Efficient Estimation of Influence Functions for SIS Model on Social Networks[*]

Masahiro Kimura
Department of Electronics and Informatics
Ryukoku University
kimura@rins.ryukoku.ac.jp

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

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

Abstract

We address the problem of efficiently estimating the influence function of initially activated nodes in a social network under the *susceptible/infected/susceptible (SIS) model*, a diffusion model where nodes are allowed to be activated multiple times. The computational complexity drastically increases because of this multiple activation property. We solve this problem by constructing a layered graph from the original social network with each layer added on top as the time proceeds, and applying the bond percolation with a pruning strategy. We show that the computational complexity of the proposed method is much smaller than the conventional naive probabilistic simulation method by a theoretical analysis and confirm this by applying the proposed method to two real world networks.

1 Introduction

Social networks mediate the spread of various information including topics, ideas and even (computer) viruses. The proliferation of emails, blogs and social networking services (SNS) in the World Wide Web accelerates the creation of large social networks. Therefore, substantial attention has recently been directed to investigating information diffusion phenomena in social networks [Adar and Adamic, 2005; Leskovec *et al.*, 2007b; Agarwal and Liu, 2008].

Overall, finding influential nodes is one of the most central problems in social network analysis. Thus, developing methods to do this on the basis of information diffusion is an important research issue. Widely-used fundamental probabilistic models of information diffusion are the *independent cascade (IC) model* and the *linear threshold (LT) model* [Kempe *et al.*, 2003; Gruhl *et al.*, 2004]. 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]. 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. 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.*, 2009].

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 nor be infected. This implies that an individual is never infected with the disease multiple times. This property holds true for the LT model as well. However, there exist phenomena for which the property does not hold. For example, consider the following propagation phenomenon of a topic in the blogosphere: A blogger who has not yet posted a message about the topic is interested in the topic by reading the blog of a friend, and posts a message about it (i.e., becoming infected). Next, the same blogger reads a new message about the topic posted by some other friend, and may post a message (i.e., becoming infected) again. Most simply, this phenomenon can be modeled by an *susceptible/infected/susceptible (SIS) model* from the epidemiology. Like this example, there are many examples of information diffusion phenomena for which the SIS model is more appropriate, including the growth of hyper-link posts among bloggers [Leskovec *et al.*, 2007b], the spread of computer viruses without permanent virus-checking programs, and epidemic disease such as tuberculosis and gonorrhea [Newman, 2003]. In this paper, we focus on an information diffusion process in a social network over a given time span on the basis of an SIS model.

Here, the SIS model is a stochastic process model, and the *influence* of a node v at time-step t, $\sigma(v,t)$, is defined as the expected number of infected nodes at time-step t when v is

[*] 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 Grant-in-Aid for Scientific Research (C) (No. 20500147).

initially infected at time-step $t = 0$. We refer to σ as the *influence function* for the SIS model. Developing an effective method for estimating σ is vital for various applications. Clearly, in order to extract influential nodes, we must estimate the value of $\sigma(v, t)$ for every node v and time-step t. Moreover, note that the method developed can be easily extended and applied to approximately solving the influence maximization problem for the SIS model by the greedy alogrithm. We can naively estimate σ by simulating the SIS model. However, this naive method is overly inefficient and not practical at all as shown in the experiments. In this paper, we propose a method for estimating influence function σ efficiently. By theoretically comparing computational complexity with the naive method, we show that the proposed method is expected to achieve a large reduction in computational cost. Further, using two large real networks, we experimentally demonstrate that the proposed method is much more efficient than the naive method with the same accuracy.

2 Information Diffusion Model

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

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

2.1 SIS Model

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

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

2.2 Influence Function

For the SIS model on G, we consider a diffusion sample from an initial active node $v \in V$ over time span T. Let $S(v, t)$ denote the set of active nodes at time-step t. Note that $S(v, t)$ is a random subset of V and $S(v, 0) = \{v\}$. Let $\sigma(v, t)$ denote the expected number of $|S(v, t)|$, where $|X|$ stands for the number of elements in a set X. We call $\sigma(v, t)$ the *influence* of node v at time-step t. Note that σ is a function defined on $V \times \{0, 1, \cdots, T\}$. We call the function σ the *influence function* for the SIS model over time span T on network G.

It is important to estimate the influence function σ efficiently. 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 $v \in V$, the diffusion process of the SIS model is simulated from the initial active node v, and the number of active nodes at time-step t, $|S(v, t)|$, is calculated for every $t \in \{0, 1, \cdots, T\}$. Then, $\sigma(v, t)$ is estimated as the empirical mean of $|S(v, t)|$'s that are obtained from M such simulations. We refer to this estimation method as the *naive method*. As shown in the experiments, the naive method is extremely inefficient, and cannot be practical.

3 Proposed Method

We propose a method for efficiently estimating the influence function σ over time span T for the SIS model on network G.

3.1 Layered Graph

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

Then, we can easily prove that the SIS model with propagation probabilities $\{p_e; e \in E\}$ on G over time span T is equivalent to the *bond percolation process (BP) with occupation probabilities* $\{q_e; e \in E^T\}$ on G^T.[1] 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 $v \in V$, let $F(v; \tilde{G}^T)$ be the set of all nodes that can be reached from v (= 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, $F(v; \tilde{G}^T) \cap V_t$ represents the set of active nodes at time-step t, $S(v, t)$.

[1] 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 [Newman, 2003; Kempe *et al.*, 2003; Kimura *et al.*, 2007].

3.2 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 $\sigma(v,t)$ for all $v \in V$. Moreover, the BP method does not fully perform the BP process, but performs it partially. Note first that all the paths from a node $v \in V$ on the graph $\tilde{G}^T$ represent a diffusion sample from the initial active node v for the SIS model on G. Let L' be the set of the links in G^T that is not in the diffusion sample. For calculating $|S(v,t)|$, 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:

1. Set $\sigma(v,t) \leftarrow 0$ for each $v \in V$ and $t \in \{1,\cdots,T\}$.

2. Repeat the following procedure M times:

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

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

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

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.

2-2c. For each $v \in A(t-1)$, compute $S(v,t) = \bigcup_{w \in S(v,t-1)} \Gamma(w;\tilde{G}_t)$, and set $\sigma(v,t) \leftarrow \sigma(v,t) + |S(v,t)|$ and $A(t) \leftarrow A(t) \cup \{v\}$ if $S(v,t) \neq \emptyset$.

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

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, $A(t) = \{v \in V;\ S(v,t) \neq \emptyset\}$. 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, $B(t-1) = \bigcup_{v \in V} S(v,t-1)$.

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 2-2a, the number of the coin-flips, $\mathcal{N}_b$, for the BP process in step 2-2b, and the number of the links, $\mathcal{N}_c$, that are followed in step 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

$$\mathcal{N}_a = \sum_{v \in A(t-1)} |S(v,t-1)|, \quad \mathcal{N}_b = \sum_{w \in B(t-1)} d(w), \quad (1)$$

and

$$\mathcal{N}_c = \sum_{v \in A(t-1)} \sum_{w \in S(v,t-1)} d'(w) \quad (2)$$

on average.

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

A method that is equivalent to the naive method:

1. Set $\sigma(v,t) \leftarrow 0$ for each $v \in V$ and $t \in \{1,\cdots,T\}$.

2. Repeat the following procedure M times:

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

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

2-2b'. For each $v \in A(t-1)$, perform the BP process for the links from $S(v,t-1)$ in G^T, and generate the graph $\tilde{G}_t(v)$ constructed by the occupied links.

2-2c'. For each $v \in A(t-1)$, compute $S(v;t) = \bigcup_{w \in S(v,t-1)} \Gamma(w;\tilde{G}_t(v))$, and set $\sigma(v,t) \leftarrow \sigma(v,t) + |S(v,t)|$ and $A(t) \leftarrow A(t) \cup \{v\}$ if $S(v,t) \neq \emptyset$.

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

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

$$\mathcal{N}_{b'} = \sum_{v \in A(t-1)} \sum_{w \in S(v,t-1)} d(w), \quad (3)$$

and the number of the links, $\mathcal{N}_{c'}$, followed in step 2-2c' is equal to $\mathcal{N}_c$ in the BP method on average. From equations (2) and (3), 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. By equations (1) and (3), 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 $\sum_{v \in A(t-1)} |S(v,t-1)| \gg |\bigcup_{v \in A(t-1)} S(v,t-1)|$ $(= |B(t-1)|)$. Therefore, the BP method is expected to achieve a large reduction in computational cost.

3.3 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 $S(u,t_0) = S(v,t_0)$ 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 $S(u,t) = S(v,t)$ for all $t > t_0$. The BP with pruning method estimates σ by the following algorithm:

BP with pruning method:

1. Set $\sigma(v,t) \leftarrow 0$ for each $v \in V$ and $t \in \{1,\cdots,T\}$.

2. Repeat the following procedure M times.

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

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

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

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.

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

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 $S(u,t) = S(v,t)$.

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

Basically, by introducing step 2-2d and reducing the size of $A(t)$, the proposed method attempts to improve the computational efficiency in comparison to the original BP method.

For the proposed method, it is important to implement efficiently the equivalence check process in step 2-2d. In our implementation, we first classify each $v \in A(t)$ according to the value of $k = |S(v,t)|$, and then perform the equivalence check process only for those nodes with the same k value. How effectively the proposed method works will depend on several conditions such as network structure, time span, values of diffusion probabilities, and so on. We will do a simple analysis later and experimentally show that it is indeed effective.

4 Experimental Evaluation

4.1 Network Data and Settings

In our experiments, we employed two datasets of large real networks used in [Kimura *et al.*, 2009], which exhibit many of the key features of social networks.

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

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

For the SIS model, we assigned a uniform probability p to the propagation probability $p_{u,v}$ for any link $(u,v) \in E$, that is, $p_{u,v} = p$. According to [Kempe *et al.*, 2003; 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 $1/\bar{d}$, where $\bar{d}$ is the mean out-degree of a network. Since the values of $\bar{d}$ were about 6.63 and 25.85 for the blog and the Wikipedia networks, respectively, the corresponding values of $1/\bar{d}$ were about 0.15 and 0.03. We decided to set $p = 0.1$ for the blog network and $p = 0.01$ for the Wikipedia network.

All our experimentation was undertaken on a single PC with an Intel Core 2 Duo E6850 3GHz processor, with 3GB of memory, running under Linux.

4.2 Estimation Accuracy Comparison

We first compared the accuracy of the estimated influence function σ of the proposed method (BP with pruning) with that of the naive method. Both methods require M to be specified in advance as a parameter. As shown in section 3.2, 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 the both methods. Thus by using the same value of M, both would estimate $\sigma(v,t)$ with the same accuracy in principle.

Table 1: Results for the naive method on the blog network.

Rank	Node ID	Influence	Node ID	Influence
1	2210	984.38	2210	985.74
2	2248	979.59	2248	980.72
3	3906	956.82	3906	956.57
4	3907	953.14	3907	953.89
5	146	931.03	146	931.62
6	155	929.68	155	930.21
7	3233	913.50	3233	911.89
8	3228	912.27	3228	910.52
9	140	910.04	140	910.37
10	2247	909.59	2247	910.00

Table 2: Results for the proposed method on the blog network.

Rank	Node ID	Influence	Node ID	Influence
1	2210	984.74	2210	984.87
2	2248	980.41	2248	979.46
3	3906	956.97	3906	955.84
4	3907	953.04	3907	952.71
5	146	929.96	146	929.30
6	155	928.77	155	928.49
7	3233	912.61	3233	911.01
8	3228	912.18	3228	910.49
9	140	909.22	140	910.31
10	2247	909.12	2247	909.59

We have experimentally confirmed that use of $M = 100,000$ gives in effect the same value of $\sigma(v,t)$, for $t = 1, \cdots, 20$. The following accuracy comparison is based on $M = 100,000$. Tables 1 and 2 show the ranking of the influential initially activated nodes v evaluated at time-step $T = 20$ for the blog network. 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 several times and listed two of them. Note that the naive method takes an order of week to return the result and we could not set T a

Table 3: Results for the naive method on the Wikipedia network.

Rank	Node ID	Influence	Node ID	Influence
1	4019	134.73	4019	133.83
2	3729	133.24	3729	132.42
3	7919	132.66	7919	131.98
4	4380	132.23	1720	131.68
5	1720	132.20	4380	131.34
6	4465	132.10	4465	131.07
7	1712	131.65	1712	130.69
8	3670	130.32	1073	129.48
9	1073	129.66	3670	129.46
10	1191	128.61	1191	128.38

Table 4: Results for the proposed method on the Wikipedia network.

Rank	Node ID	Influence	Node ID	Influence
1	4019	134.25	4019	133.67
2	3729	132.91	7919	132.17
3	7919	132.50	3729	132.02
4	4380	132.03	4380	131.84
5	4465	131.95	1720	131.63
6	1720	131.59	4465	131.12
7	1712	131.33	1712	130.90
8	3670	130.27	3670	129.78
9	1073	129.22	1073	129.12
10	1191	128.71	1191	128.40

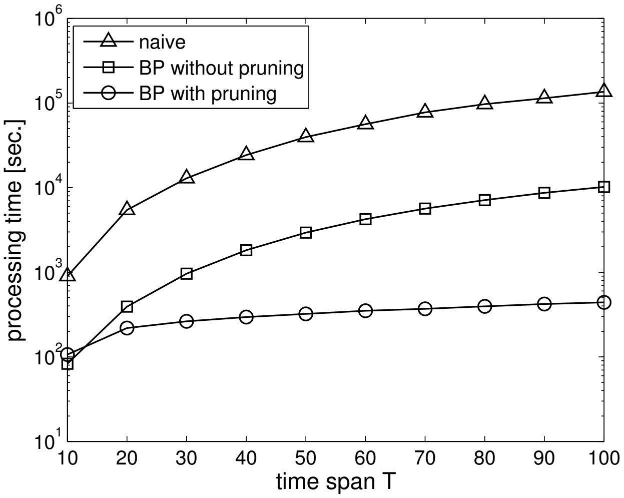

Figure 1: Results for the blog network.

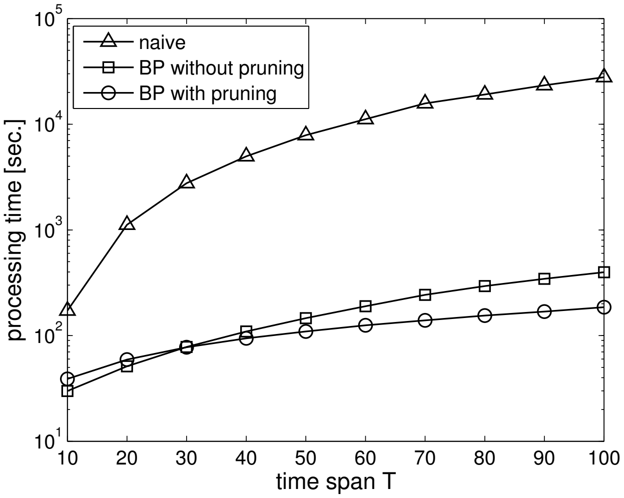

Figure 2: Results for the Wikipedia network.

larger value. We note that the ranking is exactly the same for the both methods. Tables 3 and 4 are the result for the Wikipedia network. The nodes in the 4th and the 5th ranks for the naive method, and the 5th and the 6th ranks for the proposed method are interchanged respectively, but the rests are the same. From these results we confirm that the proposed method gives the same results as the naive method with the same value of M when M is large enough.

4.3 Processing Time Comparison

Next, we compared the processing time of the proposed method (BP with pruning) with the BP method without pruning 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 1 and 2 show the total processing time to estimate $\{\sigma(v,t); v \in V, t = 0, 1, \cdots, T\}$ as a function of time span T for the blog and the Wikipedia networks, respectively. In these figures, the circles, squares and triangles indicate the results for the proposed method (BP with pruning), the BP method without pruning, and the naive method, respectively. Note that in case of the blog network, the processing time for time span $T = 100$ is about 7 minutues, 2 hours and 37 hours for the proposed method, the BP method without pruning and the naive method, respectively. Namely, the proposed method is about 20 and 310 times faster than the BP method without pruning and the naive method, respectively, for $T = 100$ in case of the blog network. Note also that in case of the Wikipedia network, the processing time for time span $T = 100$ is about 3 minutes, 6 minutes and 8 hours for the proposed method, the BP method without pruning and the naive method, respectively. Namely, the proposed method is about 2 and 150 times faster than the BP method without pruning and the naive method, respectively, for $T = 100$ in case of the Wikipedia network.

In general, the proposed method performs the best and the BP method without pruning follows with an exception that the proposed method can become slightly slower than the BP method without pruning in cases where T is small because of the overhead introduced in pruning. The two BP methods (with and without pruning) are much faster than the naive method. The performance difference between the proposed method and each of the other two methods 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 average for time-step $t+1$ (see equations (1), (2) and (3)). 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 the BP with pruning method. Then, if $ad' > d$, i.e., $ad'/d = ap > 1$ holds, the improvement ratios of the BP with pruning method over the naive method and the original BP method are respectively $asd/sd = a$ and $asd'/sd = ap$. From our experimental results, we can estimate a to be 310 for the blog network and 150 for the Wikipedia network. Then we obtain ap to be 31 and 1.5 respectively, which approximates the actual ratio each, 20 and 2.

5 Discussion

Here, we compare the method proposed in [Kimura *et al.*, 2007] that efficiently estimates the influence function also in the framework of bond percolation for the IC and the LT models. The same method is not applicable to the SIS model. The key idea there is to decompose the graph that is generated by the bond percolation into a set of strongly connected components (SCC) and efficiently calculate the node reachability. However, the layered graph in the proposed method is a directed acyclic tree and the SCC decomposition would not work effectively. The pruning technique in the proposed method is a new technique to improve the computational efficiency for the SIS model, just like the SCC decomposition is for the IC and the LT models.

In this paper we did not directly address the influential maximization problem, but only proposed a new method to efficiently estimate the influence function. We can think of two maximization problems, that is to find the initial active nodes with a specified number that maximize 1) the expected number of nodes that have been activated till the end of time-step T and 2) the expected number of active nodes at the end of time-step T. The proposed method can easily be extended to efficiently estimate the marginal gain of the objective function of each of the optimization problems when the problems are to be solved by greedy algorithms.

6 Conclusion

Finding influential nodes is one of the most central problems in the field of social network analysis. There are several models that simulate how various things, e.g., news, rumors, diseases, innovation, ideas, etc. diffuse across the network. One such realistic model is the *susceptible/infected/susceptible (SIS) model*, an information diffusion model where nodes are allowed to be activated multiple times. The computational complexity drastically increases because of this multiple activation property, e.g., compared with the *susceptible/infected/recovered (SIR) model* where once activated nodes can never be deactivated/reactivated. We addressed the problem of efficiently estimating the influence function 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 v (for all $v \in V$) at time-step $t = 0$. We solved this problem by constructing a layered graph from the original social network by adding each layer on top of the existing layers as the time proceeds, and applying the bond percolation with a pruning strategy. We showed that the computational complexity of the proposed method is much smaller than the conventional naive probabilistic simulation method by a theoretical analysis. We further confirmed this by applying the proposed method to two real world networks taken from blog and Wikipedia data. Considerable reduction of computation time was achieved without degrading the accuracy.

References

[Adar and Adamic, 2005] E. Adar and L. A. Adamic. Tracking information epidemics in blogspace. In *WI'05*, pages 207–214, 2005.

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

[Domingos and Richardson, 2001] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD'01*, pages 57–66, 2001.

[Gruhl *et al.*, 2004] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *WWW'04*, pages 107–117, 2004.

[Kempe *et al.*, 2003] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD'03*, pages 137–146, 2003.

[Kimura *et al.*, 2007] M. Kimura, K. Saito, and R. Nakano. Extracting influential nodes for information diffusion on a social network. In *AAAI'07*, pages 1371–1376, 2007.

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

[Leskovec *et al.*, 2007a] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *KDD'07*, pages 420–429, 2007.

[Leskovec *et al.*, 2007b] J. Leskovec, M. McGlohon, C. Faloutsos, , N. Glance, and M. Hurst. Patterns of cascading behavior in large blog graphs. In *SDM'07*, pages 551–556, 2007.

[Newman, 2003] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[Richardson and Domingos, 2002] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD'02*, pages 61–70, 2002.