices are 0.3333 and 0.5523, respectively.

In this paper, we extend the idea of the non-backtracking matrix to define high-order non-backtracking matrices. We first define the high-order non-backtracking matrices and study the evolution of their largest eigenvalue with respect to order (Section 3). Then, we propose that the reciprocal of the largest eigenvalue of the 2nd-order non-backtracking matrix can provide a better estimate for the percolation threshold in an arbitrary network, because it eliminates the circular dependency from triangles (Section 3). We further provide an alternative matrix sharing the same largest eigenvalue but with substantially smaller size to improve computation efficiency. Finally, we conduct extensively experiments on both the forest fire model and 42 real networks to demonstrate the effectiveness of our method (Section 5).

To summarize, our contributions include: (i) proposing the high-order non-backtracking matrices and studying their eigenvalue properties; (ii) establishing a more precise theoretical estimation for bond percolation threshold and finding a faster approach to evaluate it; and (iii) supporting our analysis by empirical evaluations on synthetic and real networks.

1.1 Related Work

For random degree-uncorrelated network models, the percolation threshold can be approximated as \( \langle d \rangle / (\langle d^2 \rangle - \langle d \rangle) \) \cite{9,5}. Here, \( \langle d \rangle \) and \( \langle d^2 \rangle \) are separately the first and the second moments of the degree distribution. This estimation is less predictive for real networks where degree correlations appear.

Bollobás et al. \cite{3} show that the percolation threshold of dense graph is reciprocal of the largest eigenvalue of the adjacency matrix. However, the conclusion requires restrictive conditions for the networks. Especially for sparse networks, this estimation can be only regarded as providing a lower bound for the true percolation threshold.

Since a lot of realistic networks are sparse, Karrer et al. \cite{15} and Hamilton et al. \cite{12} simultaneously propose that the reciprocal of the largest eigenvalue of the non-backtracking matrix is a tighter lower bound for bond and site percolation threshold, respectively, on sparse networks. This prediction is based on a message passing technique and obtained by heuristic equations and approximations on locally treelike structures. Radicchi \cite{25} further presents a mapping between the site and bond percolation to mathematically verify the predicted bond and site percolation thresholds are identical in this method. Although the estimation based on the non-backtracking matrix is more precise than that based on the adjacency matrix, it is still not close enough to the true percolation threshold on many real networks \cite{26}, since it suffers from the limitation of the treelike assumption. Radicchi et al. \cite{27} also derives an alternative matrix of the non-backtracking matrix based on triangle elimination to improve the estimate for the site percolation threshold. However, this alternative matrix would overshoot on bond percolation. That means this estimate may overestimate the percolation threshold, leading it no longer provides a lower bound for the bond percolation threshold.

2. PRELIMINARIES

We consider a finite connected undirected graph \( G = (V, E) \) with \( N \) nodes and \( E \) edges, where \( V = \{1, 2, \ldots, N\} \) is the set of \( N \) nodes and \( E = \{(i, j) \mid i, j \in V\} \) is the set of \( E \) edges. The connectivity between nodes in \( G \) is described by an adjacency matrix \( A \), in which the element \( a_{ij} = 1 \) if \((i, j) \in E\) and \( a_{ij} = 0 \) otherwise. We assume that \( G \) has no self-loops, i.e. \( a_{ii} = 0 \) for all \( i \) in \( V \). Henceforth, the adjacency and non-backtracking matrices are all refer to this graph \( G \), when the context is clear.

The bond percolation model is only controlled by one parameter, bond occupation probability \( p \). That means, in a network \( G \), each edge is independently occupied with probability \( p \). The percolation clusters are sets of nodes connected only by occupied edges. For \( p = 0 \), no edge is occupied so that there are \( N \) isolated clusters of size one. For \( p = 1 \), all edges are occupied and all nodes compose a single cluster of
size $N$. At intermediate values of $p$, the network undergoes two different phases: the non-percolating phase, where all clusters have microscopic size; the percolating phase, where a single macroscopic cluster (the giant component), whose size is comparable to the entire network, is present. The percolation threshold $p_c$ is the value above which the giant cluster appears; below $p_c$ there are only small clusters.

The transition can be monitored through two primary quantities of interest, the relative sizes of the first and second largest clusters with respect to the size of the network, denoted by $S_1(p)$ and $S_2(p)$, respectively. In order to evaluate percolation threshold numerically, there are many different estimates for $p_c$ being proposed, such as the occupation probability corresponding to the maximal value of $S_2(p)$, the peak position of the empirical variance of $S_1(p)$ and the average size of clusters except the largest one [26]. In this paper, we determine the best empirical estimate for percolation threshold as the value of $p$ where the second largest cluster reaches its maximum, namely

$$p_c = \arg \{ \max_p S_2(p) \}. \quad (1)$$

In the following, we will compare it with theoretical approaches to check their validity.

The first theoretical estimate is based on the adjacency matrix. Bollobás et al. [3] show that for percolation on dense networks, the percolation threshold can be given by

$$p_c = \frac{1}{\lambda_A},$$

where $\lambda_A$ is the largest eigenvalue of matrix $A$, which is also the spectral radius of $A$.\(^1\) Although this estimate provides a good prediction for percolation threshold on networks with high density of connections, it becomes less precise on sparse networks.

Since a lot of realistic networks are sparse, a new estimate based on non-backtracking matrix [13, 17] is proposed to give a better prediction. For any undirected network $G$, we can transform it to a directed one through replacing each undirected edge $(i,j)$ by two directed ones $i \to j$ and $j \to i$. The non-backtracking matrix, denoted by $B$, is a $2E \times 2E$ matrix with rows and columns indexed by directed edges $i \to j$ and entries are given by

$$B_{i \to j, k \to l} = \delta_{jk} (1 - \delta_{il}),$$

where $\delta_{ij}$ is the Kronecker Delta Function ($\delta_{ij} = 1$ if $i = j$, and 0 if $i \neq j$). In other words, for any two directed edges $i \to j, j \to k$, $B_{i \to j, j \to k} = 1$ if $k \neq i$, and thus $B_i \to j, j \to k$ records the walk from $i$ to $j$ and continuing to $k$ but not backtracking to $i$. Under the locally treelike assumption, which means that the local neighborhood of a node is close to a tree without redundant paths, the reciprocal of the largest eigenvalue of $B$ provides a close lower bound for sparse networks [15, 12], namely

$$p_c \geq \frac{1}{\lambda_B} \quad \text{and} \quad p_c \approx \frac{1}{\lambda_B}.$$  

As already mentioned in the introduction, the non-backtracking matrix relies on the approximation of the local
cally treelike structure and it does not eliminate triangle dependency, and thus it may not give a tight lower bound of the bond percolation threshold. In the following, we propose a more powerful tool, high-order non-backtracking matrices, to better estimate bond percolation threshold.

3. HIGH-ORDER NON-BACKTRACKING MATRICES

In this section, we first introduce the definition of high-order non-backtracking matrices, and then investigate their properties.

In a network $G$, let $i_1 \to i_2 \to \cdots \to i_{g+1}$ stand for a length-$g$ directed path composed by $g + 1$ different nodes $i_1, i_2, \ldots, i_{g+1}$, obeying $a_{i_k i_{k+1}} = 1$ ($1 \leq k \leq g$). For the sake of saving space, we use $i_1 i_2 \cdots i_{g+1}$ to stand for length-$g$ directed path in place of $i_1 \to i_2 \to \cdots \to i_{g+1}$. Let $P_g$ be the number of length-$g$ directed paths in $G$. Then, we define the $g$-th-order non-backtracking matrix as follows:

**Definition 1.** The $g$-th-order non-backtracking matrix $B^{(g)}$ is a $P_g \times P_g$ matrix with rows and columns indexed by length-$g$ directed paths. The elements in $B^{(g)}$ are

$$B^{(g)}_{i_1 i_2 \cdots i_g, j_1 j_2 \cdots j_g} = a_{i_1 j_1} (1 - \delta_{i_1, j_1}) \prod_{k=2}^{g+1} \delta_{i_k j_{k-1}}. \quad (2)$$

In other words, for each length-$(g + 1)$ directed path, $i_1 i_2 \cdots i_{g+2}$, we have $B^{(g)}_{i_1 i_2 \cdots i_g, j_1 j_2 \cdots j_g} = 1$; and all other entries of $B^{(g)}$ have value 0. For example, if $\mathbf{T}_3, \mathbf{T}_4, \mathbf{H}_5$, and $\mathbf{H}_6$ are all length-2 directed paths in a graph, then $B^{(2)}_{\mathbf{T}_3, \mathbf{T}_4} = 1$, while $B^{(2)}_{\mathbf{T}_3, \mathbf{H}_5} = B^{(2)}_{\mathbf{T}_4, \mathbf{H}_6} = 0$. We can regard matrix $B^{(g)}$ as encoding the relations between length-$g$ directed paths in $G$. It also describes a kind of non-backtracking walks with memory, avoiding going back to a node visited in the recent $g$ steps. It is easy to verify that the standard non-backtracking matrix is $B = B^{(1)}$, and adjacency matrix $A = B^{(0)} (a_{i_1 j_1} \in E (2)$ is explicitly for the case of $g = 0$).

Let $\lambda_B^{(g)}$ be the largest eigenvalue of $B^{(g)}$. By the Perron-Frobenius Theorem [20], $\lambda_B^{(g)}$ is real and non-negative, and it is zero only when the directed graph with the adjacency matrix equal to $B^{(g)}$ is a directed acyclic graph (DAG). We now study the properties of $\lambda_B^{(g)}$ as $g$ changes. A simple cycle in graph $G$ is a closed path with no repetitions of nodes and directed edges allowed, except the repetition of the starting and ending nodes. To maintain a consistent terminology, henceforth, all cycle means a simple cycle. The result is given in the following theorem:

**Theorem 1.** The $g$-th-order non-backtracking matrix $B^{(g)}$ satisfies the following properties:

(i) The largest eigenvalue of $B^{(g)}$ is non-increasing with respect to $g$, i.e., for every $g = 1, 2, 3, \ldots$, $\lambda_B^{(g)} \geq \lambda_B^{(g-1)}$.

(ii) Let $\ell$ be the length of the longest simple cycle in graph $G$. Then $\lambda_B^{(g)} = 0$ for all $g \geq \ell - 1$.

(iii) For any $\ell \geq 3$, if there is no simple cycle with length $\ell$ in $G$, then $\lambda_B^{(\ell-2)} = \lambda_B^{(\ell-1)}$.

In order to prove Theorem 1, we first present some related notations and lemmata. The $g$-th-order non-backtracking
matrix $B^{(g)}$ of network $G$ can be regarded as the adjacency matrix of a directed graph $G^{(g)} = (V^{(g)}, E^{(g)})$, where $V^{(g)} = \{ u | u$ is a length-$g$ directed path $\}$, and $E^{(g)} = \{ u \rightarrow v | u, v$ are two length-$g$ directed paths and $B^{(g)}_{uv} = 1 \}$. Note that when $g = 0$, $G^{(0)} = G$ (considering $G$ as a symmetric directed graph). The line graph of $G^{(g)}$ is $G^{(g)} = (\overline{V^{(g)}}, \overline{E^{(g)}})$, where $\overline{V^{(g)}} = \{ \alpha \rightarrow \beta | \alpha = (u, v)$ and $\beta = (v, w)$ are two different edges $u \rightarrow v$ and $v \rightarrow w$ in $E^{(g)} \}$. Let $\overline{B^{(g)}}$ denote the adjacency matrix of the line graph $G^{(g)}$, with elements

$$\overline{B}^{(g)}_{u \rightarrow v, p \rightarrow q} = \delta_{p,v},$$

where $u, v, p$ and $q$ are length-$g$ directed paths with $u \rightarrow v$, $p \rightarrow q \in E^{(g)}$. In other words, we can recast definition of $\overline{B}^{(g)}$ as

$$\overline{B}^{(g)}_{i_1 \rightarrow \ldots \rightarrow i_{g+2}, j_1 \rightarrow \ldots \rightarrow j_{g+2}} = \prod_{k=2}^{g+2} \delta_{i_k, j_k}.$$  

**Lemma 1.** $G^{(g)}$ can be obtained by deleting all edges in all length-$(g+1)$ cycles in $G^{(g-1)}$.

**Proof.** For each edge $u \rightarrow v$ ($u = i_1 \rightarrow \ldots \rightarrow i_{g+1}$ and $v = i_2 \rightarrow \ldots \rightarrow i_{g+1}$) in $G^{(g-1)}$, it must correspond to a length-$g$ directed path $i_1 \rightarrow i_2 \rightarrow \cdots \rightarrow i_{g+1}$ in $G$. On the other hand, each length-$g$ directed path also corresponds to an edge in $G^{(g-1)}$. Thus, $\overline{B}^{(g)} = \overline{E}^{(g-1)} = \overline{E}^{(g-1)}$ if we identify $i_1 i_2 \rightarrow \cdots \rightarrow i_g i_{g+1} \rightarrow i_{g+2}$ with $i_1 \rightarrow i_2 \rightarrow \cdots \rightarrow i_{g+1}$, i.e., the node set of $G^{(g)}$ is identical to the node set of $G^{(g-1)}$. Moreover, for every edge in $G^{(g)}$, $i_1 \rightarrow i_2 \rightarrow \cdots \rightarrow i_{g+1}$, it also has an edge in $G^{(g-1)}$, implying $E^{(g)} \subseteq E^{(g-1)}$. For an arbitrary edge in $E^{(g)} \setminus E^{(g-1)}$, it must have a form as $i_1 i_2 \rightarrow \cdots \rightarrow i_{g+1}$, and there must be $g - 2$ other edges with the same form together composing a length-$(g+1)$ cycle in $G^{(g)}$, namely $i_1 i_2 \rightarrow \cdots \rightarrow i_{g+1} \rightarrow i_1 \rightarrow \cdots \rightarrow i_{g+1} \rightarrow \cdots \rightarrow i_1 \rightarrow \cdots \rightarrow i_{g+1}$. Moreover, for any length-$(g+1)$ cycle in $G^{(g-1)}$, it must be of the above form. If not, then it involves an edge of form $i_1 i_2 \rightarrow \cdots \rightarrow i_{g+1} \rightarrow i_1 \rightarrow \cdots \rightarrow i_{g+1} \rightarrow i_1 \rightarrow \cdots \rightarrow i_{g+1} \rightarrow \cdots \rightarrow i_1 \rightarrow \cdots \rightarrow i_{g+1}$. Together, we know that by exactly removing all edges in all length-$(g+1)$ cycles in $G^{(g-1)}$, we obtain $G^{(g)}$. □

The next lemma is a known result saying that when we remove edges from a graph, the largest eigenvalue of the adjacency matrix will not increase.

**Lemma 2.** (Proposition 3.1.1 of [4]). For the adjacency matrix $A$ of a graph $G$, if $A'$ is a matrix obtained by replacing some of the 1’s in $A$ with 0, then we have

$$\lambda_{A'} \leq \lambda_A,$$

where $\lambda_{A'}$ and $\lambda_A$ are the largest eigenvalue of $A'$ and $A$, respectively.

We are now ready to prove Theorem 1.

**Proof of Theorem 1.**

(i) For a directed graph $G$ and its line graph, Pakoński et al. [24] show that their spectra excluding the possible eigenvalue zero are exactly the same. Therefore, we have

$$\lambda_{B^{(g-1)}} = \overline{\lambda}^{(g-1)},$$

where $\overline{\lambda}^{(g-1)}$ is the largest eigenvalue of matrix $\overline{B}^{(g-1)}$.

According to Lemma 1, $G^{(g)}$ have the same node set as $G^{(g-1)}$, and $G^{(g)}$ can be obtained by removing edges in length-$(g+1)$ cycles in $G^{(g-1)}$. Thus, applying Lemma 2, we attain the conclusion that

$$\lambda_{B^{(g-1)}} = \overline{\lambda}^{(g-1)} \geq \lambda_B^{(g)}.$$

(ii) We first claim that, $G^{(g)}$ is a DAG for all $g \geq \ell - 1$. If not, there are $k (k \geq g + 2)$ length-$g$ directed paths in $G^{(g)}$, $i_1 i_2 \rightarrow i_{g+1} \rightarrow \cdots \rightarrow i_{g+2}$, constituting a cycle in $G^{(g)}$. Then there must be a length-$k$ cycle composed by $i_1, i_2, \ldots, i_k$, since $g \geq \ell - 1$, implying that there is at least one cycle in $G$ with length larger than $\ell$, which contradicts the assumption that the length of the largest cycle in $G$ is $\ell$. Thus, network $G^{(g)}$ is a DAG. It is easy to verify that the adjacency matrix of a DAG has the normal form where all block matrices on the diagonal are one-dimensional matrices with value 0, and thus all its eigenvalues are 0. Therefore $\lambda_B^{(g)} = 0$ for all $g \geq \ell - 1$.

(iii) Consider the line graph of $G^{(g-2)}$, $\overline{G}^{(g-2)}$. By Lemma 1, graph $G^{(g-1)}$ is obtained by removing all length-$\ell$ cycles in $G^{(g-2)}$. Similar to the argument in the proof of Lemma 1, every length-$\ell$ cycle in $\overline{G}^{(g-1)}$ has the form $i_1 i_2 \rightarrow \cdots \rightarrow i_{\ell} \rightarrow i_1$; it corresponds to a length-$\ell$ simple cycle in the original graph $G$, $i_1 \rightarrow i_2 \rightarrow \cdots \rightarrow i_{\ell-1} \rightarrow i_1$. By assumption $G$ has no length-$\ell$ simple cycles, thus it implies that $\overline{G}^{(g-1)}$ is the same as $\overline{G}^{(g-2)}$. Since the line graph shares the same non-zero eigenvalues as the original graph [24], we know that $\lambda^{(g-2)} = \overline{\lambda}^{(g-2)}$. □

4. ESTIMATING PERCOLATION THRESHOLD BY 2ND-ORDER NON-BACKTRACKING MATRIX

In this section, we show that the reciprocal of the largest eigenvalue of the 2nd-order non-backtracking matrix gives a lower bound on the bond percolation threshold. According to Theorem 1, it implies that this lower bound is tighter than the previous proposed analytical lower bounds using the reciprocal of the adjacency matrix or the standard non-backtracking matrix. We then show that we can replace the 2nd-order non-backtracking matrix with a substantially smaller matrix sharing the same largest eigenvalue to improve computation efficiency.

4.1 Derivation of Lower Bound

Our derivation follows the message passing techniques proposed in [15, 12], which involves first-order analysis (ignoring higher-order terms) and heuristic equations on locally treelike graphs. Henceforth, we denote percolation threshold predicted by the adjacency matrix, non-backtracking matrix and 2nd-order non-backtracking matrix as $p^{(0)}$, $p^{(1)}$ and $p^{(2)}$, respectively.

Let $\pi_i$ be the probability that node $i$ belongs to the giant connected component, where the random events are that each edge in $G$ is independently occupied with probability
Let $\theta_{i \to j \to k}$ be the probability that node $i$ connects to
the giant component following path $i \to j \to k$, where $i, j, k$
are all different nodes. Then, we can construct the following
relation for bond percolation:

$$\pi_i = 1 - \prod_{j \in N_i, k \neq i} (1 - \theta_{i \to j \to k}), \quad (4)$$

Then, the expected size of the giant component can be given by

$$S_1(p) = \sum_{i=1}^{N} \pi_i.$$ 

In a finite-size network, there is a drastic change for $S_1(p)$
at the percolation threshold $p_c$. Next, we focus on predicting
the value of $p_c$.

According to the definition of quantity $\theta_{i \to j \to k}$, we can
construct recursive relations, which is appropriate for locally
treelike structures with triangles:

$$\theta_{i \to j \to k} = 1 - \prod_{\ell \in N_k \setminus \{i,j\}} (1 - p \cdot \theta_{j \to k \to \ell}). \quad (5)$$

The above heuristic equation intuitively means that, node
$i$ connects to the giant component through at least one of
the paths $j \to k \to \ell$, and $i$ connects to $j$ with probability
$p$ while $j$ connects to the giant component through path
$j \to k \to \ell$ with probability $\theta_{j \to k \to \ell}$. The equation is
approximately accurate when the local neighborhood of $i$ is
close to a tree without redundant paths, except that we al-
low triangles such as $i \to j \to k \to i$, since the equation
requires $\ell \neq i$, excluding such triangles. When we ignore $p^2$
and higher order terms in Eq. (5), we obtain

$$\theta_{i \to j \to k} = p \sum_{\ell \in N_k \setminus \{i,j\}} \theta_{j \to k \to \ell},$$

which can be recast in matrix notation as

$$\theta = p B^{(2)} \theta,$$

where $\theta$ is a vector in which elements indexed by length-2
directed paths. When $p \geq p_c$, percolation happens, and $\theta$
should be a non-negative vector with some strictly positive
entries. (It holds when length of the largest cycle is at least
4.) But if $p_c < 1/\lambda_B^{(2)}$, we means we could have an eigenvalue
of $1/p_c > \lambda_B^{(2)}$, violating the definition of $\lambda_B^{(2)}$. Therefore, we have

$$p_c \geq p_c^{(2)} = 1/\lambda_B^{(2)}.$$ \hspace{1cm} (6)

Theorem 1 theoretically guarantees $p_c^{(2)} \geq p_c^{(1)} \geq p_c^{(0)}$, so that $p_c^{(2)}$ provides a tighter lower bound for percolation
threshold than the ones provided by the adjacency and stan-
dard non-backtracking matrices. Moreover, consider the ex-
ample of triangle ring network shown in Figure 1b, it is easy
to obtain $\lambda_B^{(2)} = 1$ and $p_c^{(2)} = 1/\lambda_B^{(2)} = 1$, which is consistent

with the true percolation threshold. Compared with estimates $p_c^{(0)}$ and $p_c^{(1)}$, $p_c^{(2)}$ exhibits remarkable improvement in precision.

The reason why $p_c^{(2)}$ provides better prediction of $p_c$ than
$p_c^{(1)}$ can be heuristically explained as follows. In the esti-
mation based on non-backtracking matrix, it only considers
the probability node $i$ connects to the giant component
through node $j$, which can be denoted as $\theta_{i \to j}$. If there
is a triangle composed by nodes $i$, $j$ and $k$ in the network,
$\theta_{i \to j}$ grows as $\theta_{j \to k}$ increases. Similarly, $\theta_{j \to k}$ increases with
$\theta_{k \to i}$, and $\theta_{k \to i}$ increases with $\theta_{i \to j}$. This creates a triangu-
lar dependency which artificially inflates the values of $\theta_{i \to j}$,
$\theta_{j \to k}$ and $\theta_{k \to i}$, leading to a higher estimate of the proba-
bility of giant component emergence and a lower estimate on
the percolation threshold. When triangles are abundant, as
evidenced by the small-world research on many real-world
networks [30], using the standard non-backtracking matrix
may still significantly underestimate the percolation thresh-
old. By using the 2nd-order non-backtracking matrix, we
avoid the triangular dependency, so that $p_c^{(2)}$ is more precise
than $p_c^{(1)}$.

Although $p_c^{(2)}$ is a tighter lower bound, the size of $B^{(2)}$ is
usually larger than $B$, leading to higher computation com-
plexity. In the following we provide a further technique to
tackle this problem.

### 4.2 Improving Computation Efficiency

In this subsection, we illustrate that we can transfer the task of computing eigenvalues of $B^{(2)}$ to calculating eigen-
values of a new matrix $M$, defined as

$$M = \begin{pmatrix}
B^{(1)} & -\Delta B_2 & D_\Delta - I & B^{(1)} - \Delta B_1 \\
I & 0 & 0 & 0 \\
0 & I & 0 & 0 \\
0 & 0 & I & 0
\end{pmatrix}. \quad (6)$$

Here, $M$ is a square matrix of dimension $8E \times 8E$ composed
of 16 $2E \times 2E$ blocks, whose size is evidently smaller than
$B^{(2)}$. Matrix $\Delta B_1$ and $\Delta B_2$ have identical size with $B^{(1)}$
and their elements are defined as

$$(\Delta B_1)_{i \to j, k \to \ell} = \begin{cases}
1, & a_{ij} B_{i \to j, k \to \ell}^{(1)} = 1, \\
0, & \text{otherwise},
\end{cases}$$

and

$$(\Delta B_2)_{i \to j, k \to \ell} = \begin{cases}
1, & \delta_{i\ell} a_{ij} a_{jk} a_{k\ell} = 1, \\
0, & \text{otherwise},
\end{cases}$$

respectively. The matrix $D_\Delta$ is a $2E \times 2E$ diagonal matrix
and its element $D_{\Delta_{i \to j, i \to j}}$ equals the number of triangles
containing edge $(i,j)$.

**Theorem 2.** The set of non-zero eigenvalues of matrix $M$ defined in Eq. (6) consists of all non-zero eigenvalues
of the 2nd-order non-backtracking matrix $B^{(2)}$ and possibly
eigenvalues $-1, (1 + \sqrt{3}/2)$ and $(1 - \sqrt{3}/2)$.

**Proof.** The proof will include two parts. First, we prove that
every non-zero eigenvalue of $B^{(2)}$ corresponds to an eigenvalue of $M$. Second, we elaborate that every non-zero
eigenvalue of $M$ equals to one of eigenvalues of $B^{(2)}$, except
possible eigenvalues $-1, (1 + \sqrt{3}/2)$ and $(1 - \sqrt{3}/2)$.

(i) Let $\lambda$ be an arbitrary non-zero eigenvalue of $B^{(2)}$ and $\psi$
be its corresponding eigenvector. According to the definition
of the 2nd-order non-backtracking matrix, we have
\[ \lambda \psi_{i \rightarrow j \rightarrow k} = \sum_{\ell \in N_k \setminus \{i,j\}} B^{(1)}_{\ell \rightarrow k, k \rightarrow \ell} \psi_{j \rightarrow k \rightarrow \ell}. \] (7)

In order to show each non-zero eigenvalue of \( B^{(2)} \) is also an eigenvalue of \( M \), we first introduce two related quantities:
\[ x_{i \rightarrow j} = \sum_{k \in N_j \setminus \{i\}} B^{(1)}_{i \rightarrow j, j \rightarrow k} \psi_{j \rightarrow k} \] (8)
and
\[ y_{i \rightarrow j} = \sum_{k \in N_j \setminus \{i\}} B^{(1)}_{i \rightarrow j, j \rightarrow k} B^{(1)}_{j \rightarrow k, k \rightarrow j} \psi_{j \rightarrow k}. \] (9)
Let \( x \) and \( y \) be vectors with elements \( x_{i \rightarrow j} \) and \( y_{i \rightarrow j} \), respectively. We claim that \( x \) and \( y \) satisfy the following two relations:
\[ x = \frac{1}{\lambda} B^{(1)} x - \frac{1}{\lambda^2} \Delta B_2 x + \frac{1}{\lambda^3} D \Delta x - \frac{1}{\lambda^3} y \] (10)
and
\[ y = \frac{1}{\lambda} \Delta B_1 x - \frac{1}{\lambda^2} \Delta B_2 x + \frac{1}{\lambda^3} D \Delta x - \frac{1}{\lambda^3} y, \] (11)
which will be proved latter.

Combining Eq. (10) and Eq. (11), we can obtain
\[ x - y = \frac{1}{\lambda} (B^{(1)} - \Delta B_1)x, \]
which indicates
\[ y = \left( \frac{1}{\lambda} B^{(1)} - \frac{1}{\lambda} \Delta B_1 \right) x. \] (12)

Plugging Eq. (12) into Eq. (10) yields
\[ x = \frac{1}{\lambda} B^{(1)} x - \frac{1}{\lambda^2} \Delta B_2 x + \frac{1}{\lambda^3} (D \Delta - I) x + \frac{1}{\lambda^3} (B^{(1)} - \Delta B_1)x. \] (13)

Based on Eq. (13), we establish
\[ M z = \lambda z, \]
where \( z = (x^\top, \frac{1}{\lambda} x^\top, \frac{1}{\lambda^2} x^\top, \frac{1}{\lambda^3} x^\top)^\top \). Thus, every non-zero eigenvalue of \( B^{(2)} \) corresponds to an eigenvalue of \( M \).

Proof of Eq. (10). Plugging Eq. (7) into Eq. (8), we have
\[ x_{i \rightarrow j} = \frac{1}{\lambda} \sum_{k \in N_j \setminus \{i\}} B^{(1)}_{i \rightarrow j, j \rightarrow k} \psi_{j \rightarrow k} \sum_{\ell \in N_k \setminus \{j\}} B^{(1)}_{j \rightarrow k, k \rightarrow \ell} \psi_{j \rightarrow k \rightarrow \ell} - \frac{1}{\lambda} \sum_{k \in N_j \setminus N_i} B^{(1)}_{i \rightarrow j, j \rightarrow k} B^{(1)}_{j \rightarrow k, k \rightarrow j} \psi_{j \rightarrow k \rightarrow j}. \] (14)

In the r.h.s of Eq. (14), the first term can be rewritten as
\[ \frac{1}{\lambda} \sum_{k \in N_j \setminus \{i\}} B^{(1)}_{i \rightarrow j, j \rightarrow k} \sum_{\ell \in N_k \setminus \{j\}} B^{(1)}_{j \rightarrow k, k \rightarrow \ell} \psi_{j \rightarrow k \rightarrow \ell} = \frac{1}{\lambda} \sum_{k \in N_j \setminus \{i\}} B^{(1)}_{i \rightarrow j, j \rightarrow k} x_{j \rightarrow k}. \] (15)

Here, Eq. (8) is used. Utilizing Eq. (7) again, the second term of r.h.s of Eq. (14) can be represented as
\[ \frac{1}{\lambda} \sum_{k \in N_j \setminus N_i} B^{(1)}_{i \rightarrow j, j \rightarrow k} B^{(1)}_{j \rightarrow k, k \rightarrow i} \psi_{j \rightarrow k \rightarrow i} = \frac{1}{\lambda} \sum_{k \in N_j \setminus N_i} B^{(1)}_{i \rightarrow j, j \rightarrow k} B^{(1)}_{j \rightarrow k, k \rightarrow i} \psi_{j \rightarrow k \rightarrow i} + \frac{1}{\lambda} \sum_{\ell \in N_i \setminus \{k\}} B^{(1)}_{i \rightarrow j, j \rightarrow k} B^{(1)}_{j \rightarrow k, k \rightarrow i} \psi_{j \rightarrow k \rightarrow i} - \frac{1}{\lambda} \sum_{\ell \in N_i \setminus \{k\}} B^{(1)}_{i \rightarrow j, j \rightarrow k} B^{(1)}_{j \rightarrow k, k \rightarrow i} \psi_{j \rightarrow k \rightarrow i} \] (16)

where \( \Delta_{i \rightarrow j} \) is the number of length-3 loops starting from edge \( i \rightarrow j \).

Inserting Eq. (15) and Eq. (16) into Eq. (14), we have
\[ x_{i \rightarrow j} = \frac{1}{\lambda} \sum_{k \in N_j \setminus \{i\}} B^{(1)}_{i \rightarrow j, j \rightarrow k} x_{j \rightarrow k} - \frac{1}{\lambda} \sum_{k \in N_j \setminus N_i} B^{(1)}_{i \rightarrow j, j \rightarrow k} B^{(1)}_{j \rightarrow k, k \rightarrow i} x_{k \rightarrow i} + \frac{1}{\lambda} \sum_{\ell \in N_i \setminus \{k\}} B^{(1)}_{i \rightarrow j, j \rightarrow k} B^{(1)}_{j \rightarrow k, k \rightarrow i} \psi_{j \rightarrow k \rightarrow i} - \frac{1}{\lambda} \sum_{\ell \in N_i \setminus \{k\}} B^{(1)}_{i \rightarrow j, j \rightarrow k} B^{(1)}_{j \rightarrow k, k \rightarrow i} \psi_{j \rightarrow k \rightarrow i}. \] (17)

Recasting Eq. (17) in matrix notation, we can obtain Eq. (10).

Proof of Eq. (11). Analogously, substituting Eq. (7) into Eq. (9), \( y_{i \rightarrow j} \) can be represented as
\[ y_{i \rightarrow j} = \frac{1}{\lambda} \sum_{k \in N_j \setminus \{i\}} B^{(1)}_{i \rightarrow j, j \rightarrow k} B^{(1)}_{j \rightarrow k, k \rightarrow i} \psi_{j \rightarrow k \rightarrow i} - \frac{1}{\lambda} \sum_{k \in N_j \setminus N_i} B^{(1)}_{i \rightarrow j, j \rightarrow k} B^{(1)}_{j \rightarrow k, k \rightarrow i} \psi_{j \rightarrow k \rightarrow i}. \] (18)
Instituting Eq. (16) and Eq. (18) into Eq. (17), one has
\[ y_{i \rightarrow j} = \frac{1}{\lambda} \sum_{k \in \mathcal{N}_i \setminus \{i\}} B_{i \rightarrow j \rightarrow k}^{(1)} B_{j \rightarrow k \rightarrow i}^{(1)} x_{j \rightarrow k} \]
\[ - \frac{1}{\lambda^2} \sum_{k \in \mathcal{N}_i \setminus \{i\}} B_{i \rightarrow j \rightarrow k}^{(1)} B_{j \rightarrow k \rightarrow i}^{(1)} x_{j \rightarrow k} \]
\[ + \frac{1}{\lambda} \sqrt{\lambda} x_{i \rightarrow j} = \frac{1}{\lambda^2} y_{i \rightarrow j}, \]
which can be recast in matrix form to obtain Eq. (11).

(ii) Now, we prove that every non-zero eigenvalue of \( M \) equals to one of eigenvalues of \( B^{(3)} \), except possible eigenvalues \(-1, (1 + \sqrt{3})/2 \) and \((1 - \sqrt{3})/2\). Let \( \lambda \) be an arbitrary non-zero eigenvalue of \( M \) and \( z = (x^T, i \sqrt{\lambda} x^T, 1 \sqrt{\lambda} x^T)^T \) be its corresponding eigenvector. According to Eq. (7) and Eq. (8), we have
\[ \lambda \psi_{i \rightarrow j \rightarrow k} = \begin{cases} \lambda (x_{j \rightarrow k} - \psi_{j \rightarrow k \rightarrow i}, i \in \mathcal{N}_k) \\ x_{j \rightarrow k}, i \notin \mathcal{N}_k. \end{cases} \]

Thus, for the case of nodes \( i, j \) and \( k \) constituting a triangle, we can establish the following equations:
\[ \begin{align*}
\lambda \psi_{i \rightarrow j \rightarrow k} + \psi_{j \rightarrow k \rightarrow i} &= x_{j \rightarrow k} \\
\lambda \psi_{j \rightarrow k \rightarrow i} + \psi_{i \rightarrow j \rightarrow k} &= x_{i \rightarrow j} \\
\lambda \psi_{k \rightarrow i \rightarrow j} + \psi_{i \rightarrow j \rightarrow k} &= x_{i \rightarrow j}
\end{align*} \]

When \( \lambda^3 + 1 \neq 0 \), i.e. \( \lambda \neq -1, (1 + \sqrt{3})/2 \) or \((1 - \sqrt{3})/2\), we have solutions as
\[ \begin{align*}
\psi_{i \rightarrow j \rightarrow k} &= \frac{1}{\lambda^3 + 1} (\lambda^3 x_{j \rightarrow k} - \lambda x_{k \rightarrow i} + x_{i \rightarrow j}) \\
\psi_{j \rightarrow k \rightarrow i} &= \frac{1}{\lambda^3 + 1} (\lambda^3 x_{k \rightarrow i} - \lambda x_{i \rightarrow j} + x_{j \rightarrow k}) \\
\psi_{k \rightarrow i \rightarrow j} &= \frac{1}{\lambda^3 + 1} (\lambda^3 x_{i \rightarrow j} - \lambda x_{j \rightarrow k} + x_{k \rightarrow i})
\end{align*} \]

Combining Eq. (19) and Eq. (20), we have
\[ \psi_{i \rightarrow j \rightarrow k} = \frac{1}{\lambda^3 + 1} \left( \lambda^3 x_{j \rightarrow k} - \lambda x_{k \rightarrow i} + x_{i \rightarrow j} \right), i \in \mathcal{N}_k \]
\[ \frac{1}{\lambda} x_{j \rightarrow k}, i \notin \mathcal{N}_k. \]

Thus, for any non-zero eigenvalue \( \lambda \) of \( M \) except \(-1, (1 + \sqrt{3})/2 \) and \((1 - \sqrt{3})/2\), it satisfies \( E^{(i)} \psi = \lambda \psi \), and the values of elements in \( \psi \) can be determined by Eq. (21).

Exploiting Theorem 2, we give an approach to save time and space cost for computing \( p_c^{(2)} \). In particular, the numbers of edges and length-2 directed paths in a network \( G \) are \( E = \frac{1}{2} \sum_{i=1}^N d_i \) and \( P_2 = \sum_{i=1}^N d_i (d_i - 1) \), respectively, where \( d_i \) is the degree of node \( i \). Thus, we reduce the size of the matrix to be computed by a factor of \( (\sum_{i=1}^N d_i^2 - \sum_{i=1}^N d_i)/(4 \sum_{i=1}^N d_i) \approx (d_i^2)/(4\langle d_i \rangle) \).

5. EXPERIMENTS

In this section, we empirically investigate the validity of our theoretical estimate for bond percolation threshold. First, we compare \( p_c^{(2)} \) with other theoretical indicators \( p_c^{(1)} \) and \( p_c^{(0)} \) on a class of synthetic networks generated by the forest fire model [19]. Next, we perform extensive experiments on 42 real networks to further explore the performance of these estimations.

We use the peak value of the second largest component (Eq. (1)) as the ground truth of the bond percolation threshold \( p_c \). For each network, the value of the empirical ground truth \( p_c \) is computed by 1000 independent Monte Carlo simulations of the percolation process as proposed by Newman and Ziff [22].

5.1 Forest Fire Model

The forest fire model [19] is a family of evolutionary networks, controlled by burning probability \( q \).

We denote the network at time \( t \) as \( G_t \). Initially (\( t = 1 \)), there is only one node in \( G_1 \). At time \( t > 1 \), there is a new node \( v \) joining the network \( G_{t-1} \) and generating \( G_t \). The node \( v \) establishes connections with other existing nodes through the following process: (1) Node \( u \) first selects an ambassador node \( v \) in \( G_{t-1} \) uniformly at random and connects to it. (2) We sample a random number \( a \in (0, 1) \). If \( a \leq q \), node \( u \) randomly chooses a neighbor of \( v \), which is not connected to \( u \) yet, and forms a link to it. Then repeat this step. If \( a > q \), this step ends and we label all nodes linking to \( u \) at this step as \( w_1, w_2, \ldots, w_k \); (3) Let \( w_1, w_2, \ldots, w_k \) sequentially be the ambassador of node \( u \) and apply step (2) for each of them recursively. A node in the process should not be visited a second time.

The generation process of the forest fire model describes new nodes joining social networks with an epidemic fashion. In addition, forest fire model shares a number of remarkable structural features with real networks, such as densification and shrinking diameters [19]. Thus, studying percolation on the forest fire model can enhance our understanding of spreading in realistic systems.
We first consider a case that the burning probability is very small, so that the network $G_1$ is sparse. In Figure 2a, we plot bond percolation threshold estimated by different approaches during the evolution of a forest fire model with $q = 0.01$. It shows that along with the growth of the network, the empirical estimate tends to 0.901. The predicted percolation thresholds based on the adjacency and non-backtracking matrices tend to 0.210 and 0.495, respectively, which are far from the empirical estimate. Although $p_{c}^{(1)}$ performs better than $p_{c}^{(0)}$ in sparse networks, the appearance of triangle structures largely weakens its effectiveness. The estimation based on the 2nd-order non-backtracking matrix shows remarkable improvement. At the end of the evolution, $p_{c}^{(2)}$ tends to 0.778, which corresponds to an improvement of roughly 30% from $p_{c}^{(1)}$. It indicates $p_{c}^{(2)}$ is evidently more precise than $p_{c}^{(1)}$ for sparse networks with triangles.

Next we test the impact of burning probability on our indicator. In Figure 2b, we show the percolation threshold given by the theoretical and simulated estimations versus burning probability on forest fire model with 5000 nodes. Each value is the average of 100 networks. We observe that, along with the growth of the burning probability, the empirical percolation threshold decreases, because of the increase of the number of edges. For the theoretical indicators, it is not surprising that $p_{c}^{(2)}$ always performs better than $p_{c}^{(1)}$ and $p_{c}^{(0)}$, especially in the regime of small burning probability. However, as burning probability becomes large, the precision of $p_{c}^{(2)}$ decreases. In addition, the values of $p_{c}^{(0)}$, $p_{c}^{(1)}$ and $p_{c}^{(2)}$ are getting close. This is because the growth of the burning probability introduces more and more simple cycles with length larger than 3, which would lead to over-estimating probability $\theta_{i \rightarrow j \rightarrow k}$ for $i \rightarrow j \rightarrow k$ in such a cycle. Thus, $p_{c}^{(2)}$ becomes less effective in predicting the percolation threshold.

### 5.2 Real Networks

For the purpose of better understanding the predictive power of different theoretical estimations in real networks, we evaluate bond percolation threshold predicted by $p_{c}^{(0)}$, $p_{c}^{(1)}$, $p_{c}^{(2)}$ and simulations on 42 real networks. The dataset includes social, infrastructural, animal and biological networks, which exhibit various topological properties, with size from 23 to 4941. In Table 1, we report the theoretical and empirical estimations for bond percolation threshold on the 42 real networks. All networks are treated as undirected unweighted networks. We only consider the largest connected component in each network. All datasets are from the Koblenz Network Collection (http://konect.uni-koblenz.de/).

Figure 3a reports values of $p_{c}^{(0)}$, $p_{c}^{(1)}$ and $p_{c}^{(2)}$ as functions of the empirical estimate for all 42 networks. It shows that $p_{c}^{(2)}$ is always closer to the empirical estimate than $p_{c}^{(1)}$ and $p_{c}^{(0)}$, which is in agreement with our theoretical analysis. In addition, we observe that there are some networks on which $p_{c}^{(2)}$ can attain the empirical estimate. However, $p_{c}^{(2)}$ is not always close enough to the empirical estimate, especially on networks in the regime of large percolation threshold.

Next, we continue to quantitatively analyze how much improvement our proposed method attains. We define relative error of the theoretical estimate as $(p_{c} - p_{c}^{(\alpha)})/p_{c}$ ($\alpha = 0,1$ or 2). Then, we plot cumulative distribution of relative errors of $p_{c}^{(0)}$, $p_{c}^{(1)}$ and $p_{c}^{(2)}$ in the collection of 42 networks, as shown in Figure 3b. We adopt the area under the curve (AUC) as a measurement for global performance. The results is that there is an improvement of 8% from $p_{c}^{(0)}$ to $p_{c}^{(1)}$, and 3% from $p_{c}^{(1)}$ to $p_{c}^{(2)}$.

In order to further understand the influence of the true percolation threshold on our theoretical prediction, we first divide the range of possible values of the empirical estimate in eight bins. Then, we calculate the average of the empirical estimate and relative errors, respectively, in each bin. Figure 4a reports the average of relative error with respect to the average of the empirical estimate. It implies that $p_{c}^{(2)}$ works well on the networks in the range of small true percolation threshold; while it becomes less predictive on networks with large true percolation threshold.

We also analyze the impact of density of networks on our theoretical prediction. Analogously, we divide the range of possible values of average degree in seven bins and separately calculate the means of average degree and relative errors in each bin. In Figure 4b, we plot the relative error as a function of average degree for all three theoretical estimates. There are two main conclusions from this figure. First, for all three theoretical predictions, they are more precise on networks with large average degree than on networks with small average degree. Second, on networks in the regime of small average degree, $p_{c}^{(2)}$ in general has a better improvement from $p_{c}^{(1)}$ than on networks with relatively large average degree. It can be accounted as on dense net-
works, $p_c^{(0)}$ is already a good approximation for percolation threshold, and under such case $p_c^{(0)}$, $p_c^{(1)}$ and $p_c^{(2)}$ would be very close to each other.

At last, we discover the performance of theoretical predictions in different categories of networks. In Figure 4c, we give the average of relative errors in each group of networks for all three theoretical estimations. It can be found that theoretical predictions are closer to true percolation threshold in animal and social networks than in infrastructural and biological networks.

6. DISCUSSION AND FUTURE WORK

Although $p_c^{(2)}$ outperforms other theoretical estimations for bond percolation threshold, it is still not sufficiently precise on some real networks, due to the appearance of cycles larger than 3. A direct idea to further improve the precision is to use the reciprocal of the largest eigenvalue of higher order non-backtracking matrices as the predictor for the percolation threshold. According to a similar deduction as in Section 4.1, applying a $g$-th-order non-backtracking matrix should avoid overestimating probability that node $i$ connects to the giant component following a length-$g$ path. Moreover, Theorem 1 also theoretically guarantees percolation threshold estimated by $g$-th-order non-backtracking matrix is non-decreasing with respect to $g$. In order to elaborate it empirically, we consider bond percolation on a small network with only 50 nodes, generated by the Barabási-Albert network model [2]. In Figure 5, we give the expected relative sizes of the first and second largest cluster as a function of probability $p$. In addition, we derive theoretical estimations for percolation threshold based on $g$-th-order non-backtracking matrix $(0 \leq g \leq 5)$, denoted separately as $p_c^{(g)}$. We can observe $p_c^{(2)}$ is a better approximation than $p_c^{(3)}$.

However, theoretical estimation based on high-order non-backtracking matrices still suffers the following two drawbacks. First, along with the increase of $g$, the size of the $g$-th-order non-backtracking matrix grows dramatically. It would lead to a prohibitive cost for computing the largest eigenvalue for large-scale networks. Second, with a sufficiently large $g$, $p_c^{(g)}$ could be larger than the true percolation threshold. As shown in Figure 5, $p_c^{(4)}$ and $p_c^{(5)}$ are already slightly larger than the peak position of $S_2(p)$. It may be because the heuristic analysis in Section 4.1 will be no longer hold for high-order non-backtracking matrices. Thus, possible directions of future work include finding efficient techniques for generating and analyzing high-order non-backtracking matrices, and determining at which order $p_c^{(g)}$ is the closest to but still smaller than the true percolation threshold. It is also possible that we can depart from non-backtracking matrices and use some brand-new techniques to develop the theoretical assessment of the percolation threshold.

In addition, the current lower bound derivation is based on heuristic equations. A more mathematically sound technique to show the effectiveness of high-order non-backtracking matrices and the exact conditions that support this effectiveness is also a future direction.

7. ACKNOWLEDGMENTS

The authors thank anonymous reviewers for their valuable comments and helpful suggestions. Yuan Lin and Zhongzhi Zhang are supported by the National Natural Science Foundation of China (Grant No. 11275049). Wei Chen is partially supported by the National Natural Science Foundation of China (Grant No. 61433014).
References


