Extracting Influential Nodes on a Social Network for Information Diffusion

Masahiro Kimura, Kazumi Saito, Ryohei Nakano, and Hiroshi Motoda

Abstract

We address the combinatorial optimization problem of finding the most influential nodes on a large-scale social network for two widely-used fundamental stochastic diffusion models. The past study showed that a greedy strategy can give a good approximate solution to the problem. However, a conventional greedy method faces a computational problem. We propose a method of efficiently finding a good approximate solution to the problem under the greedy algorithm on the basis of bond percolation and graph theory, and compare the proposed method with the conventional method in terms of computational complexity in order to theoretically evaluate its effectiveness. The results show that the proposed method is expected to achieve a great reduction in computational cost. We further experimentally demonstrate that the proposed method is much more efficient than the conventional method using large-scale real-world networks including blog networks.

Keywords

Social network analysis, Information diffusion model, Influence maximization problem, Bond percolation

Authors' Addresses:

Masahiro Kimura
Department of Electronics and Informatics
Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp

Kazumi Saito
School of Administration and Informatics
University of Shizuoka
Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp

Ryohei Nakano
Department of Computer Science
Chubu University
Aichi 487-8501, Japan
nakano@cs.chubu.ac.jp

Hiroshi Motoda
Institute of Scientific and Industrial Research
Osaka University
Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

1 Introduction

The rise of the Internet and the World Wide Web has enabled us to investigate large-scale social networks, and there has been growing interest in social network analysis (Newman, 2001; McCallum et al., 2005; Leskovec et al., 2006). Here, a social network is the network of relationships and interactions among social entities such as individuals, groups of individuals, and organizations. Examples include blog networks, collaboration networks, and email networks.

The social network of interactions within a group of individuals plays a fundamental role in the spread of information, ideas, and innovations. In fact, a piece of information, such as the URL of a website that provides a new valuable service, can spread from one individual to another through the social network in the form of "word-of-mouth" communication. For example, the information of free email services such as Microsoft's Hotmail and Google's Gmail could spread largely through email networks. Thus, when we plan to market a new product, promote an innovation, or spread a new topic among a group of individuals, we can exploit social network effects. Namely, we can *target* a small number of influential individuals (e.g., giving free samples of the product, demonstrating the innovation, or offering the topic), and trigger a cascade of influence by which friends will recommend the product, promote the innovation, or propagate the topic to other friends. In this way, we can spread decisions in adopting the product, the innovation, or the topic through the social network from a small set of initial adopters to many individuals. Therefore, given a social network represented by a directed graph, a positive integer k, and a probabilistic model for the process by which a certain information spreads through the network, it is an important research issue in terms of sociology and *viral marketing* to find such a target set A_k^* of k nodes that maximizes the expected number of adopters of the information if A_k^* initially adopts it (Domingos and Richardson, 2001; Richardson and Domingos, 2002; Kempe et al, 2003; Kempe et al., 2005). Here, the expected number of nodes influenced by a target set is referred to as its *influence degree*, and this combinatorial optimization problem is called the *influence maximization problem* of size k.

Kempe et al. (2003) studied the influence maximization problem for two widely-used fundamental information diffusion models, the *independent cascade (IC) model* (Goldenberg, 2001; Kempe et al., 2003; Gruhl et al., 2004) and the *linear threshold (LT) model* (Watts, 2002; Kempe et al., 2003). They experimentally showed on large collaboration networks that for the influence maximization problem under the IC and LT models, the greedy algorithm significantly outperforms the high-degree and centrality heuristics that are commonly used in the sociology literature. Here, the high-degree heuristic chooses nodes in order of decreasing degrees, and the centrality heuristic chooses nodes in order of increasing average distance to other nodes in the

network. Moreover, they mathematically proved a performance guarantee of the greedy algorithm under these information diffusion models (i.e., the IC and LT models) by using an analysis framework based on submodular functions.

For the influence maximization problem of size k, the greedy algorithm iteratively finds a target set A_k of k nodes from the target set A_{k-1} of $k-1$ nodes that it has already found. Thus, it requires a method of computing all the *marginal influence degrees* of a given set A of nodes in the network. Here, for any node v that does not belong to A, the influence degree of target set $A \cup \{v\}$ is referred to as the *marginal influence degree* of A at v. However, it is an open question to compute influence degrees exactly by an efficient method, and therefore, the conventional method had to obtain good estimates for influence degrees by simulating the random process of the information diffusion model (i.e., the IC or LT model) many times (Kempe et al., 2003). Solving the influence maximization problem under the greedy algorithm needed a large amount of computation for large-scale networks.

In this paper, for the IC and LT models, we propose a method of efficiently estimating all the marginal influence degrees of a given set of nodes on the basis of bond percolation and graph theory, and apply it to approximately solving the influence maximization problem under the greedy algorithm. In order to theoretically evaluate the effectiveness of the proposed method for solving the influence maximization problem, we compare the proposed method with the conventional method in terms of computational complexity, and show that the proposed method is expected to achieve a large reduction in computational cost. Further, using large-scale real networks including blog networks, we experimentally demonstrate that the proposed method is much more efficient than the conventional method. Finally, we discuss some related work, and describe the conclusion.

2 Definitions

We examine the influence maximization problem on a network represented by a directed graph $G = (V, E)$ for the IC and LT models. Here, V and E are the sets of all the nodes and links in the network, respectively. Let N and L be the numbers of elements of V and E, respectively.

We first recall some basic notions from graph theory. Next, we define the IC and LT models on G according to the work of Kempe et al. (2003). Last, we give a mathematical definition of the influence maximization problem.

2.1 Graphs

We consider a directed graph $G = (V, E)$. If there is a directed link (u, v) from node u to node v, node v is called a *child node* of node u and node u is called a *parent node* of node v. For any $v \in V$, let $\Gamma(v)$ denote the set of all

the parent nodes of v. For a subset V' of V, graph $G' = (V', E')$ is called the *induced graph* of G to V' if $E' = E \cap (V' \times V')$.

We call $(u_0, \cdots, u_\ell)$ a *path* from node u_0 to node u_ℓ if we have $(u_{i-1}, u_i) \in E$, $(i = 1, \cdots, \ell)$. We say that node u can *reach* node v or node v is *reachable* from node u if there is a path from node u to node v. For a node v of the graph G, we define $F(v; G)$ to be the set of all the nodes that are reachable from v, and define $B(v; G)$ to be the set of all the nodes that can reach v. For any $A \subset V$, we set

$$F(A; G) = \bigcup_{v \in A} F(v; G), \ B(A; G) = \bigcup_{v \in A} B(v; G).$$

A *strongly connected component (SCC)* of G is a maximal subset C of V such that for all u, $v \in C$ there is a path from u to v. For a node v of G, we define $SCC(v; G)$ to be the SCC that contains v.

2.2 Information Diffusion Models

We consider mathematically modeling the spread of certain information through a social network $G = (V, E)$. In the IC and LT models, the following assumptions are made:

- A node is called *active* if it has adopted the information.
- The state of a node is either *active* or *inactive*.
- Nodes can switch from being inactive to being active, but cannot switch from being active to being inactive.
- The spread of the information through the network G is represented as the spread of active nodes on G.
- Given an initial set A of active nodes, we suppose that the nodes in A first become active and all the other nodes remain inactive at time-step 0.
- The diffusion process of active nodes unfolds in discrete time-steps $t \geq 0$.

2.2.1 Independent Cascade Model

First, we define the *independent cascade (IC) model*. In this model, we specify a real value $p_{u,v} \in [0, 1]$ for each directed link (u, v) in advance. Here, $p_{u,v}$ is referred to as the *propagation probability* through link (u, v). When an initial set A of active nodes is given, the diffusion process of active nodes proceeds according to the following randomized rule. When node u first becomes active at time-step t, it is given a single chance to activate

each of its currently inactive child nodes v, and succeeds with probability $p_{u,v}$. If u succeeds, then v will become active at time-step $t+1$. Here, if v has multiple parent nodes that become active at time-step t for the first time, then their activation attempts are sequenced in an arbitrary order, but performed at time-step t. Whether or not u succeeds, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

For an initial active set A ($\subset V$), let $\varphi(A)$ denote the number of active nodes at the end of the random process for the IC model. Note that $\varphi(A)$ is a random variable. Let $\sigma(A)$ denote the expected value of $\varphi(A)$. We call $\sigma(A)$ the *influence degree* of A.

2.2.2 Linear Threshold Model

Next, we define the *linear threshold (LT) model*. In this model, for any node $v \in V$, we in advance specify a *weight* $w_{u,v}$ (> 0) from its parent node u such that

$$\sum_{u \in \Gamma(v)} w_{u,v} \leq 1.$$

When an initial set A of active nodes is given, the diffusion process of active nodes proceeds according to the following randomized rule. First, for any node $v \in V$, a *threshold* θ_v is chosen uniformly at random from the interval $[0,1]$. At time-step t, an inactive node v is influenced by each of its active parent nodes u according to weight $w_{u,v}$. If the total weight from active parent nodes of v is at least threshold θ_v, that is,

$$\sum_{u \in \Gamma_t(v)} w_{u,v} \geq \theta_v,$$

then v will become active at time-step $t+1$. Here, $\Gamma_t(v)$ stands for the set of parent nodes of v that are active at time-step t. The process terminates if no more activations are possible.

Note that the threshold θ_v models the tendency of node v to adopt the information when its parent nodes do. Note also that the LT model is a probabilistic model associated with the uniform distribution on $[0,1]^N$. Further note that in the LT model it is the node thresholds that are random, while in the IC model it is the propagations through links that are random. Suppose that A is an initial set of active nodes. We define a random variable $\varphi(A)$ by the number of active nodes at the end of the random process for the LT model. Let $\sigma(A)$ denote the expected value of $\varphi(A)$. We call $\sigma(A)$ the *influence degree* of A. Note that these notations are the same as those for the IC model.

2.3 Influence Maximization Problem

We mathematically define the influence maximization problem on a network $G = (V, E)$ under the IC and LT models. Let k be a positive integer with $k < N$.

The *influence maximization problem* on G of size k is defined as follows: Find a set A_k^* of k nodes to target for initial activation such that $\sigma(A_k^*) \geq \sigma(S)$ for any set S of k nodes, that is, find

$$A_k^* \ = \ \mathrm{argmax}_{A \in \{S \subset V;\, |S|=k\}} \, \sigma(A), \tag{1}$$

where $|S|$ stands for the number of elements of set S.

3 Conventional Method

Kempe et al. (2003) showed the effectiveness of the greedy algorithm for the influence maximization problem under the IC and LT models. In this section, we introduce the greedy algorithm, and describe the conventional method for solving the influence maximization problem under the greedy algorithm. We, then, consider evaluating the computational complexity for the conventional method.

3.1 Greedy Algorithm

We approximately solve the influence maximization problem by the following greedy algorithm:

(G1) Set $A \leftarrow \emptyset$.

(G2) **for** $i = 1$ to k **do**

(G3) Choose a node $v_i \in V$ maximizing $\sigma(A \cup \{v\})$, $(v \in V \setminus A)$.

(G4) Set $A \leftarrow A \cup \{v_i\}$.

(G5) **end for**

Let A_k denote the set of k nodes obtained by this algorithm. We refer to A_k as the *greedy solution* of size k. Then, it is known that

$$\sigma(A_k) \ \geq \ \left(1 - \frac{1}{e}\right) \sigma(A_k^*),$$

that is, the quality guarantee of A_k is assured (Kempe et al., 2003). Here, A_k^* is the exact solution defined by Equation (1).

To implement the greedy algorithm, we need a method for estimating all the marginal influence degrees $\{\sigma(A \cup \{v\});\, v \in V \setminus A\}$ of A in Step (G3) of the algorithm.

3.2 Conventional Method for Estimating Marginal Influence Degrees

For Step (G3) of the greedy algorithm, the conventional method estimated all the marginal influence degrees $\{\sigma(A\cup\{v\});\, v \in V\backslash A\}$ of A in the following way (Kempe et al., 2003): First, a sufficiently large positive integer M is specified. For any $v \in V \setminus A$, the random process of the diffusion model (IC or LT model) is run from the initial active set $A \cup \{v\}$, and the number $\varphi(A \cup \{v\})$ of final active nodes is counted. Each $\sigma(A \cup \{v\})$ is estimated as the empirical mean obtained from M such simulations.

Namely, the conventional method independently estimated $\sigma(A \cup \{v\})$ for all $v \in V \setminus A$ as follows:

1. **for** $m = 1$ to M **do**
2. Compute $\varphi(A \cup \{v\})$.
3. Set $x_m \leftarrow \varphi(A \cup \{v\})$.
4. **end for**
5. Set $\sigma(A \cup \{v\}) \leftarrow (1/M) \sum_{m=1}^{M} x_m$.

Here, each $\varphi(A \cup \{v\})$ is computed as follows:

1. Set $H_0 \leftarrow A \cup \{v\}$.
2. Set $t \leftarrow 0$.
3. **while** $H_t \neq \emptyset$ **do**
4. Set $H_{t+1} \leftarrow \{$the activated nodes at time $t + 1\}$.
5. Set $t \leftarrow t + 1$.
6. **end while**
7. Set $\varphi(A \cup \{v\}) \leftarrow \sum_{j=0}^{t-1} |H_j|$

3.3 Computational Complexity of Conventional Method

We consider evaluating the computational complexity of solving the influence maximization problem. For this purpose, we introduce the notion of *examined nodes*. Here, an *examined node* is a node that is actually visited by tracing incoming or outgoing links on the graph in question for the method when all the marginal influence degrees $\{\sigma(A \cup \{v\});\, v \in V \setminus A\}$ of A are estimated in Step (G3) of the greedy algorithm. In Section 4.4, we describe the reason why we investigate the examined nodes for evaluating the computational complexity.

The computational complexity of the conventional method is evaluated in terms of the expected number of examined nodes. In order to estimate $\sigma(A \cup \{v\})$, $(v \in V \setminus A)$, it is necessary for any $v \in V \setminus A$ to simulate M times the random process of the information diffusion model (IC or LT model) from the initial active set $A \cup \{v\}$ on graph G. For each simulation, the set of examined nodes are the same as the set of active nodes in the process. Thus, we can estimate that the expected number $\mathcal{C}_0$ of examined nodes for the conventional method is

$$\mathcal{C}_0 = M \sum_{v \in V \setminus A} \sigma(A \cup \{v\}). \tag{2}$$

4 Proposed Method

We propose a method for efficiently estimating all the marginal influence degrees $\{\sigma(A \cup \{v\});\, v \in V \setminus A\}$ of A in Step (G3) of the greedy algorithm on the basis of bond percolation and graph theory, and evaluate the computational complexity, and compare it with that of the conventional method.

4.1 Bond Percolation

The IC and LT models are identified with *bond percolation models* which are defined below, and all the marginal influence degrees $\{\sigma(A \cup \{v\});\, v \in V \setminus A\}$ of A are efficiently estimated by exploiting graph theoretic methods.

A *bond percolation* process on graph $G = (V, E)$ is the process in which each link of G is randomly designated either "occupied " or "unoccupied" according to some probability distribution. Here, in terms of information diffusion on a social network, occupied links represent the links through which the information propagates, and unoccupied links represent the links through which the information does not propagate. Let us consider the following set of L-dimensional vectors,

$$R_G = \left\{ r = (r_{u,v})_{(u,v) \in E} \in \{0,1\}^L \right\},$$

where L is the number of links in G. A bond percolation process on G is determined by a probability distribution $q(r)$ on R_G. Namely, for a random vector $r \in R_G$ drawn from $q(r)$, each link $(u, v) \in E$ is designated "occupied" if $r_{u,v} = 1$, and it is designated "unoccupied" if $r_{u,v} = 0$. Let E_r denote the set of all the occupied links for $r \in R_G$, and let G_r denote the graph (V, E_r). For each $r \in R_G$, we can consider the deterministic diffusion model $\mathcal{M}_r$ on G_r such that $F(A; G_r)$ becomes the final set of active nodes when A is an initial set of active nodes, where $F(A; G_r)$ is the set that is reachable from A on G_r (see, Section 2.1). By associating the diffusion model $\mathcal{M}_r$ on G_r with a probability distribution $q(r)$ on R_G, we define a stochastic diffusion model on G. We call this diffusion model the *bond percolation model* on G, and

refer to the probability distribution $q(r)$ on R_G as the *occupation probability distribution* of the bond percolation model.

We easily see that the IC model on G can be identified with the so-called *susceptible/infective/recovered (SIR) model* (Newman, 2003) for the spread of a disease on G, where the nodes that become active at time t in the IC model correspond to the infective nodes at time t in the SIR model. We recall that in the SIR model, an individual occupies one of the three states, "susceptible", "infected" and 'recovered", where a susceptible individual becomes infected with a certain probability when s/he is encountered an infected patient and subsequently recovers at a certain rate (see, Newman, 2003; Watts and Dodds, 2007). It is known that the SIR model on a network can be exactly mapped onto a bond percolation model on the same network (Grassberger, 1983; Newman, 2002; Kempe et al., 2003; Newman, 2003). Hence, we see that the IC model on G is equivalent to a bond percolation model on G, that is, these two models have the same probability distribution for the final set of active nodes given a target set. Here, for the IC model on G, the occupation probability distribution $q(r)$ of the corresponding bond percolation model is given by

$$q(r) = \prod_{(u,v)\in E} \left\{ (p_{u,v})^{r_{u,v}} (1-p_{u,v})^{1-r_{u,v}} \right\}, \quad (r \in R_G),$$

that is, each link (u, v) of G is independently declared to be "occupied" with probability $p_{u,v}$, where $p_{u,v}$ is the propagation probability through link (u, v) in the IC model.

On the other hand, Kempe et al. (2003) proved that the LT model on G can also be equivalent to a bond percolation model on G to derive the result that the influence degree function $\sigma(A)$ is submodular in the LT model. Here, for the LT model on G, the corresponding occupation probability distribution $q(r)$ is generated by declaring "occupied" and "unoccupied" links in the following way: For any $v \in V$, we pick at most one of the incoming links to v by selecting link (u, v) with probability $w_{u,v}$ and selecting no link with probability $1 - \sum_{u\in\Gamma(v)} w_{u,v}$. After this process, the picked links are declared to be "occupied" and the other links are declared to be "unoccupied". Here, $w_{u,v}$ is the weight of link (u, v) in the LT model. Specifically, $q(r)$ is described as follows:

$$q(r) \;=\; \prod_{v\in V} \prod_{u\in\Gamma(v)} \left\{ (w_{u,v})^{r_{u,v}} \left(1 - \sum_{u\in\Gamma(v)} w_{u,v} \right)^{\left(1-\sum_{u\in\Gamma(v)} r_{u,v}\right)} \right\},$$

where if $\sum_{u\in\Gamma(v)} w_{u,v} < 1$, $\sum_{u\in\Gamma(v)} r_{u,v} \le 1$ and if $\sum_{u\in\Gamma(v)} w_{u,v} = 1$, $\sum_{u\in\Gamma(v)} r_{u,v} = 1$.

4.2 Proposed Method for Estimating Marginal Influence Degrees

We present a method of estimating all the marginal influence degrees $\{\sigma(A \cup \{v\});\ v \in V \setminus A\}$ of A in Step (G3) of the greedy algorithm. As shown in the preceding section, the IC and LT models on G can be identified with the bond percolation models on G. Therefore, we have

$$\sigma(A \cup \{v\}) = \sum_{r \in R_G} q(r)\, |F(A \cup \{v\}; G_r)|$$

for any $v \in V \setminus A$, where $q(r)$ is the corresponding occupation probability distribution, and $F(A \cup \{v\}; G_r)$ stands for the set of all the nodes that are reachable from $A \cup \{v\}$ on graph G_r (see, Section 2.1).

We estimate $\{\sigma(A \cup \{v\});\ v \in V \setminus A\}$ in the following way: First, we specify a sufficiently large positive integer M. Next, we independently generate a set $\{r_1, \cdots, r_M\}$ of M sample vectors on R_G from the probability distribution $q(r)$; that is, independently generate a set $\{G_{r_m}; m = 1, \cdots, M\}$ of M graphs. For any $v \in V \setminus A$, we approximate $\sigma(A \cup \{v\})$ by

$$\sigma(A \cup \{v\}) \simeq \frac{1}{M} \sum_{m=1}^{M} |F(A \cup \{v\}; G_{r_m})|. \tag{3}$$

Thus, we estimate $\{\sigma(A \cup \{v\});\ v \in V \setminus A\}$ on the basis of Equation (3) as follows:

1. **for** $m = 1$ to M **do**
2. Generate graph G_{r_m}.
3. Compute $\{|F(A \cup \{v\}; G_{r_m})|;\ v \in V \setminus A\}$.
4. Set $x_{v,m} \leftarrow |F(A \cup \{v\}; G_{r_m})|$ for all $v \in V \setminus A$.
5. **end for**
6. Set $\sigma(A \cup \{v\}) \leftarrow (1/M) \sum_{m=1}^{M} x_{v,m}$ for all $v \in V \setminus A$.

In particular, we evaluate $\{|F(A \cup \{v\}; G_r)|;\ v \in V \setminus A\}$ for an arbitrary $r \in R_G$ by the following algorithm:

(E1) Find the subset $F(A; G_r)$ of V.

(E2) Set $|F(A \cup \{v\}; G_r)| \leftarrow |F(A; G_r)|$ for all $v \in F(A; G_r) \setminus A$.

(E3) Find the subset $V_r^A = V \setminus F(A; G_r)$ of V, and the induced graph G_r^A of G_r to V_r^A.

(E4) Set $U \leftarrow \emptyset$.

(E5) while $V_r^A \setminus U \neq \emptyset$ **do**

(E6) Pick a node $u \in V_r^A \setminus U$.

(E7) Find the subset $F(u; G_r^A)$ of V_r^A.

(E8) Find the subset $C(u; G_r^A) = B(u; G_r^A) \cap F(u; G_r^A)$ of $F(u; G_r^A)$.

(E9) Set $|F(A \cup \{v\}; G_r)| \leftarrow |F(u; G_r^A)| + |F(A; G_r)|$ for all $v \in C(u; G_r^A)$.

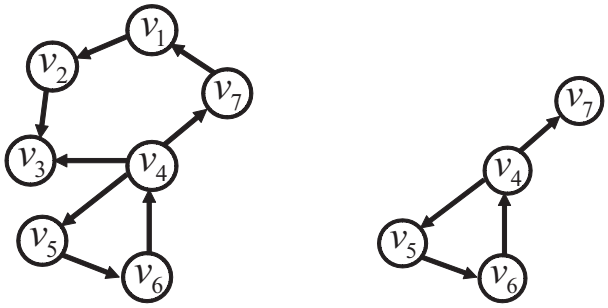
(E10) Set $U \leftarrow U \cup C(u; G_r^A)$.

(E11) end while

Now, we explain this algorithm. In Step (E1), we find the subset $F(A; G_r)$ that is reachable from A on graph G_r. In Step (E2), we use the fact that if $v \in F(A; G_r)$, the set $F(A \cup \{v\}; G_r)$ that is reachable from $A \cup \{v\}$ on G_r is equal to the set $F(A; G_r)$, and we simultaneously compute $|F(A \cup \{v\}; G_r)|$ for all $v \in F(A; G_r)$. In Step (E3), we find the subset $V_r^A = V \setminus F(A; G_r)$, and also find the induced graph G_r^A of graph G_r to V_r^A. In Steps (E4) to (E11), we use the fact that if $v \notin F(A; G_r)$, $|F(A \cup \{v\}; G_r)|$ is obtained by the sum of $|F(A; G_r)|$ and $|F(v; G_r^A)|$. This fact enables us to reduce the graph in question from G_r to G_r^A. We attempt to decompose graph G_r^A into its SCCs. In Step (E6), on graph G_r^A, we pick a node u that does not belong to the SCCs that we have already found. In Step (E7), we find the set $F(u; G_r^A)$ that is reachable from u on graph G_r^A. In Step (E8), we find the subset $C(u; G_r^A) = B(u; G_r^A) \cap F(u; G_r^A)$ of $F(u; G_r^A)$ by tracing backward all the links from u on the induced graph of G_r^A to $F(u; G_r^A)$. Note that the set $C(u; G_r^A)$ is equal to the SCC $SCC(u; G_r^A)$ that contains u. In Step (E9), we use the fact that $|F(v; G_r^A)| = |F(u; G_r^A)|$ if $v \in C(u; G_r^A)$, and simultaneously compute $|F(A \cup \{v\}; G_r)|$ for all $v \in C(u; G_r^A)$. We illustrate the flow of the algorithm in the following example:

Example: We consider the graph G_r shown in Figure 1a, where $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$. We set $A = \{v_1\}$. In this case, the process of the algorithm proceeds as follows.

In Step (E1), we find $F(A; G_r) = \{v_1, v_2, v_3\}$. In Step (E2), we find $|F(A \cup \{v_2\}; G_r)| = |F(A \cup \{v_3\}; G_r)| = 3$. In Step (E3), we find $V_r^A = \{v_4, v_5, v_6, v_7\}$ and G_r^A as shown in Figure 1b. In Step (E4), we set $U = \emptyset$. In Step (E5), we check $V_r^A \setminus U = \{v_4, v_5, v_6, v_7\} \neq \emptyset$. In Step (E6), we pick $v_4 \in V_r^A \setminus U$. In Step (E7), we find $F(v_4; G_r^A) = \{v_4, v_5, v_6, v_7\}$. In Step (E8), we find $C(v_4; G_r^A) = B(v_4; G_r^A) \cap F(v_4; G_r^A) = \{v_4, v_5, v_6\}$ in $F(v_4; G_r^A)$. In Step (E9), we find $|F(A \cup \{v_4\}; G_r)| = |F(A \cup \{v_5\}; G_r)| = |F(A \cup \{v_6\}; G_r)| = 7$. In Step (E10), we set $U = \{v_4, v_5, v_6\}$. In Step (E11), we return to Step (E5). In Step (E5), we check $V_r^A \setminus U = \{v_7\} \neq \emptyset$. In Step (E6), we pick $v_7 \in V_r^A \setminus U$. In Step (E7), we find $F(v_7; G_r^A) = \{v_7\}$. In Step (E8), we find $C(v_7; G_r^A) = \{v_7\}$. In Step (E9), we find $|F(A \cup \{v_7\}; G_r)|$



(a) An example of graph G_r.

(b) Graph G_r^A.

Figure 1: An illustration of the flow of the proposed algorithm for evaluating $\{|F(A \cup \{v\}; G_r)|;\, v \in V \setminus A\}$, where $r \in R_G$ and $A = \{v_1\}$.

$= 4$. In Step (E10), we set $U = \{v_4, v_5, v_6, v_7\}$. In Step (E11), we return to Step (E5). In Step (E5), we check $V_r^A \setminus U = \emptyset$. Then, the process of the algorithm ends.

4.3 Computational Complexity of Proposed Method

In the same way as in Section 3.3, we evaluate the computational complexity of the proposed method as the expected number of examined nodes for estimating all the marginal influence degrees $\{\sigma(A \cup \{v\});\, v \in V \setminus A\}$ of A in Step (G3) of the greedy algorithm.

Let G_r be a graph generated from the occupation probability distribution $q(r)$ of the corresponding bond percolation model. We consider evaluating the expected number $\overline{Z(A, G_r)}$ of examined nodes for computing $\{|F(A \cup \{v\}; G_r)|;\, v \in V \setminus A\}$ by the proposed method (see, Section 4.2). First, the number of examined nodes for finding $F(A; G_r)$ is given by $|F(A; G_r)|$. Let

$$V_r^A = \bigcup_{u \in U_r^A} SCC(u; G_r^A)$$

be the SCC decomposition of the induced graph G_r^A of G_r to $V_r^A = V \setminus F(A; G_r)$, where U_r^A stands for the set of all the representative nodes for SCCs. For any $u \in U_r^A$, the number of examined nodes for finding $F(u; G_r^A)$ is $|F(u; G_r^A)|$. Suppose now that $F(u; G_r^A)$ is found. Then, the number of examined nodes for finding $C(u; G_r^A)$ $(= SCC(u; G_r^A))$ is $|SCC(u; G_r^A)|$, since $C(u; G_r^A) = B(u; G_r^A) \cap F(u; G_r^A)$ is calculated on the induced graph of graph G_r^A to $F(u; G_r^A)$. Therefore, the number $Z(A, G_r)$ of examined nodes for computing $\{|F(A \cup \{v\}; G_r)|;\, v \in V \setminus A\}$ by the proposed method is as follows:

$$Z(A, G_r) = |F(A; G_r)| + \sum_{u \in U_r^A} \left(|F(u; G_r^A)| + |SCC(u; G_r^A)| \right).$$

By the definition of graph G_r^A, we have

$$\sum_{u \in U_r^A} |SCC(u; G_r^A)| = N - |F(A; G_r)|,$$

where $N = |V|$. Thus, we have

$$Z(A, G_r) = N + \sum_{u \in U_r^A} |F(u; G_r^A)|. \tag{4}$$

Since $|F(u; G_r^A)| = |F(A \cup \{u\}; G_r)| - |F(A; G_r)|$, we can estimate the expected value of $|F(u; G_r^A)|$ as $\sigma(A \cup \{u\}) - \sigma(A)$. Hence, by Equation (4), we can estimate the expected number $\overline{Z(A, G_r)}$ of examined nodes for computing $\{|F(A \cup \{v\}; G_r)|;\, v \in V \setminus A\}$ as

$$\overline{Z(A, G_r)} = N + \left\langle \sum_{u \in U_r^A} (\sigma(A \cup \{u\}) - \sigma(A)) \right\rangle_r,$$

where $\langle f(r) \rangle_r$ stands for the operation that averages $f(r)$ with respect to r under $q(r)$, that is,

$$\langle f(r) \rangle_r = \sum_{r \in R(G)} f(r)\, q(r).$$

From the above results, we can estimate that the expected number $\mathcal{C}_1$ of examined nodes for the proposed method is

$$\mathcal{C}_1 = M \left\{ N + \left\langle \sum_{u \in U_r^A} (\sigma(A \cup \{u\}) - \sigma(A)) \right\rangle_r \right\}. \tag{5}$$

4.4 Computational Complexity Comparison

We compare the proposed method with the conventional method in terms of computational complexity. Both methods need M to be specified as a parameter, and we use the same value for both. We note that more coin-flips are used in the conventional method. In fact, if we think of a single run, i.e., any one of the M runs, the expected number of coin-flips for the conventional method is $O(|V|\sigma(v))$ for both the IC and LT models, whereas that for the proposed method is $O(|E|)$ for the IC model and $O(|V|)$ for the LT model. Note that in case of LT model for the proposed method, the coin-flip is realized by roulette for each node, i.e., picking at most one incoming link. However, if we focus on a single node v for initial activation from which to propagate the information, the number of coin-flips are $O(\sigma(v))$ for both the conventional and the proposed methods and for both the IC and the LT models because only the activated nodes (the expected number is $\sigma(v)$) are on the paths that lead to reachable nodes from v in the proposed

method. Thus by using the same value of M, both would estimate $\sigma(v)$ with the same accuracy in principle (see Appendix A). The biggest difference is that in the conventional method, when A is not empty, many of the coin-flips are redundant; that is, the diffusion process from A is repeatedly performed, whereas in the proposed method, no such repetition is made. This contributes to the stability of the proposed method. Below we begin by explaining the reason why we investigate the examined nodes to compare the proposed and the conventional methods.

First, we consider the case of IC model. Both the proposed and the conventional methods flip a coin with a bias $p_{u,v}$ on a link (u, v) to decide whether to propagate the information through the link (u, v) or not. Here, if we assume that all the coins are flipped in advance for the conventional method and ignore the computational complexity for flipping a coin and deciding whether or not to propagate the information, then for both the proposed and the conventional methods, the major computation is to trace forward or backward the links the information propagates and identify the nodes to visit. Therefore, we evaluate the computational complexities of the both methods for the IC model in terms of the expected number of examined nodes.

Next, we consider the case of LT model. For the proposed method, we ignore the computational complexity for the process of choosing at most one incoming link of each node in the original graph. For the conventional method, we ignore the computational complexity for the process of choosing the threshold θ_v of each node v in the original graph. Note that the proposed method performs the process M times, whereas the conventional method performs the process MN times. Moreover, for the conventional method, we further ignore the computational complexity for adding the weights from the neighboring active nodes to a node and deciding whether the node becomes active or not. Then, the major computation for the conventional method is to trace forward the links the information propagates and identify the nodes to visit. Therefore, we also evaluate the computational complexities of the both methods for the LT model in terms of the expected number of examined nodes.

Now, we compare the proposed and the conventional methods in terms of the expected number of examined nodes. We use the results in Sections 3.3 and 4.3. By Equation (2), the expected number $\mathcal{C}_0$ of examined nodes for the conventional method can be estimated as

$$\mathcal{C}_0 = M \left\{ N - |A| + \sum_{u \in V \setminus A} (\sigma(A \cup \{u\}) - 1) \right\}, \tag{6}$$

since $\sum_{V \setminus A} 1 = N - |A|$. In Equation (6), we can expect that $|A| \ll N$ $(= |V|)$, and $\sigma(A \cup \{u\}) - 1$ is summed up for almost all $u \in V$, since $k \ll N$. On the other hand, we can generally expect $|U_r^A| \ll N$ in Equation (5).

Also, we have $\sigma(A) > 1$ in the greedy algorithm if $A \neq \emptyset$. Moreover, for any $u \in V \setminus A$, $\sigma(A \cup \{u\}) - \sigma(A)$ decreases as $|A|$ increases, since $\sigma(A)$ is a submodular function. Hence, we can generally expect that in Step (G3) of the greedy algorithm, the proposed method has much smaller expected number of examined nodes than the conventional method.

From the above results, we can expect that compared with the conventional method, the proposed method will achieve a large reduction in computational cost.

5 Experimental Evaluation

Using large-scale real networks, we experimentally evaluated the performance of the proposed method.

5.1 Network Datasets

In the evaluation experiments, we should desirably use large-scale networks that exhibit many of the key features of real social networks. Here, we show the experimental results for two different datasets of such real networks.

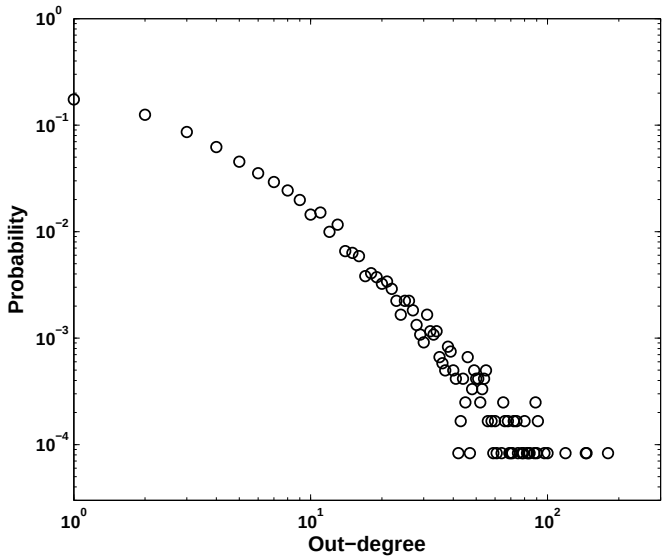


Figure 2: The out-degree distribution for the blog dataset.

First, we employed a trackback network of blogs, since a piece of information can propagate from one blog author to another blog author through a trackback, where a trackback is a kind of hyperlink with a *linkback* (i.e., link notification) function. We exploited the blog "Theme salon of blogs"

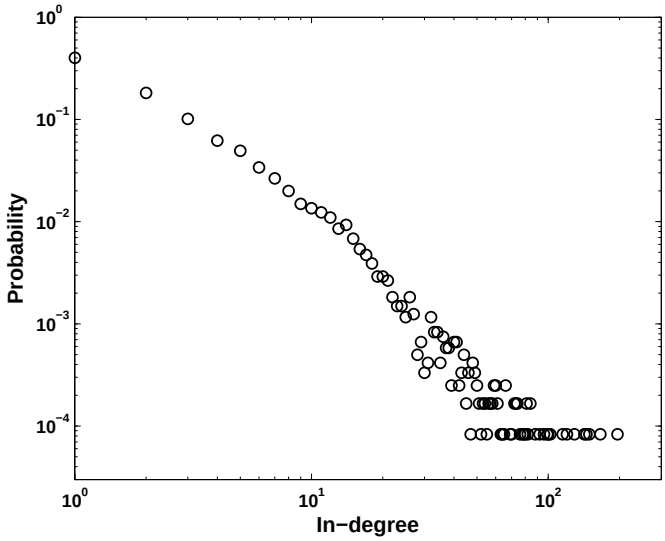


Figure 3: The in-degree distribution for the blog dataset.

in the site "goo" (`http://blog.goo.ne.jp/usertheme/`), where blog authors could recruit trackbacks of other blog authors by registering interesting themes. We collected a large-scale connected trackback network in May, 2005 by the following breadth first search process:

1. We started the process from the blog of the theme "JR Fukuchiyama Line Derailment Collision" in the site "goo", analyzed its HTML file, and extracted the list of the URLs of the source blogs of the trackbacks to this blog.

2. For each list obtained, we collected the blogs of the URLs in the list.

3. For each blog collected, we analyzed its HTML file, and constructed the list of the URLs of the source blogs of the trackbacks to the blog.

4. We repeated from Step 2 until depth ten from the original blog.

We call this network data the blog dataset. This network was a directed graph of $12,047$ nodes and $53,315$ links, and is expected to have a feature of real world social network in light of the way it is generated. To confirm this, the out-degree and in-degree distributions are plotted in Figures 2 and 3, from which it is understood that these are "heavy-tailed" distributions that most large real networks exhibit. Here, the out-degree and in-degree distributions are the distributions of the number of outgoing and incoming links for every node, respectively. Thus, we believe that the blog dataset is a typical example of a large real social network represented by a directed

graph, and can be used as the network data to evaluate the performance of the proposed method.

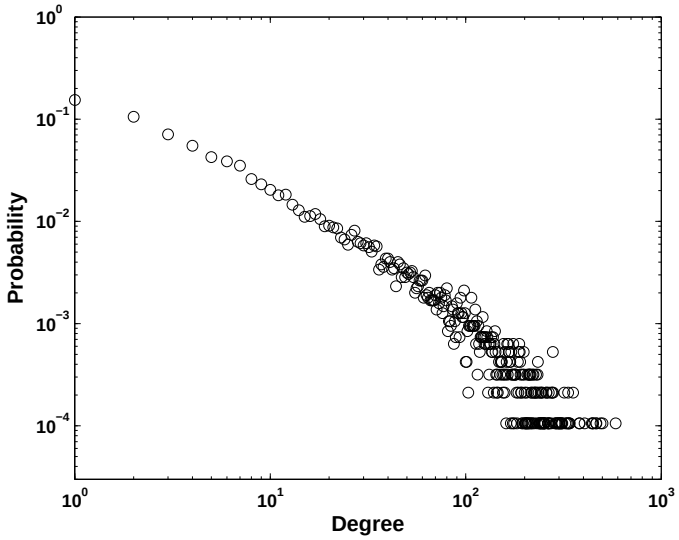


Figure 4: The degree distribution for the Wikipedia dataset.

Next, we employed a network of people that was derived from the "list of people" within Japanese Wikipedia. Specifically, we extracted the maximal connected component of the undirected graph obtained by linking two people in the "list of people" if they co-occur in six or more Wikipedia pages, and constructed a directed graph by regarding those undirected links as bidirectional ones. We call this network data the Wikipedia dataset. The total numbers of nodes and directed links were $9,481$ and $245,044$, respectively. Compared with the blog network, the way this network is generated is rather synthetically. Figure 4 shows the degree distribution of the undirected graph. We also observe that the degree distribution is a "heavy-tailed" distribution.

For social networks represented as undirected graphs, Newman and Park (2003) observed that they generally have the following two statistical properties that non-social networks do not have. First, they show positive correlations between the degrees of adjacent nodes. Second, they have much higher values of the *clustering coefficient* than the corresponding *configuration models* (i.e., random network models). Here, the clustering coefficient C for an undirected graph is defined by

$$C = \frac{3 \times \text{number of triangles on the graph}}{\text{number of connected triples of nodes}},$$

where a "triangle" means a set of three nodes each of which is connected to each other, and a "connected triple" means a node connected directly to

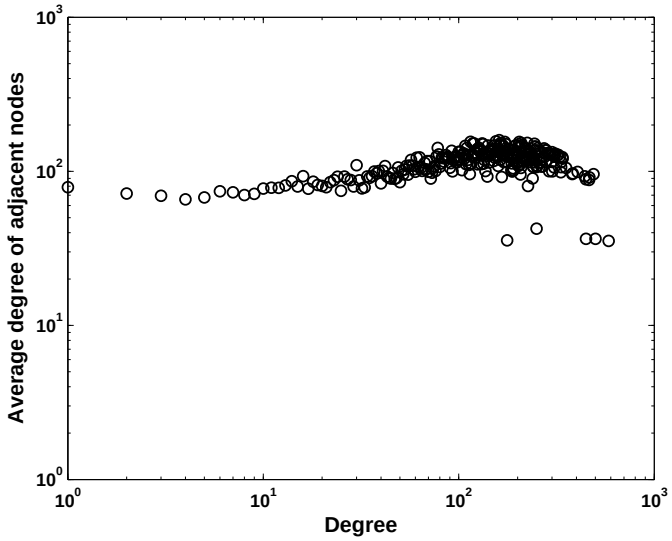


Figure 5: The degree correlation for the Wikipedia dataset.

unordered other pair nodes. Note that in terms of sociology, C measures the probability that two of your friends will also be friends each other. Given a degree distribution $\{\lambda_d\}$, the corresponding configuration model of a random network of N nodes is defined as the ensemble of all possible undirected graphs of N nodes that possess the degree distribution $\{\lambda_d\}$, where λ_d is the fraction of nodes in the network having degree d. It is known [18] that the value of C for the configuration model is exactly calculated by

$$C \;=\; \frac{1}{Nz_1}\left(\frac{z_2}{z_1}\right)^2,$$

where

$$z_1 \;=\; \sum_d \; d\lambda_d$$

is the average number of neighbors of a node and

$$z_2 \;=\; \sum_d \; d^2\lambda_d \;-\; \sum_d \; d\lambda_d$$

is the average number of second neighbors. For the undirected graph of the Wikipedia dataset, the value of C of the corresponding configuration model was 0.046, while the actual measured value of C was 0.39. Namely, the undirected graph of the Wikipedia dataset had a much higher value of the clustering coefficient than the corresponding configuration model. Moreover, we can see from Figure 5 that the Wikipedia dataset had weakly positive degree correlation. Therefore, we believe that the Wikipedia dataset is also

a typical example of a large real social network represented by an undirected graph, and can be used as the network data to evaluate the performance of the proposed method.

5.2 Experimental Settings

The proposed and the conventional methods are equipped with parameter M. We refer to the conventional method with $M = 1,000$ for the IC model as the *IC1000*. In the same way, we define the *LT1000* and *LT10000* for the conventional method with the LT model. We also refer to the proposed method using $M = 1,000$ and $M = 10,000$ for the IC model as the *ICBP1000* and *ICBP10000*, respectively. In the same way, we define the *LTBP1000* and *LTBP10000* for the proposed method with the LT model. As described in Section 4.4, we compare these methods for the same value of M.

The IC and LT models have parameters to be specified in advance. In the IC model, we assigned a uniform probability p to the propagation probability $p_{u,v}$ for any directed link (u, v) of the network, that is, $p_{u,v} = p$. In the LT model, we uniformly set weights as follows: For any node v of the network, the weight $w_{u,v}$ from a parent node $u \in \Gamma(v)$ is given by $w_{u,v} = 1/|\Gamma(v)|$.

We implemented all our programs of both the conventional and proposed methods for the IC and LT models in C language. Of course, the basic structure of these programs is the same, except that the routines of active node calculation used in the conventional method are replaced with those of bond percolation and SCC decomposition used in the proposed method.

5.3 Experimental Results

We compared the proposed method with the conventional method in terms of both the performance of the approximate solution A_k and the processing time for solving the influence maximization problem of size k. The performance of A_k is measured by the influence degree $\sigma(A_k)$. We estimated $\sigma(A_k)$ by using $300,000$ simulations according to the work of Kempe et al. (2003). All our experimentation was undertaken on a single Dell PC with an Intel 3.4GHz Xeon processor, with 2GB of memory, running under Linux.

In order to keep computational time at a reasonable level for the conventional method, we mainly compared these two methods using $M = 1,000$. Note that if a large enough M is taken, these two methods should produce the same solution. We conjecture that $M = 1,000$ is not large enough, that is, these two methods with $M = 1,000$ cannot necessarily obtain good approximate values for the marginal influence degrees $\{\sigma(A \cup \{v\});\ v \in V \setminus A\}$ of A, (see Appendices A and B). Thus, we iterated the same experiment five times independently. Tables 1 and 2 show the experimental results for the IC model with $p = 10\%$ and the LT model for the blog dataset, respectively, where the values are rounded to three significant figures. Note that

Table 1: Performance of approximate solutions for the influence maximization problem under the IC model with $p = 10\%$ for the blog dataset. Upper: IC1000 (the conventional method). Lower: ICBP1000 (the proposed method).

k	$\sigma(A_k)$ (IC1000)				
1	1.74×10^2	1.74×10^2	1.74×10^2	1.74×10^2	1.74×10^2
10	6.93×10^2	6.98×10^2	6.93×10^2	6.91×10^2	6.95×10^2
20	8.58×10^2	8.61×10^2	8.57×10^2	8.58×10^2	8.60×10^2
30	9.59×10^2	9.69×10^2	9.68×10^2	9.66×10^2	9.78×10^2

k	$\sigma(A_k)$ (ICBP1000)				
1	1.74×10^2	1.74×10^2	1.74×10^2	1.74×10^2	1.74×10^2
10	7.02×10^2	7.01×10^2	7.00×10^2	7.01×10^2	7.02×10^2
20	8.74×10^2	8.75×10^2	8.73×10^2	8.74×10^2	8.73×10^2
30	9.91×10^2	9.92×10^2	9.90×10^2	9.92×10^2	9.92×10^2

in these tables and later ones, too, the values are reestimated with $300,000$ simulations once A_k has been obtained by each method with a specified M. Since the true solution $\sigma(A_k^*)$ is by definition the maximum among all $\sigma(A_k)$, if $\sigma(A_k)$ is estimated accurately, it makes sense to argue that the larger the value is, the closer it is to the true solution and thus it is of better quality. We first observe that the results for the proposed method were relatively stable over the iterations, while the results for the conventional method somewhat fluctuated for large k in particular. Here, we note that the proposed method using $M = 10,000$ was stable and always produced the same solution for $k = 30$ over the iterations (not shown in the tables). We also observe that for $k = 30$, the solutions by the ICBP1000 and LTBP1000 outperforms those by the IC1000 and LT1000, respectively.

Table 3 shows the processing time to obtain A_k by the IC1000, ICBP1000, LT1000 and LTBP1000 for the blog dataset, where the values are rounded to three significant figures. We observe from Table 3 that the ICBP1000 and LTBP1000 are much more efficient than the IC1000 and LT1000, respectively. For example, to obtain the approximate solution A_{30} for $k = 30$, both the IC1000 and LT1000 needed about 2.5 days, while the ICBP1000 and LTBP1000 needed about 2.5 and 1.5 minutes, respectively. Namely, for $k = 30$, the ICBP1000 was 1.8×10^3 times faster than the IC1000, and the LTBP1000 was 4.6×10^3 times faster than the LT1000. We also examined the LT10000 and LTBP10000 on the blog dataset. In order to obtain approximate solution A_{30}, the LT10000 needed about 27 days, while the LTBP10000 needed only about 14 minutes.

Table 2: Performance of approximate solutions for the influence maximization problem under the LT model for the blog dataset. Upper: LT1000 (the conventional method). Lower: LTBP1000 (the proposed method).

k	$\sigma(A_k)$ (LT1000)				
1	2.86×10^2	2.86×10^2	2.86×10^2	2.86×10^2	2.86×10^2
10	1.59×10^3	1.61×10^3	1.61×10^3	1.59×10^3	1.58×10^3
20	2.41×10^3	2.40×10^3	2.42×10^3	2.42×10^3	2.38×10^3
30	3.02×10^3	3.05×10^3	3.01×10^3	3.01×10^3	3.00×10^3

k	$\sigma(A_k)$ (LTBP1000)				
1	2.86×10^2	2.86×10^2	2.86×10^2	2.86×10^2	2.86×10^2
10	1.60×10^3	1.61×10^3	1.61×10^3	1.59×10^3	1.60×10^3
20	2.44×10^3	2.44×10^3	2.44×10^3	2.44×10^3	2.44×10^3
30	3.07×10^3	3.07×10^3	3.06×10^3	3.06×10^3	3.06×10^3

Table 3: Processing time (sec.) for the blog dataset.

k	IC1000	ICBP1000	LT1000	LTBP1000
1	3.70×10^2	7.07	6.57×10^2	3.19
10	4.69×10^4	5.68×10^1	4.24×10^4	2.96×10^1
20	1.24×10^5	1.09×10^2	1.25×10^5	5.64×10^1
30	2.13×10^5	1.60×10^2	2.32×10^5	8.20×10^1

Tables 4, 5 and 6 show the experimental results for the Wikipedia dataset. We see that the results were qualitatively very similar to the ones for the blog dataset. First, the solutions by the ICBP1000 and LTBP1000 outperformed those by the IC1000 and LT1000, respectively. We also note that the proposed method using $M = 10,000$ was stable and always produced the same solution for $k = 30$ over the iterations (not shown in the tables). Next, the ICBP1000 and LTBP1000 were much more efficient than the IC1000 and LT1000, respectively. For example, for obtaining the approximate solution A_{30} for $k = 30$, the ICBP1000 was 1.9×10^3 times faster than the IC1000, and the LTBP1000 was 8.3×10^3 times faster than the LT1000. We also conducted experiments on some other large-scale real networks including a blogroll network of blogs, and confirmed the effectiveness of the proposed method.

Table 4: Performance of approximate solutions for the influence maximization problem under the IC model with $p = 1\%$ for the Wikipedia dataset. Upper: IC1000 (the conventional method). Lower: ICBP1000 (the proposed method).

k	$\sigma(A_k)$ (IC1000)				
1	1.39×10^2	1.39×10^2	1.36×10^2	1.36×10^2	1.36×10^2
10	3.91×10^2	3.97×10^2	3.98×10^2	4.02×10^2	4.01×10^2
20	4.56×10^2	4.64×10^2	4.62×10^2	4.64×10^2	4.66×10^2
30	4.97×10^2	5.02×10^2	4.95×10^2	5.00×10^2	4.98×10^2

k	$\sigma(A_k)$ (ICBP1000)				
1	1.39×10^2	1.39×10^2	1.39×10^2	1.36×10^2	1.36×10^2
10	4.05×10^2	4.06×10^2	4.07×10^2	4.06×10^2	4.07×10^2
20	4.75×10^2	4.76×10^2	4.76×10^2	4.75×10^2	4.77×10^2
30	5.16×10^2	5.17×10^2	5.17×10^2	5.16×10^2	5.17×10^2

5.4 Discussion

These experimental results show that the proposed method is much more efficient than the conventional method.

First, we investigate the reason why the proposed method outperforms the conventional method in the case of $M = 1,000$ for our network datasets. If we take a sufficiently large M (e.g., $M = 100,000$), the proposed and the conventional methods should produce the same solution. As shown in the experiments, the estimation accuracy of influence degree function σ with $M = 1,000$ is not so high for the both methods. Now, consider estimating all the marginal influence degrees $\{\sigma(A_k \cup \{v\});\ v \in V \setminus A_k\}$ of solution A_k, and choosing the node v_{k+1} that maximizes $\sigma(A_k \cup \{v\})$, $(v \in V \setminus A_k)$. It should be reemphasized that the influence set of A_k is equally evaluated for all $v \in V \setminus A_k$ for the proposed method. In fact, when $\sigma(A_k \cup \{v\})$ is estimated using Equation (3), each $|F(A_k \cup \{v\}; G_{r_m})|$ is basically computed by

$$|F(A_k \cup \{v\}; G_{r_m})| = \left|F(v; G_{r_m}^{A_k})\right| + |F(A_k; G_{r_m})|\,.$$

Thus, for the proposed method, a node that is relatively optimal for A_k can be selected as v_{k+1}. On the other hand, for the conventional method, the influence set of A_k is not equally evaluated for all $v \in V \setminus A_k$ since $\sigma(A_k \cup \{v\})$ is independently estimated for every v each by a distinct simulation. We also note that the number of final active nodes for a given target set greatly varied for every simulation in the IC and LT models (see, Appendix B). Thus, unlike the proposed method, the selection of v_{k+1} in the conventional method

Table 5: Performance of approximate solutions for the influence maximization problem under the LT model for the Wikipedia dataset. Upper: LT1000 (the conventional method). Lower: LTBP1000 (the proposed method).

k	$\sigma(A_k)$ (LT1000)				
1	3.41×10^2	3.41×10^2	3.41×10^2	3.41×10^2	3.41×10^2
10	1.72×10^3	1.72×10^3	1.67×10^3	1.66×10^3	1.72×10^3
20	2.55×10^3	2.55×10^3	2.45×10^3	2.53×10^3	2.55×10^3
30	3.12×10^3	3.03×10^3	2.99×10^3	3.01×10^3	3.11×10^3

k	$\sigma(A_k)$ (LTBP1000)				
1	3.41×10^2	3.41×10^2	3.41×10^2	3.41×10^2	3.41×10^2
10	1.72×10^3	1.72×10^3	1.72×10^3	1.72×10^3	1.71×10^3
20	2.58×10^3	2.58×10^3	2.59×10^3	2.59×10^3	2.59×10^3
30	3.18×10^3	3.18×10^3	3.18×10^3	3.18×10^3	3.18×10^3

Table 6: Processing time (sec.) for the Wikipedia dataset.

k	IC1000	ICBP1000	LT1000	LTBP1000
1	6.63×10^2	1.91×10^1	5.41×10^2	5.17
10	1.94×10^5	1.74×10^2	9.60×10^4	4.64×10^1
20	4.82×10^5	3.42×10^2	3.03×10^5	8.57×10^1
30	8.03×10^5	5.10×10^2	5.69×10^5	1.21×10^2

using $M = 1,000$ by necessity completely depends on how the influence set of A_k is evaluated by chance for each $v \in V \setminus A_k$. Therefore, we believe that the proposed method outperforms the conventional method in the case of $M = 1,000$ for our network datasets.

Here, to explain the point of the reason described above more clearly, we consider the following method as an extended version of the conventional method:

1. **for** $m = 1$ to M **do**
2. Find the set $D(A_k)$ of active nodes at the end of the random process of the IC or the LT models for initial active set A_k by simulation.
3. **for** each $v \in V \setminus A_k$ **do**
4. Find the set $D(v)$ of active nodes at the end of the random process of the IC or the LT models for initial active set $\{v\}$ by simulation.

5. Set $x_{v,m} \leftarrow |D(A_k) \cup D(v)|$.

6. **end for**

7. **end for**

8. **for** each $v \in V \setminus A_k$ **do**

9. Set $\sigma(A_k \cup \{v\}) \leftarrow (1/M) \sum_{m=1}^{M} x_{v,m}$

10. **end for**

The extended method should improve the conventional method because the influence set of A_k is now equally evaluated for all $v \in V \setminus A_k$, and should be comparable to the proposed method in quality of solution. However, it cannot be as efficient as the proposed method since it does not incorporate the SCC-finding technique.

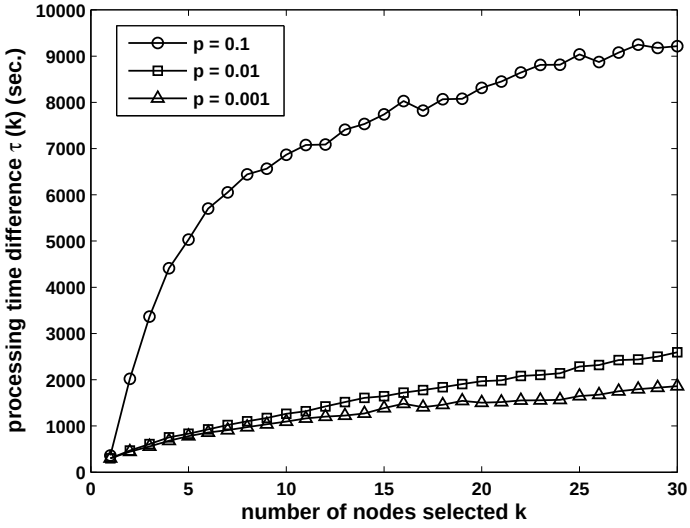


Figure 6: Processing time difference $\tau(k)$ between the proposed and conventional methods for the blog dataset in the case of the IC model.

Next, we discuss the sources of the difference between the proposed and conventional methods in processing time. Note that we use the same value of parameter M for both methods. Let $\tau_1(k)$ and $\tau_0(k)$ respectively denote the processing times of the proposed and the conventional methods for obtaining solution A_{k+1} when solution A_k is given. We define the processing time difference $\tau(k)$ by $\tau_0(k) - \tau_1(k)$ for k, the number of nodes selected. We believe the essential sources of speed-up in the proposed method is that we compute $\{|F(A_k \cup \{v\}; G_r)|;\, v \in V \setminus A_k\}$ on graph G_r as follows:

- By first identifying $F(A_k; G_r)$, we reduce the graph in question from G_r to the induced graph $G_r^{A_k}$ of G_r to $V \setminus F(A_k; G_r)$
- By decomposing $G_r^{A_k}$ into the SCCs, we compute $|F(A_k \cup \{v\}; G_r)|$ for many nodes v at once.

Namely, we believe that the larger the size of $F(A_k; G_r)$ is, the larger the value of $\tau(k)$ is. Moreover, we believe that the larger the sizes of the SCCs of graph $G_r^{A_k}$ are, the larger the value of $\tau(k)$ is. Here, we demonstrate these characteristics for the IC model. Note that the size of $F(A_k; G_r)$ monotonically increases with the value of k. Thus, we can expect that the value of $\tau(k)$ also monotonically increases with the value of k. Note also that graph G_r becomes denser when the value of the propagation probability p is larger, and the sizes of the SCCs of G_r also become larger. Thus, we can also expect that the value of $\tau(k)$ monotonically increases with the value of p. Figure 6 shows $\tau(k)$ for $p = 0.1\%$, 1% and 10% as a function of k for the blog dataset, where circles, squares and diamonds indicate $\tau(k)$ for $p = 0.1\%$, 1% and 10%, respectively. Here, we used $M = 1,000$ for both the proposed and the conventional methods. The results support our conjectures.

6 Related Work

6.1 Calculation of Influence Degrees

First, we describe work related to the calculation of influence degrees in the IC model. Let us recall that the SIR model for the spread of a disease on a network is equivalent to a bond percolation model on the same network, and the size of a disease outbreak from a node corresponds to the size of the cluster that can be reached from the node by traversing only the "occupied" links. There are a series of work that uses this correspondence to develop a method for theoretically calculating the probability distribution of the size of a disease outbreak that starts with a randomly chosen node in the configuration model (i.e., a random network model) with a given degree distribution (Callaway et al., 2000; Newman, 2002; Newman, 2003), and to derive a condition for the disease outbreak from a randomly chosen node to give an *epidemic outbreak* that affects a non-zero fraction on the network in the limit of very large network. Mathematically more rigorous treatments of similar results can be found in the work of Molloy and Reed (1998) and Chung and Lu (2002).

Next, we describe work related to the calculation of influence degrees in the LT model. Watts (2002) investigated the LT model on a network to explain large but rare cascade phenomena triggered by small initial shocks. Using the concept of *site percolation*, he theoretically derived a condition for the cascade from a randomly chosen seed node to give a *global cascade* that affects a non-zero fraction on the network in the limit of infinitely large

network for the configuration model (i.e., a random network model) with a given degree distribution.

The above mentioned studies focused on global properties averaged over a random network in the limit of very large size, while our primary interest is to practically answer which nodes are most influential for information diffusion on a given real-world network of a finite size. We also note that those studies dealt with undirected graphs, while our work investigates information diffusion on networks represented by directed graphs. Moreover, the theories developed in those studies assumed that the loop structure on a network of interest can be essentially ignored in the limit of large network size. However, this property is not true of many large-scale social networks, and it is an open question whether or not those theories are effective for such networks (Newman, 2003). In fact, the clustering coefficient C quantifies the loop structure in a network, and it was indeed observed that many social networks have much higher values of C than the corresponding configuration models (i.e., random network models) (Newman and Park, 2003).

6.2 Solving the Influence Maximization Problem

The influence degree function σ is submodular (see, Kempe et al., 2003). For solving a combinatorial optimization problem of a submodular function f on V by the greedy algorithm, Leskovec et al. (2007) have recently presented a lazy evaluation method that leads to far fewer (expensive) evaluations of the marginal increments $f(A \cup \{v\}) - f(A)$ $(v \in V \setminus A)$ in the greedy algorithm for $A \neq \emptyset$, and achieved an improvement in speed. Note here that their method requires evaluating $f(v)$ for all $v \in V$ at least. Thus, we can apply their method to the influence maximization problem for the IC or LT models, where the influence degree function σ is evaluated through the simulations of the corresponding random process. It is clear that this method is more efficient than the conventional method. However, the proposed method for $k = 30$ was faster than the conventional method for $k = 1$ as shown in Tables 3 and 6. Therefore, it is evident that the proposed method can be faster than the method by Leskovec et al. (2007) for the influence maximization problem for the IC or LT models. To quantify the difference we implemented the Lazy evaluation method. The processing time for $k = 30$ in case of the blog dataset was 2.12×10^3 and 8.28×10^2 seconds for the IC and the LT models, respectively, and the corresponding processing time in case of Wikipedia dataset was 1.46×10^4 and 2.65×10^3 seconds for the IC and the LT models, respectively. Here, $M = 1,000$ are used as the number of simulations (see, Section 3.2), and the values are rounded to three significant figures. From these results, we can see that the proposed method was more than ten times faster than the method by Leskovec et al. (2007) for $k = 30$ in the blog and Wikipedia datasets (see, Tables 3 and 6).

Beyond the IC and LT models, Kempe et al. (2003) proposed the *trig-*

gering model as an yet another diffusion model on a network. It is proved that the triggering model can be identified with a bond percolation model (see, Kempe et al., 2003). The proposed method can be applied to this model because it can be applied to any diffusion model that can be identified with a bond percolation model. The future work includes presenting a large number of realistic examples of such diffusion models.

In this paper, we have considered the *progressive* case in which nodes cannot switch from being active to being inactive. However, there are many information diffusion phenomena that non-progressive diffusion models are required. Examples include the spread of posts for a topic in blogspace (Gruhl et al, 2004). Kempe et al. (2003) proved that *non-progressive* case can be reduced to the progressive case. More specifically, it is proved that the influence maximization problem for a non-progressive diffusion model on graph G in time-limit T is equivalent to the ordinary influence maximization problem on the *layered graph* G_T for the progressive diffusion model, where G_T is the directed acyclic graph (DAG) constructed by time-forwardly connecting $(T+1)$ copies of G (see, Kempe et al. 2003). Therefore, building effective methods for fundamental progressive models such as the IC and LT models is indeed important and crucial for the non-progressive case.

From a realistic point of view, the IC and LT models are by no means a complete model, but are at best a simplified and partial representation of a complex reality (see, Kempe et al, 2003; Gruhl et al., 2004; Leskovec et al., 2006). However, in the field of sociology, Watts and Dodds (2007) recently examined the "influentials hypothesis" in the contexts of the LT model and the SIR model (i.e., an extended model of the IC model), that is, they investigated by computer simulations whether large cascades of influence are actually driven by influentials or not. On the other hand, Even-Dar and Shapira (2007) mathematically studied the influence maximization problem in the context of another fundamental model called the voter model. We also believe that it is important to investigate information diffusion phenomena for the IC and LT models (i.e., fundamental diffusion models) to deepen our understanding of these models. The future work includes proposing effective methods for solving the influence maximization problem in the contexts of various realistic diffusion models.

6.3 Applications

As is easily understood, the conventional method is not practical unless we rely on high-performance computers and sophisticated techniques such as parallel computing (see, Tables 3 and 6) to solve the kind of problems such as influence maximization problem as addressed in this paper. In contrast, the proposed method enables us to obtain a practical solution to this kind of problems on a single standard PC in a reasonable processing time. Thus, we can apply the proposed method to a variety of real problems.

The work of Watts and Dodds (2007) briefly described above needs a method to efficiently estimate $\sigma(A)$ and the proposed method can readily be applicable.

As mentioned in the introduction, the influence maximization problem finds many realistic applications. The most straightforward application would be viral marketing. When we wish to promote a new product (e.g., an email service or a search engine), and are given a relevant social network, we can easily find a limited number of key (influential) persons first to adopt the new product by the proposed method, and enjoy the diffusion effect for the IC or LT models (i.e., fundamental diffusion models) through the social network. We admit that the diffusion models we discussed are oversimplified but still it is useful to obtain approximate solutions as a first step toward an effective marketing without using classical advertising channels.

The proposed method has an application of different flavor which is the visualization of information flow. Understanding the flow of information through a complex network is important in terms of sociology and marketing. We devised a new node embedding method for visualizing the information diffusion process from the target nodes selected to be a solution of the influence maximization problem (Saito et al., 2008). This visualization method is characterized by 1) utilization of the target nodes as a set of pivot objects for visualization, 2) application of a probabilistic algorithm for embedding all the nodes in the network into an Euclidean space to conserve the posterior information diffusion probability, and 3) varying appearance of the embedded nodes on the basis of two label assignment strategies, one with emphasis on influence of initially activated nodes, and the other on degree of information reachability.

7 Conclusion

We have considered the influence maximization problem for the IC and LT models on a large-scale social network represented as a directed graph $G = (V, E)$. Due to the computational complexity, the greedy search algorithm is the only practical approach, but still the conventional method needed a high amount of computation. We have proposed a method of efficiently finding a good approximate solution to the problem under the greedy algorithm. In particular, in order to improve the computational efficiency, we have estimated all the marginal influence degrees $\{\sigma(A \cup \{v\});\ v \in V \setminus A\}$ of a given target set A in the following way:

- We identify the IC and LT models with the corresponding bond percolation models.
- For any $v \in V \setminus A$, we estimate the influence degree $\sigma(A \cup \{v\})$ of $A \cup \{v\}$ as the empirical mean of the number $|F(A \cup \{v\}; G_r)|$ of the

nodes that are reachable from $A \cup \{v\}$ on a graph G_r generated from the corresponding occupation probability distribution $q(r)$ of the bond percolation.

In particular, we estimate $\{|F(A \cup \{v\}; G_r)|;\ v \in V \setminus A\}$ as follows:

- We find the set $F(A; G_r)$ that is reachable from A on graph G_r, and simultaneously compute $\{|F(A \cup \{v\}; G_r)|;\ v \in F(A; G_r)\}$.
- We find the induced graph G_r^A of G_r to $V \setminus F(A; G_r)$, and decompose G_r^A into its SCCs (Strongly Connected Components).
- For each SCC $SCC(u; G_r^A)$ of G_r^A, $(u \in V \setminus F(A; G_r))$, we simultaneously compute $\{|F(A \cup \{v\}; G_r)|;\ v \in SCC(u; G_r^A)\}$.

We have compared the proposed method with the conventional method in terms of computational complexity and quality of the solution, and have shown that the proposed method is expected to achieve a large amount of reduction in computational cost. Moreover, using large-scale networks including a real blog network, we have experimentally demonstrated the effectiveness of the proposed method. For example, we obtained the following results for the influence maximization problem of size $k = 30$ on the blog and Wikipedia datasets that are real networks with about $10,000$ nodes: In the case of the IC model, the proposed method was 1800 times faster than the conventional method, and in the case of the LT model, the proposed method was 4600 times faster than the conventional method.

Acknowledgement

This work was partly supported by JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147), and Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-08-4027.

Appendix

A Convergence Speed

As described in Section 4.4, by using the same value of M, both the proposed and the conventional methods would estimate $\sigma(v)$ with the same accuracy in principle. Here, we experimentally demonstrate this conjecture.

According to the work of Kempe et al. (2003), we set $M = 300,000$ as a sufficiently large value of M, that is, we assume that $\sigma(v)$ for any $v \in V$ is well approximated by $300,000$ simulations of the information diffusion model (i.e., the conventional method using $M = 300,000$). For any $v \in V$, let $\sigma_0(v; M)$ and $\sigma_1(v; M)$ denote the estimates of $\sigma(v)$ by the conventional and the proposed methods using parameter value M, respectively. For the blog and Wikipedia datasets, we investigated

$$\mathcal{E} = \frac{1}{N} \sum_{v \in V} |\sigma_0(v; 300, 0000) - \sigma_1(v; 300, 000)|,$$

$$\mathcal{E}_0(M) = \frac{1}{N} \sum_{v \in V} |\sigma_0(v; M) - \sigma_0(v; 300, 000)|,$$

$$\mathcal{E}_1(M) = \frac{1}{N} \sum_{v \in V} |\sigma_1(v; M) - \sigma_1(v; 300, 000)|.$$

We first consider the case of the IC model. Then, the value of $\mathcal{E}$ was 0.03 and 0.04 for the blog and Wikipedia datasets, respectively. Thus, we can assume that the values of $\sigma_0(v; 300, 000)$ and $\sigma_1(v; 300, 000)$ are almost the same for any $v \in V$.

Table 7: Convergence speed for the blog dataset.

M	$\mathcal{E}_0(M)$	$\mathcal{E}_1(M)$
100	1.16	1.12
1,000	0.36	0.36
10,000	0.11	0.12
100,000	0.03	0.03

Table 8: Convergence speed for the Wikipedia dataset.

M	$\mathcal{E}_0(M)$	$\mathcal{E}_1(M)$
100	1.28	1.23
1,000	0.42	0.42
10,000	0.13	0.14
100,000	0.03	0.03

Tables 7 and 8 show the values of $\mathcal{E}_0(M)$ and $\mathcal{E}_1(M)$ for the blog and Wikipedia datasets, respectively. These results imply that the proposed and the conventional methods estimate $\{\sigma(v);\, v \in V\}$ with almost the same

accuracy for the IC model. We also obtained similar results for the case of the LT model. For example, the value of $\mathcal{E}$ was 0.03 and 0.09 for the blog and Wikipedia datasets, respectively. For the blog dataset, the values of $\mathcal{E}_0(10,000)$ and $\mathcal{E}_1(10,000)$ were 0.13 and 0.12, respectively. Also, for the Wikipedia datasets, the values of $\mathcal{E}_0(10,000)$ and $\mathcal{E}_1(10,000)$ were 0.36 and 0.37, respectively. These results support our conjecture.

B Fluctuation in Simulations of Information Diffusion Models

For each $v \in V$, we examine fluctuation in the number $\varphi(v)$ of the final active nodes for a target initially activated node v through $1,000$ simulations in the IC and LT models. Let $\mu(v)$ and $s(v)$ denote the empirical mean and the standard deviation of $\varphi(v)$ for $1,000$ simulations, respectively. We define $\overline{\mu}$ and $\overline{s}$ by the empirical means of $\{\mu(v);\ v \in V\}$ and $\{s(v);\ v \in V\}$, respectively. For the blog dataset, $\overline{\mu}$ and $\overline{s}$ were as follows:

IC model ($p = 10\%$): $\overline{\mu} = 8.6$, $\overline{s} = 14.3$.

LT model: $\overline{\mu} = 6.8$, $\overline{s} = 14.9$.

For the Wikipedia dataset, $\overline{\mu}$ and $\overline{s}$ were as follows:

IC model ($p = 1\%$): $\overline{\mu} = 8.1$, $\overline{s} = 16.1$,

LT model: $\overline{\mu} = 12.6$, $\overline{s} = 42.4$,

Here, the values are rounded to the first decimal place. We can observe that compared with $\overline{\mu}$, $\overline{s}$ is very large. Therefore, we see that the number of final active nodes for a given target set can greatly vary for every simulation in the IC and LT models.

References

[1] Callaway, D. S., Newman, M. E. J., and Strogatz, S. H. 2000. Network robustness and fragility: Percolation on random graphs. Physical Review Letters, 85:5468–5471.

[2] Chung, F. and Lu, L. 2002. Connected components in a random graph with given expected degree sequences. Annals of Combinatorics, 6:125-145.

[3] Domingos, P. 2005. Mining social networks for viral marketing. IEEE Intelligent Systems, 20:80–82.

[4] Domingos, P. and Richardson, M. 2001. Mining the network value of customers. Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA, pp. 57–66.

[5] Even-Dar, E. and Shapira, A. 2007. A note on maximizing the spread of influence in social networks. Internet and Network Economics: WINE 2007, LNCS 4858, pp. 281–286.

[6] Goldenberg, J., Libai, B., and Muller, E. 2001. Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters, 12:211–223.

[7] Grassberger, P. 1983. On the critical behavior of the general epidemic process and dynamical percolation. Mathematical Bioscience, 63:157–172.

[8] Gruhl, D., Guha, R., Liben-Nowell, D., and Tomkins, A. 2004. Information diffusion through blogspace. Proceedings of the 7th International World Wide Web Conference, New York, USA, pp. 107–117.

[9] Kempe, D., Kleinberg, J., and Tardos, E. 2003. Maximizing the spread of influence through a social network. Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, pp. 137–146.

[10] Kempe, D., Kleinberg, J., and Tardos, E. 2005. Influential nodes in a diffusion model for social networks. Automata, Languages and Programming: ICALP 2005, LNCS 3580, pp. 1127–1138.

[11] Leskovec, J., Adamic, L. A., and Huberman, B. A. 2006. The dynamics of viral marketing. Proceedings of the 7th ACM Conference on Electronic Commerce, Ann Arbor, Michigan, USA, pp. 228–237.

[12] Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., and Glance, N. 2007. Cost-effective outbreak detection in networks. Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, pp. 420–429.

[13] McCallum, A., Corrada-Emmanuel, A., and Wang, X. 2005. Topic and role discovery in social networks. Proceedings of the 19th International Joint Conference on Artificial Intelligence, Edinburugh, Scotland, pp. 786–791.

[14] Molloy, M. and Reed, B. 1998. The size of the giant component of a random graph with a given degree sequence. Combinatorics, Probability and Computing, 7:295–305.

[15] Newman, M. E. J. and Park, J. 2003. Why social networks are different from other types of networks. Physical Review E, 68:036122.

[16] Newman, M. E. J. 2001. The structure of scientific collaboration networks. Proceedings of the National Academy of Sciences of the United States of America, 98:404–409.

[17] Newman, M. E. J. 2002. Spread of epidemic disease on networks. Physical Review E, 66:016128.

[18] Newman, M. E. J. 2003. The structure and function of complex networks. SIAM Review, 45:167–256.

[19] Richardson, M. and Domingos, P. 2002. Mining knowledge-sharing sites for viral marketing. Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, pp. 61–70.

[20] Saito, K., Kimura, M., and Motoda, H. 2008. Effective visualization of information diffusion process over complex networks. Machine Learning and Knowledge Discovery in Databases: ECML PKDD 2008, LNAI 5212, pp. 326–341.

[21] Watts, D. J. 2002. A simple model of global cascades on random networks. Proceedings of the National Academy of Sciences of the United States of America, 99:5766–5771.

[22] Watts, D. J. and Dodds, P. S. 2007. Influence, networks, and public opinion formation. Journal of Consumer Research, 34:441–458.