Discovering Influential Nodes for SIS models in Social Networks

Kazumi Saito[1], Masahiro Kimura[2], and Hiroshi Motoda[3]

[1] School of Administration and Informatics, University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp
[2] Department of Electronics and Informatics, Ryukoku University
Otsu, Shiga 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[3] Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. We address the problem of efficiently discovering the influential nodes in a social network under the *susceptible/infected/susceptible (SIS) model*, a diffusion model where nodes are allowed to be activated multiple times. The computational complexity drastically increases because of this multiple activation property. We solve this problem by constructing a layered graph from the original social network with each layer added on top as the time proceeds, and applying the bond percolation with pruning and burnout strategies. We experimentally demonstrate that the proposed method gives much better solutions than the conventional methods that are solely based on the notion of centrality for social network analysis using two large-scale real-world networks (a blog network and a wikipedia network). We further show that the computational complexity of the proposed method is much smaller than the conventional naive probabilistic simulation method by a theoretical analysis and confirm this by experimentation. The properties of the influential nodes discovered are substantially different from those identified by the centrality-based heuristic methods.

1 Introduction

Social networks mediate the spread of various information including topics, ideas and even (computer) viruses. The proliferation of emails, blogs and social networking services (SNS) in the World Wide Web accelerates the creation of large social networks. Therefore, substantial attention has recently been directed to investigating information diffusion phenomena in social networks [1–3].

Overall, finding influential nodes is one of the most central problems in social network analysis. Thus, developing methods to do this on the basis of information diffusion is an important research issue. Widely-used fundamental probabilistic models of information diffusion are the *independent cascade (IC) model* and the *linear threshold (LT) model* [4, 5]. Researchers investigated the problem of finding a limited number of influential nodes that are effective for the spread of information under the above models [4,

6]. This combinatorial optimization problem is called the *influence maximization problem*. Kempe et al. [4] experimentally showed on large collaboration networks that the greedy algorithm can give a good approximate solution to this problem, and mathematically proved a performance guarantee of the greedy solution (i.e., the solution obtained by the greedy algorithm). Recently, methods based on bond percolation [6] and submodularity [7] were proposed for efficiently estimating the greedy solution. The influence maximization problem has applications in sociology and "viral marketing" [3], and was also investigated in a different setting (a descriptive probabilistic model of interaction) [8, 9]. The problem has recently been extended to influence control problems such as a contamination minimization problem [10].

The IC model can be identified with the so-called *susceptible/infected/recovered (SIR) model* for the spread of a disease [11, 5]. In the SIR model, only infected individuals can infect susceptible individuals, while recovered individuals can neither infect nor be infected. This implies that an individual is never infected with the disease multiple times. This property holds true for the LT model as well. However, there exist phenomena for which the property does not hold. For example, consider the following propagation phenomenon of a topic in the blogosphere: A blogger who has not yet posted a message about the topic is interested in the topic by reading the blog of a friend, and posts a message about it (i.e., becoming infected). Next, the same blogger reads a new message about the topic posted by some other friend, and may post a message (i.e., becoming infected) again. Most simply, this phenomenon can be modeled by an *susceptible/infected/susceptible (SIS) model* from the epidemiology. Like this example, there are many examples of information diffusion phenomena for which the SIS model is more appropriate, including the growth of hyper-link posts among bloggers [2], the spread of computer viruses without permanent virus-checking programs, and epidemic disease such as tuberculosis and gonorrhea [11].

We focus on an information diffusion process in a social network $G = (V, E)$ over a given time span T on the basis of an SIS model. Here, the SIS model is a stochastic process model, and the *influence* of a set of nodes H at time-step t, $\sigma(H, t)$, is defined as the expected number of infected nodes at time-step t when all the nodes in H are initially infected at time-step $t = 0$. We refer to σ as the *influence function* for the SIS model. Developing an effective method for estimating $\sigma(\{v\}, t)$, $(v \in V, t = 1, \ldots, T)$ is vital for various applications. Clearly, in order to extract influential nodes, we must estimate the value of $\sigma(\{v\}, t)$ for every node v and time-step t. Thus, we proposed a novel method based on the bond percolation with an effective pruning strategy to efficiently estimate $\{\sigma(\{v\}, t);\ v \in V, t = 1, \ldots, T\}$ for the SIS model in our previous work [12].

In this paper, we consider solving the influence maximization problems on a network $G = (V, E)$ under the SIS model. Here, unlike the cases of the IC and the LT models, we define two influence maximization problems, the *final-time maximization problem* and the *accumulated-time maximization problem*, for the SIS model. We introduce the greedy algorithm for solving the problems according to the work of Kempe et al. [4] for the IC and the LT models. Now, let us consider the problem of influence maximization at the final time step T (i.e., final-time maximization problem) as an example. We then note that for solving this problem by the greedy algorithm, we need a method for not only evaluating $\{\sigma(\{v\}, T);\ v \in V\}$, but also evaluating the *marginal influence*

gains $\{\sigma(H \cup \{v\}, T) - \sigma(H, T);\ v \in V \setminus H\}$ for any non-empty subset H of V. Needless to say, we can naively estimate the marginal influence gains for any non-empty subset H of V by simulating the SIS model[2]. However, this naive simulation method is overly inefficient and not practical at all. In this paper, by incorporating the new techniques (the pruning and the burnout methods) into the bond percolation method, we propose a method to efficiently estimate the marginal influence gains for any non-empty subset H of V, and apply it to approximately solve the two influence maximization problems for the SIS model by the greedy alogrithm. We show that the proposed method is expected to achieve a large reduction in computational cost by theoretically comparing computational complexity with other more naive methods. Further, using two large real networks, we experimentally demonstrate that the proposed method is much more efficient than the naive greedy method based on the bond percolation method. We also show that the discovered nodes by the proposed method are substantially different from and can result in considerable increase in the influence over the conventional methods that are based on the notion of various centrality measures.

2 Information Diffusion Model

Let $G = (V, E)$ be a directed network, where V and E ($\subset V \times V$) stand for the sets of all the nodes and (directed) links, respectively. For any $v \in V$, let $\Gamma(v; G)$ denote the set of the child nodes (directed neighbors) of v, that is,

$$\Gamma(v; G) = \{w \in V;\ (v, w) \in E\}.$$

2.1 SIS Model

An SIS model for the spread of a disease is based on the cycle of disease in a host. A person is first *susceptible* to the disease, and becomes *infected* with some probability when the person encounters an infected person. The infected person becomes susceptible to the disease soon without moving to the immune state. We consider a discrete-time SIS model for information diffusion on a network. In this context, infected nodes mean that they have just adopted the information, and we call these infected nodes *active* nodes.

We define the SIS model for information diffusion on G. In the model, the diffusion process unfolds in discrete time-steps $t \geq 0$, and it is assumed that the state of a node is either active or inactive. For every link $(u, v) \in E$, we specify a real value $p_{u,v}$ with $0 < p_{u,v} < 1$ in advance. Here, $p_{u,v}$ is referred to as the *propagation probability* through link (u, v). Given an initial set of active nodes X and a time span T, the diffusion process proceeds in the following way. Suppose that node u becomes active at time-step t ($< T$). Then, node u attempts to activate every $v \in \Gamma(u; G)$, and succeeds with probability $p_{u,v}$. If node u succeeds, then node v will become active at time-step $t + 1$. If multiple active nodes attempt to activate node v in time-step t, then their activation attempts are sequenced in an arbitrary order. On the other hand, node u will become or remain inactive at time-step $t + 1$ unless it is activated from an active node in time-step t. The process terminates if the current time-step reaches the time limit T.

[2] Note that the method we proposed in [12] does not perform simulation.

2.2 Influence Function

For the SIS model on G, we consider a diffusion sample from an initially activated node set $H \subset V$ over time span T. Let $S(H,t)$ denote the set of active nodes at time-step t. Note that $S(H,t)$ is a random subset of V and $S(H,0) = H$. Let $\sigma(H,t)$ denote the expected number of $|S(H,t)|$, where $|X|$ stands for the number of elements in a set X. We call $\sigma(H,t)$ the *influence* of node set H at time-step t. Note that σ is a function defined on $2^{|V|} \times \{0,1,\cdots,T\}$. We call the function σ the *influence function* for the SIS model over time span T on network G. In view of more complex social influence, we need to incorporate a number of social factors with social networks such as rank, prestige and power. In our approach, we can encode such factors as diffusion probabilities of each node.

It is important to estimate the influence function σ efficiently. In theory we can simply estimate σ by the simulations based on the SIS model in the following way. First, a sufficiently large positive integer M is specified. For each $H \subset V$, the diffusion process of the SIS model is simulated from the initially activated node set H, and the number of active nodes at time-step t, $|S(H,t)|$, is calculated for every $t \in \{0,1,\cdots,T\}$. Then, $\sigma(H,t)$ is estimated as the empirical mean of $|S(H,t)|$'s that are obtained from M such simulations. However, this is extremely inefficient, and cannot be practical.

3 Influence Maximization Problem

We mathematically define the influence maximization problems on a network $G = (V,E)$ under the SIS model. Let K be a positive integer with $K < |V|$. First, we define the *final-time maximization problem*: Find a set H_K^* of K nodes to target for initial activation such that $\sigma(H_K^*;T) \geq \sigma(H;T)$ for any set H of k nodes, that is, find

$$H_K^* = \arg \max_{\{H \subset V;\, |H|=K\}} \sigma(H;T). \tag{1}$$

Second, we define the *accumulated-time maximization problem*: Find a set H_K^* of K nodes to target for initial activation such that $\sigma(H_K^*;1) + \cdots + \sigma(H_K^*;T) \geq \sigma(H;1) + \cdots + \sigma(H;T)$ for any set H of k nodes, that is, find

$$H_K^* = \arg \max_{\{H \subset V;\, |H|=K\}} \sum_{t=1}^{T} \sigma(H;t). \tag{2}$$

The first problem cares only how many nodes are influenced at the time of interest. For example, in an election campaign it is only those people who are convinced to vote the candidate at the time of voting that really matter and not those who were convinced during the campaign but changed their mind at the very end. Maximizing the number of people who actually vote falls in this category. The second problem cares how many nodes have been influenced throughout the period of interest. For example, maximizing the amount of product purchase during a sales campaign falls in this category.

4 Proposed Method

Kempe et al. [4] 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 for the SIS model, and describe some techniques (the bond percolation method, the pruning method, and the burnout method) for efficiently solving the influence maximization problem under the greedy algorithm, together with some arguments for evaluating the computational complexity for these methods.

4.1 Greedy Algorithm

We approximately solve the influence maximization problem by the greedy algorithm. Below we describe this algorithm for the final-time maximization problem:

Greedy algorithm for the final-time maximization problem:
$\mathcal{A}$**1.** Set $H \leftarrow \emptyset$.
$\mathcal{A}$**2.** For $k = 1$ to K do the following steps:
$\mathcal{A}$**2-1.** Choose a node $v_k \in V \setminus H$ maximizing $\sigma(H \cup \{v\}, T)$.
$\mathcal{A}$**2-2.** Set $H \leftarrow H \cup \{v_k\}$.
$\mathcal{A}$**3.** Output H.

Here we can easily modify this algorithm for the accumulated-time maximization problem by replacing step $\mathcal{A}$2-1 as follows:

Greedy algorithm for the accumulated-time maximization problem:
$\mathcal{A}$**1.** Set $H \leftarrow \emptyset$.
$\mathcal{A}$**2.** For $k = 1$ to K do the following steps:
$\mathcal{A}$**2-1'.** Choose a node $v_k \in V \setminus H$ maximizing $\sum_{t=1}^{T} \sigma(H \cup \{v\}, t)$.
$\mathcal{A}$**2-2.** Set $H \leftarrow H \cup \{v_k\}$.
$\mathcal{A}$**3.** Output H.

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

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

that is, the quality guarantee of H_k is assured [4]. Here, H_k^* is the exact solution defined by Equation (1) or (2).

To implement the greedy algorithm, we need a method for estimating all the marginal influence degrees $\{\sigma(H \cup \{v\}, t);\ v \in V \setminus H\}$ of H in step $\mathcal{A}$2-1 or $\mathcal{A}$2-1' of the algorithm. In the subsequent subsections, we propose a method for efficiently estimating the influence function σ over time span T for the SIS model on network G.

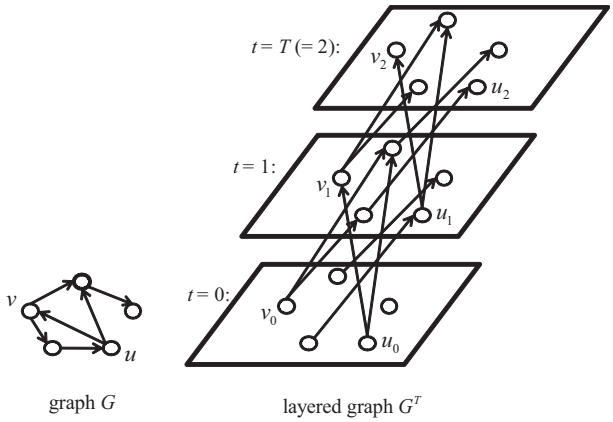

Fig. 1. An example of a layered graph.

4.2 Layered Graph

We build a layered graph $G^T = (V^T, E^T)$ from G in the following way (see Figure 1). First, for each node $v \in V$ and each time-step $t \in \{0, 1, \cdots, T\}$, we generate a copy v_t of v at time-step t. Let V_t denote the set of copies of all $v \in V$ at time-step t. We define V^T by $V^T = V_0 \cup V_1 \cup \cdots \cup V_T$. In particular, we identify V with V_0. Next, for each link $(u, v) \in E$, we generate T links (u_{t-1}, v_t), $(t \in \{1, \cdots, T\})$, in the set of nodes V^T. We set $E_t = \{(u_{t-1}, v_t); (u, v) \in E\}$, and define E^T by $E^T = E_1 \cup \cdots \cup E_T$. Moreover, for any link (u_{t-1}, v_t) of the layered graph G^T, we define the occupation probability q_{u_{t-1},v_t} by $q_{u_{t-1},v_t} = p_{u,v}$.

Then, we can easily prove that the SIS model with propagation probabilities $\{p_e; e \in E\}$ on G over time span T is equivalent to the *bond percolation process (BP) with occupation probabilities* $\{q_e; e \in E^T\}$ on G^T.[3] Here, the BP process with occupation probabilities $\{q_e; e \in E^T\}$ on G^T is the random process in which each link $e \in E^T$ is independently declared "occupied" with probability q_e. We perform the BP process on G^T, and generate a graph constructed by occupied links, $\tilde{G}^T = (V^T, \tilde{E}^T)$. Then, in terms of information diffusion by the SIS model on G, an occupied link $(u_{t-1}, v_t) \in E_t$ represents a link $(u, v) \in E$ through which the information propagates at time-step t, and an unoccupied link $(u_{t-1}, v_t) \in E_t$ represents a link $(u, v) \in E$ through which the information does not propagate at time-step t. For any $v \in V \setminus H$, let $F(H \cup \{v\}; \tilde{G}^T)$ be the set of all nodes that can be reached from $H \cup \{v\} \in V_0$ through a path on the graph $\tilde{G}^T$. When we consider a diffusion sample from an initial active node $v \in V$ for the SIS model on G, $F(H \cup \{v\}; \tilde{G}^T) \cap V_t$ represents the set of active nodes at time-step t, $S(H \cup \{v\}, t)$.

[3] The SIS model over time span T on G can be exactly mapped onto the IC model on G^T [4]. Thus, the result follows from the equivalence of the BP process and the IC model [11, 4, 6].

4.3 Bond Percolation Method

Using the equivalent BP process, we present a method for efficiently estimating influence function σ. We refer to this method as the *BP method*. Unlike the naive method, the BP method simultaneously estimates $\sigma(H \cup \{v\}, t)$ for all $v \in V \setminus H$. Moreover, the BP method does not fully perform the BP process, but performs it partially. Note first that all the paths from nodes $H \cup \{v\}$ ($v \in V \setminus H$) on the graph $\tilde{G}^T$ represent a diffusion sample from the initial active nodes $H \cup \{v\}$ for the SIS model on G. Let L' be the set of the links in G^T that is not in the diffusion sample. For calculating $|S(H \cup \{v\}, t)|$, it is unnecessary to determine whether the links in L' are occupied or not. Therefore, the BP method performs the BP process for only an appropriate set of links in G^T. The BP method estimates σ by the following algorithm:

BP method:
$\mathcal{B}$1. Set $\sigma(H \cup \{v\}, t) \leftarrow 0$ for each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$.
$\mathcal{B}$2. Repeat the following procedure M times:
$\mathcal{B}$2-1. Initialize $S(H \cup \{v\}, 0) = H \cup \{v\}$ for each $v \in V \setminus H$, and set $A(0) \leftarrow V \setminus H$, $A(1) \leftarrow \emptyset, \cdots, A(T) \leftarrow \emptyset$.
$\mathcal{B}$2-2. For $t = 1$ to T do the following steps:
$\mathcal{B}$2-2a. Compute $B(t-1) = \bigcup_{v \in A(t-1)} S(H \cup \{v\}, t-1)$.
$\mathcal{B}$2-2b. Perform the BP process for the links from $B(t-1)$ in G^T, and generate the graph $\tilde{G}_t$ constructed by the occupied links.
$\mathcal{B}$2-2c. For each $v \in A(t-1)$, compute $S(H \cup \{v\}, t) = \bigcup_{w \in S(H \cup \{v\}, t-1)} \Gamma(w; \tilde{G}_t)$, and set $\sigma(H \cup \{v\}, t) \leftarrow \sigma(H \cup \{v\}, t) + |S(H \cup \{v\}, t)|$ and $A(t) \leftarrow A(t) \cup \{v\}$ if $S(H \cup \{v\}, t) \neq \emptyset$.
$\mathcal{B}$3. For each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$, set $\sigma(H \cup \{v\}, t) \leftarrow \sigma(H \cup \{v\}, t)/M$, and output $\sigma(H \cup \{v\}, t)$.

Note that $A(t)$ finally becomes the set of information source nodes that have at least an active node at time-step t, that is, $A(t) = \{v \in V \setminus H;\, S(H \cup \{v\}, t) \neq \emptyset\}$. Note also that $B(t-1)$ is the set of nodes that are activated at time-step $t-1$ by some source nodes, that is, $B(t-1) = \bigcup_{v \in V} S(H \cup \{v\}, t-1)$.

Now we estimate the computational complexity of the BP method in terms of the number of the nodes, $\mathcal{N}_a$, that are identified in step $\mathcal{B}$2-2a, the number of the coin-flips, $\mathcal{N}_b$, for the BP process in step $\mathcal{B}$2-2b, and the number of the links, $\mathcal{N}_c$, that are followed in step $\mathcal{B}$2-2c. Let $d(v)$ be the number of out-links from node v (i.e., out-degree of v) and $d'(v)$ the average number of occupied out-links from node v after the BP process. Here we can estimate $d'(v)$ by $\sum_{w \in \Gamma(v;G)} p_{v,w}$. Then, for each time-step $t \in \{1, \cdots, T\}$, we have

$$\mathcal{N}_a = \sum_{v \in A(t-1)} |S(H \cup \{v\}, t-1)|, \quad \mathcal{N}_b = \sum_{w \in B(t-1)} d(w), \quad \mathcal{N}_c = \sum_{v \in A(t-1)} \sum_{w \in S(H \cup \{v\}, t-1)} d'(w) \quad (3)$$

on average.

In order to compare the computational complexity of the BP method to that of the naive method, we consider mapping the naive method onto the BP framework, that is, separating the coin-flip process and the link-following process. We can easily verify that the following algorithm in the BP framework is equivalent to the naive method:

A method that is equivalent to the naive method:
$\mathcal{B}$**1.** Set $\sigma(H \cup \{v\}, t) \leftarrow 0$ for each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$.
$\mathcal{B}$**2.** Repeat the following procedure M times:
$\mathcal{B}$**2-1.** Initialize $S(H \cup \{v\}, 0) = H \cup \{v\}$ for each $v \in V \setminus H$, and set $A(0) \leftarrow V \setminus H$, $A(1) \leftarrow \emptyset, \cdots, A(T) \leftarrow \emptyset$.
$\mathcal{B}$**2-2.** For $t = 1$ to T do the following steps:
$\mathcal{B}$**2-2b'.** For each $v \in A(t-1)$, perform the BP process for the links from $S(H \cup \{v\}, t-1)$ in G^T, and generate the graph $\tilde{G}_t(v)$ constructed by the occupied links.
$\mathcal{B}$**2-2c'.** For each $v \in A(t-1)$, compute $S(H \cup \{v\}; t) = \bigcup_{w \in S(H \cup \{v\}, t-1)} \Gamma(w; \tilde{G}_t(v))$, and set $\sigma(H \cup \{v\}, t) \leftarrow \sigma(H \cup \{v\}, t) + |S(H \cup \{v\}, t)|$ and $A(t) \leftarrow A(t) \cup \{v\}$ if $S(H \cup \{v\}, t) \neq \emptyset$.
$\mathcal{B}$**3.** For each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$, set $\sigma(H \cup \{v\}, t) \leftarrow \sigma(H \cup \{v\}, t)/M$, and output $\sigma(H \cup \{v\}, t)$.

Then, for each $t \in \{1, \cdots, T\}$, the number of coin-flips, $\mathcal{N}_{b'}$, in step $\mathcal{B}$2-2b' is

$$\mathcal{N}_{b'} = \sum_{v \in A(t-1)} \sum_{w \in S(H \cup \{v\}, t-1)} d(w), \tag{4}$$

and the number of the links, $\mathcal{N}_{c'}$, followed in step $\mathcal{B}$2-2c' is equal to $\mathcal{N}_c$ in the BP method on average. From equations (3) and (4), we can see that $\mathcal{N}_{b'}$ is much larger than $\mathcal{N}_{c'} = \mathcal{N}_c$, especially for the case where the diffusion probabilities are small. We can also see that $\mathcal{N}_{b'}$ is generally much larger than each of $\mathcal{N}_a$ and $\mathcal{N}_b$ in the BP method for a real social network. In fact, since such a network generally includes large clique-like subgraphs, there are many nodes $w \in V$ such that $d(w) \gg 1$, and we can expect that $\sum_{v \in A(t-1)} |S(H \cup \{v\}, t-1)| \gg |\bigcup_{v \in A(t-1)} S(H \cup \{v\}, t-1)|$ $(= |B(t-1)|)$. Therefore, the BP method is expected to achieve a large reduction in computational cost.

4.4 Pruning Method

In order to further improve the computational efficiency of the BP method, we introduce a pruning technique and propose a method referred to as the *BP with pruning method*. The key idea of the pruning technique is to utilize the following property: Once we have $S(H \cup \{u\}, t_0) = S(H \cup \{v\}, t_0)$ at some time-step t_0 on the course of the BP process for a pair of information source nodes, u and v, then we have $S(H \cup \{u\}, t) = S(H \cup \{v\}, t)$ for all $t > t_0$. The BP with pruning method estimates σ by the following algorithm:

BP with pruning method:
$\mathcal{B}$**1.** Set $\sigma(H \cup \{v\}, t) \leftarrow 0$ for each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$.
$\mathcal{B}$**2.** Repeat the following procedure M times:
$\mathcal{B}$**2-1".** Initialize $S(H \cup \{v\}; 0) = H \cup \{v\}$ for each $v \in V \setminus H$, and set $A(0) \leftarrow V \setminus H$, $A(1) \leftarrow \emptyset, \cdots, A(T) \leftarrow \emptyset$, and $C(v) \leftarrow \{v\}$ for each $v \in V \setminus H$.
$\mathcal{B}$**2-2.** For $t = 1$ to T do the following steps:
$\mathcal{B}$**2-2a.** Compute $B(t-1) = \bigcup_{v \in A(t-1)} S(H \cup \{v\}, t-1)$.
$\mathcal{B}$**2-2b.** Perform the BP process for the links from $B(t-1)$ in G^T, and generate the graph $\tilde{G}_t$ constructed by the occupied links.
$\mathcal{B}$**2-2c".** For each $v \in A(t-1)$, compute $S(H \cup \{v\}, t) = \bigcup_{w \in S(H \cup \{v\}, t-1)} \Gamma(w; \tilde{G}_t)$, set $A(t) \leftarrow A(t) \cup \{v\}$ if $S(H \cup \{v\}, t) \neq \emptyset$, and set $\sigma(H \cup \{u\}, t) \leftarrow \sigma(H \cup \{u\}, t) + |S(H \cup \{v\}, t)|$ for each $u \in C(v)$.

$\mathcal{B}$**2-2d.** Check whether $S(H \cup \{u\}, t) = S(H \cup \{v\}, t)$ for $u, v \in A(t)$, and set $C(v) \leftarrow C(v) \cup C(u)$ and $A(t) \leftarrow A(t) \setminus \{u\}$ if $S(H \cup \{u\}, t) = S(H \cup \{v\}, t)$.

$\mathcal{B}$**3.** For each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$, set $\sigma(H \cup \{v\}, t) \leftarrow \sigma(H \cup \{v\}, t)/M$, and output $\sigma(H \cup \{v\}, t)$.

Basically, by introducing step $\mathcal{B}$2-2d and reducing the size of $A(t)$, the proposed method attempts to improve the computational efficiency in comparison to the original BP method. For the proposed method, it is important to implement efficiently the equivalence check process in step $\mathcal{B}$2-2d. In our implementation, we first classify each $v \in A(t)$ according to the value of $n = |S(H \cup \{v\}, t)|$, and then perform the equivalence check process only for those nodes with the same n value.

4.5 Burnout Method

In order to further improve the computational efficiency of the BP with pruning method, we additionally introduce a burnout technique and propose a method referred to as the *BP with pruning and burnout method*. More specifically, we focus on the fact that maximizing the marginal influence degree $\sigma(H \cup \{v\}, t)$ with respect to $v \in V \setminus H$ is equivalent to maximizing the marginal influence gain $\phi_H(v, t) = \sigma(H \cup \{v\}, t) - \sigma(H, t)$. Here on the course of the BP process for a newly added information source node v, maximizing $\phi_H(v, t)$ reduces to maximizing $|S(H \cup \{v\}, t) \setminus S(H, t)|$ on average. The BP with pruning and burnout method estimates ϕ_H by the following algorithm:

BP with pruning and burnout methods:

$\mathcal{C}$**1.** Set $\phi_H(v, t) \leftarrow 0$ for each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$.

$\mathcal{C}$**2.** Repeat the following procedure M times:

$\mathcal{C}$**2-1.** Initialize $S(H; 0) = H$, and $S(\{v\}; 0) = \{v\}$ for each $v \in V \setminus H$, and set $A(0) \leftarrow V \setminus H$, $A(1) \leftarrow \emptyset, \cdots, A(T) \leftarrow \emptyset$, and $C(v) \leftarrow \{v\}$ for each $v \in V \setminus H$.

$\mathcal{C}$**2-2.** For $t = 1$ to T do the following steps:

$\mathcal{C}$**2-2a.** Compute $B(t-1) = \bigcup_{v \in A(t-1)} S(\{v\}, t-1) \cup S(H, t-1)$.

$\mathcal{C}$**2-2b.** Perform the BP process for the links from $B(t-1)$ in G^T, and generate the graph $\tilde{G}_t$ constructed by the occupied links.

$\mathcal{C}$**2-2c.** Compute $S(H, t) = \bigcup_{w \in S(H, t-1)} \Gamma(w; \tilde{G}_t)$, and for each $v \in A(t-1)$, compute $S(\{v\}, t) = \bigcup_{w \in S(\{v\}, t-1)} \Gamma(w; \tilde{G}_t) \setminus S(H, t)$, set $A(t) \leftarrow A(t) \cup \{v\}$ if $S(\{v\}, t) \neq \emptyset$, and set $\phi_H(\{u\}, t) \leftarrow \phi_H(\{u\}, t) + |S(\{v\}, t)|$ for each $u \in C(v)$.

$\mathcal{C}$**2-2d.** Check whether $S(\{u\}, t) = S(\{v\}, t)$ for $u, v \in A(t)$, and set $C(v) \leftarrow C(v) \cup C(u)$ and $A(t) \leftarrow A(t) \setminus \{u\}$ if $S(\{u\}, t) = S(\{v\}, t)$.

$\mathcal{C}$**3.** For each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$, set $\phi_H(\{v\}, t) \leftarrow \phi_H(\{v\}, t)/M$, and output $\phi_H(\{v\}, t)$.

Intuitively, compared with the BP with pruning method, by using the burnout technique, we can substantially reduce the size of the active node set from $S(H \cup \{v\}, t)$ to $S(\{v\}, t)$ for each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$. Namely, in terms of computational costs described by Equation (3), we can expect to obtain smaller numbers for $\mathcal{N}_a$ and $\mathcal{N}_c$ when $H \neq \emptyset$. However, how effectively the proposed method works will depend on several conditions such as network structure, time span, values of diffusion probabilities, and so on. We will do a simple analysis later and experimentally show that it is indeed effective.

5 Experimental Evaluation

In the experiments, we report our evaluation results on the final-time maximization problem due to the space limitation.

5.1 Network Data and Settings

In our experiments, we employed two datasets of large real networks used in [10], which exhibit many of the key features of social networks.

The first one is a trackback network of Japanese blogs. The network data was collected by tracing the trackbacks from one blog in the site "goo (http://blog.goo.ne.jp/)" in May, 2005. We refer to the network data as the blog network. The blog network was a strongly-connected bidirectional network, where a link created by a trackback was regarded as a bidirectional link since blog authors establish mutual communications by putting trackbacks on each other's blogs. The blog network had $12,047$ nodes and $79,920$ directed links.

The second one is a network of people that was derived from the "list of people" within Japanese Wikipedia. We refer to the network data as the Wikipedia network. The Wikipedia network was also a strongly-connected bidirectional network, and had $9,481$ nodes and $245,044$ directed links.

For the SIS model, we assigned a uniform probability p to the propagation probability $p_{u,v}$ for any link $(u,v) \in E$, that is, $p_{u,v} = p$. According to [4, 2], we set the value of p relatively small. In particular, we set the value of p to a value smaller than $1/\bar{d}$, where $\bar{d}$ is the mean out-degree of a network. Since the values of $\bar{d}$ were about 6.63 and 25.85 for the blog and the Wikipedia networks, respectively, the corresponding values of $1/\bar{d}$ were about 0.15 and 0.03. We decided to set $p = 0.1$ for the blog network and $p = 0.01$ for the Wikipedia network. Also, for the time span T, we set $T = 30$.

For the bond percolation method, we need to specify the number M of performing the bond percolation process. According to [12], we set $M = 10,000$ for estimating influence degrees for the blog and Wikipedia networks.

All our experimentation was undertaken on a single PC with an Intel Dual Core Xeon X5272 3.4GHz processor, with 32GB of memory, running under Linux.

5.2 Comparison Methods

First, we compared the proposed method with three heuristics from social network analysis with respect to the solution quality. They are based on the notions of "degree centrality", "closeness centrality", and "betweenness centrality" that are commonly used as influence measure in sociology [13]. Here, the betweenness of node v is defined as the total number of shortest paths between pairs of nodes that pass through v, the closeness of node v is defined as the reciprocal of the average distance between v and other nodes in the network, and the degree of node v is defined as the number of links attached to v. Namely, we employed the methods of choosing nodes in decreasing order of these centralities. We refer to these methods as the *betweenness method*, the *closeness method*, and the *degree method*, respectively.

Next, to evaluate the effectiveness of the pruning and the burnout strategies, we compared the proposed method with the naive greedy method based on the BP method with respect to the processing time. Hereafter, we refer to the naive greedy method based on the BP method as the BP method for short.

5.3 Solution Quality Comparison

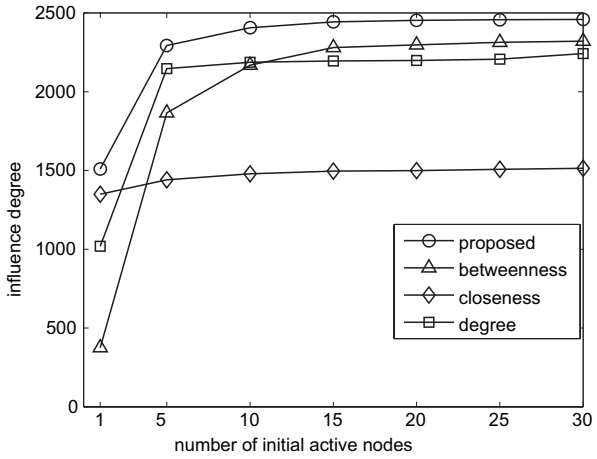

Fig. 2. Comparison of solution quality for the blog network.

We first compared the quality of the solution H_K of the proposed method with that of the betweenness, the closeness, and the degree methods for solving the problem of the influence maximization at the final time step T. Clearly, the quality of H_K can be evaluated by the influence degree $\sigma(H_K, T)$. We estimated the value of $\sigma(H_K, T)$ by using the bond percolation method with $M = 10,000$ according to [12].

Figures 2 and 3 show the influence degree $\sigma(H_K, T)$ as a function of the number of initial active nodes K for the blog and the Wikipedia networks, respectively. In the figures, the circles, triangles, diamonds, and squares indicate the results for the proposed, the betweenness, the closeness, and the degree methods, respectively. The proposed method performs the best for both networks, while the betweenness method follows for the blog dataset and the degree method follows for the Wikipedeia dataset. Note that how each of the conventional heuristics performs depends on the characteristics of the network structure. These results imply that the proposed method works effectively, and outperforms the conventional heuristics from social network analysis.

It is interesting to note that the k nodes ($k = 1, 2, ..., K$) that are discovered to be most influential by the proposed method are substantially different from those that are found by the conventional centrality-based heuristic methods. For example, the best node ($k = 1$) chosen by the proposed method for the blog dataset is ranked 118 for the

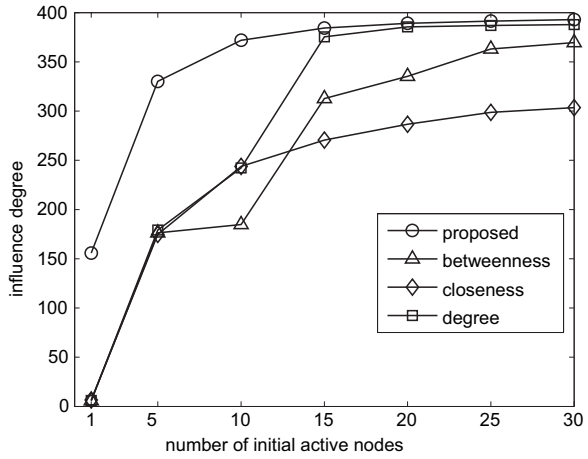

Fig. 3. Comparison of solution quality for the Wikipedia network.

betweenness method, 659 for the closeness method and 6 for the degree method, and the 15th node ($k = 15$) by the proposed method is ranked 1373, 8848 and 507 for the corresponding conventional methods, respectively. The best node ($k = 1$) chosen by the proposed method for the Wikipedia dataset is ranked 580 for the betweenness method, 2766 for the closeness method and 15 for the degree method, and the 15th node ($k = 15$) by the proposed method is ranked 265, 2041, and 21 for the corresponding conventional methods, respectively. It is hard to find a correlation between these rankings, but for the smaller k, it appears that degree centrality measure is better than the other centrality measures, which can be inferred from Figures 2 and 3.

5.4 Processing Time Comparison

Next, we compared the processing time of the proposed method (BP with pruning and burnout method) with that of the BP method. Let $\tau(K, T)$ denote the processing time of a method for solving the problem of the influece maximization at the final time step T, where K is the number of initial active nodes. Figures 4 and 5 show the processing time difference $\Delta\tau(K, T) = \tau(K, T) - \tau(K - 1, T)$ as a function of the number of initial active nodes K for the blog and the Wikipedia networks, respectively. In these figures, the circles, and crosses indicate the results for the proposed and the BP methods, respectively. Note that $\Delta\tau(K, T)$ decreases as K increases for the proposed method, whereas $\Delta\tau(K, T)$ increases for the BP method. This means that the difference in the total processing time becomes increasingly larger as K increases. In case of the blog dataset, the total processing time for $K = 5$ is about 2 hours for the proposed method and 100 hours for the BP methods. Namely, the proposed method is about 50 times faster than the BP method for $K = 5$. The same is true for the Wikipedia dataset. The total processing time for $K = 5$ is about 0.5 hours for the proposed method and 9 hours the BP methods, and the proposed method is about 18 times faster than the BP method for $K = 5$. These results

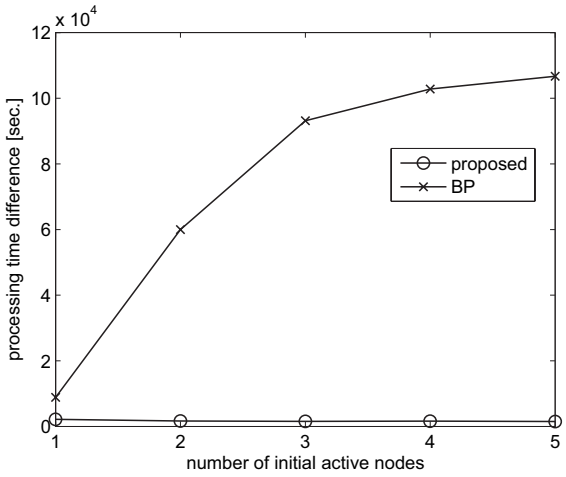

Fig. 4. Comparison of processing time for the blog network.

confirm that the proposed method is much more efficient than the BP method, and can be practical.

6 Discussion

The influence function $\sigma(\cdot, T)$ is submodular [4]. For solving a combinatorial optimization problem of a submodular function f on V by the greedy algorithm, Leskovec et al. [7] have recently presented a lazy evaluation method that leads to far fewer (expensive) evaluations of the marginal increments $f(H \cup \{v\}) - f(H)$, $(v \in V \setminus H)$ in the greedy algorithm for $H \neq \emptyset$, 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 SIS model, where the influence function $\sigma(\cdot, T)$ is evaluated by simulating the corresponding random process. It is clear that 1) this method is more efficient than the naive greedy method that does not employ the BP method and instead evaluates the influence degrees by simulating the diffusion phenomena, and 2) further the both methods become the same for $K = 1$ and empirically estimate the influence function $\sigma(\cdot, T)$ by probabilistic simulations. These methods also require M to be specified in advance as a parameter, where M is the number of simulations. Note that the BP and the simulation methods can estimate influence degree $\sigma(v, t)$ with the same accuracy by using the same value of M (see [12]). Moreover, as shown in [12], estimating influence function $\sigma(\cdot, 30)$ by $10,000$ simulations needed more than 35.8 hours for the blog dataset and 7.6 hours for the Wikipedia dataset, respectively. However, the proposed method for $K = 30$ needed less than 7.0 hours for the blog dataset and 3.2 hours for the Wikipedia dataset, respectively. Therefore, it is clear that the proposed method can be faster than the method by Leskovec [7] for the influence maximization problem for the SIS model.

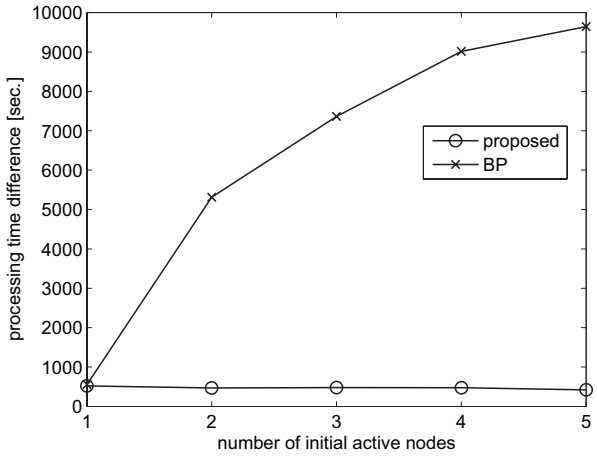

Fig. 5. Comparison of processing time for the Wikipedia network.

7 Conclusion

Finding influential nodes is one of the most central problems in the field of social network analysis. There are several models that simulate how various things, e.g., news, rumors, diseases, innovation, ideas, etc. diffuse across the network. One such realistic model is the *susceptible/infected/susceptible (SIS) model*, an information diffusion model where nodes are allowed to be activated multiple times. The computational complexity drastically increases because of this multiple activation property, e.g., compared with the *susceptible/infected/recovered (SIR) model* where once activated nodes can never be deactivated/reactivated. We addressed the problem of efficiently discovering the influential nodes under the SIS model, i.e., estimating the expected number of activated nodes at time-step t for $t = 1, \cdots, T$ starting from an initially activated node set $H \in V$ at time-step $t = 0$. We solved this problem by constructing a layered graph from the original social network by adding each layer on top of the existing layers as the time proceeds, and applying the bond percolation with a pruning strategy. We showed that the computational complexity of the proposed method is much smaller than the conventional naive probabilistic simulation method by a theoretical analysis. We applied the proposed method to two different types of influence maximization problem, i.e. discovering the K most influential nodes that together maximize the expected influence degree at the time of interest or the expected influence degree over the time span of interest. Both problems are solved by the greedy algorithm taking advantage of the submodularity of the objective function. We confirmed by applying to two real world networks taken from blog and Wikipedia data that the proposed method can achieve considerable reduction of computation time without degrading the accuracy compared with the naive simulation method, and discover nodes that are more influential than the nodes identified by the conventional methods based on the various centrality measures. Just as a key task on biology is to find some important groups of genes or proteins by performing

biologically plausible simulations over regulatory networks or metabolic pathways, our proposed method can be a core technique for the discovery of influential persons over real social networks, which can contribute to a progress on social science.

Acknowledgments

This work was partly supported by 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, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

1. Adar, E., Adamic, L.A.: Tracking information epidemics in blogspace. In: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05). (2005) 207–214
2. Leskovec, J., McGlohon, M., Faloutsos, C., Glance, N., Hurst, M.: Patterns of cascading behavior in large blog graphs. In: Proceedings of the 2007 SIAM International Conference on Data Mining (SDM'07). (2007) 551–556
3. Agarwal, N., Liu, H.: Blogosphere: Research issues, tools, and applications. SIGKDD Explorations **10** (2008) 18–31
4. 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-2003). (2003) 137–146
5. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. In: Proceedings of the 13th International World Wide Web Conference (WWW 2004). (2004) 107–117
6. 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). (2007) 1371–1376
7. 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 2007). (2007) 420–429
8. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001). (2001) 57–66
9. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002). (2002) 61–70
10. Kimura, M., Saito, K., Motoda, H.: Blocking links to minimize contamination spread in a social network. ACM Transactions on Knowledge Discovery from Data **3** (2009) 9:1–9:23
11. Newman, M.E.J.: The structure and function of complex networks. SIAM Review **45** (2003) 167–256
12. Kimura, M., Saito, K., Motoda, H.: Efficient estimation of influence functions for sis model on social networks. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09). (2009)
13. Wasserman, S., Faust, K.: Social network analysis. Cambridge University Press, Cambridge, UK (1994)