Int J Data Sci Anal (2016) 1:3–16
DOI 10.1007/s41060-015-0001-y

REGULAR PAPER

Speeding-up node influence computation for huge social networks

Masahiro Kimura[1] · Kazumi Saito[2] · Kouzou Ohara[3] · Hiroshi Motoda[4,5]

Received: 3 October 2015 / Accepted: 22 November 2015 / Published online: 7 January 2016

Abstract We address the problem of efficiently estimating the influence degree for all the nodes simultaneously in the network under the SIR setting. The proposed approach is a further improvement over the existing work of the bond percolation process which was demonstrated to be very effective, i.e., three orders of magnitude faster than direct Monte Carlo simulation, in approximately solving the influence maximization problem. We introduce two pruning techniques which improve computational efficiency by an order of magnitude. This approach is generic and can be instantiated to any specific diffusion model. It does not require any approximations or assumptions to the model that were needed in the existing approaches. We demonstrate its effectiveness by extensive experiments on two large real social networks. Main finding includes that different network structures have different epidemic thresholds and the node influence can identify influential nodes that the existing centrality measures cannot. We analyze how the performance changes when the network structure is systematically changed using synthetically generated networks and identify important factors that affect the performance.

1 Introduction

Studies of the structure and functions of large complex networks have attracted a great deal of attention in many different fields such as sociology, biology, physics and computer science [23]. It has been recognized that developing new methods/tools that enable us to quantify the importance of each individual node in a network is crucially important in pursuing fundamental network analysis. Networks mediate the spread of information, and it sometimes happens that a small initial seed cascades to affect large portions of networks [29]. Such information cascade phenomena are observed in many situations: for example, cascading failures can occur in power grids (e.g., the August 10, 1996 accident in the western US power grid), diseases can spread over networks of contacts between individuals, innovations and rumors can propagate through social networks, and large grass-roots social movements can begin in the absence of centralized control (e.g., the Arab Spring). Understanding these phenomena involves dynamic analysis of diffusion process. Thus, the node influence with respect to information cascade is a useful measure of node importance, and it is different from the existing centralities because diffusion dynamics are involved.

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research—United States, under Grant No. AOARD-13-4042, and JSPS Grant-in-Aid for Scientific Research (C) (No. 26330352), Japan.

✉ Masahiro Kimura
kimura@rins.ryukoku.ac.jp

Kazumi Saito
k-saito@u-shizuoka-ken.ac.jp

Kouzou Ohara
ohara@it.aoyama.ac.jp

Hiroshi Motoda
motoda@ar.sanken.osaka-u.ac.jp

[1] Department of Electronics and Informatics, Ryukoku University, Otsu 520-2194, Japan

[2] School of Administration and Informatics, University of Shizuoka, Shizuoka 422-8526, Japan

[3] Department of Integrated Information Technology, Aoyama Gakuin University, Kanagawa 229-8558, Japan

[4] Institute of Scientific and Industrial Research, Osaka University, Osaka 567-0047, Japan

[5] School of Computing and Information Systems, University of Tasmania, Hobart, TAS 7005, Australia

Basic models of information diffusion over a network often assume that each node has three states, *susceptible*, *infective*, and *recovered* from the analogy of epidemiology. A node in the susceptible state means that it has not yet been influenced with the information. A node in the infective state means that it is influenced and can propagate the information to its neighbor nodes. A node in the recovered state means that it can no longer propagate the information once it has been influenced with the information, i.e., immune. The *SIR model* is a typical one among such basic models and well exploited in many fields [23]. To be more concrete, the SIR model is a discrete-time stochastic process model, and assumes that a susceptible node becomes infective with a certain probability when its neighbor nodes get infective, and becomes subsequently recovered. In particular, it is known that the SIR model on a network can be exactly mapped onto a *bond percolation process* on the same network [15,23].

The dynamical behaviors of the SIR model have been widely studied in physics literature. One such important analysis is to examine the *epidemic threshold* p^*_G of a network G, where most nodes of the network remain uninfected (i.e., a small outbreak) if the probability that a susceptible node receives information from its infective neighbor is smaller than p^*_G, and the number of infected (recovered) nodes rapidly increase (i.e., a large outbreak) if the probability becomes greater than p^*_G [23]. We must be able to estimate node influence very efficiently to make this kind of analysis feasible because we need to estimate the average influence degree. In this paper, we focus on the node influence based on the SIR model, and regard it as one of the centrality measures and refer to it as the *influence degree centrality* for convenience sake.

Let $G = (V, E)$ be a directed network, where V and E $(\subset V \times V)$ stand for the sets of all nodes and links, respectively. For the SIR model over G, the *influence degree* $\sigma_G(v)$ of a node $v \in V$ is defined as the expected number of recovered nodes at the end of the information diffusion process (i.e., when there are no nodes in the infective state), assuming that at the initial time $t = 0$, only v is in infective state and all other nodes are in susceptible state. In order to examine the influence degree centrality in G, it is necessary to estimate the influence degree $\sigma_G(v)$ for every single node $v \in V$. We refer to $\sum_{v \in V} \sigma_G(v)/|V|$ as the *average influence degree* of G. In order to examine the epidemic threshold of G, we must calculate the average influence degree of G for various values of diffusion probability of the SIR model. Note that it is difficult to calculate the influence degree exactly since the SIR model is defined by a stochastic process [9,17,18]. In general, the influence degree is approximately estimated through a number of simulations, while the existing centrality measures are exactly calculated once the network structure is given. Thus, it is an important research issue to estimate the influence degree $\{\sigma_G(v) \mid v \in V\}$ quite efficiently.

In this paper,[1] we propose an improved method of efficiently estimating the influence degree of all the nodes in network G, $\{\sigma_G(v) \mid v \in V\}$ simultaneously under the SIR model setting. Many of the existing techniques (see Sect. 2 for more details) are designed for a specific diffusion model, e.g., independent cascade or linear threshold models, and introduce approximations to the influence estimation, e.g., use of sampling and/or assumptions to the model chosen, e.g., assuming that the diffusion probability is small enough to allow for linear approximation, considering only the shortest diffusion path or the maximum influence path between a pair of nodes is enough, etc. To the best of our knowledge, two groups of work, one [17,18] (called bond percolation) and the other [9] (called new greedy algorithm) are the only ones that do not introduce any approximations and/or assumptions to the model. Both use the same idea, and in this paper we call it *BP method* for short.

The BP method was shown to be very efficient, three orders of magnitude faster than direct Monte Carlo simulation in computing the node influence degree [17,18]. Our contribution is to have made the influence degree centrality $\{\sigma_G(v) \mid v \in V\}$ estimation in network G even faster by an order of magnitude by introducing two new pruning techniques: the *redundant-edge pruning (REP) technique* and the *marginal-component pruning (MCP) technique*. The REP technique prunes redundant edges for reachability analysis among three vertices and the MCP technique recursively prunes vertices with in-degree 1 or out-degree 1 from the quotient graph which is obtained by decomposing the graph (realized by the corresponding bond percolation process) into the strongly connected components (SCCs).

We extensively evaluate the proposed method using two large real social networks, compare the computation time,[2] and show that the proposed method significantly outperforms the existing BP method. The MCP technique is found to be more effective than the REP technique. Use of both techniques is always better than the single use of either technique. We further examine how the performance of the two pruning techniques changes as the network structure changes. For this purpose we extend the BA and CNN methods, and systematically generate synthetic networks with different structure. We reconfirm the above results and identify the important factors that are decisive in controlling the performance.

The proposed method inherits the good feature of the BP method. It is a generic framework to estimate the influence degree centrality under the SIR model setting without need for any approximations and assumptions. With this improved efficiency, it is now possible to estimate the node influence

[1] This paper extends the work [19] presented in the 2014 International Conference on Data Science and Advanced Analytics (DSAA'14).

[2] The estimation accuracy of $\{\sigma_G(v) \mid v \in V\}$ is the same because of no new approximations and assumptions introduced.

of every single node of a network with one million nodes and analyze the existence of epidemic threshold. We further confirm that the inluence degree centrality can identify nodes that are deemed indeed influential which are not identifiable by the existing centrality measures.

The paper is organized as follows. We briefly explain the related work in Sect. 2 and the BP method in Sect. 3. We then introduce the proposed method (REP and MCP techniques) in Sect. 4. The experimental results for real networks are given in Sect. 5, and the performance analysis for synthetic networks is given in Sect. 6. We conclude the paper in Sect. 7 summarizing the main achievement and future plans.

2 Related work

Developing efficient methods that enable us to find influential nodes in a social network is a fundamental problem in social network analysis, and many studies have been made on this problem.

Several centraility measures have been proposed in the field of social science. The well-known centrality measures include, but not limited to, degree centrality [12], eigenvector centrality [3], Katz centrality [14], PageRank [5], closeness centrality [12], betweenness centrality [12], and topological centrality [32]. However, some centrality measures (e,g., closeness centrality and betweenness centrality) require to use the global structure of a network for computing the value of each node, and their computation become harder as the size of a network increases. Thus, several researchers try to efficiently approximate such centralities [2,11,25]. Notable feature of the existing centrality measures is that they all are defined only by network topology. Node influence is different from them in that it is defined through dynamical processes of a network. Therefore, it can provide new insights into information diffusion phenomena such as existence of epidemic threshold which the topology-based centrality measures can never do.

Estimating influence degree is a sub-problem in the *influence maximization problem*, which has recently attracted tremendous interest in the field of social network mining [7]. The task of the influence maximization problem is to identify a limited number of seed nodes that together maximize the expected spread of influence over G. Kempe et al. [15] first formalized this problem and presented a polynomial solution by using a greedy search strategy. Since then, many researchers have proposed various techniques for improving the efficiency in finding high-quality approximate solutions [8–10,13,17,20,24,30]. Recently, Borgs et al. [4] provided a fast algorithm running in quasilinear time, and mathematically proved its high performance. Song et al. [27] introduced a diffusion model to accommodate link weights, and investigated the influence maximization problem for a mobile social network where individuals communicate with one another using mobile phones. Zhou et al [31] established new upper bounds to significantly reduce the number of Monte-Carlo simulations in greedy algorithms and presented a fast algorithm based on the upper bounds. The techniques developed so far include both of those that aim at improving the efficiency of estimating the expected spread for a given seed node set and those that aim at improving the efficiency of the search for the seed node set. The proposed method belongs to the former, but differs from the others in that it can obtain the influence degree of all the nodes simultaneously. Thus, it can naturally be applied to the influence maximization problem through the greedy search. It can also be utilized for identifying super-mediators of information diffusion in social networks [26].

3 BP method

We briefly revisit the BP method (see [18] for more detail). A bond percolation process on a given network $G = (V, E)$ is the process in which each link of G is stochastically designated either “occupied ” or “unoccupied” according to some probability distribution. The occupation probability distribution is determined according to the assumed information diffusion model and its associated parameter values. Now, we consider M times of bond percolation processes. Let E_m ($\subset E$) denote the set of occupied links at the m-th bond percolation process and let G_m denote the network (V, E_m).

Figure 1 illustrates a bond percolation process and a resulting network. The solid arrows in the network at the left in Fig. 1 denote occupied links, while the broken arrows denote unoccupied ones. This process results in the network at the right in Fig. 1. For any node $v \in V$, we define $\bar{\sigma}_G(v)$ by

$$\bar{\sigma}_G(v) = \frac{1}{M} \sum_{m=1}^{M} |R_{G_m}(v)|, \tag{1}$$

where $R_{G_m}(v)$ stands for the set of *reachable* nodes from v on G_m, and $|R_{G_m}(v)|$ is the number of nodes in $R_{G_m}(v)$.

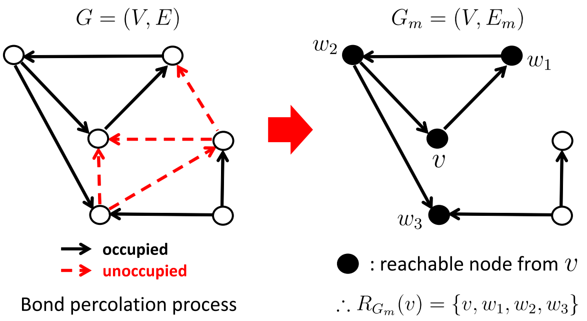


Fig. 1 A network resulted from a bond percolation

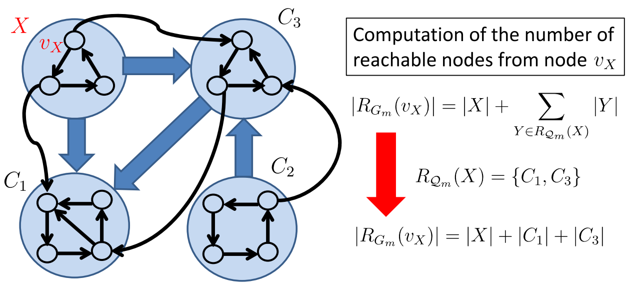

X
v_X
C_3
C_1
C_2
Computation of the number of reachable nodes from node v_X
$|R_{G_m}(v_X)| = |X| + \sum_{Y \in R_{\mathcal{Q}_m}(X)} |Y|$
$R_{\mathcal{Q}_m}(X) = \{C_1, C_3\}$
$|R_{G_m}(v_X)| = |X| + |C_1| + |C_3|$

Fig. 2 An example of quotient graph

Here, we say that a node $w \in V$ is *reachable* from node v on G_m if there exists a path from v to w in the network G_m. For example, in the network at the right in Fig. 1, the reachable nodes from node v are v, w_1, w_2, w_3. Thus, $R_{G_m}(v) = \{v, w_1, w_2, w_3\}$, and $|R_{G_m}(v)| = 4$.

It is known [23] that the influence degree $\sigma_G(v)$ can be estimated by $\bar{\sigma}_G(v)$ with a reasonable accuracy if M is sufficiently large. [3] Here note that the bond percolation technique decomposes each network G_m into its SCCs, where an SCC (strongly connected component) is a maximal subset C of V such that for all $v, w \in C$ there is a path from v to w on G_m. Note that $R_{G_m}(v) = R_{G_m}(w)$ $(v, w \in C)$. Thus, we can obtain $R_{G_m}(v)$ for any node $v \in V$ by calculating $R_{G_m}(v)$ for only one node v in each component C. Let $\mathcal{Q}_m = (\mathcal{C}_m, \mathcal{E}_m)$ be the quotient graph obtained by the SCC decomposition of $G_m = (V, E_m)$, where $\mathcal{C}_m$ is the set of all the SCCs of G_m, and $\mathcal{E}_m$ $(\subset \mathcal{C}_m \times \mathcal{C}_m)$ is the set of edges in $\mathcal{Q}_m$, i.e., $(C, D) \in \mathcal{E}_m$ if there exist some pair of nodes $v \in C$ and $w \in D$ which satisfies $(v, w) \in E_m$. Note that the quotient graph $\mathcal{Q}_m$ is a DAG (directed acyclic graph). For each component $C \in \mathcal{C}_m$, we can also consider the set of *reachable* components from C on $\mathcal{Q}_m$, which is denoted by $R_{\mathcal{Q}_m}(C)$. Here, a component $D \in \mathcal{C}_m$ is an element of $R_{\mathcal{Q}_m}(C)$ when there exists a path from vertex C to vertex D on the graph $\mathcal{Q}_m$. Then, for any node $v \in C$, we can calculate the number of reachable nodes from v on the network G_m by

$$|R_{G_m}(v)| = |C| + \sum_{D \in R_{\mathcal{Q}_m}(C)} |D|. \tag{2}$$

For example, Fig. 2 shows a quotient graph consisting of four components X, C_1, C_2, and C_3, in which block arrows are edges in this quotient graph that connect components and narrow arrows are links in the original networks. Then, the number of reachable nodes from node $v_X \in X$ is given as $|R_{G_m}(v_X)| = |X| + |C_1| + |C_3|$ because a set of reachable components from X are $R_{\mathcal{Q}_m}(X) = \{C_1, C_3\}$.

In case of the MCP technique as described later, Eq. (2) is replaced as follows:

$$|R_{G_m}(v)| = h_m(C) + \sum_{D \in R_{\mathcal{Q}_m}(C)} h_m(D), \tag{3}$$

where $h_m(D)$ is initially set to $h_m(D) = |D|$ for any component $D \in \mathcal{C}_m$, and it is to be updated iteratively. Note that in general,

$$|R_{G_m}(v)| \neq |C| + \sum_{D \in \mathcal{F}_m(C)} |R_{G_m}(w_D)|$$

for any node $v \in C$, unless $\mathcal{Q}_m$ is a tree. Here, $\mathcal{F}_m(C)$ denotes the set of *child components* of a component C in G_m, defined by

$$\mathcal{F}_m(C) = \{D \in \mathcal{C}_m \mid (C, D) \in \mathcal{E}_m\},$$

and w_D stands for a representative node of a component $D \in \mathcal{C}_m$.

In summary, the existing BP method first computes the subset $R_{\mathcal{Q}_m}(C)$ of $\mathcal{C}_m$ for each component $C \in \mathcal{C}_m$ by following the edges on the quotient graph $\mathcal{Q}_m$, then calculates $|R_{G_m}(v_C)|$ for only one node $v_C \in C$ by using Eq. (2), and finally sets $|R_{G_m}(v)|$ as follows:

$$|R_{G_m}(v)| \leftarrow |R_{G_m}(v_C)|, \quad (\forall v \in C \setminus \{v_C\}).$$

4 Proposed method

We enhance the existing BP method by introducing two techniques: redundant-edge pruning (REP) and marginal-component pruning (MCP). Again, we focus on the quotient graph $\mathcal{Q}_m = (\mathcal{C}_m, \mathcal{E}_m)$ of the network $G_m = (V, E_m)$ constructed through the m-th bond percolation process.

The REP technique performs pruning redundant edges for reachability analysis among three components in G_m, i.e., three vertices on $\mathcal{Q}_m$. For each component $C \in \mathcal{C}_m$ in G_m, an edge $(C, D) \in \mathcal{E}_m$ is called a *redundant edge* with respect to C if a component D is reachable from C via another component $X \in \mathcal{C}_m$. This situation is illustrated in Fig. 3, in which a component D is reachable from a component C via two edges (C, X) and (X, D). Let $\mathcal{EP}_{\mathcal{Q}_m}(C)$ denote the set of all redundant edges with respect to $C \in \mathcal{C}_m$. Then, we have

$$\mathcal{EP}_{\mathcal{Q}_m}(C) = \left\{ (C, D) \in \mathcal{E}_m \;\middle|\; D \in \bigcup_{X \in \mathcal{F}_m(C)} \mathcal{F}_m(X) \right\}. \tag{4}$$

[3] It is shown that setting M to a few thousands usually gives good accuracy in experiments using real social networks (see [18]).

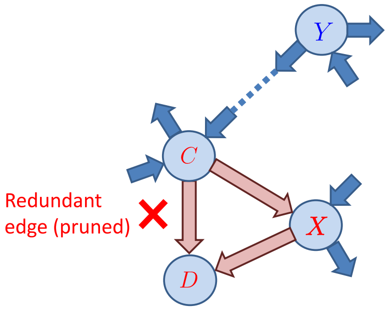


Fig. 3 Redundant edge pruned by the REP technique

Note that if an edge $(C, D) \in \mathcal{E}_m$ is a redundant edge with respect to a component C, i.e., $(C, D) \in \mathcal{EP}_{\mathcal{Q}_m}(C)$, then it is possible to correctly compute $R_{\mathcal{Q}_m}(C)$ without using the edge (C, D). For example, in Fig. 3, $R_{\mathcal{Q}_m}(Y)$, reachable components from a component Y can be correctly computed without using the redundant edge (C, D). Thus, the REP technique prunes the set of redundant edges $\mathcal{EP}_{\mathcal{Q}_m}(C)$ when computing $R_{\mathcal{Q}_m}(C)$ for any component $C \in \mathcal{C}_m$. If interpreted as a network motif [22], the REP technique detects such 3-vertices $\{C, X, D\}$ on graph $\mathcal{Q}_m$ that form a feedforward motif pattern $\{(C, X), (X, D), (C, D)\}$, and prunes its short-cut edge (C, D) from them. Let $\mathcal{EP}_{\mathcal{Q}_m}$ denote the set of all the redundant edges, i.e.,

$$\mathcal{EP}_{\mathcal{Q}_m} = \bigcup_{C \in \mathcal{C}_m} \mathcal{EP}_{\mathcal{Q}_m}(C).$$

In summary, the REP technique computes the set of all the redundant edges $\mathcal{EP}_{\mathcal{Q}_m}$, and replaces the set of edges on $\mathcal{Q}_m$ as follows:

$$\mathcal{E}_m \leftarrow \mathcal{E}_m \setminus \mathcal{EP}_{\mathcal{Q}_m}.$$

The MCP technique recursively performs pruning components with in-degree 1 or out-degree 1 in the network G_m. Here, we define the sets of components with in-degree 1 and out-degree 1 by Eqs. (5) and (6), respectively:

$$\mathcal{CPI}_{\mathcal{Q}_m} = \{C \in \mathcal{C}_m \mid |\mathcal{B}_m(C)| = 1, |\mathcal{F}_m(C)| = 0\}, \quad (5)$$
$$\mathcal{CPO}_{\mathcal{Q}_m} = \{C \in \mathcal{C}_m \mid |\mathcal{F}_m(C)| = 1, |\mathcal{B}_m(C)| = 0\}. \quad (6)$$

Here, $\mathcal{B}_m(C)$ denotes the set of all parent components of C,

$$\mathcal{B}_m(C) = \{D \in \mathcal{C}_m \mid (D, C) \in \mathcal{E}_m\}.$$

We define the set $\mathcal{CP}_{\mathcal{Q}_m}$ of components with in-degree 1 or out-degree 1 in G_m by

$$\mathcal{CP}_{\mathcal{Q}_m} = \mathcal{CPI}_{\mathcal{Q}_m} \cup \mathcal{CPO}_{\mathcal{Q}_m}.$$

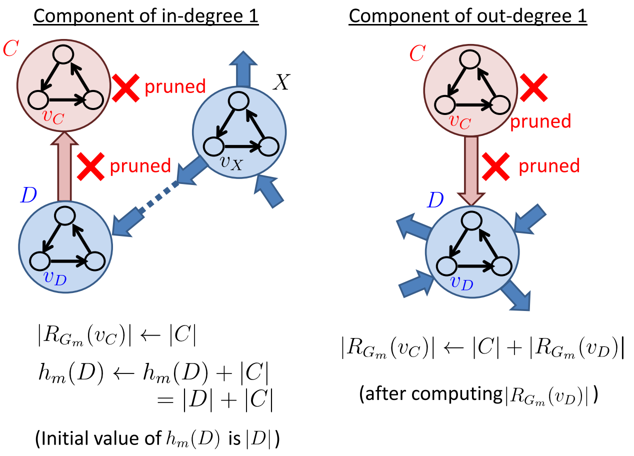

Component of in-degree 1
Component of out-degree 1
C
pruned
X
pruned
v_C
v_X
D
v_D
$|R_{G_m}(v_C)| \leftarrow |C|$
$h_m(D) \leftarrow h_m(D) + |C|$
$= |D| + |C|$
(Initial value of $h_m(D)$ is $|D|$)
C
pruned
pruned
v_C
D
v_D
$|R_{G_m}(v_C)| \leftarrow |C| + |R_{G_m}(v_D)|$
(after computing $|R_{G_m}(v_D)|$)

Fig. 4 Pruning components and edges by the MCP technique

Below we explain two basic ideas of the MCP technique. First, for any component $C \in \mathcal{CPI}_{\mathcal{Q}_m}$ with in-degree 1, we can easily prove the following properties:

1. $|R_{G_m}(v)| = |C|$ for any $v \in C$.
2. Setting $h_m(D) \leftarrow h_m(D) + |C|$ for the unique parent component $D \in \mathcal{B}_m(C)$, $|R_{G_m}(v_X)|$ is obtained by

$$|R_{G_m}(v_X)| = h_m(X) + \sum_{Y \in R_{\mathcal{Q}_m}(X) \setminus \{C\}} h_m(Y)$$

(see Eq. (3)) for any component $X \in \mathcal{C}_m \setminus \{C\}$, where v_X stands for a representative node of X.

For example, at the left in Fig. 4, component C is the one with in-degree 1, and $|R_{G_m}(v_C)| = |C|$ for its representative node $v_C \in C$. Then, even if we prune C and its unique edge (D, C), we can correctly compute the number of nodes reachable from the representative node of component X, according to the above definition, by setting $h_m(D)$ as $h_m(D) \leftarrow |D| + |C|$.

Second, for any component $C \in \mathcal{CPO}_{\mathcal{Q}_m}$ of out-degree 1, we can easily prove that if $|R_{G_m}(v_D)|$, $(v_D \in D)$ is given for the unique child component $D \in \mathcal{F}_m(C)$, then $|R_{G_m}(v_C)|$, $(v_C \in C)$ is obtained by

$$|R_{G_m}(v_C)| = |C| + |R_{G_m}(v_D)|$$

without computing $R_{\mathcal{Q}_m}(C)$ by following the edges on $\mathcal{Q}_m$. This is illustrated at the right in Fig. 4, in which component C is the one with out-degree 1 and its unique child is component D. Then, it is obvious that even if we prune C and its unique edge (C, D) from this quotient graph, it does not affect computation of $R_{\mathcal{Q}_m}(X)$ for any component $X \in \mathcal{C}_m$. Therefore, it is possible to prune the components with in-degree 1 or out-degree 1 in G_m from $\mathcal{C}_m$ when computing $R_{\mathcal{Q}_m}(C)$ for any component $C \in \mathcal{C}_m$.

For a component $X \in \mathcal{C}_m$, let $\mathcal{IE}_{\mathcal{Q}_m}(X)$ be the set of all edges attached to X in $\mathcal{Q}_m$. We define the operation of pruning a component $C \in \mathcal{C}_m$ in graph $\mathcal{Q}_m$ by

$$\mathcal{Q}_m \ominus C \ = \ (\mathcal{C}_m \setminus \{C\}, \ \mathcal{E}_m \setminus \mathcal{IE}_{\mathcal{Q}_m}(C)).$$

Evidently, after pruning a component C, there might exist some component $D \in \mathcal{C}_m$ such that $D \notin \mathcal{CP}_{\mathcal{Q}_m}$ and $D \in \mathcal{CP}_{\mathcal{Q}_m \ominus C}$. Thus, the MCP technique need recursively perform pruning components. In summary, unless $|\mathcal{CP}_{\mathcal{Q}_m}| = 0$, the MCP technique recursively selects a component $C \in \mathcal{CP}_{\mathcal{Q}_m}$, and prunes C by

$$\mathcal{Q}_m \leftarrow \mathcal{Q}_m \ominus C$$

after setting first,

$$|R_{G_m}(v_C)| \leftarrow |C|, \ \ (v_C \in C)$$
$$h_m(D) \leftarrow h_m(D) + |C|$$

for the unique parent component $D \in \mathcal{B}_m(C)$ if $C \in \mathcal{CPI}_{\mathcal{Q}_m}$, and second,

$$|R_{G_m}(v_C)| \leftarrow |C| + |R_{G_m}(v_D)|$$

when $|R_{G_m}(v_D)|$, $(v_D \in D)$ has been computed for the unique child component $D \in \mathcal{F}_m(C)$ if $C \in \mathcal{CPO}_{\mathcal{Q}_m}$.

In our proposed method, the REP technique is applied before the MCP techniques, because it is naturally conceivable that the REP technique increases the number of components with in-degree 1 or out-degree 1. Clearly we can individually incorporate these techniques into the existing BP method. Hereafter, we refer to the proposed method without the MCP technique as the REP method, and the proposed method without the REP technique as the MCP method. Since it is difficult to analytically examine the effectiveness of these techniques, we empirically evaluate the computational efficiency of these three methods in comparison with the existing BP method.

5 Experiments

We evaluated the effectiveness of the proposed method using large real networks.

5.1 Network datasets

We employed two large social networks, where all the networks are represented as directed graphs. Here, we adopt the notation for a link in which the link creator is the target node in order to emphasize the direction of information flow.

The first one is a network extracted from "@cosme",[4] a Japanese word-of-mouth communication site for cosmetics, in which each user page can have *fan links*. A fan link (u, v) means that user v registers user u as her favorite user. We traced up to ten steps in the fan-link network from a randomly chosen user in December 2009 and extracted a large weakly connected network consisting of 45,024 nodes and 351,299 directed links. We refer to this directed network as the Cosme network.

The second one is a network extracted from a set of message posts from "Japanese Twitter",[5] which totally consists of 201,297,161 messages (tweets) made by 1,088,040 active users (micro-bloggers or twitters who posted no less than 200 messages) during the period of almost three weeks (from March 5, 2011 to March 24, 2011), when the massive earthquake and consequent tsunami in eastern Japan occurred on March 11, 2011. We used the network constructed from the *follower links* between these users, which resulted in a network consisting of 1,088,040 nodes and 157,371,628 directed links. We refer to this huge network as the Twitter network.

5.2 Experimental settings

One of the simplest models of the SIR framework is the *independent cascade (IC) model* [15], where nodes have two states (*active* and *inactive*) and can switch their states only from inactive to active. The IC model on a network $G = (V, E)$ has a *diffusion probability* $p_{u,v}$ with $0 < p_{u,v} < 1$ for each link $(u, v) \in E$ as a parameter. Suppose that a node $u \in V$ first becomes active at time-step t, it is given a single chance to activate each currently inactive child node $v \in V$ with $(u, v) \in E$, and succeeds with probability $p_{u,v}$. If u succeeds, then v will become active at time-step $t + 1$. If multiple parent nodes of v first become active at time-step t, then their activation trials are sequenced in an arbitrary order, but all performed at time-step t. Whether u succeeds or not, it cannot make any further trials to activate v in subsequent rounds. The process terminates if no more activations are possible. It is well known [15] that the IC model on G for diffusion probabilities $\{p_{u,v} \mid (u, v) \in E\}$ is equivalent to the bond percolation process on G for occupation probabilities $\{p_{u,v} \mid (u, v) \in E\}$, that is, these two models have the same probability distribution for the final active (recovered) nodes. In the experiments, we employed the IC model.

Now, we explain the setting of diffusion probabilities $\{p_{u,v} \mid (u, v) \in E\}$ for the IC model. We draw $\{p_{u,v} \mid (u, v) \in E\}$ independently assuming a generative model according to

[4] http://www.cosme.net/.

[5] http://twitter.jp.

the beta distribution with a mean of μ. Note that the beta distribution is the conjugate prior probability distribution for the Bernoulli distribution corresponding to a single toss of a coin. Then, the average occupied probability of the corresponding bond percolation process over G reduces to μ. Actually, this formulation is equivalent to assigning a uniform value μ to the diffusion probability $p_{u,v}$ for any link, i.e., $p_{u,v} = \mu, \forall(u, v) \in E$. In the experiments, we investigated the four cases of very low, low, medium, and high diffusion probabilities:

$$\mu = \frac{r}{\bar{d}_G}, \quad (r = 0.25, 0.5, 1.0, 2.0),$$

where $\bar{d}_G$ is the mean out-degree of network G. We refer r to the *diffusion probability factor*.

For the parameter M of the proposed method, we found $M = 1000$ to be a reasonable value for estimating the influence degree for the Cosme and Twitter networks through our preliminary experiments. Thus, we used $M = 1000$ unless otherwise stated.

In the next subsection, we explain experimental results for computation time. All our experimentation was undertaken on a single PC with Intel(R) Xeon(R) CPU X5690 @ 3.474 GHz, with 198 GB of memory, running under Linux.

5.3 Efficiency evaluation

First, we evaluated the efficiency of the proposed method. We compared the computation time of the proposed, REP, MCP, and existing BP methods. All of them are based on the bond percolation process on the same network G, and have the same accuracy for the same M (see Eq. (1)). Here, we used $M = 100$ trials and evaluated the time for each trial (corresponding to $M = 1$), because the existing BP method needed much time for the Twitter network. Figure 5 shows the computation time of each method as a function of diffusion probability factor r, where the average values are plotted and the standard deviations are indicated by the error bars. The results show that the MCP technique can always be useful although the REP technique is not necessarily effective alone. However, the proposed method, which incorporates both techniques, always performs the best. The Twitter network requires much longer computation time than the Cosme network since the former is much larger than the latter. It is in particular important to reduce the processing time in case of large diffusion probability μ since the processing time in general increases as μ becomes larger. In case of $r = 2.0$, the proposed method is about 18 times faster than the existing BP method on average for the Cosme network. Moreover, when using $M = 1$ in the Twitter network for $r = 2.0$, the proposed method requires only about 2 min while the existing BP method needs about 20 min. Thus, for $M = 1000$, the existing BP method would have needed about two weeks, while the proposed method would have required only about one day and a half. Compared to the existing BP method, the proposed method has smaller standard deviations, especially for the diffusion probabilities with medium and high values. When the diffusion probability takes a large value, the information diffusion path length changes substantially for each trial as seen in the next experiment (see Fig. 6). This fluctuation is attributed to whether or not information diffusion paths in network G arrive at several marginal components of G, that is, we conjecture that the structure of quotient graph $\mathcal{Q}_m$ substantially changes for each trial m. In general, it takes more time to trace down longer paths for identifying $R_{\mathcal{Q}_m}(C)$ in the BP framework. Since the MCP technique attempts to prune such marginal components in advance, we can expect that the MCP method has smaller standard deviations than the existing BP method. Further, since the REP technique finds candidates of marginal components, we can conjecture

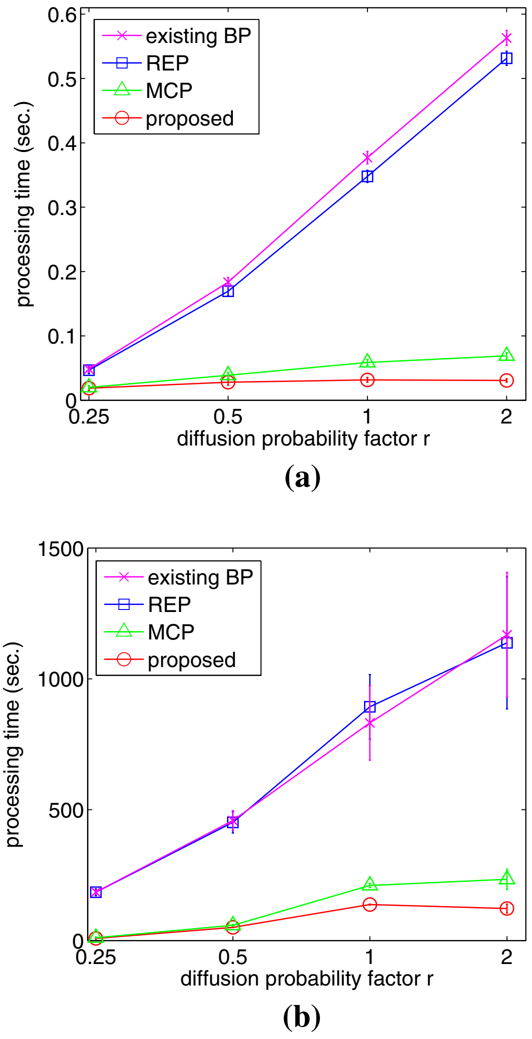
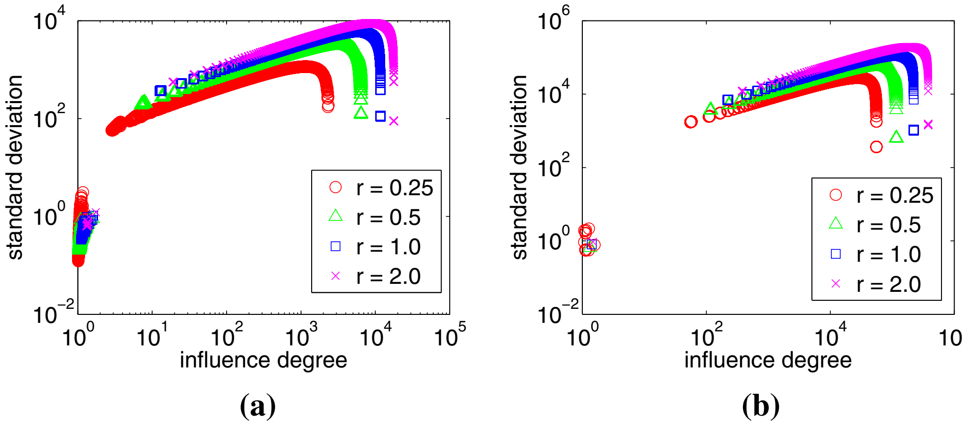
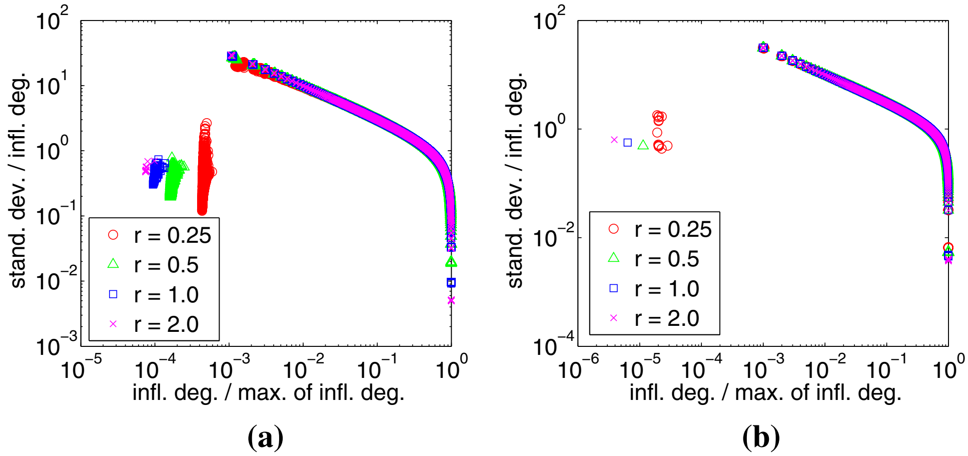


Fig. 5 Computation time comparison. **a** Cosme network, **b** Twitter network



(a) **(b)**

Fig. 6 Results for "influence degree versus standard deviation". **a** Cosme network, **b** Twitter network



(a) **(b)**

Fig. 7 Relation between $\bar{\sigma}_G^1(v)$ and $\bar{s}_G^1(v)$.**a** Cosme network, **b** Twitter network

that the proposed method combining both the REP and MCP techniques is more stable than the other three methods in terms of computation time. These results demonstrate the effectiveness of the proposed method.

Next, we investigated a global picture of the node influence estimation of the BP method framework with $M = 1000$ for the Cosme and Twitter networks. Using the proposed method with $M = 1000$, we estimated the influence degree of each node v in network G by $\bar{\sigma}_G(v)$ (see Eq. (1)), and then calculated the standard deviation $\bar{s}_G(v)$ of samples $\{|R_{G_m}(v)|\}$ for each $v \in V$. Figure 6 plots the pair $(\bar{\sigma}_G(v), \bar{s}_G(v))$ for all $v \in V$. We first see that all the results are qualitatively very similar, and these plots can provide a tool of network structure analysis. In fact, there exists a critical influence degree $\bar{\sigma}_G(v_*)$ for network G such that standard deviation $\bar{s}_G(v)$ is an increasing function of influence degree $\bar{\sigma}_G(v)$ if $\bar{\sigma}_G(v) \leq \bar{\sigma}_G(v_*)$, but $\bar{s}_G(v)$ is a rapidly decreasing function of $\bar{\sigma}_G(v)$ if $\bar{\sigma}_G(v) > \bar{\sigma}_G(v_*)$. Moreover, influence degree $\bar{\sigma}_G(v)$ and its standard deviation $\bar{s}_G(v)$ increase as the diffusion probability becomes larger. We also investigated the relation between ratios $\bar{\sigma}_G^1(v)$ and $\bar{s}_G^1(v)$,

$$\bar{\sigma}_G^1(v) = \frac{\bar{\sigma}_G(v)}{\max_{u \in V} \bar{\sigma}_G(u)}, \qquad \bar{s}_G^1(v) = \frac{\bar{s}_G(v)}{\bar{\sigma}_G(v)},$$

for all $v \in V$. Figure 7 plots the pair $(\bar{\sigma}_G^1(v), \bar{s}_G^1(v))$ for all $v \in V$. We observe that $\bar{s}_G^1(v)$ is essentially a decreasing function of $\bar{\sigma}_G^1(v)$, and the function form does not primarily depend on the value of diffusion probability although it does depend on network structure. Moreover, roughly speaking, $\bar{s}_G^1(v)$ becomes almost equal to or less than $10^0 = 1.0$ when the ratio $\bar{\sigma}_G^1(v)$ is larger than 10^{-1} for both the networks, which means that standard deviation $\bar{s}_G(v)$ becomes almost equal to or less than $\bar{\sigma}_G(v)$ for nodes whose influence degree $\bar{\sigma}_G(v)$ is greater than 10% of the maximum value of influence degree. These results imply that the estimation accuracy with $M = 1000$ is acceptable from a statistical point of view.

5.4 Average influence degree

We consider finding the epidemic threshold p_G^* of the IC model for the Cosme and Twitter networks. To this end, we examined the relation between the diffusion probability $p_{u,v} = \mu$ and the average influence degree $\sum_{v \in V} \sigma_G(v)/|V|$. Since this is a computationally heavy task, we estimated the average influence degree using the proposed method with $M = 100$. Figure 8 shows the estimated average influence degree as a function of diffusion probability factor r, where

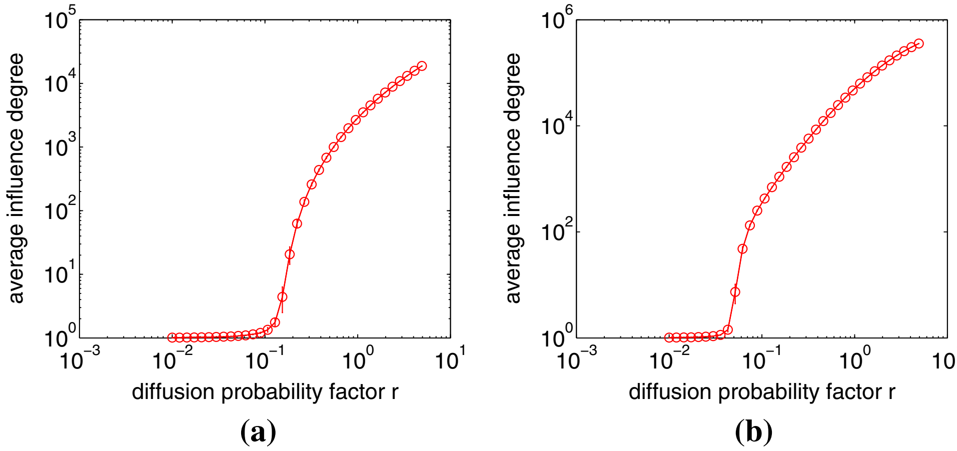


Fig. 8 Average influence degree curves. **a** Cosme network, **b** Twitter network

the standard deviations (see Eq. (1)) are indicated by the error bars. Here, we investigated $r = r_1 a^{k-1}$, $(r_1 = 0.01, a = 1.2, k = 1, \ldots, 35)$, that is, $1.3 \times 10^{-3} \leq \mu \leq 6.3 \times 10^{-1}$ for the Cosme network and $6.9 \times 10^{-5} \leq \mu \leq 3.4 \times 10^{-2}$ for the Twitter network. We first observe that the standard deviations are relatively small, and the accuracy with $M = 100$ is acceptable when the goal is to estimate the average influence degree. We needed about 1.1 min for the Cosme network and about 9.1 hours for the Twitter network to obtain the results shown in Fig. 8. From Fig. 8, we can find that the epidemic threshold $p_G^* = r_G^* / \bar{d}_G$ is given by $p_G^* = 1.9 \times 10^{-2}$ $(r_G^* = 0.15)$ for the Cosme network and $p_G^* = 2.8 \times 10^{-4}$ $(r_G^* = 0.04)$ for the Twitter network. These results imply that the epidemic threshold depends on network structure and the Twitter network spreads information more easily than the Cosme network.

5.5 Comparison with conventional centralities

Although estimating influence degree centrality for large networks is a time-consuming and difficult task, the proposed method enabled us to approximately calculate the influence degree within a reasonable time even for huge social networks. Thus, for the huge Twitter network, we evaluated whether or not the influence degree centrality can actually provide a novel concept in comparison with conventional centralities.

As conventional centralities, we examined the betweenness centrality, the closeness centrality, the hub centrality, and the PageRank centrality for network G. Here, the betweenness $\mathrm{betw}(v)$ of a node v is defined as

$$\mathrm{betw}(v) = \sum_{u \in V} \sum_{w \in V} \frac{\mathrm{spath}_{u,w}^{G}(v)}{\mathrm{spath}_{u,w}^{G}},$$

where $\mathrm{spath}_{u,w}^{G}$ is the total number of the shortest paths between node u and node v in G and $\mathrm{spath}_{u,w}^{G}(v)$ is the number of the shortest paths between node u and node v in G that passes through node v. The closeness $close(v)$ of a node v is defined as

$$\mathrm{close}(v) = \frac{1}{|V|} \sum_{u \in V} \frac{1}{\mathrm{dist}_G(v, u)},$$

where $\mathrm{dist}_G(v, u)$ stands for the graph distance from v to u in G, that is, the length of the shortest path from v to u in G. Also, the hub centrality score of a node is obtained by the HITS algorithm [6] that defines the hub and authority centrality, and the PageRank score of a node is provided by applying the PageRank algorithm with random jump factor 0.15 [5] to the reverse network $G^- = (V, E^-)$ that is constructed through reversing any link of G, that is,

$$E^- = \{(u, v) \in V \times V \mid (v, u) \in E\}.$$

Tables 1 and 2 show the top five nodes in the degree, betweenness, closeness, hub, PageRank, and influence degree $(r = 0.25, 0.5, 1.0, 2.0)$ centralities for the Twitter network. We can first observe that each centrality measure actually extracts its own proper nodes. For the influence degree centrality, while the diffusion probability setting affects the result, the top two nodes coincided. They were “masason” and “GachapinBlog”, which also appeared in the top five of the degree, closeness and PageRank centralities. Here, “masason” is the Twitter account of Masayoshi Son who is a famous Japanese businessman and CEO of SoftBank (a big IT company), and “GachapinBlog” is the Twitter account of Gachapin who is a popular Japanese TV character in a children’s program. These are very influential in Japanese Twitter. Unlike other centralities, the hub centrality extracted the representatives of a certain big community in Japanese Twitter, where “tomo7272” is the Twitter account of an ordinary person who often posts nice tweets. Note that “shuzo_matsuoka” is a famous bot in Japanese Twitter, and was extracted by the degree, betweenness and close-

Table 1 Ranking results for conventional centralities in the huge Twitter network

Rank	Degree	Betweenness	Closeness	Hub	PageRank
1	**masason**	**shuzo_matsuoka**	**masason**	tomo7272	**masason**
2	**GachapinBlog**	SNOOPYbot	**GachapinBlog**	ktamiya	natalie_mu
3	higashimototiji	NHK_PR	**shuzo_matsuoka**	euro_tour	JAXA_jp
4	**shuzo_matsuoka**	moomin_valley	higashimototiji	rakko001	Hayabusa_jaxa
5	555hamako	shuumai	takapon_jp	mabou77	**GachapinBlog**

Table 2 Ranking results for the influence degree centrality in the huge Twitter network

Rank	$r = 0.25$	$r = 0.5$	$r = 1.0$	$r = 2.0$
1	**masason**	**masason**	**masason**	**masason**
2	**GachapinBlog**	**GachapinBlog**	**GachapinBlog**	**GachapinBlog**
3	higashimototiji	itoi_shigesato	itoishigesato	**utadahikaru**
4	itoi_shigesato	higashimototiji	higashimototiji	shiro_tsubuyaki
5	555hamako	Astro_Soichi	**utadahikaru**	tenkijp

ness centralities. However, it did not appear in the top ten of the influence degree ranking. The tweet of bot attracts many people but dies out very rapidly. Thus, it is not identified as influential by the proposed method. On the other hand, "utadahikaru" was extracted only by the influence degree centrality with medium and high diffusion probabilities, while it did not appear in the top ten of other rankings. Here, "utadahikaru" is the Twitter account of Hikaru Utada who is a Japanese American singer known as one of the most influential artists in Japan. These results demonstrate that the influence degree centrality can serve as a novel measure that extracts influential nodes in terms of information diffusion which are not identified by existing measures.

6 Performance analysis of proposed techniques

The results of the previous section supported the usefulness of the proposed approach. However, analysis for networks of fixed structure alone is not sufficient enough to understand the effects of the REP and MCP techniques. Here, we extended our analysis using synthetic network with varying structures. The performance of these two pruning techniques should depend on the structure of the quotient graph $\mathcal{Q}_m = (\mathcal{C}_m, \mathcal{E}_m)$ which is derived from the SCC decomposition of an underlying network G_m. Clearly, if there are many feedforward motif patterns (i.e., $\{(C, X), (X, D), (C, D) \in \mathcal{E}_m\}$), the REP technique must be useful. Also, the MCP technique must be effective if $\mathcal{C}_m$ has a large number of components with in-degree 1 or out-degree 1, and a small number of components of large size. For simplicity, we consider roughly controlling the size of SCCs and the number of feedforward motif patterns for an original network G. In this section, we first describe such network generation methods, and next present the analysis results using those synthetic networks.

6.1 Network generation methods

For a given DAG expressed as $G = (V, E)$, we first note that any pair of two nodes $v, w \in V$ is classified into one of the following three cases: (1) w is reachable from v, i.e., $w \in R_G(v) \wedge v \notin R_G(w)$, (2) v is reachable from w, i.e., $v \in R_G(w) \wedge w \notin R_G(v)$, and (3) v (or w) is not reachable from w (or v), i.e., $v \notin R_G(w) \wedge w \notin R_G(v)$. Moreover, even when we add a link (v, w) for the case (1), and (w, v) for the case (2), it is guaranteed that the modified network still has the property of DAG. In what follows, for a given arbitrary network $G = (V, E)$, we will say that a pair of nodes $v, w \in V$ has a *DAG property* if the pair of nodes is classified into one of the above first two cases, (1) and (2), and a link (v, w) has a *DAG direction* if the pair of nodes $v, w \in V$ still has a DAG property after creation of this link. Now, we consider controlling the size of SCCs by changing a rate q of DAG direction link creation. Here note that each size of SCCs is minimized as 1 for a DAG.

In order to prepare networks having substantially different numbers of feedforward motif patterns, we focus on two network generation methods, CNN (Connecting Nearest-Neighbors) [28] and BA (Barabási-Albert) [1], and extend them so as to control the size of SCCs according to the rate q. Hereafter, these extended methods are referred to as the DCNN and DBA methods. Here, we will say that a pair of nodes $\{v, w\}$ is a *potential pair* if they are not directly connected, but have at least one common neighbor node, i.e., $(v, w) \notin E \wedge (w, v) \notin E$ and $\exists x \in V$ $((v, x) \in E \vee (x, v) \in E) \wedge ((w, x) \in E \vee (x, w) \in E)$. Then, we can summarize the DCNN method as an algorithm which repeats the following steps L times from a single node and an empty set of links:

1. With probability $1 - \epsilon$, create a new node $u \in V$, select a node $v \in V$ at random, and create a link (u, v) or (v, u) arbitrary.

2. With probability ϵ, select a potential pair $\{v, w\}$ at random, and create a link (v, w) or (w, v) to be a DAG direction with probability q if the pair of nodes $v, w \in V$ has a DAG property; otherwise create a link (v, w) or (w, v) arbitrary.

Clearly, we can easily see that the DCNN method generates a DAG by setting $q = 1$. In our experiments, we set $L = 360{,}000$ and $\epsilon = 1/8$ for the sake that the size of the generated networks can be roughly equal to that of the Cosme network, and their average degree can be around $\bar{d}_G = 8$.

Next, we describe the DBA method. Here, we will say that a node is selected by *preferential attachment* if its selection probability is proportional to the number of adjacent nodes. Then, we can summarize the DBA method as an algorithm which repeats the following steps $L - H$ times from a DAG having H links generated by the DCNN method:

1. With probability $1 - \epsilon$, create a new node $u \in V$, select a node $v \in V$ by preferential attachment, and create a link (u, v) or (v, u) arbitrary.
2. With probability ϵ, select a node $v \in V$ at random, select another node $w \in V$ by preferential attachment, and create a link (v, w) or (w, v) to be a DAG direction with probability q if the pair of nodes $v, w \in V$ has a DAG property; otherwise create a link (v, w) or (w, v) arbitrary.

Again, we can easily see that the DBA method generates a DAG by setting $q = 1$. In our experiments, we also set $L = 360{,}000$, $\epsilon = 1/8$, and $H = 800$. Here note that numbers of feedforward motif patterns appearing in the networks generated by the DCNN method inevitably become larger than those generated by the DBA method because the DCNN method has a link creation mechanism between potential pairs.

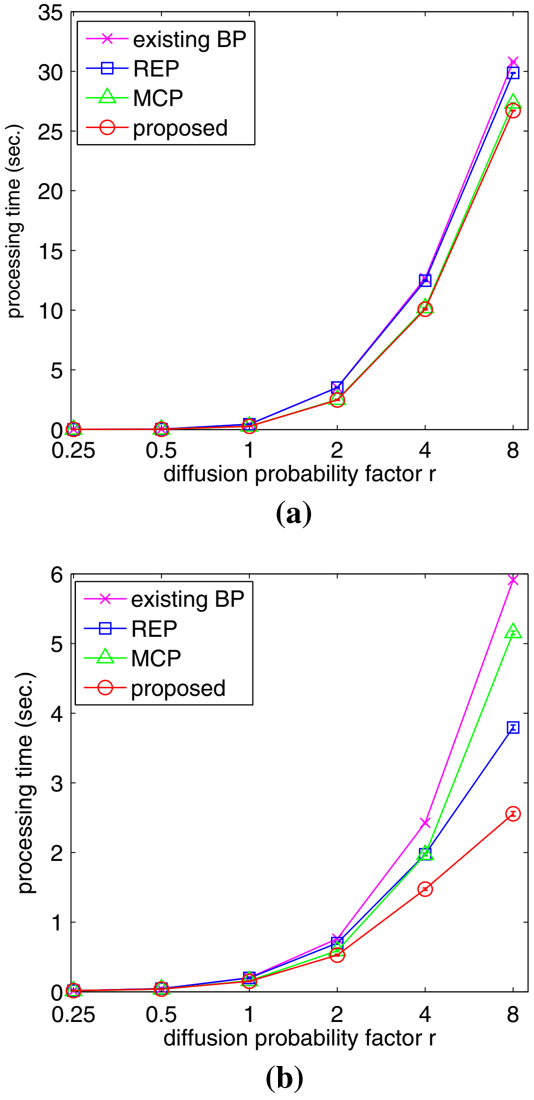


Fig. 9 Computation time comparison for DAGs. **a** BA DAG, **b** CNN DAG

6.2 Analysis results

We compared the computation time of the proposed, REP, MCP and existing BP methods in the same way as the case of real networks in Sect. 5.3 (see Fig. 5) for the synthetic networks generated in Sect. 6.1. Here, the cases of $r = 0.25, 0.5, 1.0, 2.0, 4.0, 8.0$ were investigated since the mean out-degree of each synthetic network is set at $\bar{d}_G = 8$. For each setting of respective method, 100 trials ($M = 100$) were performed, and the time for each trial was evaluated. The results are shown in Figs. 9, 10 and 11, where the average values are plotted and the standard deviations are indicated by the error bars.

Figure 9 displays the results for DAGs, where the size of each SCC component of an original network G is one, and the quotient graph $\mathcal{Q}_m$ coincides with G_m. We first observe that all the methods are comparable when $r \leq 1$, and the existing BP method always performs the worst when $r \geq 2$. Thus, the proposed REP and MCP methods can be helpful. As expected, the REP technique is more effective for the CNN DAG than for the BA DAG, since the CNN DAG encourages constructing feedforward motif patterns, while the BA DAG does not. In fact, the generated CNN DAG had 20 times more feedforward motif patterns than the generated BA DAG. Thus, in particular, the REP method outperforms the MCP method for the CNN DAG. Compared to the case of real networks (see Fig. 5), the MCP method is not so useful for these DAGs, since there are not that many components with in-degree 1 or out-degree 1, and the size of such components is also very small (equal to one). For the BA DAG, the proposed method combining both the REP and MCP techniques is comparable to the MCP method, and these two

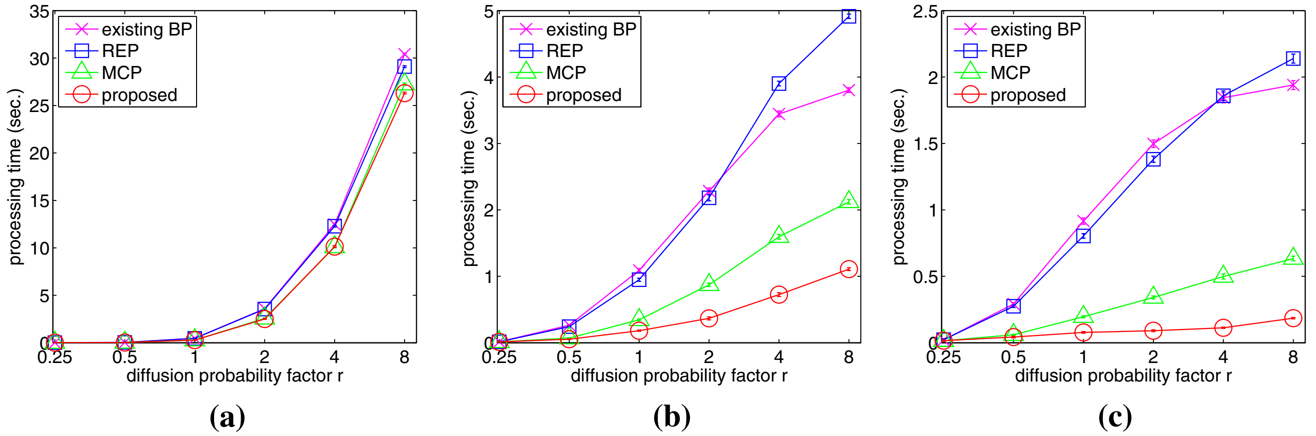


Fig. 10 Computation time comparison for BA networks. **a** $q = 10^{-5}$, **b** $q = 10^{-3}$, **c** $q = 10^{-1}$

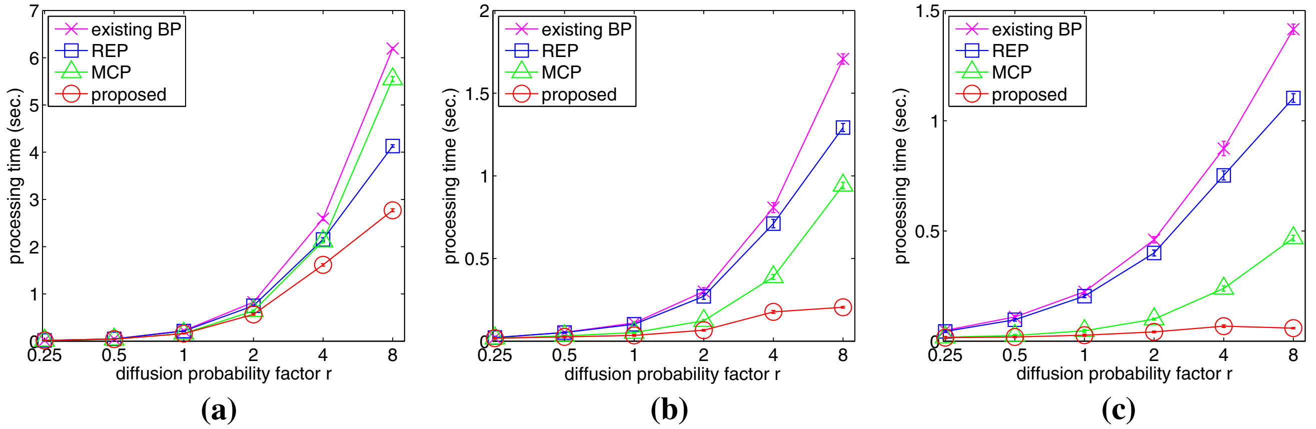


Fig. 11 Computation time comparison for CNN networks. **a** $q = 10^{-5}$, **b** $q = 10^{-3}$, **c** $q = 10^{-1}$

methods slightly outperform the REP and existing BP methods. This is attributed to the fact that the REP technique is not so useful for the BA DAG. However, for the CNN DAG, the proposed method significantly outperforms other three methods for large r since the REP technique becomes effective.

Next, we tried to increase the size of SCC components of a generated network G by increasing the value of q. Figures 10 and 11 show the results for the BA and CNN networks, respectively. When $q = 10^{-5}$, the generated network G is expected to be close to a DAG. From Figs. 10a and 11a, we first confirm that the results for $q = 10^{-5}$ is almost identical to those for the cases of DAGs (see Fig. 9). When the value of q becomes large, i.e., $q = 10^{-3}$ and $q = 10^{-1}$, components of large size can emerge for large r. Also, many components with in-degree 1 or out-degree 1 can be created. Thus, the MCP technique becomes useful, which is the same as the case of real networks (see Fig. 5). From Figs. 11b and 11c, we see that the REP technique is indeed effective for the CNN network. On the other hand, we see that the REP method is worse than the existing BP method for large r (see Figs. 10b and 10c) in case of the BA network. This is because there are not many feedforward motif patterns and the number of edges to be explored also becomes large as r is large. However, the proposed method always significantly outperforms other three methods for large r (see Figs. 10b, c, 11b, c). When $q = 10^{-1}$ and $r = 8$, the proposed method is about 10 and 25 times faster than the existing BP method for the BA and CNN networks, respectively. Note that the REP technique not only contributes to pruning redundant edges, but also encourages creating components with in-degree 1 or out-degree 1. Thus, the proposed method combining both the REP and MCP techniques can be effective even for the BA network. These analysis results support the effectiveness of the proposed method.

7 Conclusion

We view the dynamic process of information diffusion as an important ingredient to evaluate the importance of a node in a social network and consider that the node influence degree shares the same role that other existing topology-based centrality measures have. Unlike the existing centrality measures, the influence degree centrality is not easily computable because it is defined to be the expected number of information spread. We proposed a method that can estimate the influence degree of every single node in a large network simultaneously under the framework of *SIR model* setting. More specifically, we proposed two new pruning techniques called redundant-edge pruning (REP) and marginal-component pruning (MCP) on top of the existing bond percolation approach which reduces the node influence estimation problem to the problem of counting the reachable nodes from each single node in the directed graph realized by bond percolation on the original directed graph.

We, first, tested our algorithm using two real-world networks, one with 40K nodes and the other with 1000K nodes. The experimental results confirmed that the new pruning techniques improve the computational efficiency by an order of magnitude over the existing bond percolation method which is already three orders of magnitude faster than direct Monte Carlo simulations.

We, second, demonstrated that the proposed method can estimate the epidemic threshold of the IC model even for a huge Twitter network with 1000K nodes in reasonable time by examining the relation between the diffusion probability and the average influence degree, and showed that the epidemic threshold depends on network structure and for the two real-world networks, we tested the Twitter network spreads information more easily than the Cosme network. Further, it is confirmed that the nodes identified as influential by the influence degree centrality based on the SIR model are not necessarily the same or similar to those identified by the other existing centralities, and the influence degree centrality can identify those nodes that are deemed indeed influential but are not identifiable by the other existing methods.

We, third, examined how the performance of the two pruning techniques changes as the network structure changes using many different networks that are synthetically and systematically generated by extending the BA and CNN method in addition to the verification by the two real networks. We confirmed that the REP technique is effective when the quotient graph (a DAG obtained after decomposing the graph realized by applying the bond percolation to the original directed graph) has a large number of feed forward motif patterns and the MCP technique is effective when the quotient graph has a large number of components with in-degree 1 or out-degree 1 and a small number of components of large size. In general the MCP technique is more effective than the REP technique. Use of both techniques is always better than the single use of either techniques.

The bond percolation is a generic approach for the SIR model and can be instantiated to any specific diffusion model. Its advantage over other methods is that it allows us to estimate the influence degree of all the nodes in the network simultaneously regardless of the size of network. It does not require any approximations or assumptions to the model to improve the computational efficiency, e.g., small diffusion probability, shortest path, maximum influence path, etc., that were needed in the existing approaches. We instantiated it to the independent cascade (IC) model, but the same technique can be applied to other instantiations, e.g., linear threshold (LT) model.

Our immediate future work is to extensively evaluate the proposed method for various instantiations of the SIR framework including the LT model by using large real networks in a variety of fields. Needless to say, it is also necessary to mathematically clarify the performance difference between the proposed method and the existing BP method in terms of computational efficiency. Our results obtained by the synthetic networks has laid a basis toward this direction. In several real-world networks, there exist phenomena in which the *SIS model* is more suitable than the SIR model [21,23], where every node is allowed to be activated multiple times. It is known that the SIS-type independent cascade model on a network can be exactly mapped onto the IC model on a layered network built from the original network [15,16]. Thus, note that the proposed method developed for the SIR setting can also be applied to the SIS setting. Our future work includes evaluating the proposed method in the SIS framework.

References

1. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**, 509–512 (1999)
2. Boldi, P., Vigna, S.: In-core computation of geometric centralities with hyperball: a hundred billion nodes and beyond. In: Proceedings of the 2013 IEEE 13th International Conference on Data Mining Workshops (ICDMW'13), pp. 621–628 (2013)
3. Bonacich, P.: Power and centrality: a family of measures. Am. J. Sociol. **92**, 1170–1182 (1987)
4. Borgs, C., Brautbar, M., Chayes, J., Lucier, B.: Maximizing social influence in nearly optimal time. In: Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'14), pp. 946–957 (2014)
5. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. Comput. Netw. ISDN Syst. **30**, 107–117 (1998)
6. Chakrabarti, S., Dom, B., Kumar, R., Raghavan, P., Rajagopalan, S., Tomkins, A., Gibson, D., Kleinberg, J.: Mining the web's link structure. IEEE Comput. **32**, 60–67 (1999)
7. Chen, W., Lakshmanan, L., Castillo, C.: Information and influence propagation in social networks. Synthesis Lectures on Data Management, vol. 5(4), pp. 1–177. Morgan & Claypool Publishers, California, USA (2013)
8. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'10), pp. 1029–1038 (2010)

9. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09), pp. 199–208 (2009)
10. Chen, W., Yuan, Y., Zhang, L.: Scalable influence maximization in social networks under the linear threshold model. In: Proceedings of the 10th IEEE International Conference on Data Mining (ICDM'10), pp. 88–97 (2010)
11. Chierichetti, F., Epasto, A., Kumar, R., Lattanzi, S., Mirrokni, V.: Efficient algorithms for public-private social networks. In: Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'15), pp. 139–148 (2015)
12. Freeman, L.: Centrality in social networks: conceptual clarification. Social Netw. **1**, 215–239 (1979)
13. Goyal, A., Bonchi, F., Lakshmanan, L.: A data-based approach to social influence maximization. Proc. VLDB Endow. **5**(1), 73–84 (2011)
14. Katz, L.: A new status index derived from sociometric analysis. Psychometrika **18**, 39–43 (1953)
15. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03), pp. 137–146 (2003)
16. Kimura, M., Saito, K., Motoda, H.: Efficient estimation of influence functions fot sis model on social networks. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09), pp. 2046–2051 (2009)
17. Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI-07), pp. 1371–1376 (2007)
18. Kimura, M., Saito, K., Nakano, R., Motoda, H.: Extracting influential nodes on a social network for information diffusion. Data Min. Knowl. Discov. **20**, 70–97 (2010)
19. Kimura, M., Saito, K., Ohara, K., Motoda, H.: Efficient analysis of node influence based on sir model over huge complex networks. In: Proceedings of the 2014 International Conference on Data Science and Advanced Analytics (DSAA'14), pp. 216–222 (2014)
20. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'07), pp. 420–429 (2007)
21. Leskovec, J., McGlohon, M., Faloutsos, C., Glance, N., Hurst, M.: Patterns of cascading behavior in large blog graphs. In: Proceedings of 2007 SIAM International Conference on Data Mining (SDM'07), pp. 551–556 (2007)
22. Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., Alon, U.: Network motifs: simple building blocks of complex networks. Science **298**, 824–827 (2002)
23. Newman, M.: The structure and function of complex networks. SIAM Rev. **45**, 167–256 (2003)
24. Nguyen, H., Zheng, R.: Influence spread in large-scale social networks—a belief propagation approach. In: Proceedings of 2012 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD'12), pp. 515–530. LNAI 7524 (2012)
25. Ohara, K., Saito, K., Kimura, M., Motoda, H.: Resampling-based framework for estimating node centrality of large social network. In: Proceedings of the 17th International Conference on Discovery Science (DS'14), pp. 228–239. LNAI 8777 (2014)
26. Saito, K., Kimura, M., Ohara, K., Motoda, H.: Identifying super-mediators of information diffusion in social networks. In: Proceedings of the 16th International Conference on Discovery Science (DS'13), pp. 170–184. LNAI 8140 (2013)
27. Song, G., Zhou, X., Wang, Y., Xie, K.: Influence maximization on large-scale mobile social network: a divide-and-conquer method. IEEE Trans. Parallel Distrib. Syst. **26**, 1379–1392 (2015)
28. Vázquez, A.: Growing network with local rules: preferential attachment, clustering hierarchy, and degree correlations. Phys. Rev. E **67**, 056–104 (2003)
29. Watts, D.: A simple model of global cascades on random networks. Proc. Natl. Acad. Sci. USA **99**, 5766–5771 (2002)
30. Yang, Y., Chen, E., Liu, Q., Xiang, B., Xu, T., Shad, S.: On approximation of real-world influence spread. In: Proceedings of 2012 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD'12), pp. 548–564. LNAI 7524 (2012)
31. Zhou, C., Zhang, P., Zang, W., Guo, L.: On the upper bounds of spread for greedy algorithms in social network influence maximization. IEEE Trans. Knowl. Data Eng. **27**, 2770–2783 (2015)
32. Zhuge, H., Zhang, J.: Topological centrality and its e-science applications. J. Am. Soc. Inf. Sci. Technol. **61**, 1824–1841 (2010)