Blocking Links to Minimize Contamination Spread in a Social Network

MASAHIRO KIMURA
Ryukoku University
KAZUMI SAITO
University of Shizuoka
and
HIROSHI MOTODA
Osaka University

We address the problem of minimizing the propagation of undesirable things, such as computer viruses or malicious rumors, by blocking a limited number of links in a network, which is converse to the influence maximization problem in which the most influential nodes for information diffusion is searched in a social network. This minimization problem is more fundamental than the problem of preventing the spread of contamination by removing nodes in a network. We introduce two definitions for the contamination degree of a network, accordingly define two contamination minimization problems, and propose methods for efficiently finding good approximate solutions to these problems on the basis of a naturally greedy strategy. Using large social networks, we experimentally demonstrate that the proposed methods outperform conventional link-removal methods. We also show that unlike the case of blocking a limited number of nodes, the strategy of removing nodes with high out-degrees is not necessarily effective for these problems.

Categories and Subject Descriptors: G.2.2 [**Discrete Mathematics**]: Graph Theory—*network problems*; H.2.8 [**Database Management**]: Database Applications—*data mining*; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*sociology*

General Terms: Algorithms

Additional Key Words and Phrases: Contamination diffusion, link analysis, social networks

1. INTRODUCTION

Considerable attention has recently been devoted to investigating the structure and function of various networks including computer networks, social networks and the

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).
Authors' addresses: M. Kimura, Department of Electronics and Informatics, Ryukoku University, Otsu 520-2194, Japan; email: kimura@rins.ryukoku.ac.jp; K. Saito, School of Administration and Informatics, University of Shizuoka, Shizuoka 422-8526, Japan; email: k-saito@u-shizuoka-ken.ac.jp; H. Motoda, Institute of Scientific and Industrial Research, Osaka University, Osaka 567-0047, Japan; email: motoda@ar.sanken.osaka-u.ac.jp.

World Wide Web [Newman 2003]. From a functional point of view, networks can mediate diffusion of various things such as innovation and topics. However, undesirable things can also spread through networks. For example, computer viruses can spread through computer networks and email networks, and malicious rumors can spread through social networks among individuals. Thus, developing effective strategies for preventing the spread of undesirable things through a network is an important research issue. Previous work studied strategies for reducing the spread size by removing nodes from a network. It has been shown in particular that the strategy of removing nodes in decreasing order of out-degree can often be effective [Albert et al. 2000; Broder et al. 2000; Callaway et al. 2000; Newman et al. 2002]. Here notice that removal of nodes by necessity involves removal of links. Namely, the task of removing links is more fundamental than that of removing nodes, and this is the problem we address in the paper.

In contrast, finding influential nodes that are effective for the spread of information through a social network is also an important research issue in terms of sociology and "viral marketing" [Domingos and Richardson 2001; Richardson and Domingos 2002; Gruhl et al. 2004]. Recent studies include attempts to solve a combinatorial optimization problem called the *influence maximization problem* on a network under the *independent cascade (IC) model*, a widely-used fundamental probabilistic model of information diffusion[Kempe et al. 2003; Kimura et al. 2007]. Here, the influence maximization problem is the problem of extracting a set of K nodes to target for initial activation such that it yields the largest expected spread of information, where K is a given positive integer. Note also that the IC model can be identified with the so-called *susceptible/infective/recovered (SIR) model* for the spread of disease in a network [Gruhl et al. 2004].

As we see, what we address in this paper is a problem that is converse to the influence maximization problem. The problem is to minimize the spread of undesirable things by blocking a limited number of links in a network. More specifically, we consider, when some undesirable thing starts with any node and diffuses through the network under the IC model, finding a set of K links such that the resulting network obtained by blocking those links minimizes the *contamination degree* for the undesirable thing, where K is a given positive integer. We refer to this combinatorial optimization problem as a *contamination minimization problem*. We introduce two definitions for the contamination degree of a network; the *average contamination degree* and the *worst contamination degree*. According to these definitions, we formalize two contamination minimization problems; the *average contamination minimization problem* and the *worst contamination minimization problem*. The former aims to minimize the expected number of contaminated nodes (i.e., the average case), and the latter aims to minimize the maximum number of contaminated nodes (i.e., the worst case).

We presented in [Kimura et al. 2008] a method for efficiently finding a good approximate solution on the basis of a naturally greedy strategy for the average contamination minimization problem. In this paper, we explain the method in more detail, and propose a novel method for efficiently finding a good approximate solution for the worst contamination minimization problem on the basis of the same greedy strategy

Furthermore, we compare the proposed methods with a naive greedy strategy in terms of computational complexity for both the average and the worst contamination minimization problems, and show that the proposed methods can achieve a great deal of reduction in computational cost. We also present strategies for making the proposed methods computationally more efficient in practice. Finally, using large real networks that exhibit many of the key features of social networks, we experimentally demonstrate that the proposed methods outperform link-removal heuristics that rely on the well-studied notions of betweenness and out-degree in the field of complex network theory. In particular, we show that unlike the case of blocking a limited number of nodes, the strategy of removing nodes with high out-degrees is not necessarily effective for our problems.

2. INFORMATION DIFFUSION MODEL

We assume the IC model to be a mathematical model for the diffusion process of some undesirable thing on a network. We call nodes *active* if they have been contaminated by the undesirable thing.

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. Throughout this paper, a network means a directed network, a link means a directed link, and we also call a network a graph. According to the work of Kempe et al. [2003], we define the IC model on graph G in Section 2.1, and recall a mathematical definition of the influence maximization problem for the IC model on graph G in Section 2.2.

2.1 Independent Cascade Model

First, we define the IC model on graph G. In the IC model, the diffusion process unfolds in discrete time-steps $t \geq 0$, and it is assumed that nodes can switch their states only from inactive to active, but not from active to inactive. Given an initial set A of active nodes, we assume that the nodes in A have first become active at time-step 0, and all the other nodes are inactive at time-step 0. For every $e \in E$, we specify a real value p_e with $0 < p_e < 1$ in advance. Here, p_e is referred to as the *propagation probability* through link e.

The diffusion process proceeds from a given initial active set A in the following way. When a node u first becomes active at time-step t, it is given a single chance to activate each currently inactive child node w, and succeeds with probability p_e, where $e = (u, w) \in E$. Here, for a link $e' = (u', w') \in E$, nodes u' and w' are called the *parent* and *child* nodes of w' and u', respectively. If u succeeds, then w will become active at time-step $t + 1$. If multiple parent nodes of w first become active at time-step t, then their activation attempts are sequenced in an arbitrary order, but all performed at time-step t. Whether or not u succeeds, it cannot make any further attempts to activate w in subsequent rounds. The process terminates if no more activations are possible.

For an initial active set A, let $\varphi(A; G)$ denote the number of active nodes at the end of the random process for the IC model on G. Note that $\varphi(A; G)$ is a random variable. Let $\sigma(A; G)$ denote the expected value of $\varphi(A; G)$. We call $\sigma(A; G)$ the *influence degree* of node set A on graph G. When A is in particular equal to a set of single node $\{v\}$, we simply denote $\sigma(A; G)$ by $\sigma(v; G)$, and call $\sigma(v; G)$ the influence degree of node v on graph G.

2.2 Influence Maximization Problem

Next, we recall a mathematical definition of the influence maximization problem on a network. Here, we consider maximizing the spread of desirable information through graph $G = (V, E)$. Let K be a given positive integer with $K < |V|$. Here, $|X|$ stands for the number of elements of a set X. The influence maximization problem on G for the IC model is defined as follows: Find a subset A^* of V with $|A^*| = K$ such that $\sigma(A^*; G) \geq \sigma(A; G)$ for every $A \subset V$ with $|A| = K$.

3. PROBLEM FORMULATION

We assume that some undesirable thing starts with any node in a network and diffuses through the network under the IC model. For preventing it from spreading through the network, we aim to minimize the *contamination degree* by appropriately removing a fixed number of links. Here, the contamination degree of a network is a measure of how badly the undesirable thing will contaminate the network. We give two definitions for contamination degree, and mathematically formalize two contamination minimization problems on a network.

3.1 Contamination Degree

For any graph $G = (V, E)$, we introduce two definitions for contamination degree of G.

3.1.1 *Average Contamination Degree.* We define the *average contamination degree* $c_0(G)$ of graph G as the average of influence degrees of all the nodes in G,

$$c_0(G) = \frac{1}{|V|} \sum_{v \in V} \sigma(v; G). \tag{1}$$

3.1.2 *Worst Contamination Degree.* We define the *worst contamination degree* $c_+(G)$ of graph G as the maximum of influence degrees of all the nodes in G,

$$c_+(G) = \max_{v \in V} \sigma(v; G). \tag{2}$$

3.2 Contamination Minimization Problem

According to the above definitions of contamination degree, we mathematically define the contamination minimization problems on a network, which are converse to the influence maximization problem on the network.

For any graph $G = (V, E)$, we denote by $c(G)$ both the average contamination degree $c_0(G)$ and the worst contamination degree $c_+(G)$. For any link $e \in E$, let $G(e)$ denote the graph $(V, E \setminus \{e\})$. We refer to $G(e)$ as the graph constructed by *blocking* e in G. Similarly, for any $D \subset E$, let $G(D)$ denote the graph $(V, E \setminus D)$. We refer to $G(D)$ as the graph constructed by *blocking* D in G.

We define the *contamination minimization problems* on a graph $G = (V, E)$ as follows: Given a positive integer K with $K < |E|$, find a subset D^* of E with $|D^*| = K$ such that $c(G(D^*)) \leq c(G(D))$ for any $D \subset E$ with $|D| = K$. The contamination minimization problem for $c = c_0$ is referred to as the *average contamination minimization problem*, and the contamination minimization problem for $c = c_+$ is referred to as the *worst contamination minimization problem*.

For a large network, any straightforward method for exactly solving the contamination minimization problems suffers from combinatorial explosion. Therefore, we consider approximately solving the problems.

4. PROPOSED METHOD

We propose methods for efficiently finding good approximate solutions to our contamination minimization problems. Let K be the number of links to be blocked in the problems.

4.1 Greedy Algorithm

We approximately solve the contamination minimization problems on a given graph $G_0 = (V_0, E_0)$ by the following greedy algorithm:

A1. Initialize a subset D of E_0 as $D \leftarrow \emptyset$.
A2. Initialize a graph $G = (V, E)$ as $V \leftarrow V_0$ and $E \leftarrow E_0$.
A3. Choose a link $e_* \in E$ minimizing $c(G(e))$, $(e \in E)$.
A4. Update D as $D \leftarrow D \cup \{e_*\}$.
A5. Update $G = (V, E)$ as $E \leftarrow E \setminus \{e_*\}$.
A6. Return to Step **A3** if $|D| < K$.
A7. Set $D_K \leftarrow D$.
A8. Set $G_K \leftarrow G$.

Here, D_K is the set of links blocked, and represents the approximate solution obtained by this algorithm. We refer D_K to as the *greedy solution*. G_K is the graph constructed by blocking D_K in the graph G_0, that is, $G_K = G_0(D_K)$.

To implement this greedy algorithm, we need methods for calculating

$$e_* = \arg\min_{e \in E} c(G(e)) \tag{3}$$

for a given graph $G = (V, E)$ in Step **A3** of the algorithm. The IC model is a stochastic process model, and it is an open question to exactly calculate influence degrees by an efficient method [Kempe et al. 2003]. Therefore, we must develop methods for efficiently estimating $\{c(G(e));\ e \in E\}$ for graph $G = (V, E)$.

Kimura et al. [2007] presented the bond percolation method that efficiently estimates the influence degrees $\{\sigma(v; \tilde{G});\ v \in \tilde{V}\}$ for any graph $\tilde{G} = (\tilde{V}, \tilde{E})$. Thus, in the greedy algorithm, we can estimate $c(G(e))$ for each $e \in E$ by applying the bond percolation method for the graph $G(e)$ and using Equations (1) or (2). Namely, we can simply estimate the greedy solution D_K by implementing Step **A3** of the greedy algorithm as follows:

(1) Estimate $\{c(G(e));\ e \in E\}$ by straightforwardly performing the bond percolation method $|E|$ times.
(2) Find $e_* \in E$ such that $c(G(e_*)) \leq c(G(e))$ for any $e \in E$.

We refer this strategy to as the *naive greedy strategy*. However, $|E|$ becomes very large for a large network in the greedy algorithm unless K is very large. Namely, the naive greedy strategy is not practical for large networks. Therefore, we propose more efficient methods for estimating $e_* \in E$ satisfying Equation (3) on the basis of the bond percolation method.

4.2 Bond Percolation Method

First, we revisit the bond percolation method [Kimura et al. 2007]. Here, we consider estimating the influence degrees $\{\sigma(v;G);\ v \in V\}$ for the IC model with propagation probabilities $\{p_e; e \in E\}$ on a graph $G = (V, E)$.

The *bond percolation process with occupation probabilities* $\{p_e; e \in E\}$ on graph G is the random process in which each link $e \in E$ is independently declared "occupied" with probability p_e. Note that in terms of information diffusion on a network, the occupied links represent the links through which the information propagates, and the unoccupied links represent the links through which the information does not propagate. For a positive integer M, we perform the bond percolation process M times, and sample a set of M graphs constructed by the occupied links,

$$\{G^m = (V, E^m);\ m = 1, \cdots, M\}.$$

For any $v \in V$, we define $s(v; G, M)$ by

$$s(v; G, M) = \frac{1}{M} \sum_{m=1}^{M} |F(v; G^m)|. \tag{4}$$

Here, for any graph $\tilde{G} = (\tilde{V}, \tilde{E})$ and any node $v \in \tilde{V}$, $F(v; \tilde{G})$ stands for the set of all the nodes that are *reachable* from node v on graph $\tilde{G}$. We say that node u is reachable from node v on graph $\tilde{G}$ if there is a path from u to v along the links on graph $\tilde{G}$.

It is known [Newman 2003] that the IC model with propagation probabilities $\{p_e; e \in E\}$ on graph G can be exactly mapped onto the bond percolation process with occupation probabilities $\{p_e; e \in E\}$ on graph G, and the influence degree $\sigma(v; G)$ of node $v \in V$ can well be approximated by $s(v; G, M)$,

$$\sigma(v; G) \simeq s(v; G, M), \quad (v \in V), \tag{5}$$

if M is sufficiently large. We decompose each graph G^m into the strongly connected components (SCCs) as follows:

$$V = \bigcup_{j=1}^{J^m} SCC(u_j^m; G^m), \tag{6}$$

where J^m is the number of the strongly connected components of graph G^m, each u_j^m is an element of V, and $SCC(u_j^m; G^m)$ denotes the SCC of graph G^m that contains node u_j^m. Note that

$$|F(v; G^m)| = \left|F(u_j^m; G^m)\right|, \quad \text{if } v \in SCC(u_j^m; G^m). \tag{7}$$

Thus, by calculating $\{|F(u_j^m; G^m)|; j = 1, \cdots, J^m\}$ in advance and using Equation (7), we efficiently calculate $|F(v; G^m)|$ for all $v \in V$. Once we have $\{|F(v; G^m)|; v \in V, m = 1, \cdots, M\}$, we can calculate $s(v; G, M)$ for all $v \in V$ from Equation (4).

Namely, the bond percolation method estimates all the influence degrees $\{\sigma(v; G); v \in V\}$ on graph G as follows: It first specifies the value of integer M, calculates $s(v; G, M)$ for all $v \in V$ by performing the above procedure, and estimates $\sigma(v; G)$ for all $v \in V$ by using Equation (5).

4.3 Estimation Method

Now, we give methods for efficiently estimating $e_* \in E$ satisfying Equation (3) for a given graph $G = (V, E)$ to implement Step **A3** of the greedy algorithm for the average and the worst contamination minimization problems.

First, we perform the bond percolation process M times on graph $G = (V, E)$, and sample a set of M graphs constructed by the occupied links,

$$\{G^m = (V, E^m);\ m = 1, \cdots, M\},$$

where M is a given positive integer. Next, we calculate

$$\mathcal{B}_M(e) = \{m \in \{1, \cdots, M\};\ e \notin E^m\}, \quad (e \in E). \tag{8}$$

Note that $\mathcal{B}_M(e)$ represents the subset of the M trials for the bond percolation process on graph G such that e is not an occupied link.

Here, we consider performing the bond percolation process $|\mathcal{B}_M(e)|$ times on the graph $G(e) = (V, E \setminus \{e\})$ for any $e \in E$, and sampling a set of $|\mathcal{B}_M(e)|$ graphs constructed by the occupied links,

$$\{G(e)^m;\ m = 1, \cdots, |\mathcal{B}_M(e)|\}.$$

We assume that M is large enough so that $|\mathcal{B}_M(e)|$ is also sufficiently large. Then, by Equation (5), we have

$$\sigma(v; G(e)) \simeq s\left(v; G(e), |\mathcal{B}_M(e)|\right), \quad (v \in V). \tag{9}$$

Note from Equation (4) that

$$s\left(v; G(e), |\mathcal{B}_M(e)|\right) = \frac{1}{|\mathcal{B}_M(e)|} \sum_{m=1}^{|\mathcal{B}_M(e)|} |F(v; G(e)^m)|, \quad (v \in V). \tag{10}$$

In order to efficiently estimate $\{c(G(e));\ e \in E\}$ without applying the bond percolation method on the graph $G(e)$ for every $e \in E$, we alternatively calculate

$$\overline{s}_M(v, e) = \frac{1}{|\mathcal{B}_M(e)|} \sum_{m \in \mathcal{B}_M(e)} |F(v; G^m)|, \quad (v \in V,\ e \in E), \tag{11}$$

for the graph G on the basis of the bond percolation method. Since each link of graph G is independently declared "occupied" in the bond percolation process, we can obtain the following theorem from Equations (8), (9), (10) and (11).

THEOREM 4.1. *Let $G = (V, E)$ be a graph. For every $v \in V$ and $e \in E$, we have*

$$\overline{s}_M(v, e) \to \sigma(v; G(e))$$

as $M \to \infty$.

From Theorem 4.1, we can apply the approximation

$$\sigma(v; G(e)) \simeq \overline{s}_M(v, e), \quad (v \in V,\ e \in E), \tag{12}$$

for a sufficiently large M. Therefore, by Equations (1) and (2), we propose estimating $e_* \in E$ satisfying Equation (3) for a given graph $G = (V, E)$ as follows:

$$e_* = \arg\min_{e \in E} \left(\frac{1}{|V|} \sum_{v \in V} \overline{s}_M(v, e) \right) \tag{13}$$

for the average contamination minimization problem (i.e., $c = c_0$), and

$$e_* = \arg\min_{e \in E} \left(\max_{v \in V} \ \overline{s}_M(v, e) \right) \tag{14}$$

for the worst contamination minimization problem (i.e., $c = c_+$). Note that for the proposed method, the value of M is specified in advance.

4.4 Computational Complexity and Implementational Strategy

For both the average and the worst contamination minimization problems, we compare the proposed methods with the naive greedy strategy in terms of computational complexity. We focus on the computational complexity of estimating $e_* \in E$ satisfying Equation (3) for a given graph $G = (V, E)$.

Let Q be the expected computational complexity for calculating the values of $\{s(v; G, 1);\ v \in V\}$ on graph $G = (V, E)$ on the basis of the bond percolation method (see, Equation (4)). Then, the expected computational complexity of the proposed method for calculating $\{\overline{s}_M(v, e);\ v \in V,\ e \in E\}$ amounts to MQ, since the values of $\{|F(v; G^m)|;\ v \in V,\ m = 1, \cdots, M\}$ are calculated on the basis of the bond percolation method (see, Equations (4) and (11)). Note that for any $e \in E$, calculating $\{\overline{s}_M(v, e);\ v \in V\}$ for the proposed methods corresponds to estimating $c(G(e))$ through $|\mathcal{B}_M(e)|$ trials of the bond percolation process on graph $G(e)$ (see, Equation (11)). For the naive greedy strategy, we consider estimating $c(G(e))$ through $|\mathcal{B}_M(e)|$ trials of the bond percolation process on graph $G(e)$ (see, Equations (9) and (10)). Then, in order to estimate the values of $\{c(G(e);\ e \in E\}$, the naive greedy strategy requires the computational complexity of $Q \sum_{e \in E} |\mathcal{B}_M(e)|$. Here we assumed that the computational complexities of $s(v; G, 1)$ and $s(v; G(e), 1)$ are the same because $|E|$ is sufficiently large in general. By noting that the expected value of $|\mathcal{B}_M(e)|$ is $(1 - p_e)M$, the expected computational complexity of the naive greedy strategy for estimating $\{c(G(e);\ e \in E\}$ becomes $MQ \sum_{e \in E}(1 - p_e)$. Thus, we can see that the proposed methods are $\sum_{e \in E}(1 - p_e)$ times faster than the naive greedy strategy on the average. For instance, when the number of links is $100,000$ and each propagation probability p_e for the IC model is a uniform probability $p = 0.2$, the value of $\sum_{e \in E}(1 - p_e)$ is $80,000$. Namely, the proposed methods can achieve a great deal of reduction in computational cost, compared with the naive greedy strategy.

Furthermore, the following strategies can be used to efficiently find $e_* \in E$ satisfying Equations (13) or (14) for a given graph $G = (V, E)$ in actual practice.

First, as for the worst contamination minimization problem, we apply the idea of lazy evaluations for marginal increments of a submodular function by Leskovec et al. [2007]. More specifically, we efficiently calculate Equation (14) by appropriately pruning the evaluations for $\{\overline{s}_M(v, e);\ v \in V,\ e \in E\}$. By Equations (4) and (11), we have

$$M\, s(v; G, M) \ = \ |\mathcal{B}_M(e)|\ \overline{s}_M(v, e) \ + \sum_{m \in \{1, \cdots, M\} \setminus \mathcal{B}_M(e)} |F(v; G^m)|$$

for any $v \in V$ and $e \in E$. Thus, we can derive the following upper bound with

respect to $\overline{s}_M(v,e)$:

$$\frac{M}{|\mathcal{B}_M(e)|}s(v;G,M) \geq \overline{s}_M(v,e), \quad (v \in V, e \in E). \tag{15}$$

We arbitrarily fix a link $e \in E$. Then, we first sort all the nodes $\{v \in V\}$ of graph G by the value $Ms(v;G,M)/|\mathcal{B}_M(e)|$ in descending order as follows: $\langle v_i;\, i = 1, \cdots, |V|\rangle$. We next calculate the value of $\overline{s}_M(v,e)$ in this order, until the current maximum value $\overline{s}_M(v_i^*,e)$ exceeds the value $Ms(v_{i+1};G,M)/|\mathcal{B}_M(e)|$ for the head v_{i+1} of the remaining nodes. By Equation (15), this pruning guarantees that the current maximum value attains the maximum without necessarily evaluating $\overline{s}_M(v,e)$ for all $v \in V$. In our experiments, the computational efficiency was greatly improved by using this strategy, just as reported in [Leskovec et al. 2007].

Next, as for the average contamination minimization problem, we efficiently calculate Equation (13) without evaluating the value of $\overline{s}_M(v,e)$ for every pair of node v and link e. Our strategy is to exploit the relation

$$\frac{1}{|V|}\sum_{v\in V}\overline{s}_M(v,e) \;=\; \frac{1}{|\mathcal{B}_M(e)|}\sum_{m\in\mathcal{B}_M(e)}\frac{1}{|V|}\sum_{v\in V}|F(v;G^m)|, \quad (v \in V, e \in E), \tag{16}$$

(see, Equation (11)). More specifically, we evaluate $\sum_{v\in V}|F(v;G^m)/|V|$ for each m on the basis of the bond percolation method in advance (see, Equations (6) and (7)), and then calculate Equation (13) by evaluating $\sum_{v\in V}\overline{s}_M(v,e)$ for every $e \in E$ using Equation (16).

5. EXPERIMENTAL EVALUATION

Using two large real networks that exhibit many of the key features of social networks, we experimentally evaluated the performance of the proposed method.

5.1 Network Data

First, we employed a trackback network of blogs because a piece of information can propagate from one blog author to another blog author through a trackback. Since bloggers (i.e., blog authors) discuss various topics and establish mutual communications by putting trackbacks on each other's blogs, we regarded a link created by a trackback as a bidirectional link. By tracing up to ten steps back in the trackbacks from the blog of the theme "JR Fukuchiyama Line Derailment Collision" in the site "goo" [1], we collected a large connected trackback network in May, 2005. The resulting network was a directed graph of $12,047$ nodes and $79,920$ links, which features the so-called "power-law" degree distribution that most large real networks exhibit (see, Figure 1). Here, the degree distribution is the distribution of the number of links for every node. We refer to this network data as the blog network.

Next, we employed a network of people that was derived from the "list of people" within Japanese Wikipedia. Specifically, we extracted the maximal connected component of the undirected graph obtained by linking two people in the "list of people" if they co-occur in six or more Wikipedia pages, and constructed a directed

[1] `http://blog.goo.ne.jp/usertheme/`

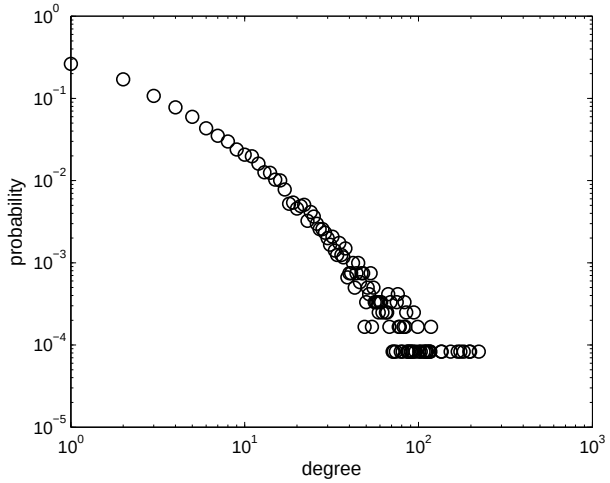

Fig. 1. The degree distribution for the blog network.

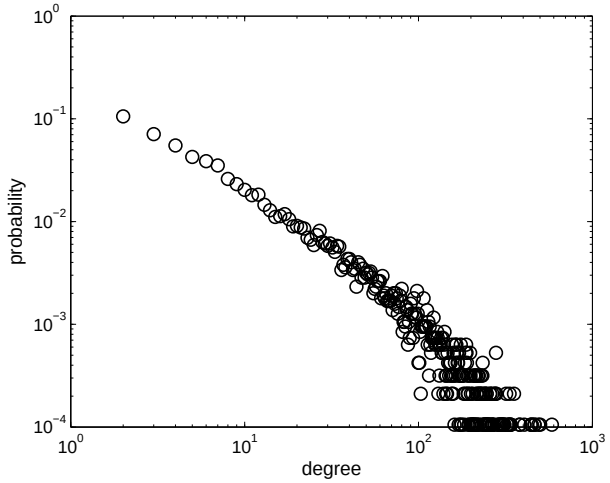

Fig. 2. The degree distribution for the Wikipedia network.

graph regarding those undirected links as bidirectional ones. We refer to this network data as the Wikipedia network. Here, the total numbers of nodes and directed links were $9,481$ and $245,044$, respectively. The network also showed the power-law degree distribution (see, Figure 2).

Newman and Park [2003] observed that social networks represented as undirected graphs generally have the following two statistical properties that are different from non-social networks. First, they show positive correlations between the degrees of adjacent nodes. Second, they have much higher values of the *clustering coefficient* CC than the corresponding *configuration models* (i.e., random network models).

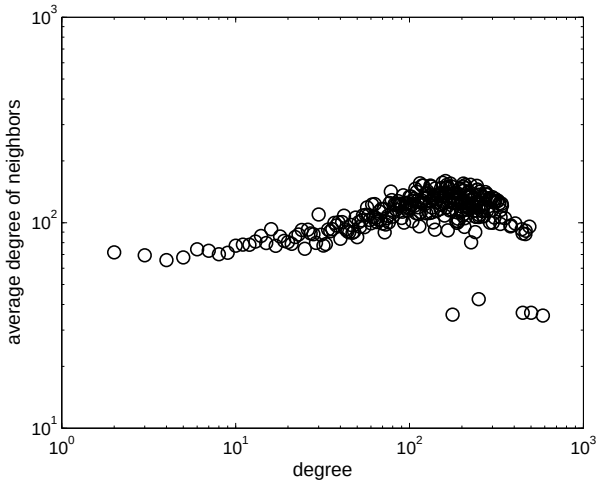

Fig. 3. The degree correlation for the Wikipedia network.

Here, the clustering coefficient CC for an undirected graph is defined by

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

where a "triangle" means a set of three nodes each of which is connected to each other, and a "connected triple" means a node connected directly to unordered other pair nodes. For the undirected graph of the Wikipedia network, the value of CC of the corresponding configuration model was 0.046, while the actual measured value of CC was 0.39. Namely, the undirected graph of the Wikipedia network had a much higher value of the clustering coefficient than the corresponding configuration model. Moreover, we can see from Figure 3 that the Wikipedia network had weakly positive degree correlation. Therefore, we believe that the Wikipedia network is a typical example of a large real social network represented by an undirected graph, and can be used as the network data to evaluate the performance of the proposed method.

5.2 Experimental Settings

For the bond percolation method, we need to specify the number M of performing the bond percolation process. It is reported [Kimura et al. 2007] that setting the value of M at several thousand is good enough for estimating influence degrees for the blog and Wikipedia networks. The following is the basis of assessing the value of M in the experiments in this paper. We estimated the average and the worst contamination degrees for the two networks with $M = 8,000$ and $M = 300,000$, where we assigned a uniform probability p to each propagation probability p_e for the IC model (how the value of p is determined for each network is described in detail in the next paragraph). The difference in the estimated average contamination degree for $M = 8,000$ and $M = 300,000$ was about 0.01% for the blog network and 0.02% for the Wikipedia network. Also, the corresponding difference in the estimated

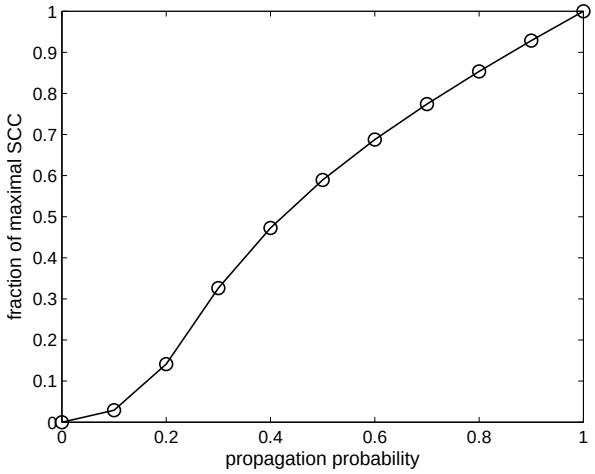

Fig. 4. Fragmentation of the blog network for the IC model. The fraction H of the maximal SCC as a function of the propagation probability p.

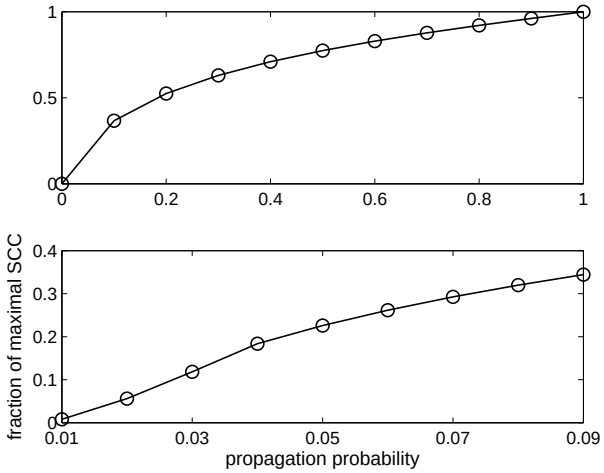

Fig. 5. Fragmentation of the Wikipedia network for the IC model. The fraction H of the maximal SCC as a function of the propagation probability p. The upper and lower frames show the network fragmentation curves for the whole range of p and the range of $0.01 \leq p \leq 0.09$, respectively.

worst contamination degree was about 0.02% for the blog network and 0.01% for the Wikipedia network. Thus, we concluded that the estimated contamination degrees for these networks with $M = 8,000$ are comparable to those with $M = 300,000$. By considering the assigned values of the propagation probabilities, we decided to use $M = 10,000$ through the experiments (i.e., $10,000\ (1 - 0.2) = 8,000$, see the next paragraph).

Because we assigned a uniform probability p to the propagation probability p_e for any directed link e of a network, the IC model had a single parameter p, and we determined the typical value of p for each of the blog and Wikipedia networks, and used them in the experiments. Let us consider the bond percolation process corresponding to the IC model with propagation probability p on a graph $G = (V, E)$. Let H be the expected fraction of the maximal SCC in the network constructed by occupied links. H is a function of p, and as the value of p decreases, the value of H decreases. In other words, as the value of p decreases, the original graph G gradually fragments into small clusters under the corresponding bond percolation process. Figures 4 and 5 show the network fragmentation curves for the blog and Wikipedia networks, respectively. Note that $H \to 1$ as $p \to 1$ since the blog and Wikipedia networks are strongly connected. Here, given the value of p, we estimated H as follows (see, Equation (6)):

$$H = \frac{1}{M|V|} \sum_{m=1}^{M} \max_{1 \le j \le J^m} |SCC(u_j^m; G^m)|,$$

where $M = 10,000$. We focus on the point p_* at which the average rate dH/dp of change of H attains the maximum, and regard it as the typical value of p for the network. Note that p_* is a critical point of dH/dp, and defines one of the features intrinsic to the network. From Figures 4 and 5, we estimated p_* to be $p_* = 0.2$ for the blog network and $p_* = 0.03$ for the Wikipedia network.

5.3 Comparison Methods

We compared the proposed method with three other heuristic methods. Two of them are based on the well-studied notions of betweenness and out-degree in the field of complex network theory and the other one is the crude baseline of blocking links randomly. We refer to these methods as *betweenness method, out-degree method* and *random method*, respectively.

5.3.1 *Betweenness Method.* The *betweenness score* $b_G(e)$ of a link e in a graph $G = (V, E)$ is defined as follows:

$$b_G(e) = \sum_{u,v \in V} \frac{n_G(e; u, v)}{N_G(u, v)},$$

where $N_G(u, v)$ denotes the number of the shortest paths from node u to node v on graph G, and $n_G(e; u, v)$ denotes the number of those paths that pass e. Here, we set $n_G(e; u, v)/N_G(u, v) = 0$ if $N_G(u, v) = 0$. Newman and Girvan [2004] successfully extracted community structure in a network using the following link-removal algorithm based on betweenness:

B1. Calculate betweenness scores for all links in the network.

B2. Find the link with the highest score and remove it from the network.

B3. Recalculate betweenness scores for all remaining links.

B4. Repeat from Step **B2**.

In particular, the notion of betweenness can be interpreted in terms of signals traveling through a network. If signals travel from source nodes to destination

nodes along the shortest paths in a network, and all nodes send signals at the same constant rate to all others, then the betweenness score of a link is a measure of the rate at which signals pass along the link. Thus, we naively expect that blocking the links with the highest betweenness score can be effective for preventing the spread of contamination in the network. Therefore, we apply the method of Newman and Girvan [2004] to the contamination minimization problems.

5.3.2 *Out-degree Methods.* Previous work has shown that simply removing nodes in order of decreasing *out-degrees* works well for preventing the spread of contamination in most real networks [Albert et al. 2000; Broder et al. 2000; Callaway et al. 2000; Newman et al. 2002]. Here, the out-degree $d(v)$ of a node v means the number of outgoing links from the node v. Therefore, as a comparison method, we consider the straightforward application of this node removal method. Namely, we employ the method of choosing nodes in decreasing order of out-degree and blocking simultaneously all the links attached to the chosen nodes. We refer to this method as the *node out-degree method*. Note that the node out-degree method cannot be applied for all values of positive integer K ($\leq |E|$) to the contamination minimization problems of blocking K links.

We also consider the method of blocking links between nodes with high out-degrees as an alternative comparison method. We define the link out-degree $\overline{d}(e)$ of a link $e = (u, v)$ from node u to node v by

$$\overline{d}(e) \;=\; d(u)\, d(v),$$

and recursively block links in decreasing order of link out-degree. We refer to this method as the *link out-degree method*.

5.4 Experimental Results

We evaluated the performance of the proposed method and compared it with that of the betweenness, the node out-degree, the link out-degree and the random methods. Clearly, the performance can be evaluated by the average contamination degree c_0 and the worst contamination degree c_+. We estimated these values by using the bond percolation method with $M = 300,000$, that is,

$$c_0(G_K) \;=\; \frac{1}{|V|} \sum_{v \in V} s(v; G_K, M),$$
$$c_+(G_K) \;=\; \max_{v \in V} s(v; G_K, M),$$

(see, Equation (4)), where $M = 300,000$. Note that this evaluation is done separately from the approximation used to search for the link to be deleted, i.e., Equation (11).

5.4.1 *Average Contamination Minimization Problem.* Figures 6 and 7 show the average contamination degree c_0 as a function of the number K of links blocked for the blog network and Figures 8 and 9 show the corresponding results for the Wikipedia network. In these figures the circles, squares, diamonds, triangles and crosses indicate the results for the proposed, the betweenness, the node out-degree, the link out-degree and the random methods, respectively. For each dataset, there are two figures, one comparing the proposed method with the betweenness method

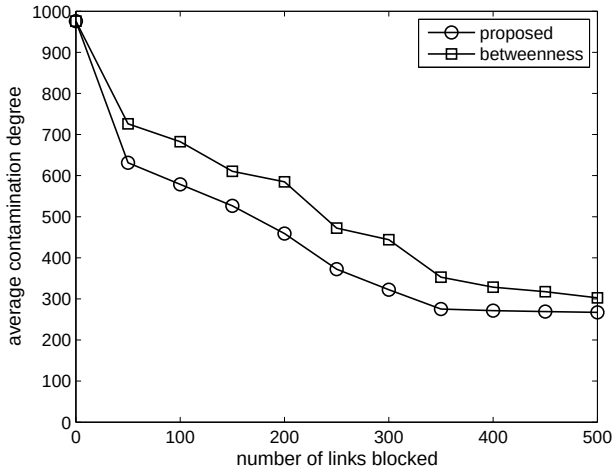

Fig. 6. Performance comparison between the proposed and the betweenness methods in the blog network for the average contamination minimization problem.

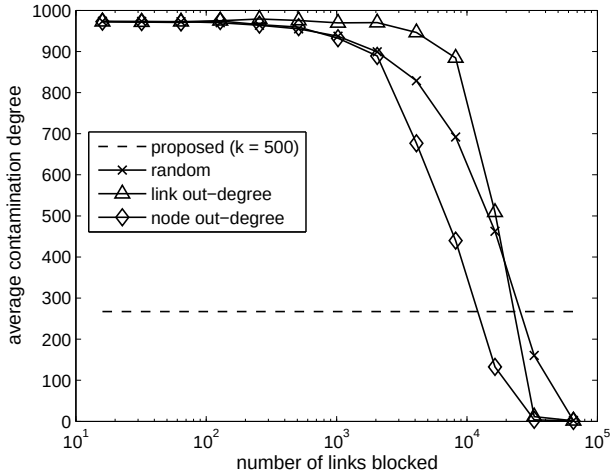

Fig. 7. Performance comparison of the proposed method at $K = 500$ with the node out-degree, the link out-degree and the random methods in the blog network for the average contamination minimization problem.

and the other comparing the proposed method at a fixed value of $K = 500$ with the node out-degree, the link out-degree and the random methods.

First, note that the average contamination degree c_0 at $K = 0$ is 976 for the blog network and 403 for the Wikipedia network, which is 8.2% and 4.2% of the total nodes, respectively. The average contamination degree as defined by Equation (1) is less than 10%. The fact that this value for Wikipedia network is about half of

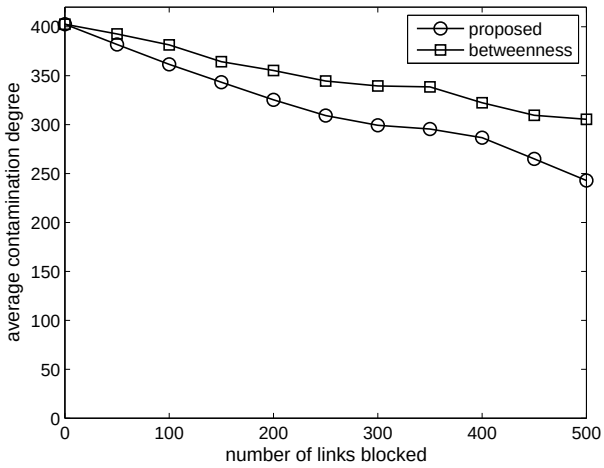

Fig. 8. Performance comparison between the proposed and the betweenness methods in the Wikipedia network for the average contamination minimization problem.

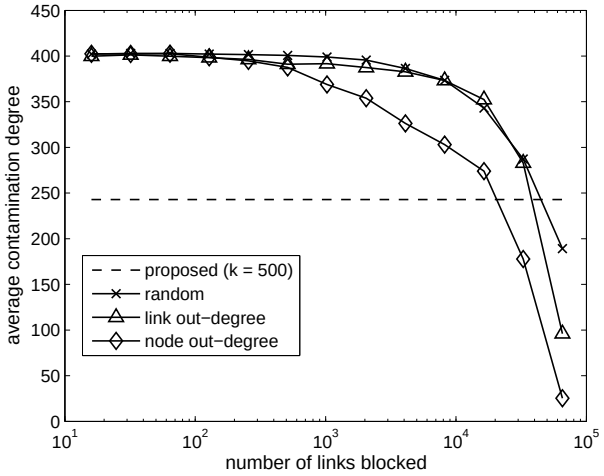

Fig. 9. Performance comparison of the proposed method at $K = 500$ with the node out-degree, the link out-degree and the random methods in the Wikipedia network for the average contamination minimization problem.

that of the blog network is explained by the smaller value of p for the Wikipedia network with the difference in network sizes considered. As expected the proposed method performs the best and the betweenness method follows. The other three methods are much worse than these two in the networks used.

The number of links blocked: $K = 500$ corresponds to 0.63% of the total links for the blog network and 0.2% for the Wikipedia network. Inversely, 0.2% of the total

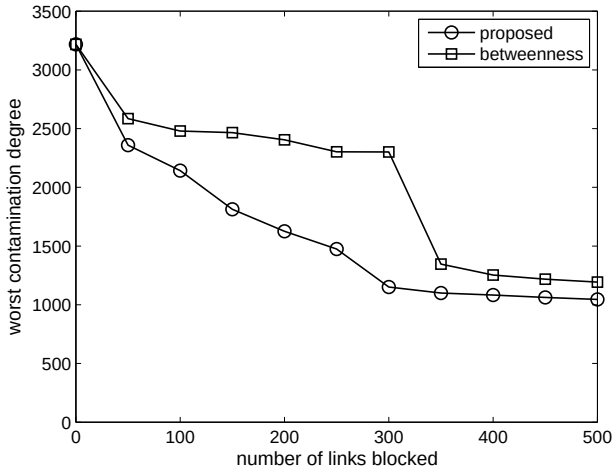

Fig. 10. Performance comparison between the proposed and betweenness methods in the blog network for the worst contamination minimization problem.

links corresponds to 163 links for the blog network. The average contamination degree at 0.2% link block, i.e., $K = 163$ for the blog network and $K = 500$ for the Wikipedia network is 495 and 243 for the proposed method, which is equivalent to 49% and 40% reduction in the degree, respectively, and 607 and 306 for the betweenness method, which is equivalent to 38% and 24% reduction in the degree, respectively. The difference between the two methods is 11% for the blog network and 16% for the Wikipedia network, respectively. The average contamination degree at 0.63% link block for the blog network, i.e., $K = 500$ is 267 for the proposed method and 303 for the betweenness method, which is equivalent to 73% and 69% reduction in the degree, respectively, and the difference between the two methods is 4%.

Differently from the above, the proposed method as well as the betweenness method outperform by far the other three methods (the node out-degree, the link out-degree and the random) for both the blog and the Wikipedia networks. Blocking 500 links by the proposed methods is equivalent to blocking more than 10,000 links for the blog network and 20,000 links for the Wikipedia network by the other three methods, meaning that the proposed method is 20 to 40 times more effective.

5.4.2 *Worst Contamination Minimization Problem.* Figures 10 and 11 show the worst contamination degree c_+ as a function of the number K of links blocked for the blog network, and Figures 12 and 13 show the corresponding results for the Wikipedia network. The meaning of the symbols in captions and the layout of the figures are the same as before.

First note that the worst contamination degree c_+ at $K = 0$ is 3218 for the blog network and 1929 for the Wikipedia network, which is 27% and 20% respectively. They are about 3 and 5 times larger than the average contamination degrees. The difference of the values between the two networks is consistent with the average

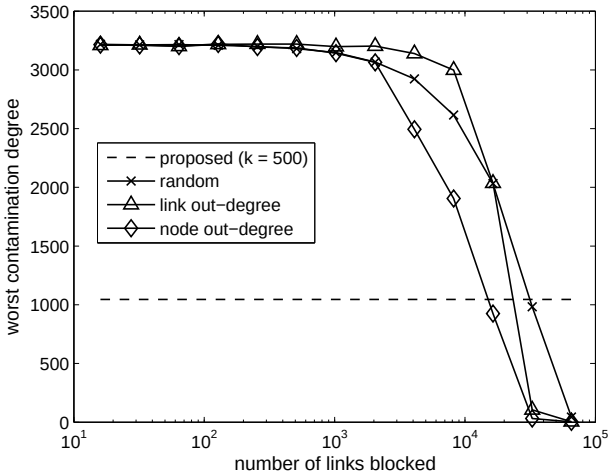

Fig. 11. Performance comparison of the proposed method for $K = 500$ with the node out-degree, link out-degree and random methods in the blog network for the worst contamination minimization problem.

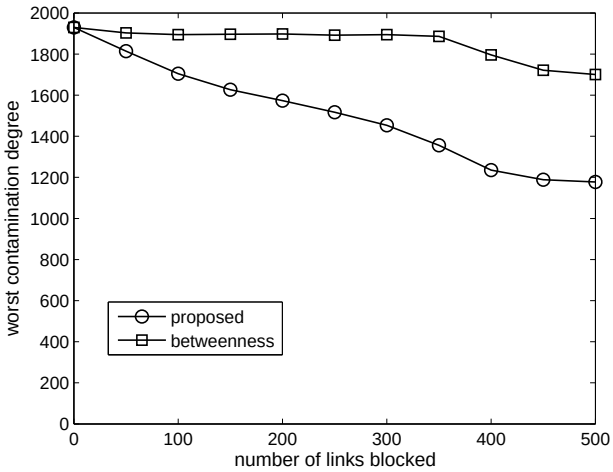

Fig. 12. Performance comparison between the proposed and betweenness methods in the Wikipedia network for the worst contamination minimization problem.

contamination case. The overall performance difference among the four methods is also consistent with the average contamination case.

The worst contamination degree at 0.2% link block, i.e., $K = 163$ for the blog network and $K = 500$ for the Wikipedia is 1763 and 1177 for the proposed method, which is equivalent to 45% and 39% reduction in the degree, respectively, and 2455 and 1700 for the betweenness method, which is equivalent to 24% and 12%

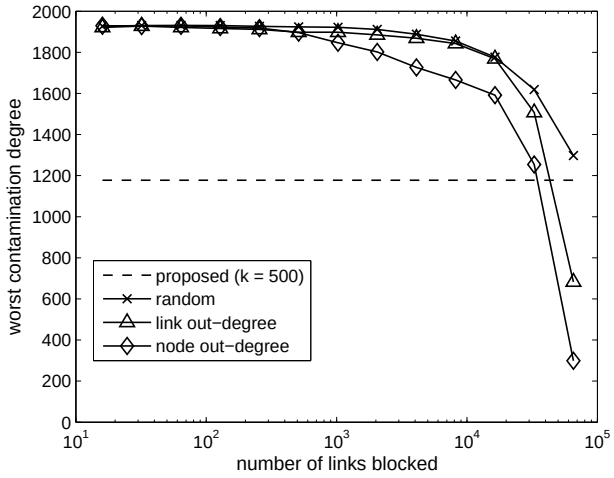

Fig. 13. Performance comparison of the proposed method for $K = 500$ with the node out-degree, link out-degree and random methods in the Wikipedia network for the worst contamination minimization problem.

reduction in the degree, respectively. The difference between the two methods is 21% for the blog network and 27% for the Wikipedia network, respectively. The worst contamination degree at 0.63% link block for the blog network, i.e., $K = 500$ is 1045 for the proposed method and 1193 for the betweenness method, which is equivalent to 78% and 63% reduction in the degree, respectively, and the difference between the two methods is 15%.

Again differently from the above, the proposed method as well as the betweenness method outperform by far the other three methods (the node out-degree, the link out-degree and the random) for both the blog and the Wikipedia networks. Blocking 500 links by the proposed method is equivalent to blocking more than 10,000 links for the blog network and 30,000 links by the other three methods, meaning that the proposed method is 20 to 60 times more effective.

5.4.3 *Discussion.* These results imply that the proposed method works effectively as expected, and outperforms the conventional link-removal heuristics. There is no big difference in the comparative performance results between the two networks. For both of them, the betweenness method performs reasonably well but the other three methods (the node out-degree, the link out-degree and the random) perform very poorly. There is no out-degree myth observed.

Of course how each of the conventional link-removal heuristics performs depends on the characteristics of the network structure. In general a network consists of multiple communities, and the members of each community are tightly connected and the members of different communities are less tightly connected. Thus, it is reasonable to assume that blocking the links between the different communities is effective in suppressing the contaminant to diffuse from one community to others. This is particularly true when there is a small number of nodes that play a key

role of connecting different communities. Blocking these small number of paths is quite effective. The fact that the betweenness method performed reasonably well implies that the networks we analyzed may have this type of community structure. On the other hand, if the network is hierarchically structured, blocking the nodes, equivalently blocking the links attached to them, in the upper hierarchy should be quite effective. The fact that the node out-degree method does not do well suggests that there may not be such a structure in the networks we analyzed. Among the poorly performing three methods, the link out-degree method performs most poorly. It performs worse than the random methods for the blog network. This would indicate that it is mainly blocking the links within the communities.

With all these different factors affecting the performance of each method taken, the proposed method exhibits its strength of explicitly minimizing the contamination by considering the dynamics of information diffusion process, thereby making its performance less sensitive to the structure of the network.

Considering the fact that all the methods can eventually block the contamination when all of the links are blocked, it is important to have a method which is effective when the number of links to be blocked is limited to be small, and the proposed method has this property. It is noticeable that blocking only 0.2% of the links by the proposed method can reduce the contamination by nearly 50%.

We have devised two measures: the average contamination degree and the worst contamination degree. It is expected that the performance difference between the proposed method and the betweenness method is larger for the latter than the former, and the results is consistent. Our formulation does not assume the origins of contamination to be known and fixed. If they are known in advance, the problem is much easier computationally.

6. FUTURE WORK

We note that the analysis we showed in this paper is the simplest case where p_e takes a single value for all the links e in E. In a more realistic setting we can divide E into subsets $E_1, E_2, ..., E_N$ and assign a different value p_n for all the links in each E_n. For example, we may divide the nodes into two groups: those that strongly influence others and those not, or we may divide the nodes into another two groups: those that are easily influenced by others and those not. We can further divide the nodes into multiple groups. From a different angle, when we deal with online social interactions that are represented as a latent social network, it is possible to assign an adequate probability to each link according to the communication density between the nodes.

Especially, when using multiple link probabilities, we need to carefully check that some probabilities are not too large. In fact, when a link probability p_e is close to 1, $|\mathcal{B}_M(e)|$ becomes a quite small number. Here recall that $\mathcal{B}_M(e)$ represents the subset of the M trials for the bond percolation process on graph G such that e is not an occupied link. Thus for such a case, the proposed method sometimes produces unreliable result. One simple remedy to this case is to adopt a substantially large trial number M.

As mentioned earlier, we consider that one important research direction is to explore the relationships between information diffusion processes and community

structure. To this end, we need to perform empirical evaluation by using a wide range of both real and synthetic networks with elaborated diffusion probability settings to information diffusion models. Other research direction includes applications of our proposed method to more practical problems, such as preventing the spread of computer virus. We believe that the proposed method greatly contributes to efficiently performing such experiments.

7. CONCLUSION

Just as good things, e.g., innovation, important topics, etc. spread through a network and bring positive affects to people, undesirable things, e.g., computer virus, malicious rumors, etc. also spread and affect people badly. We addressed the problem of minimizing the spread of undesirable things by blocking links in a social network, which is converse to the influence maximization problem for the same network. In particular, we have considered two contamination minimization problems, one minimizing the average contamination degree and the other minimizing the worst (maximum) contamination degree. We chose to block "links" rather than "nodes" because deleting nodes necessitates deleting links, but not vice versa.

We have proposed novel methods for efficiently finding good approximate solutions to these problems on the basis of a naturally greedy algorithm and the bond percolation method. Using large-scale blog and Wikipedia networks, we have experimentally demonstrated that the proposed method works effectively, and also outperforms the conventional link-removal heuristics. The betweenness method performed reasonably well but the out-degree methods performed very poorly almost as badly as the random method. No out-degree myth was observed for the networks we analyzed. The performance of the link-removal heuristics is strongly affected by the network structure, but the proposed method shows that it is important to explicitly minimize the contamination by considering the dynamics of information diffusion process, which would make the performance less sensitive to the structure of the network.

REFERENCES

Albert, R., Jeong, H., and Barabási, A.-L. 2000. Error and attack tolerance of complex networks. *Nature 406*, 378–382.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J. 2000. Graph structure in the web. In *Proceedings of the 9th International World Wide Web Conference*. Elsevier, Amsterdam, Netherland, 309–320.

Callaway, D. S., Newman, M. E. J., Strogatz, S. H., and Watts, D. J. 2000. Network robustness and fragility: Percolation on random graphs. *Physical Reveiw Letters 85*, 5468–5471.

Domingos, P. and Richardson, M. 2001. Mining the network value of customers. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, 57–66.

Gruhl, D., Guha, R., Liben-Nowell, D., and Tomkins, A. 2004. Information diffusion through blogspace. In *Proceedings of the 13th International World Wide Web Conference*. ACM, New York, 107–117.

Kempe, D., Kleinberg, J., and Tardos, E. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, 137–146.

Kimura, M., Saito, K., and Motoda, H. 2008. Minimizing the spread of contamination by blocking links in a network. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*. AAAI, Menlo Park, CA, 1175–1180.

Kimura, M., Saito, K., and Nakano, R. 2007. Extracting influential nodes for information diffusion on a social network. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*. AAAI, Menlo Park, CA, 1371–1376.

Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., and Glance, N. 2007. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, 420–429.

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

Newman, M. E. J., Forrest, S., and Balthrop, J. 2002. Email networks and the spread of computer viruses. *Physical Review E 66*, 035101.

Newman, M. E. J. and Girvan, M. 2004. Finding and evaluating community structure in networks. *Physical Review E 69*, 026113.

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

Richardson, M. and Domingos, P. 2002. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, 61–70.

Received September 2008; December 2008