Under consideration for publication in Knowledge and Information Systems

Efficient Discovery of Influential Nodes for SIS Models in Social Networks

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

[1]School of Administration and Informatics, University of Shizuoka, Shizuoka 422-8526, Japan;
[2]Department of Electronics and Informatics, Ryukoku University, Otsu 520-2194, Japan;
[3]Department of Integrated Information Technology, Aoyama Gakuin University, Kanagawa 229-8558, Japan;
[4]Institute of Scientific and Industrial Research, Osaka University, Osaka 567-0047, Japan

Abstract. We address the problem of discovering the influential nodes in a social network under the *susceptible/infected/susceptible (SIS) model* which allows multiple activation of the same node, by defining two influence maximization problems:*final-time* and *integral-time*. We solve this problem by constructing a layered graph from the original network with each layer added on top as the time proceeds and applying the bond percolation with two effective control strategies: pruning and burnout. We experimentally demonstrate that the proposed method gives much better solutions than the conventional methods that are based solely on the notion of centrality using two real-world networks. The pruning is most effective when searching for a single influential node, but burnout is more powerful in searching for multiple nodes which together are influential. We further show that the computational complexity is much smaller than the naive probabilistic simulation both by theory and experiment. The influential nodes discovered are substantially different from those identified by the centrality measures. We further note that the solutions of the two optimization problems are also substantially different, indicating the importance of distinguishing these two problem characteristics and using the right objective function that best suits the task in hand.

Keywords: Information diffusion; SIS model; Influence maximization; Pruning method; Burnout method

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.

Received May 22, 2010
Revised Mar 01, 2011
Accepted Mar 20, 2011

Therefore, substantial attention has recently been directed to investigating information diffusion phenomena in social networks (Newman, 2001; Adar and Adamic, 2005; Domingos, 2005; McCallum et al, 2005; Leskovec et al, 2007b; Watts and Dodds, 2007; Agarwal and Liu, 2008), and other aspects such as analyses of social networking sites (Mislove et al, 2007; Muhlestein and Lim, 2009), topic evolution (Zhou et al, 2006; Peng and Li, 2010), and privacy issues (Backstrom et al, 2007; Zhou and Pei, 2010).

Finding influential nodes is one of the central problems in social network analysis[1]. Thus, developing efficient and practical methods of doing 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* (Goldenberg et al, 2001; Kempe et al, 2003; Gruhl et al, 2004) and the *linear threshold (LT) model* (Watts, 2002; Kempe et al, 2003). Researchers investigated the problem of finding a limited number of influential nodes that are effective for the spread of information under the above models (Kempe et al, 2003; Kimura et al, 2007; Kimura et al, 2010). This combinatorial optimization problem is called the *influence maximization problem*. Kempe et al (2003) 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 (Kimura et al, 2007) and submodularity (Leskovec et al, 2007a) were proposed for efficiently estimating the greedy solution. Succeeding work further improved the efficiency by approximating the solution using a heuristic (Chen et al, 2009). The influence maximization problem has applications in sociology and "viral marketing" (Agarwal and Liu, 2008), and was also investigated in a different setting (a descriptive probabilistic model of interaction) (Domingos and Richardson, 2001; Richardson and Domingos, 2002). The problem has recently been extended to influence control problems such as a contamination minimization problem (Kimura et al, 2009a).

The IC model can be identified with the so-called *susceptible/infected/recovered (SIR) model* for the spread of a disease (Newman, 2003; Gruhl et al, 2004). In the SIR model, only infected individuals can infect susceptible individuals, while recovered individuals can neither infect others nor be infected by others. 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 are many phenomena for which this property does not hold. A typical example would be 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 (activated)[2]). 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. Note here that we regard the act of "posting" to be the state change from "susceptible" to "infected". The blogger can read the next blog and respond to it anytime after the completion of the previous posting. Most simply, this phenomenon can be modeled by a *susceptible/infected/susceptible (SIS) model* from the epidemiology. Other examples include the growth of hyper-link posts among bloggers (Leskovec et al, 2007b), the spread of computer viruses without permanent virus-checking programs, and epidemic disease such as tuberculosis and gonorrhea (Newman, 2003). There are

[1] "Influence" means many things and there are many factors which make a node influential. In this paper, as we describe later in this section and define more formally in subsection 2.2, influence of a node simply means the expected number of activated nodes as a result of information diffusion that starts from the node.

[2] We use "infected" and "activated" interchangeably.

many more examples of information diffusion phenomena for which the SIS model is more appropriate.

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. When we want to find an influential node, we need to know $\sigma(\{v\}, t)$, ($v \in V$, $t = 1, \ldots, T$), but when we want to solve influence maximization problem, we need to know $\sigma(H, t)$, ($H \subseteq V$, $t = 1, \ldots, T$). It is vital, first of all, to have an effective method for estimating $\sigma(\{v\}, t)$. Clearly, in order to extract influential nodes, we must estimate the value of $\sigma(\{v\}, t)$ for every node v and every time-step t. Solving influence maximization problem is much more difficult because we have to find the optimal subset of nods H^*_K with a fixed cardinality K. Here it is vital to have an effective method for 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. We have reported our preliminary work on efficiently estimating $\{\sigma(\{v\}, t); v \in V, t = 1, \ldots, T\}$ for the SIS model based on the bond percolation with a pruning strategy (Kimura et al, 2009b), and extended it to influential maximization problem in which we introduced a new technique called burnout to efficiently estimate $\{\sigma(H \cup \{v\}, T) - \sigma(H, T); v \in V \setminus H\}$ (Saito et al, 2009).

In this paper, we describe these two techniques in details and conduct extensive experiments to evaluate how these two affect the efficiency of solving the influence maximization problems on a network $G = (V, E)$ under the SIS model. Needless to say, we can naively estimate the marginal influence gains for any non-empty subset H of V by simulating the SIS model. However, this naive simulation method is overly inefficient and not practical at all. Here, we define two influence maximization problems: the *final-time maximization problem* and the *integral-time maximization problem*. The latter problem does not make sense for the SIR model and is only meaningful for the SIS model. We adopt the greedy algorithm, to reduce the computational complexity, for approximately solving the problems according to the work of Kempe et al (2003) which was conducted for the IC and the LT models, ensuring that submodularity holds in the SIS model setting, too. We show theoretically that the proposed method is expected to achieve a large reduction in computational cost by comparing computational complexity with the naive probabilistic simulation method. Further, using two large real networks, we experimentally demonstrate that the proposed method is much more efficient than the naive greedy method that uses only the bond percolation without employing both the pruning and the burnout. We show that the pruning is effective when searching for a single influential node, but the burnout is more powerful and eventually takes over the pruning as we increase the number of nodes to search. Thus, it is advisable to use both the pruning and the burnout only in the initial few iterations and stop using the pruning and use the burnout alone in the succeeding iterations in the greedy algorithm. The computational cost reduces by 2 orders of magnitudes comparing the naive bond percolation which itself is 2 to 3 orders of magnitudes more efficient than the naive simulation. We also show that the nodes discovered by the proposed method are substantially different from the nodes discovered by the conventional methods that are based on the notion of various centrality measures which does not consider the information diffusion phenomena and can be evaluated from the network topology alone. The proposed method results in a substantial increase in the expected influence. We further find that the two optimization problems give also substantially different solutions and it is important to use the right objective function which reflects the problem characterization.

The paper is organized as follows. We define the information diffusion model in sec-

tion 2 and the two influential maximization problems we want to solve in section 3. We then give details of the algorithms to solve this problem (greedy algorithm, bond percolation, pruning, burnout and their combinations) in section 4. The experimental results are given in section 5 (network data, quality of the solutions and computation time for both influence function estimation and influence maximization estimation), followed by some discussions in section 6. We end this paper by summarizing the conclusion in section 7.

2. Information Diffusion Model

Let $G = (V, E)$ be a directed network, where V and E stand for the sets of all the nodes and (directed) links, respectively. Here, note that E is a subset of $V \times V$. 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 has contact with 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 *diffusion probability* through link (u, v). Given an initial set of active nodes H 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 at time-step t, then their activation attempts are sequenced in an arbitrary order. On the other hand, node u becomes or remains inactive at time-step $t + 1$ unless it is activated from other active node at time-step t. The process terminates if the current time-step reaches the time limit T.

2.2. Influence Function

For the SIS model on G, we consider an information diffusion 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 assume that we can encode such factors as diffusion

probabilities of each node[3]. As emphasized in section 1, 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 *integral-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 (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 for the SIS model, and describe three techniques (the bond percolation method, the pruning method, and the burnout method) for efficiently solving the influence maximization problem under the greedy algorithm. We also discuss the computational complexity of these methods and show the merit of the pruning and the burnout.

[3] Such factors as rank, prestige and power exert influence in a cumulative way, i.e. richer gets richer phenomena. We need some reinforcement mechanism outside the SIS model to deal with such feedback which is beyond the scope of our framework.

4.1. Greedy Algorithm

We approximately solve the influence maximization problem by the greedy algorithm. Below we describe this algorithm first for the final-time maximization problem and then for the integral-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.

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

Greedy algorithm for the integral-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

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

where H_k^* is the exact solution defined by Equation (1) or (2), that is, the expected influence of the greedy solution is lower bounded and it is guaranteed that it is at worst 63% of the optimal expected influence (Kempe et al, 2003).

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 above algorithms. 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.

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 diffusion probabilities $\{p_e; e \in E\}$ on G over time span T is equivalent to the *bond percolation process (BP) with occu-*

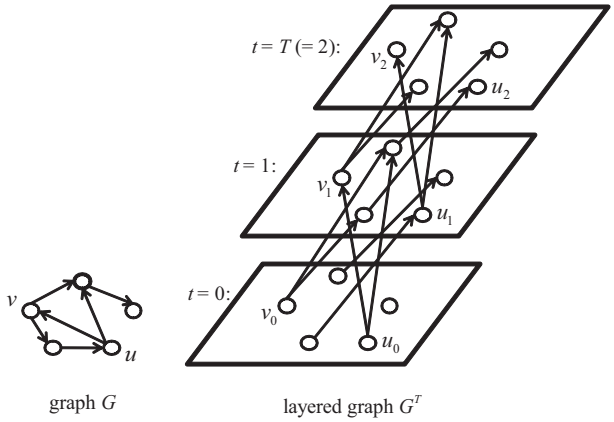


Fig. 1. An example of a layered graph.

pation probabilities $\{q_e; e \in E^T\}$ on G^T.[4] 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, $S(H \cup \{v\}, t)$.

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 start from the non-activated nodes 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:

[4] The SIS model over time span T on G can be exactly mapped onto the IC model on G^T (Kempe et al, 2003). Thus, the result follows from the equivalence of the BP process and the IC model (Grassberger, 1983; Newman, 2002; Kempe et al, 2003; Kimura et al, 2007).

$\mathcal{B}$**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 the 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:

Naive method expressed in the framework of 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:

$\mathcal{B}$**2-2b'.** 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

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

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 the 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 over 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 scan each $v \in A(t)$ according to the value of $n = |S(H \cup \{v\}, t)|$, and identify 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 introduce another technique called burnout and propose a method which is referred to as the *BP with pruning and burnout method* [5]. More specifically, we focus on the fact that maximizing the marginal influence degree $\sigma(H \cup \{v\}, t)$ with respect to $v \in$

[5] Here we integrated these two techniques, but it is also possible to combine the BP method with only the burnout method. We skipped this one because it is self-evident.

$V \setminus H$ is equivalent to maximizing the marginal influence gain $\phi_H(v,t) = \sigma(H \cup \{v\}, t) - \sigma(H,t)$. Here in terms 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 the average. The BP with pruning and burnout method estimates ϕ_H by the following algorithm:

BP with pruning and burnout methods:

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

*C***2.** Repeat the following procedure M times:

*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 $v \in V \setminus H$.

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

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

*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.

*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 $u \in C(v)$.

*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 $S(\{u\},t) = S(\{v\},t)$.

*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, 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 $t \in \{1, \cdots, T\}$ compared with the BP with pruning method. 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, etc. We will do a simple analysis later and experimentally show that it is indeed effective.

5. Experimental Evaluation

We have carried out extensive experiments and evaluated the effects of the two techniques that were implemented on top of the bond percolation on the quality of the solution and the computation time, using two real world social networks. The baseline to compare the quality of the solution is the naive simulation method which is confirmed to be prohibitively inefficient.

5.1. Network Data and Basic Settings

In our experiments, we employed two datasets of large real networks used in Kimura et al (2009a), which exhibit many of the key features of social networks (Newman and Park, 2003).

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. 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 refer to the network data as the Wikipedia network. Thus, the Wikipedia network was also a strongly-connected bidirectional network, and had $9,481$ nodes and $245,044$ directed links.

We assigned a uniform value p to the diffusion probability $p_{u,v}$ for any link $(u,v) \in E$, that is, $p_{u,v} = p$ for the SIS model we used. According to Kempe et al (2003) and Leskovec et al (2007b), we set the value of p relatively small. In particular, we set the value of p to a value smaller than $1/d$, where d is the mean out-degree of a network. Since the values of d were about 6.63 and 25.85 for the blog and the Wikipedia networks, respectively, the corresponding values of $1/d$ were about 0.15 and 0.039. In view of these values we decided to set $p = 0.1$ for the blog network and $p = 0.03$ for the Wikipedia network. Time span T can be arbitrarily set but it is constrained by the inefficiency of the naive simulation method. We found $T = 30$ is good enough to evaluate the performance of our method. We also need to specify the number M of performing the bond percolation process. The larger, the better, but we have to compromise between the solution quality and the computational cost. We set $M = 10,000$ for estimating influence degrees for the blog and Wikipedia networks (See 5.2.1).

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

5.2. Performance for Influence Function Estimation

5.2.1. Accuracy of Estimated Influence Function

We first investigated how accurately the proposed method can estimate the value of influence function in terms of node ranking. Since, in this case, the information diffusion starts with every single node $v \in V$ independently with all the other nodes remaining inactive, i.e. $H = \emptyset$, there is no room for burnout to come in. Thus, we compared the BP with pruning method (BPP for short) with the naive method (naive for short) which we consider as the baseline. Both methods require M to be specified in advance as a parameter. If M is set at ∞, both BPP and naive should give the correct expected influence degree. For a finite value of M, the results may seem different. In fact, as shown in section 4.3, the number of coin flips is different in these two methods and it is much larger in the naive method. However, this does not mean that there is more randomness introduced in the naive method and thus the convergence of the naive method is faster. In fact for each single (initially activated) node v from which to propagate the information, the number of independent coin-flips is effectively the same for both the methods. Thus by using the same value of M, both would estimate $\sigma(v,t)$ with the same accuracy in principle.

We have first experimentally confirmed that use of $M = 100,000$ gives a very stable identical converged solution for both methods for a few selected initial nodes, but the naive method took an order of week to return the result and thus is not practical to perform the comparative study. Then we found that further reducing the value to $M = 10,000$ still gives reliable results, i.e., in effect the same ranking and value of $\sigma(v,t)$, for $t = 1, \cdots, 20$ for the high ranked nodes. The following results were obtained by using

Table 1. Results for the top 10 nodes v and the values of $\sigma(v, 20)$ based on the proposed method (BPP) for the blog network. Left: The result of the first experiment. Right: The result of the second experiment.

Rank	v	$\sigma(v, 20)$	Rank	v	$\sigma(v, 20)$
1	2210	984.74	1	2210	984.87
2	2248	980.41	2	2248	979.46
3	3906	956.97	3	3906	955.84
4	3907	953.04	4	3907	952.71
5	146	929.96	5	146	929.30
6	155	928.77	6	155	928.49
7	3233	912.61	7	3233	911.01
8	3228	912.18	8	3228	910.49
9	140	909.22	9	140	910.31
10	2247	909.12	10	2247	909.59

Table 2. Results for the top 10 nodes v and the values of $\sigma(v, 20)$ based on the naive method for the blog network. Left: The result of the first experiment. Right: The result of the second experiment.

Rank	v	$\sigma(v, 20)$	Rank	v	$\sigma(v, 20)$
1	2210	984.38	1	2210	985.74
2	2248	979.59	2	2248	980.72
3	3906	956.82	3	3906	956.57
4	3907	953.14	4	3907	953.89
5	146	931.03	5	146	931.62
6	155	929.68	6	155	930.21
7	3233	913.50	7	3233	911.89
8	3228	912.27	8	3228	910.52
9	140	910.04	9	140	910.37
10	2247	909.59	10	2247	909.59

$M = 10,000$. Tables 1 and 2 show the ranking of the initially activated influential nodes v evaluated at time-step $T = 20$ for the blog network. We had to limit T to 20 because of the prohibitive computation cost for the naive simulation. The value of influence function $\sigma(v, 20)$ is sorted in the decreasing order and the top 10 nodes are listed. We repeated the experiment twice for each method (BPP and naive) and the results for both are shown side by side. We note that the ranking is exactly the same for the two runs and this is also true between the two methods. We further note that the values of corresponding influence degrees are very similar. The influence degree varies slowly and it decreases only by less than 10% in going from the top to the 10th. Tables 3 and 4 are the results for the Wikipedia network. The results are slightly less stable than for the

Table 3. Results for the top 10 nodes v and the values of $\sigma(v, 20)$ based on the proposed method (BPP) for the Wikipedia network. Left: The result of the first experiment. Right: The result of the second experiment.

Rank	v	$\sigma(v, 20)$	Rank	v	$\sigma(v, 20)$
1	790	2121.52	1	790	2120.45
2	279	2120.52	2	279	2119.32
3	8340	2119.33	3	8340	2118.42
4	323	2118.86	4	323	2117.81
5	326	2117.98	5	326	2117.15
6	772	2117.06	6	772	2116.66
7	325	2116.12	7	325	2114.85
8	2441	2113.09	8	4924	2112.72
9	2465	2112.52	9	1407	2112.44
10	1407	2112.19	10	2498	2111.35

Table 4. Results for the top 10 nodes v and the values of $\sigma(v, 20)$ based on the naive method for the Wikipedia network. Left: The result of the first experiment. Right: The result of the second experiment.

Rank	v	$\sigma(v, 20)$	Rank	v	$\sigma(v, 20)$
1	790	2122.14	1	790	2120.84
2	279	2119.62	2	323	2118.81
3	8340	2119.10	3	279	2118.76
4	323	2117.97	4	8340	2118.52
5	326	2117.84	5	326	2117.75
6	772	2116.37	6	772	2117.32
7	325	2115.84	7	325	2116.39
8	1407	2113.85	8	1407	2114.42
9	4294	2112.79	9	2465	2114.34
10	3149	2112.57	10	4924	2113.55

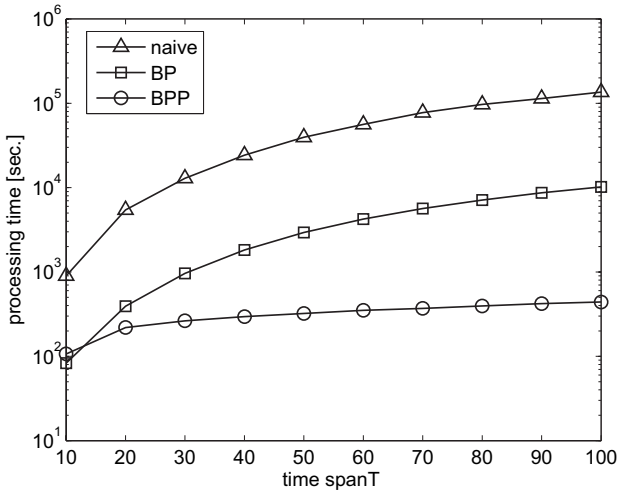


Fig. 2. Results for the blog network.

blog network. However, the rankings of top 7 are the same for the two runs of BPP and the first run of the naive. We note that the values of the influence degrees change much more slowly and the value only reduces by less than 0.5% in going from the top to the 10th. The Wikipedia network is much more difficult in terms of correctly identifying the ranking. From the overall experimental results, we confirm that for the same and large enough values of M, the proposed method (BPP) gives the same results as the naive method.

We have not evaluated the integral influence function over the time span T:$\sum_{t=1}^{T} \sigma(v, t)$ because if it is confirmed that each component $\sigma(v, t)$ can be well approximated, its sum is equally well approximated.

5.2.2. Computational Cost for Influence Function Estimation

Next, we compared the processing time of the proposed method (BPP) with the BP method without pruning (BP for short) and the naive method. Here, we used $M = 1,000$ in order to keep the computational time for the naive method at a reasonable level so that it runs for a larger T. Figures 2 and 3 show the processing time to estimate $\{\sigma(v, t); v \in$

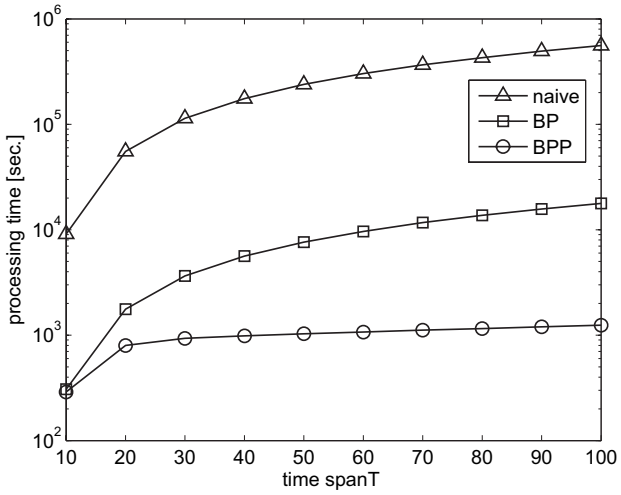


Fig. 3. Results for the Wikipedia network.

$V, t = 0, 1, \cdots, T\}$ as a function of the time span T for the blog and the Wikipedia networks, respectively. In these figures, the circles, squares and triangles indicate the results for BPP, BP and naive, respectively. Note that in case of the blog network, the processing time for the time span $T = 100$ is about 7 minutes, 2.8 hours, and 1.5 days for BPP, BP, and naive, respectively. Namely, BPP is about 25 and 310 times faster than BP and naive, respectively. Note also that in case of the Wikipedia network, the processing time for the time span $T = 100$ is about 21 minutes, 5 hours, and 155 hours for BPP, BP and naive, respectively. Namely, BPP is about 14 and 440 times faster than BP and naive, respectively.

The reduction of the processing time due to the pruning is large. The processing time is about 20 times less when evaluated for $T = 100$. However, when T is small the pruning adversely affects the processing time because of the computational overhead. The two BP methods (with and without pruning) are much faster than the naive method. The performance difference between BPP and each of BP and naive increases as time-step (or time span) increases. Moreover, the same performance difference becomes larger for the blog network than the Wikipedia network. The following simple analysis explains this. Consider the extreme case where $S(u,t) = S(v,t)$ for $\forall u, v \in A(t)$ and $d(w) = d$ for $\forall w \in S(v,t)$ ($v \in A(t)$) at some time-step t. We denote $|A(t)| = a$ and $|S(v,t)| = s$. Then, we have $\mathcal{N}_a = as$, $\mathcal{N}_b = sd$, $\mathcal{N}_{b'} = asd$ and $\mathcal{N}_c = asd'$ on the average for time-step $t+1$. Recall that d' is the expected number of the occupied links, which is calculated as pd, where p is the common diffusion probability for all links. Further assume that the pruning was ideal such that $\tilde{\mathcal{N}}_a = s$ and $\tilde{\mathcal{N}}_c = sd'$, which respectively denote the number of nodes identified in step 2-2a and the average number of links followed in step 2-2c" for BPP. Then, if $ad' > d$, i.e., $ad'/d = ap > 1$ holds, the improvement ratios of BPP over BP and naive are respectively $asd'/sd = ap$ and $asd/sd = a$. From our experimental results, we can estimate a as 310 for the blog network and 440 for the Wikipedia network. Then we obtain ap as 31 and 13 respectively, which approximates the actual ratio ($\text{Proc\_time}_{BP}/\text{Proc\_time}_{BPP}$), 25 and 14. The similar discussion applies to the processing time for the integrated influence function over the time span T.

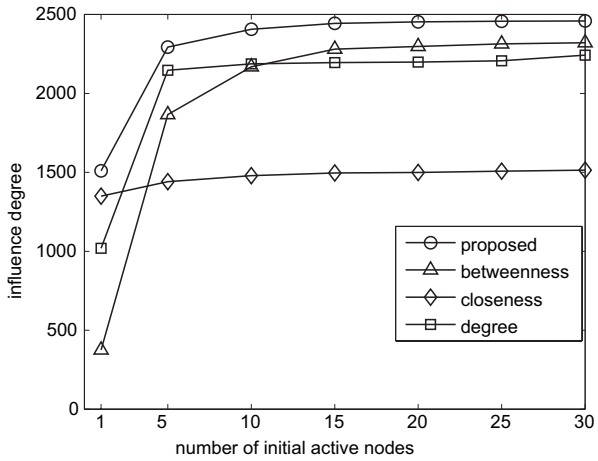


Fig. 4. Comparison of solution quality for the blog network (final-time maximization problem).

5.3. Performance of Influence Maximization Problem

5.3.1. Comparison of Accuracy of the Proposed Methods with Centrality Measures

We compared the quality of the solution of the proposed method, i.e. the BP with pruning and burnout method (BPPB for short) with the three well known centrality measures: "degree centrality", "closeness centrality", and "betweenness centrality" that are commonly used as the influence measure in sociology (Wasserman and Faust, 1994). 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*. We evaluated the value of these measures for each node and ranked the nodes in decreasing order, and calculated the influence degree (both the final-time value and the integral-time value) using the top K nodes with $K = 1, 2, ..., 30$. We refer to these methods as the *betweenness method*, the *closeness method*, and the *degree method*, respectively.

The solution H_K of the proposed method is calculated by the bond percolation algorithm described in 4.5 using both pruning and burnout. Clearly, the quality of H_K can be evaluated by the influence degree $\sigma(H_K, T)$ for the final-time maximization problem and the influence degree $\sum_{t=1}^{T} \sigma(H_K, t)$ for the integral-time maximization problem. We estimated the values of $\sigma(H_K, T)$ and $\sum_{t=1}^{T} \sigma(H_K, t)$ with $M = 10,000$ and $T = 30$. Figures 4 and 5 show the influence degree $\sigma(H_K, T)$ (solution of the final-time maximization problem) as a function of the number of initial active nodes K for the blog and the Wikipedia networks, respectively. In the same way, Figures 6 and 7 show the influence degree $\sum_{t=1}^{T} \sigma(H_K, t)$ (solution of the integral-time maximization problem) 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 (BPPB), the betweenness, the closeness, and the degree methods, respectively. Evidently, the proposed method performs the best for both networks and for both maximization problems. The shapes of the curves are different for

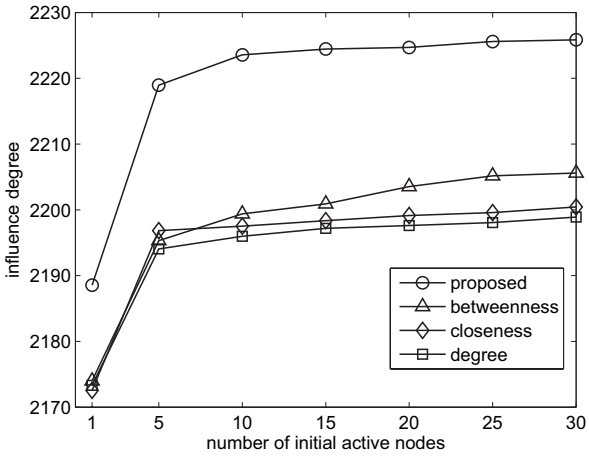


Fig. 5. Comparison of solution quality for the Wikipedia network (final-time maximization problem).

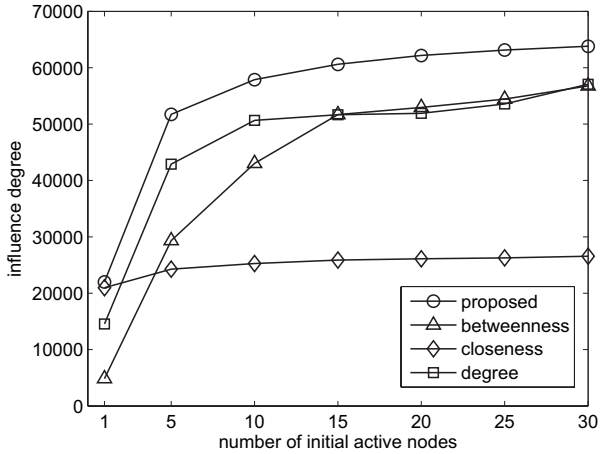


Fig. 6. Comparison of solution quality for the blog network (integral-time maximization problem).

the two problems. In the final-time maximization problem, only the first top 5 to 10 nodes are influential and the succeeding nodes do not contribute to increasing the influence degree. As a rule of thumb, this is true for all the four methods. In the integral-time maximization problem, nodes after the top 10 are also influential and contribute to increasing the influence degree. This is also true for all the four methods as a rule of thumb. There is no clear indication as to which centrality measures rank higher for a wide range of nodes. For example, betweenness measure appears to be the next best for the both networks in case of the final-time maximization problem, but degree measure is also good for the both networks (slightly better for the blog and slightly worse for the Wikipedia network) in case of the integral-time maximization problem. If we focus only the first 10 nodes, degree method appears to be the best among the three conventional

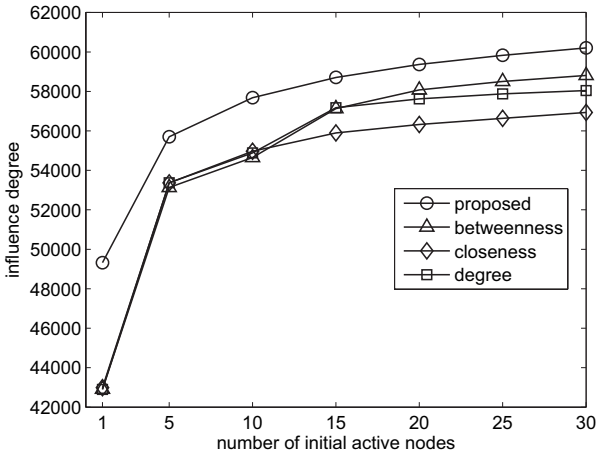


Fig. 7. Comparison of solution quality for the Wikipedia network (integral-time maximization problem).

methods. How well or badly each of the conventional heuristics performs depends on the characteristics of the network structure and the type of the maximization problem. Note that there are substantial differences in the amount of the influence degree (value of the objective function). These results clearly indicate that it is indeed important to obtain the optimal solution. The proposed method can be effectively used for this purpose, and outperforms the conventional heuristics centrality measures from social network analysis.

It is interesting to note that the k nodes ($k = 1, 2, ..., K$) that are discovered to be the most influential by the proposed method are substantially different from those that are found by the conventional centrality measures. For example, in the case of the final-time maximization problem, the best node ($k = 1$) chosen by the proposed method for the blog dataset is ranked 118 for the 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. In the case of the integral-time maximization problem, the difference is not that much but is similar by no means. The best node ($k = 1$) chosen by the proposed method for the blog dataset is ranked 17, 5 and 3 for the corresponding conventional methods, and the 15th node ($k = 15$) by the proposed method is ranked 31, 653 and 27, respectively. The best node ($k = 1$) chosen by the proposed method for the Wikipedia dataset is ranked 15, 6 and 3, and the 15th node ($k = 15$) by the proposed method is ranked 84, 23, and 12.

What these results imply is that the influential nodes strongly depend on the objective functions to be maximized, which in turn implies that taking the diffusion process into consideration is crucially important. The results would be affected not only by the network structure but also by the values of diffusion parameters, *i.e.*, even if the network structure remains the same, assigning different diffusion probabilities changes the influence degree of each node. Said differently, any centrality measure that is solely based on network topology has an intrinsic limitation to correctly evaluate the node

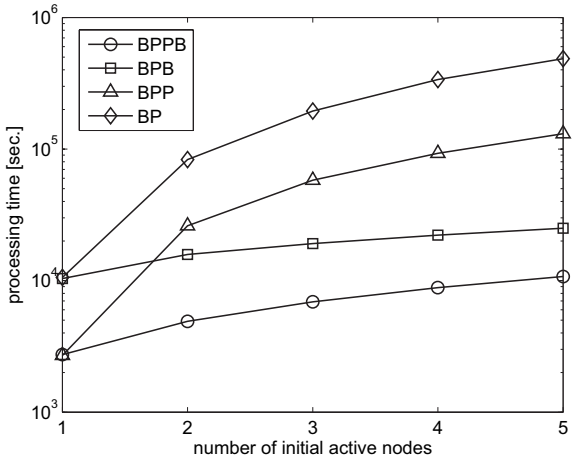


Fig. 8. Comparison of processing time for the blog network (final-time maximization problem).

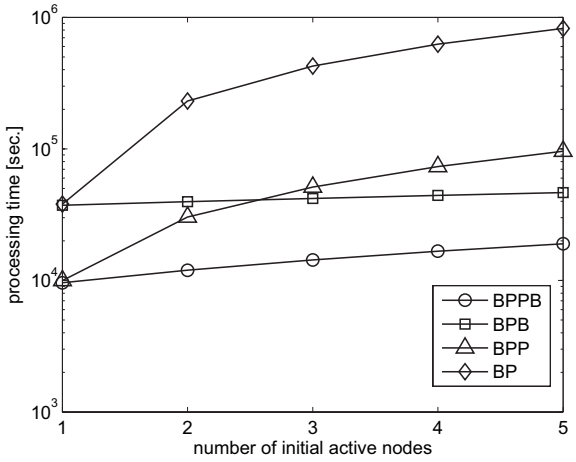


Fig. 9. Comparison of processing time for the Wikipedia network (final-time maximization problem).

influence as defined in this paper. We realize that these centrality measures are not necessarily designed to infer the influential nodes. They have their own advantages, *e.g.*, degree centrality can be used to identify the core nodes of a community and betweenness centrality can be used to study community structure. Indeed, the recently proposed topological centrality (Zhuge and Zhang, 2010) is shown to be very useful to understand the structure of network by distinguishing the roles of nodes, discovering communities and finding underlying backbone networks.

5.3.2. Comparison of Computational Cost among Different Combinations of Component Techniques

Next, we compared the processing time of the proposed method (BPPB) with three other methods with different combinations of component techniques (with/without Pruning and Burnout), i.e. bond percolation only (BP), bond percolation with pruning (BPP) and bond percolation with burnout (BPB) to see the effect of each component. We only show the results for the final-time maximization problem because it is self-evident that the processing time for the integral-time maximization problem is almost the same from the algorithm in 4.1. Figures 8 and 9 show the processing time of these four methods as a function of the number of initial active nodes K for the blog and the Wikipedia networks, respectively. In these figures, circles, triangles, squares and crosses indicate the results of BPPB, BPB, BPP and BP, respectively. The effect of the pruning is shown by the difference of the processing time at $K = 1$ (difference between BP and BPP). The pruning reduces the processing time to about $1/5$, which is consistent with Figs. 2 and 3 for $T = 30$ in 5.2.2. At $K = 2$ the effect of burnout starts appearing and it surpasses the effect of pruning for the blog network (BPB < BPP) but it still does not do so for the Wikipedia network (BPP < BPB). However, after $K \geq 3$ the effect of burnout surpasses the effect of pruning, and burnout plays a key role of reducing the computational cost. Combining the both, i.e., BPPB, always gives the best results within the region where the experiments were performed, i.e. $K \leq 5$. The amount of reduction in processing time by BPPB is large. The processing time of BP and BPPB for $K = 5$ is 5.8 days and 2.8 hours, respectively, for the blog network, and 9.3 days and 5.6 hours, respectively, for the Wikipedia network. The processing time reduces to $1/50$ for the blog network and $1/40$ for the Wikipedia network for $K = 5$. However, it is seen that the difference between BPB and BPPB becomes smaller as K becomes larger and it is predicted that eventually BPB will surpass BPPB, meaning that the overhead of pruning exceeds the saving by pruning. Thus, it is advisable to use both the strategies only in the initial few iterations, and stop using the pruning and use the burnout alone in the succeeding iterations in the greedy algorithm. Note that the above reduction is for $T = 30$. It is expected that the reduction is much larger for a larger T, e.g., $T = 100$, and also for a larger K, e.g. $K = 30$. Needless to say, the naive method needs an order of month to return the results and is prohibitively inefficient. From these results, we can conclude that the proposed method is much more efficient than the simple BP method and can be practical.

6. Discussion

The influence function $\sigma(\cdot, T)$ is submodular (Kempe et al, 2003). For solving a combinatorial optimization problem of a submodular function f on V by the greedy algorithm, Leskovec et al. (Leskovec et al, 2007a) 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 both the 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. Moreover, estimating influence function $\sigma(\cdot, 30)$ by $10,000$ simulations needed more than 35.8 hours for the blog dataset and 13.2 days for the Wikipedia dataset, respectively. However, the proposed method for $K = 30$ needed less than 7.0 hours for the blog dataset and 13.1 hours for the Wikipedia dataset, respectively. Therefore, it is clear that the proposed method can be faster than the method by Leskovec et al (2007a) for the influence maximization problem for the SIS model. In fact, we have confirmed in Kimura et al (2010) that the bond percolation method is 10 times faster than the lazy evaluation for the SIR model for $K = 30$. Since the SIS model can be mapped to the SIR model by introducing the layered graph, the result above is consistent to our previous result.

We discussed the accuracy and the computational cost of the proposed method in 5.2 and 5.3. Here we look into the solutions of the final-time maximization problem and the integral-time maximization problem. We found that these two different maximization problems give almost totally different nodes although the objective function to be maximized for the latter is the sum of the objective function of the former over the final time T. There is only one common node out of 30 influential nodes in case of the blog network and there are only five common nodes in case of the Wikipedia network. In general the identified influential nodes for the final-time maximization problem reflects the diffusion characteristics of one time slot but those for the integral-time maximization problem reflects the global diffusion characteristics. Intermediate process does not matter and what matters is only the final situation for the former, whereas the whole process does matter for the latter. It is important to distinguish these two different problem characteristics and use the right objective function that best suits the task in hand.

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 nodes once activated 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 $t = 0$ and finding the optimal subset H^* to maximize the expected influence. 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 two control strategies: pruning and burnout. 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 (final-time maximization problem) or the expected influence degree over the time span of interest (integral-time maximization problem). Both problems are solved by the greedy algorithm taking advantage of the submodu-

larity of the objective function. We confirmed by applying the proposed method to two real world networks taken from the blog and Wikipedia data that the proposed method can achieve considerable reduction in computation time without degrading the accuracy compared with the naive simulation method as predicted by the theory. Use of the two control strategies contributes to reducing the computational cost by a factor of 50 compared with the naive bond percolation which itself is 2 to 3 orders of magnitudes faster than the naive simulation method. The proposed method can discover nodes that are more influential than the nodes identified by the conventional methods based on the various centrality measures. The results of the two influence maximization problems are totally different in terms of the identified influential nodes and thus it is crucial to choose the right objective function that meets the need for the task. We further found that the pruning is effective when searching for a single influential node, but gradually its overhead surpasses its saving and the burnout is more powerful when searching for multiple influential nodes. Use of both is most effective for the initial few iterations. Thus, we recommend to use both the pruning and the burnout only in the initial few iterations, and stop using the pruning and use the burnout alone in the succeeding iterations in the greedy algorithm. 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.

Acknowledgements. 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-10-4053, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

Adar E, Adamic LA (2005) Tracking information epidemics in blogspace. In Skowron A, Agrawal R, Luck M, Yamaguchi T, Morizet-Mahoudeaux P, Liu J, Zhong N (eds). Proceedings of 2005 IEEE/WIC/ACM international conference on Web intelligence, Compiegne, France, September 2005, pp 207–214

Agarwal N and Liu H (2008) Blogosphere: research issues, tools, and applications. SIGKDD Explorations 10(1):18–31

Backstrom L, Dwork C, Kleinberg J (2007) Wherefore art thou r3579x?: anonymized social networks, hidden patterns, and structural steganography. In Williamson CL, Zurko ME, Patel-Schneider PF, Shenoy PJ (eds). Proceedings of the 16th international conference on World Wide Web, Banff, Alberta, Canada, May 2007, pp 181–190

Chen W, Wang Y, Yang S (2009) Efficient influence maximization in social networks. Elder IV JF, Fogelman-Soulié F, Flach PA, Zaki MJ (eds). Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining, Paris, France, June 2009, pp 199–208

Domingos P (2005) Mining social networks for viral marketing. IEEE Intelligent Systems 20(1):80–82

Domingos P, Richardson M (2001) Mining the network value of customers. In Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining, San Francisco, CA, August 2001, pp 57–66

Goldenberg J, Libai B, Muller E (2001) Talk of the network: a complex systems look at the underlying process of word-o-mouth. Marketing Letters 12(3):211–223

Grassberger P (1983) On the critical behavior of the general epidemic process and dynamical percolation. Mathematical Bioscience 63(2):157–172

Gruhl D, Guha R, Liben-Nowell D, Tomkins A (2004) Information diffusion through blogspace. In Feldman SI, Uretsky M, Najork M, Wills CE (eds). Proceedings of the 13th international conference on World Wide Web, New York, NY, May 2004, pp 107–117

Kempe D, Kleinberg J, Tardos E (2003) Maximizing the spread of influence through a social network. In Getoor L, Senator TE, Domingos P, Faloutsos C (eds). Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining, Washington DC, August 2003, pp 137–146

Kimura M, Saito K, Motoda H (2009a) Blocking links to minimize contamination spread in a social network. ACM Transactions on Knowledge Discovery from Data 3(2):9:1–9:23

Kimura M, Saito K, Motoda H (2009b) Efficient estimation of influence functions for SIS model on social networks. In Boutilier C (ed). Proceedings of the 21st international joint conference on artificial intelligence, Pasadena, CA, July 2009, pp 2046–2051

Kimura M, Saito K, Nakano R (2007) Extracting influential nodes for information diffusion on a social network. In Proceedings of the 22nd AAAI conference on artificial intelligence, Vancouver, British Columbia, Canada, July 2007, pp 1371–1376

Kimura M, Saito K, Nakano R, Motoda H (2010) Extracting influential nodes on a Social Network for information. Data Mining and Knowledge Discovery 20(1):70–97

Leskovec J, Krause A, Guestrin C, Faloutsos C, VanBriesen J, Glance N (2007a) Cost-effective outbreak detection in networks. In Berkhin P, Caruana R, Wu X (eds). Proceedings of the 13th ACM SIGKDD international conference on knowledge discovery and data mining, San Jose, CA, August 2007, pp 420–429

Leskovec J, McGlohon M, Faloutsos C, Glance N, Hurst M (2007b) Patterns of cascading behavior in large blog graphs. In Proceedings of the Seventh SIAM international conference on data mining, Minneapolis, MN, April 2007, pp 551–556

McCallum A, Corrada-Emmanuel A, Wang X (2005) Topic and role discovery in social networks. In Kaelbling LP, Saffiotti A (eds). Proceedings of the 19th international joint conference on artificial intelligence, Edinburgh, Scotland, UK, July - August 2005, pp 786–791

Mislove A, Marcon M, Gummadi KP, Druschel P, Bhattacharjee B (2007) Measurement and analysis of online social networks. In Dovrolis C, Roughan M (eds). Proceedings of the seventh ACM SIGCOMM conference on internet measurement, San Diego, CA, October 2007, pp 29–42

Muhlestein D, Lim S (2009) Online learning with social computing based interest sharing. Knowledge and Information Systems, Published online: November 2009

Newman MEJ (2001) The structure of scientific collaboration networks. Proceedings of the National Academy of Sciences of the United States of America 98(2):404–409

Newman MEJ (2002) Spread of epidemic disease on networks. Physical Review E 66:016128

Newman MEJ (2003) The structure and function of complex networks. SIAM Review 45(2):167–256

Newman MEJ, Park J (2003) Why social networks are different from other types of networks. Physical Review E 68:036122

Peng W, Li T (2010) Temporal relation co-clustering on directional social network and author-topic evolution. Knowledge and Information Systems, Published online: March 2010

Richardson M, Domingos P (2002) Mining knowledge-sharing sites for viral marketing. In Proceedings of the Eighth ACM SIGKDD international conference on knowledge discovery and data mining, Edmonton, Alberta, Canada, July 2002, pp 61–70

Saito K, Kimura M, Motoda H (2009) Discovering influential nodes for SIS models in social networks. In Gama J, Costa VS, Jorge AM, Brazdil P (eds). Proceedings of the 12th International Conference of Discovery Science, Porto, Portugal, October 2009. Lecture Notes in Computer Science 5808, Springer, pp 302–316

Wasserman S, Faust K (1994) Social network analysis. Cambridge University Press, Cambridge, UK

Watts DJ (2002) A simple model of global cascade on random networks. Proceedings of the National Academy of Sciences of the United States of America 99(9):5766–5771

Watts DJ, Dodds PS (2007) Influence, networks, and public opinion formation. Journal of Consumer Research 34(4):441–458

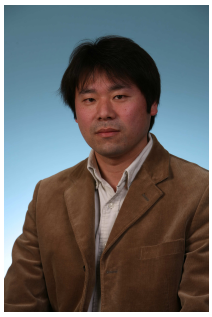
Zhou B, Pei J (2010) The *k*-anonymity and *l*-diversity approaches for privacy preservation in social networks against neighborhood attacks. Knowledge and Information Systems, Published online: June 2010

Zhou D, Ji X, Zha H, Giles CL (2006) Topic evolution and social interactions: how authors effect research. In Yu PS, Tsotras VJ, Fox EA, Liu B (eds). Proceedings of the 2006 ACM CIKM international conference on information and knowledge management, Arlington, VA, November 2006, pp 248–257

Zhuge H, Zhang J (2010) Topological centrality and its applications. Journal of the American Society for Information Science and Technology 61(9):1824-1841

Author Biographies

Kazumi Saito received a BS degree in mathematics from Keio University, Kanagawa, Japan, in 1985, and a PhD in engineering from University of Tokyo, Tokyo, Japan, in 1998. In 1985, he joined the NTT Electrical Communication Laboratories, Kanagawa, Japan. In 1991, he joined the NTT Communication Science Laboratories, Kyoto, Japan. In 2007, he joined the University of Shizuoka, Shizuoka, Japan. He is a professor at the School of Administration and Informatics. From 1991 to 1992, he was a visiting scholar at the University of Ottawa, Ontario, Canada. His current research interests are machine learning and statistical analysis of complex networks. He is a member of the Institute of Electronics, Information, and Communication Engineers (IEICE), the Information Processing Society of Japan (IPSJ), the Japanese Society of Artificial Intelligence (JSAI), the Japanese Neural Network Society (JNNS).



Masahiro Kimura received his BS, MS, and PhD degrees in mathematics from Osaka University, Osaka, Japan, in 1987, 1989, and 2000, respectively. In April 1989, he joined Nippon Telegraph and Telephone (NTT) Corporation, Tokyo, Japan. He mainly worked at NTT Human Interface Laboratories and NTT Communication Science Laboratories. In April 2005, he joined Ryukoku University, Kyoto, Japan. Currently, he serves as a professor of the Department of Electronics and Informatics. His research interests include complex networks science, data mining, and machine learning. He is a member of the Japanese Society for Artificial Intelligence (JSAI), the Mathematical Society of Japan (MSJ), the Japan Society for Industrial and Applied Mathematics (JSIAM), the Japanese Neural Networks Society (JNNS), and the Institute of Electronics, Information and Communication Engineers (IEICE).

Kouzou Ohara received the Master of Engineering degree from Osaka University, Osaka, Japan in 1995. He also received the Ph. D. degree in engineering from Osaka University in 2002. He is currently an Associate Professor in the department of Integrated Information Technology at the college of Science and Engineering of Aoyama Gakuin University. His research interests include machine learning, data mining, social network analysis, and personalization of intelligent systems. He is a member of the Institute of Electrical and Electronics Engineers (IEEE), the Association for the Advancement of Artificial Intelligence (AAAI), the Institute of Electronics, Information, and Communication Engineers (IEICE), the Information Processing Society of Japan (IPSJ) and the Japanese Society of Artificial Intelligence (JSAI).

Hiroshi Motoda is a professor emeritus of Osaka University and a scientific advisor of AFOSR/AOARD (Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, US Air Force Research Laboratory). His research interests include information diffusion in social network, data mining, machine learning, knowledge acquisition, scientific knowledge discovery and artificial intelligence in general. He received his Bs, Ms and PhD degrees all in nuclear engineering from the University of Tokyo. He is a member of the steering committee of PAKDD, PRICAI, DS and ACML. He received the best paper awards from Atomic Energy Society of Japan (1977, 1984) and from Japanese Society of Artificial Intelligence (1989, 1992, 2001), the outstanding achievement awards from JSAI (2000) and Okawa Publication Prize from Okawa Foundation (2007).

Correspondence and offprint requests to: Masahiro Kimura, Department of Electronics and Informatics, Ryukoku University, Otsu 520-2194, Japan. Email: kimura@rins.ryukoku.ac.jp