Probabilistic Solutions of Influence Propagation on Social Networks

Miao Zhang
University of Texas at Arlington
Texas, USA
miao.zhang@mavs.uta.edu

Chunni Dai
Shanghai Jianqiao College
Shanghai, China
daicn@126.com

Chris Ding
University of Texas at Arlington
Texas, USA
chqding@uta.edu

Enhong Chen
University of Science and Technology of China
Hefei, China
cheneh@ustc.edu.cn

ABSTRACT

Given fixed budgets, companies attempt to obtain maximum coverage on a social network by targeting at influential individuals. This viral marketing is often modeled by the *independent cascade* model. However, identifying the most influential people by computing influence spread is NP-hard, and various approximate algorithms are developed. In this paper, we emphasize the probabilistic nature of influence propagation. We propose to use exact probabilistic solutions and prove an inclusion-exclusion principle for computing influence spread. Our probabilistic solutions can significantly speed up the computation of influence spread. We also give a probabilistic-additive incremental search strategy to solve the influence maximization problem, i.e., to find a subset of individuals that has the largest influence spread in the end. Experiments on real data sets demonstrated the effectiveness and efficiency of our methods.

Categories and Subject Descriptors

F.2.2 [**Analysis and Algorithms and Problem Complexity**]: Non-numerical Algorithms and Problems

Keywords

social influence, viral marketing, inclusion-exclusion theorem, probabilistic additive

1. INTRODUCTION

Different from other marketing strategies, viral marketing targets at a certain number of consumers at the beginning, and relies on communications and trust between individuals within close social networks [10] [11] to enlarge the social influence. Nowadays, web 2.0 enables convenient communications among people within or between different social circles through online social networks, such as Twitter, Facebook, Linkedin, and so on. They provide a large and facilitating platform for viral marketing strategies. Information or innovations can propagate from a small number of individuals to a huge number of users of social networks in a short time.

Before companies can apply real viral marketing strategies on those online social networks, some challenges need to be addressed: (1) how to determine the edge weights between different users; (2) how to calculate the social influence given a set of activated nodes (seed set); (3) how to select the optimal seed set, which has the maximum social influence, i.e., the number of activated nodes in the end is the largest. This problem is defined as influence maximization problem in [7]. In this paper, we concentrate to solve the second and third challenges. To address the problem of how to calculate the social influence given each seed set, we first need to present a social influence model defining how the propagation proceeds under some circumstances. There are several influence models which have been proposed and studied, and the most popular ones are linear threshold model (LT) and independent cascade model (IC), which were presented by Kempe et al. in [7]. We study the influence propagation process under IC model in this paper. IC Model can be described as a stochastic process based on some probabilistic settings. For details, social network can be modeled as one graph $G(V, E)$ with edge weights P. IC model starts with an initial active seed node set; in the first step, those active nodes try to influence their inactivated out-bound neighbors with probability of the corresponding edge weights; each active node only has one chance to influence its each inactivated out-bound neighbor; in next step, the newly activated nodes continue to influence their own inactivated out-bound neighbors with one single chance to each neighbor; this process proceeds until no more inactivated nodes become activated.

Kempe et al[7] proved the influence maximization problem under IC model is NP-hard. and Wei Chen et al [2] proved that calculating the influence spread of a seed set under IC model is NP-hard too. Kempe et al. applied Monte Carlo simulation to approximate the influence spread, which is time-consuming, because Monte Carlo simulation needs to be run at least thousands of times to reach a good approximation of the true influence spread. Therefore, as the first challenge, proposing an efficient approximation method to

CIKM'13, Oct. 27–Nov. 1, 2013, San Francisco, CA, USA.

http://dx.doi.org/10.1145/2505515.2505718.

calculate the influence spread of a seed set is urgent. To that end, we present probabilistic solutions to calculate the influence spread under IC model and propose a probabilistic additive strategy to reuse the influence spread obtained in previous steps when calculating the optimal seed set using greedy methods. Our main contributions are:
(1) We analyze exact solutions of small networks; one key finding from these analysis is the inclusion-exclusion principle which we prove vigorously. We further propose exact probabilistic solutions to influence spread for Directed Acyclic Graph (DAG) under IC model, which can be generalized to approximate the generic case of IC model.
(2) We propose a probabilistic additive strategy to reuse the influence spread calculated in the previous steps when determining the optimal seed set using greedy methods, which speedup influence spread computation.
(3) We also propose an incremental search strategy to continue refining the seed set, which is first obtained by two greedy methods. After incremental search, the influence spread of the selected seed set is improved.

1.1 Related Work

In the past decades, there has been a lot of research work studying and analyzing the different aspects of social influence, we group these related work into three categories. The first category includes research work on influence models. The second category includes the related work on how to compute social influence spread. The third category focuses on solving the ultimate viral marketing goal - find a set of seed nodes those have the maximal social influence. Usually, the second and the third categories are solved together, but with different emphases.

For the first category, Domingos et al. [4] [12] first proposed to mine the customers' network value, and then based on customers' network value to solve the social influence maximization problem. Kempe et al. [7] first presented the two basic influence models - LT and IC model [6] [9]. Aggarwal et al. proposed a stochastic information flow model to determine the authoritative individuals in [1], which is closely related to IC model. Other aspects of influence models, such as the edge weights between individuals were also studied in [5]. Tang et al. [13] [14] proposed a Topical Factor Graph model to analyze social influence.

For the second and third categories, Kempe et al. [7] presented to use Monte Carlo Simulation to estimate the influence spread for a given seed set, and proposed a greedy method to find a good seed set, which is not scalable to large scale networks, because Monte Carlo Simulation needs to be run at least tens of thousand times to get a good estimation. Then many heuristic algorithms were introduced for the IC model. Kimura et al. proposed two influence cascade models based on shortest-path to approximate the influence spread of a seed set, and present algorithms to give good approximations to IC model for finding good seed sets [8]. Chen et al. proposed a heuristic algorithm using degree discount for a limited version of IC model, in which the edge wieghts/probabilities between any two connected individuals are the same in [3]. Chen et al. also proposed a maximum influence arborescence (MIA) heuristic model for the generic IC model in [2]. In MIA model, maximum influence paths (MIP) between every pair of two nodes need to be pre-computed, and then based on these MIPs, local MIA structures can be formed. The algorithms introduced above are mostly heuristic methods. We are trying to present an exact solution on how to compute the influence spread given a seed set under IC model, which is very challenging. In this paper we first analyze the exact probabilistic solution to small networks and then we propose an probabilistic solution for DAG under IC model, which can be applied on directed graph to approximate the influence spread. We also present a probabilistic additive strategy to speed up the influence spread calculation when using greedy method to select the most influential seed set.

2. INDEPENDENT CASCADE MODEL

In this section, we introduce the IC model.

A social network can be represented by a directed graph $G(V, E)$ with edge weight/probability P, i.e., $P(u, v)$ or P_{uv} in short denotes the propagation probability through edge $(u, v) \in E$ from node u to node v. The total number of nodes in G is $n = |V|$.

Given an activated seed set S, the independent cascade model works as follows, $S_0 = S$ is the activated node set at step 0, and S_t denotes the activated node set at step t. At step $t + 1$, every newly activated node u in S_t, i.e., $\{u|u \in S_t \setminus S_{t-1}\}$, is trying to influence its out-bound non-activated neighbors v, which don't belong to S_t, i.e. $\{v|(u \rightsquigarrow v) \cap (v \in V \setminus S_t)\}$ with an independent probability $P(u, v)$. The process stops when an equilibrium state is reached, i.e. there are no more nodes being activated in next propagation step. In independent cascade model, each node $\{u|u \in S_t(t \geq 0)\}$ can only influence its out-bound neighbors once right after it is activated, and the activated nodes will stay activated ever after. Now, we are ready to define the influence spread of activated seed set S, denoted by $\sigma(S)$, which is the number of activated nodes in the final step (in the stationary/equilibrium state).

3. EXACT INFLUENCE SPREAD FOR SMALL NETWORKS

As explained above, most current research on IC Model focused on developing efficient **approximate** algorithms to compute the influence propagation. Our focus here is to provide **exact** solutions. Due to the NP-hard complexity, we obtain exact solution on small networks. In this section, we give three small network examples to illustrate the exact influence propagation process. The three small networks are shown in Figure 1, where node 1 is the seed (shaded in green) in each case. We present the exact propagation solution for each network. These exact solutions can be extended to larger networks.

From these exact results we obtain two important benefits:

(1) We learn the rules of adding contributions from different path of influence propagation. At first glance, these contributions seems to be statistically *independent*. But the exact results show they may not be independent and why. This introduces the inclusion-exclusion principle we found useful in correctly enumerating contributions from different paths.

(2) The rules we learned in this process are helpful to formulate an exact computational algorithm in §4.

(3) Exact solutions obtained can be used to evaluate approximate algorithms in previous studies [3, 8, 1, 15]. This may lead to refined methods to further improve these existing approximate algorithms.

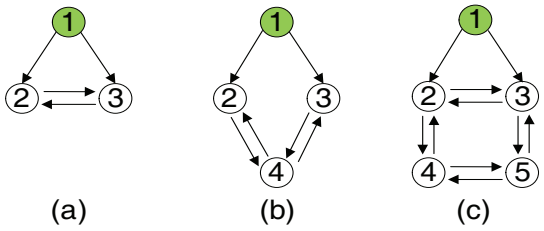


Figure 1: Small networks

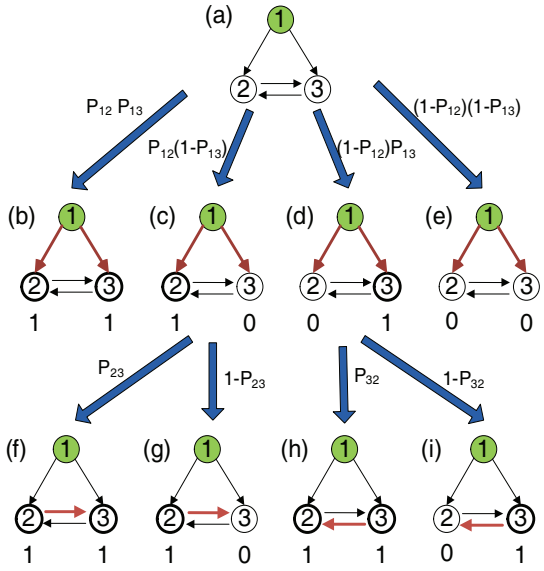


Figure 2: Different stages of influence propagation for a 3-node network in (a). Graphs (b),(c),(d),(e) are first stage results of seed node 1 attempting to influence nodes $\{2, 3\}$ with corresponding probabilities given. Thick red edges indicate the influence action. Thick circle means the node is successfully influenced, also indicated by a number 1 or 0 underneath. From graph (c), node 2 tries to influence node 3; results are given in (f),(g).

For those networks, we assume that transition probabilities on the edges already exist and remain fixed during the influence propagation.

3.1 Solution for 3-node network

The IC influence propagation process for the 3-node network in Figure 1(a) can be illustrated in Figure 2.

We start with Figure 2(a), where node 1 is a seed node and thus always activated. The four networks of Figure 2(b, c, d, e) are the four possibilities of node 1 attempts to activate nodes 2 and 3. The 4 probabilities are indicated next to the arrow. For example the case where nodes 2 and 3 are both been successfully activated is shown in Figure 2(b), with probability $P_{12}P_{13}$.

The cases in Figure 2(b) and 2(e) are terminal, i.e., there are no further possibilities. In 2(c), node 2 (been successfully activated by node 1) will attempt to activate node 3. The results are shown in Figure 2(f) and Figure 2(g) with the appropriate probabilities indicated next to the arrows. Similarly, in Figure 2(d), node 3 (been successfully activated by node 1 will attempt to activate node 2. The results are shown in Figure 2(h) and Figure 2(i) with the appropriate probabilities indicated next to the arrows

Now, we can compute the activation probabilities. Let's consider node 2. There are 3 cases where node 2 becomes activated:
(i) Figure 2(b) with probability $P_{12}P_{13}$.
(ii) Figure 2(c) with probability $P_{12}(1-P_{13})$. Note that this is equal to the sum of probabilities of Figure 2(f) and Figure 2(g).
(iii) Figure 2(h) with probability $(1-P_{12})P_{13}P_{32}$. This probability for the influence flow path to Figure 2(h) equals to the probability to reach Figure 2(d) multiplied by the probability to further reach Figure 2(h).
Therefore, by adding these 3 probabilities, the activation probability for node 2 is,

$$\pi_2 = P_{12} + P_{13}P_{32} - P_{12}P_{13}P_{32}. \quad (1)$$

Another way to compute π_2 is by directly counting influence flow paths. First, node 2 can be influenced by node 1 directly, with probability P_{12} (this is the sum of probabilities of Fig.2(b),(c)). Second, If node 1 fails to influence node 2, there is another path node 2 can be activated, which is illustrated by Figure 2(a) $\rightarrow$ 2(d) $\rightarrow$ 2(h). For this influence path the probability is $(1 - P_{12})P_{13}P_{32}$. Adding these two we get the same result in Eq.(1).

Inclusion-exclusion. The above two counting methods rely on the detailed influence propagation stages shown in Figure 2. The result of Eq.(1) can be obtained without relying on Figure 2. We compute probabilities of *different paths* together with an **inclusion-exclusion** principle. For node 2, there are two paths:
(i) $1 \rightarrow 2$, with probability $P_{1\rightarrow 2} = P_{12}$.
(ii)$1 \rightarrow 3 \rightarrow 2$, with probability $P_{1\rightarrow 3\rightarrow 2} = P_{13}P_{32}$.
These two events are not **independent** because in (ii) we did not include the factor $(1 - P_{12})$.

We use inclusion-exclusion principle to correct for overcounting, i.e., we set

$$\pi_2 = P_{1\rightarrow 2} + P_{1\rightarrow 3\rightarrow 2} - P_{1\rightarrow 2}P_{1\rightarrow 3\rightarrow 2}. \quad (2)$$

This gives the same result of Eq.(1).

For node 3, the probability can be calculated similarly,

$$\pi_3 = P_{13} + P_{12}P_{23} - P_{12}P_{13}P_{23}. \quad (3)$$

3.2 Solution for 4-node network

Let us look at a more complicated case — the 4-node network of Figure 1(b). The IC influence propagation process is illustrated in Figure 3. The settings in Figure 3 are the same as those of Figure 2.

We start with Figure 3(a), where node 1 is a seed node. The four networks of Figure 3(b, c, d, e) are the four possibilities of node 1 attempts to activate nodes 2 and 3. The 4 probabilities are indicated next to the arrow.

The cases in Figure 3(e, f, i, j) are terminal, i.e., there are no further possibilities. In Figure 3(b), node 2 and 3 (been successfully activated by node 1) will attempt to activate node 4. The successful result is shown in Figure 3(f) with the appropriate probabilities indicated next to the arrows, figures for failure results are not shown here. We will compute the activation probabilities by directly counting influence flow paths, so the failure results will reach irrelevant terminal cases. In Figure 3(c), node 2 (been successfully

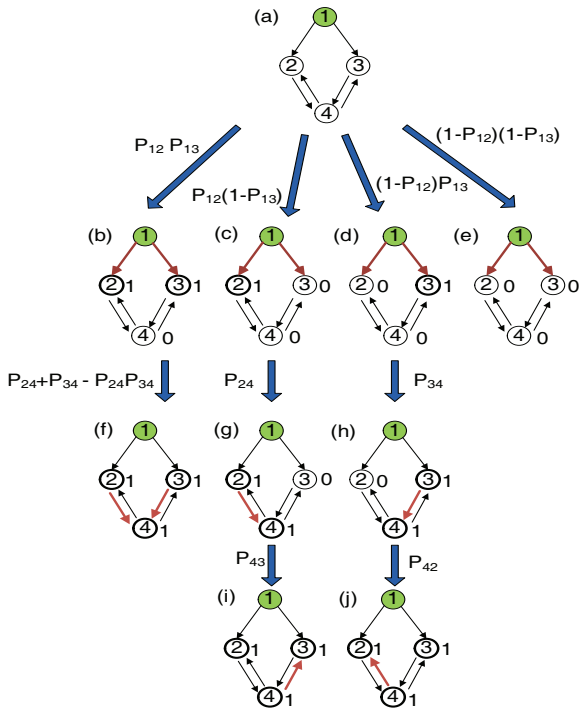


Figure 3: Different stages of influence propagation from a 4-node network of (a). Node 1 is the seed. Symbols are same as Figure 2. From graph (c), node 2 attempts to influence node 4. Only successful result graph (g) is shown. Results of all unsuccessful attempts are skipped.

activated by node 1) will attempt to activate node 4. The successful result is shown in Figure 3(g) with the appropriate probabilities indicated next to the arrows, and the failure result will reach an irrelevant terminal case, so we didn't show the figure here. In Figure 3(g), node 4 will attempt to activate node 3. The successful result is shown in Figure 3(i). Similarly, in Figure 3(d), node 3 (been successfully activated by node 1 will attempt to activate node 4. The successful result is shown in Figure 2(h) with the appropriate probabilities indicated next to the arrows, while the failure result is not shown here. In Figure 3(h), node 4 will attempt to activate node 2. The successful result is shown in Figure 3(j).

Now, we compute the activation probabilities. We compute π_2 by counting influence flow paths. For node 2, first, it can be influenced by node 1, with results given in Fig. 3(b, c). The corresponding probability is $\pi_2^{(1)} = P_{12}$.

If node 1 fails to influence node 2, there is another path node 2 can be activated, which is illustrated by Figure 3(a) $\rightarrow$ 3(d) $\rightarrow$ 3(h) $\rightarrow$ 3(j). For this influence path, the probability is $\pi_2^{(2)} = (1 - P_{12})P_{13}P_{34}P_{42}$. Adding these two we get the following result,

$$\pi_2 = P_{12} + P_{13}P_{34}P_{42} - P_{12}P_{13}P_{34}P_{42}. \quad (4)$$

We note again that we may directly compute the probabilities of two paths (i) $P_{1\rightarrow 2} = P_{12}$, and (ii) $P_{1\rightarrow 3\rightarrow 4\rightarrow 2} = P_{13}P_{34}P_{42}$ and use *inclusion-exclusion* to correct for the non-independence to obtain

$$\pi_2 = P_{1\rightarrow 2} + P_{1\rightarrow 3\rightarrow 4\rightarrow 2} - P_{1\rightarrow 2}P_{1\rightarrow 3\rightarrow 4\rightarrow 2}. \quad (5)$$

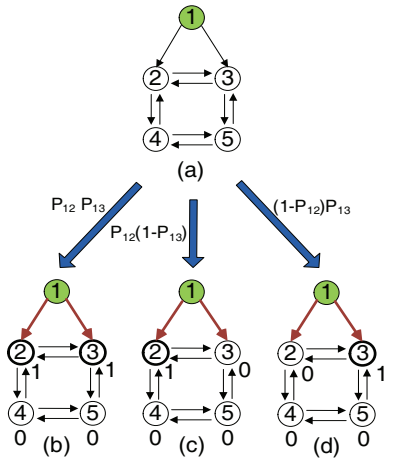


Figure 4: First stages of influence propagation of the 5-node network. Node 1 is the seed. Symbols are same as in Figure 2.

which gives the same result.

For node 3, the probability can be calculated symmetrically,

$$\pi_3 = P_{13} + P_{12}P_{24}P_{43} - P_{12}P_{13}P_{24}P_{43}.$$

For node 4, it can be activated by
(i) node 2 only, shown in Figure 3(g);
(ii) node 3 only, shown in Figure 3(h);
(iii) nodes 2 and 3 simultaneously, shown in Figure 3(f).
The total activation probability for node 4 is

$$\begin{aligned}\pi_4 =& (1 - P_{13})P_{12}P_{24} + (1 - P_{12})P_{13}P_{34} + P_{12}P_{13}(P_{24} + P_{34} - P_{24}P_{34}) \\ =& P_{12}P_{24} + P_{13}P_{34} - (P_{12}P_{24})(P_{13}P_{34}).\end{aligned} \quad (6)$$

Once again, this results can be derived using the inclusion-exclusion principle mentioned above, without counting detailed influence propagation stages in Figure 3.

3.3 Solution for 5-node network

As the last example, we compute activation probabilities for the 5-node network in Figure 1(c). The first stages of node 1 attempts to influence nodes 2,3 are shown in Figure 4(b,c,d).

Let us compute the activation probability for node 2. The contributions are shown in graphs Fig.4(b,c,d). The contributions of Fig.4(b,c) is P_{12}.

The contribution of Fig.4(d) is computed as the following. The probability to reach Fig.4(d) is $(1 - P_{12})P_{13}$. Starting from Fig.4(d), we may ignore node 1 and consider the remaining network with nodes $\{2, 3, 4, 5\}$, and node 3 is activated. This situation is identical to Figure 3, and we need to compute π_2. Following the results of Eq.(5), we obtain

$$P_{32} + P_{35}P_{54}P_{42} - P_{32}(P_{35}P_{54}P_{42}).$$

The contribution to node 2 of Fig.4(d) is a result of combining two paths $3 \rightarrow 2$ and $3 \rightarrow 5 \rightarrow 4 \rightarrow 2$ with the inclusion-exclusion principle. Therefore the final score for node 2 is

$$\pi_2 = P_{12} + (1 - P_{12})P_{13}\Big[P_{32} + P_{35}P_{54}P_{42} - P_{32}(P_{35}P_{54}P_{42})\Big]. \quad (7)$$

Now we compute activation probability for node 4. It can be activated by the following 3 paths:

(1) Starting from the situation in Figure 4(b) and activate node 4;

(2) Starting from the situation in Figure 4(c) and activate node 4; This is the same as the 4-node graph in Figure 3 and the node of interest is node 2.

(3) Starting from the situation in Figure 4(d) and activate node 4; This is the same as the 4-node graph in Figure 3 and the node of interest is node 4.

The total probability for node 4 being activated is

$$\begin{aligned}\pi_4 =& P_{12}P_{13}(P_{24}+P_{35}P_{54}-P_{24}P_{35}P_{54})\\ &+P_{12}(1-P_{13})(P_{24}+P_{23}P_{35}P_{54}-P_{24}P_{23}P_{35}P_{54})\\ &+(1-P_{12})P_{13}(P_{32}P_{24}+P_{35}P_{54}-P_{32}P_{24}P_{35}P_{54})\end{aligned} \tag{8}$$

We note that the 5-node network in Figure 1(c) includes both the 3-node network and the 4-node network cases, for example, 5-node network results become the results of the 3-node network, when $P_{24} = P_{35} = P_{54} = P_{42} = P_{53} = P_{45} = 0$; when $P_{54} = P_{45} = 1$, the results become those of the 4-node network.

When it comes to much more complicated large-scale network, the exact propagation solution is hard to derive. It's an exponential growth case along the number of nodes in V. Therefore, in this paper, we propose an approximation solution to IC model, which is exact solution for directed acyclic graph (DAG) and approximation solution for generic graphs. The exact solution for generic case is under development.

4. INCLUSION-EXCLUSION THEOREM

The lessons we learned in previous section on exact solution are useful. One of the most important lessons is the inclusion-exclusion principle that we briefly mentioned in §3. Here we formalize the concept and prove it vigorously.

Let $\pi_v (0 \le \pi_v \le 1)$ denote the activation probability for each node $\{v|v \in V \backslash S_0\}$, which means to what extent v is activated.

We have the following Theorem, which is called inclusion-exclusion theorem,

THEOREM 1. *Given fixed seed set S and edge weights P, for every non-seed node v and its in-bound neighbors $\mathcal{N}_v = (u_1, \cdots, u_k)$, i.e., $\mathcal{N}_v = \{u_i | u_i \rightsquigarrow v\}$, π_v, the stationary probability of v being activated, is related to $\{\pi_{u_i}\}$, stationary probabilities of its in-bound neighbors, with the following relationship,*

$$\begin{aligned}\pi_v =& \sum_{u_i \rightsquigarrow v} \pi_{u_i}P_{u_iv} - \sum_{\substack{u_i,u_j \rightsquigarrow v,\\ i<j}} (\pi_{u_i}P_{u_iv})(\pi_{u_j}P_{u_jv})\\ &+ \sum_{\substack{u_i,u_j,u_l \rightsquigarrow v,\\ i<j<l}} (\pi_{u_i}P_{u_iv})(\pi_{u_j}P_{u_jv})(\pi_{u_l}P_{u_lv})\\ &+ \cdots + (-1)^k(\pi_{u_1}P_{u_1v})(\pi_{u_2}P_{u_2v})\cdots(\pi_{u_k}P_{u_kv}).\end{aligned} \tag{9}$$

To better understand this result, we compare it to a simpler model of random walk. In this random-walk model, all neighbors of v can activate v any time it walks towards v. In this model, the activated probability would be

$$\pi_v^{(k)} = \sum_{u_i \rightsquigarrow v} \pi_{u_i}P_{u_iv}. \tag{10}$$

In contrast to the random walk model, in the IC model, any actor can only attempt to affect v *once*. Thus the activation probability in IC model is lower than that in random walk model. Comparing Eq.(9) and Eq.(10), we see that the reduction from random walk model to IC model are the second term and later terms in Eq.(9). They are exactly the inclusion-exclusion principle.

Figure 5 illustrates the propagation process of IC model. In this figure, node v is the target node. Its in-bound neighbors $\mathcal{N}_v = \{u_1, \cdots, u_k\}$ attempts to influence it. Suppose there is only one activated neighbor u_1 in step 1, then u_1 tries to influence v. If u_1 fails to influence v in step 1, then newly activated u_2 tries to activate v in step 2. If u_2 fails to influence v in step 2, then newly activated u_3 tries to activate v in step 3, and so on so forth.

Proof of Theorem 1

To simplify the notations, we define $\sigma_{u_i} = \pi_{u_i}P_{u_iv}$.

Step 1 : The probability that v is activated by u_1 is

$$\pi_v^{(1)} = \pi_{u_1}P_{u_1v} = \sigma_{u_1}. \tag{11}$$

Step 2: If u_1 failed to activate v in step 1, the failure probability is $1-\pi_v^{(1)}$. Now u_2 attempts to influence v; the probability that u_2 succeed in this is $\pi_{u_2}P_{u_2v} = \sigma_{u_2}$; Therefore, the conditional probability that u_1 failed but u_2 succeed in activating v is $(1-\pi_v^{(1)})\sigma_{u_2}$. This should be added to the total probability that v becomes activated. Thus

$$\pi_v^{(2)} = \pi_v^{(1)} + (1-\pi_v^{(1)})\sigma_{u_2} = \sigma_{u_1} + \sigma_{u_2} - \sigma_{u_1}\sigma_{u_2}. \tag{12}$$

Step 3: Now u_3 attempts to activate v under the condition that neither u_1 nor u_2 activated v. The probability that u_3 succeed in this is $\pi_{u_3}P_{u_3v} = \sigma_{u_3}$; The probability that neither u_1 nor u_2 activated v is $1-\pi_v^{(2)}$. Therefore, the conditional probability that u_1, u_2 failed but u_3 succeed in activating v is $(1-\pi_v^{(2)})\sigma_{u_3}$. This should be added to the total probability that v becomes activated. Thus

$$\begin{aligned}\pi_v^{(3)} &= \pi_v^{(2)} + (1-\pi_v^{(2)})\sigma_{u_3}\\ &= \sigma_{u_1} + \sigma_{u_2} + \sigma_{u_3} - \sigma_{u_1}\sigma_{u_2} - \sigma_{u_1}\sigma_{u_3} - \sigma_{u_2}\sigma_{u_3}\\ &\quad + \sigma_{u_1}\sigma_{u_2}\sigma_{u_3}.\end{aligned} \tag{13}$$

Using induction, we can include the value of π_v in step k.

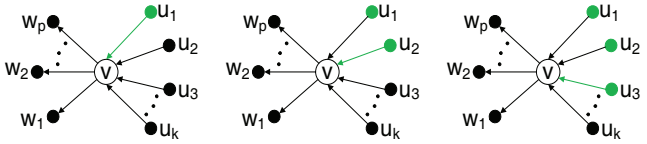
Step k: Now u_k attempts to activate v under the condition that none of v's previous in-neighbors activated v. The total probability that v becomes activated is,

$$\begin{aligned}\pi_v^{(k)} &= \pi_v^{(k-1)} + (1-\pi_v^{(k-1)})\sigma_{u_k}\\ &= \sum_{u_i \in \mathcal{N}_v} \sigma_{u_i} - \sum_{\substack{u_i,u_j \in \mathcal{N}_v,\\ i<j}} \sigma_{u_i}\sigma_{u_j} + \sum_{\substack{u_i,u_j,u_l \in \mathcal{N}_v,\\ i<j<l}} \sigma_{u_i}\sigma_{u_j}\sigma_{u_l}\\ &\quad + \cdots + (-1)^{k-1}\sigma_{u_1}\sigma_{u_2}\cdots\sigma_{u_k}.\end{aligned} \tag{14}$$

This completes the proof.

4.1 Computing Activation Probability on a Single Node

Based on the inclusion-exclusion theorem, we can get the following Lemma,



(a) step 1 (b) step 2 (c) step 3

Figure 5: Illustration of the propagation process of IC model: v is the target node

LEMMA 2. *Given a single node v's in-bound neighbors $\{u_i|u_i \rightsquigarrow v\}$ and edge weights P, to calculate the probability that node v becomes activated, we have the following iterative update equation,*

$$\pi_v^{(i+1)} = \pi_v^{(i)} + (1 - \pi_v^{(i)})\sigma_{u_{i+1}}, \quad i = 1, \cdots, k-1. \tag{15}$$

Thus, now we can present the algorithm to compute the activation probability of single node v in the following,

Input: $\mathcal{N}_v = \{u_i|u_i \rightsquigarrow v\}$, $\{\sigma_{u_i}\}$, $k = |\mathcal{N}_v|$ denotes the number of in-bound neighbors of v
Output: $\pi_v^* = \pi_v^{(k)}$
Initialize $i = 1, \pi_v^{(1)} = \sigma_{u_1}$
for $i = 1 : k-1$ **do**
  $\pi_v^{(i+1)} = \pi_v^{(i)} + (1 - \pi_v^{(i)})\sigma_{u_{i+1}}$
end

Algorithm 1: Activation probability on v

4.2 Computing Activation Probability on Entire Network

Now we are ready to describe the algorithm to compute activation probability on entire network. Given fixed seed set S and edge weights P, the activation probability to IC model for each node v can be represented in the following,

$$\pi_v^* = F(\pi_{\mathcal{N}_v}^*), \quad v = 1, \cdots, n. \tag{16}$$

which can be obtained by the following updating strategy, where $\pi_{\mathcal{N}_v}$ denotes a vector whose elements are the activation probabilities of v's in-bound neighbors.

$$\pi_v^{(t+1)} = F(\pi_{\mathcal{N}_v}^{(t)}), \quad v = 1, \cdots, n. \tag{17}$$

where, $F(\cdot)$ denotes a function, which is represented in Eq.(15).

The detailed algorithm for computing activation probabilities for all nodes V is given in Algorithm 2, where π denotes the activation probability vector whose elements are activation probabilities for all nodes.

4.3 Inclusion-Exclusion Theorem Provides Approximate Solution for Generic Networks

Let's consider the 3-node undirected network in Figure 1(a) and the IC propagation process given seed set $\{1\}$ for this network in Figure 2, suppose π_2^∞ and π_3^∞ are the probabilities that node 2 and node 3 are activated, respectively, they have exact solution in the following,

$$\pi_2^\infty = P_{12} + P_{13}P_{32} - P_{12}P_{13}P_{32} \tag{18}$$
$$\pi_3^\infty = P_{13} + P_{12}P_{23} - P_{12}P_{13}P_{23} \tag{19}$$

However, if we solve this propagation problem using Inclusion-Exclusion Theorem, π_2^∞ and π_3^∞ have the following relationship,

$$\begin{aligned} \pi_2^\infty &= P_{12} + \pi_3^\infty P_{32} - P_{12}\pi_3^\infty P_{32} \\ \pi_3^\infty &= P_{13} + \pi_2^\infty P_{23} - P_{13}\pi_2^\infty P_{23} \end{aligned} \tag{20}$$

By solving the above binary linear equation group, we get the following solution,

$$\pi_2^\infty = \frac{P_{12} + P_{13}P_{32} - P_{12}P_{13}P_{32}}{P_{23}P_{32}(1 - P_{13})(1 - P_{12})} \tag{21}$$

$$\pi_3^\infty = \frac{P_{13} + P_{12}P_{23} - P_{12}P_{13}P_{23}}{P_{23}P_{32}(1 - P_{13})(1 - P_{12})} \tag{22}$$

We note that the numerator in Eq.(21) is the same with that of Eq.(18) and the numerator in Eq.(22) is the same with that of Eq.(19). The denominators are apparently both less than 1; this indicates the Inclusion-Exclusion Theorem over-counts the influence for non-DAG network. However, experiments show that the results using Inclusion-Exclusion Theorem is very close to those using Monte Carlo simulation on real world networks.

4.4 Selecting Seed Set for Viral Marketing

To this end, we can define the objective function of maximizing the social influence as follows,

$$\begin{aligned} \max_S \quad & \sigma(S) = \sum_{v=1}^{n} \pi_v \\ s.t. \quad & |S| = m, \\ & \pi_v = F(\pi_{\mathcal{N}_v}), \quad v = 1, \cdots, n. \end{aligned} \tag{23}$$

where, m is the size of seed set. We will present an probabilistic additive strategy to solve the above social influence maximization problem using greedy methods.

5. GREEDY METHOD TO SOLVE SOCIAL INFLUENCE MAXIMIZATION

As discussed above, influence maximization is to determine m activated seeds at the beginning of the information propagation, in order to maximize the social influence in

Input: $G(V, E)$, $n = |V|$, edge weight P, activated seed set $S_0 = S$, $m = |S|$, $maxIter$
Initialize $\pi_{v\in S}^{(0)} = 1$, $\pi_{v\notin S}^{(0)} = 0$
for $l = 1 : maxIter$ **do**
  for $i = 1 : n$, $v_i \notin S$ **do**
    $\pi_{v_i}^{(l)} \leftarrow$ Activation probability on v_i
  end
  if $\|\pi^{(l)} - \pi^{(l-1)}\|_1 < \delta$ **then**
    break;
  end
end
Output: stationary activation probability for each node $\pi = \pi^{(l)}$

Algorithm 2: Activation probability on entire network

the end. We first propose an probabilistic additive strategy and two greedy methods, then based on these two greedy methods, another efficient incremental search strategy will be introduced.

5.1 Greedy Method 1

The basic idea is to select each node v_i as a single seed, i.e., $S = \{v_i\}$, and then compute the stationary activation probability using Algorithm 2 for all the other nodes $\{u|u \notin S\}$. We let $\beta_{\{i\}}$ denote the stationary activation probability for every node when v_i is selected as the seed node. Let $\mathbf{B} = (\beta_{\{1\}}, \beta_{\{2\}}, \cdots, \beta_{\{n\}})$ Obviously, the elements on diagonal of matrix $\mathbf{B}$ are all 1. After getting every stationary activation probability vector in $\mathbf{B}$, we calculate the influence spread of each node, denoted by $\sigma(v_i)$, which is the sum of $\beta_{\{i\}}$. The following Algorithm 3 describes the detailed process on how to calculate $\mathbf{B}$.

Input: Edge weight P
for $i = 1 : n$, $S = \{v_i\}$ **do**
  $\pi^{(0)}_{v_i} = 1$, $\pi^{(0)}_{V \setminus v_i} = 0$,
  $\pi \longleftarrow$ Call Algorithm 2(P,$S = \{v_i\}$),
  $\beta_{\{i\}} = \pi$.
end
Output: Stationary activation probability matrix $\mathbf{B}$

Algorithm 3: Computing stationary activation probability vector when each node is selected as the seed node

After we get the social influence spread for each node being seed node, we sort the influence spread scores in descending order, i.e., $\sigma(v_1) > \sigma(v_2) > \cdots > \sigma(v_n)$, and then select the top m nodes with largest influence scores as the initialization seed set S.

Greedy method 1 is based on our inclusion-exclusion theorem, and it's much faster than greedy method using Monte Carlo Simulation, which needs at least thousands of simulations even for calculating the influence score of each v_i, not mention the entire stationary activation probability matrix $\mathbf{B}$. We will present the time needed for both methods in the experiment section.

5.2 Greedy Method 2

Before presenting Greedy Method 2, we first introduce a probabilistic additive strategy when adding more nodes to a current seed set.

5.2.1 Probabilistic Additive

Suppose we have a current seed set S_c and its activation probability vector β_{S_c}, calculated by Algorithm 2, and now we are going to add v_i to the current seed set, note, v's activation probability vector $\beta_{\{i\}}$ can be get from matrix $\mathbf{B}$. We have the following definition,

Definition 1. Probabilistic Additive of vector β_{S_c} and $\beta_{\{i\}}$ is defined as follows,

$$\beta_{S_c \cup \{i\}} = \beta_{S_c} \biguplus \beta_{\{i\}} = 1 - (1 - \beta_{S_c}) .* (1 - \beta_{\{i\}}) \quad (24)$$

where $.*$ means element wise multiplication. Similarly, Probabilistic Additive of m vectors is defined as follows,

$$\beta_{\{1\} \cup \{2\} \cdots \cup \{m\}} = \beta_{\{1\}} \biguplus \beta_{\{2\}} \cdots \biguplus \beta_{\{m\}} = 1 - \Pi_{i=1}^{m} (1 - \beta_{\{i\}}) \quad (25)$$

where Π also means element wise multiplication.

We can use the Probabilistic Additive of each node in S $\beta_{\cup\{i|i \in S\}}$ as the initialization when calculating the activation probability for entire network using Algorithm 2 with seed set S, which is much faster than 0 or 1 initialization used in Algorithm 2.

As Greedy Method 1 did, Greedy Method 2 first uses algorithm 3 to calculate the stationary probability distribution matrix $\mathbf{B}$, and then calculate influence score for each node $\sigma(v_i)$. Different with Greedy Method 1, Greedy Method 2 adds only one node to seed set at a time, which is described in the following,

(1) Add node i_1 with the largest influence score to seed set S_1, $S_1 = \{i_1\}$, and the activation probability vector is $\beta_{S_1} = \beta_{\{i_1\}}$.

(2) Add node i_2, which can lead to the largest influence score of seed set S_2, $S_2 = S_1 \cup \{i_2\} = \{i_1, i_2\}$. Then we use probabilistic additive of β_{S_1} and $\beta_{\{i_2\}}$ to initialize π^0 when calculating the activation probability vector for S_2 using Algorithm 2, $\pi^{(0)} = \beta_{S_1 \cup \{i_2\}} = \beta_{S_1} \biguplus \beta_{\{i_2\}}$;

(3) Repeat the above process until we need to add node i_m, and node i_m should lead to the largest influence score of seed set S_m, $S_m = S_{m-1} \cup \{i_m\} = \{i_1, \cdots, i_m\}$. Then we use probabilistic additive of $\beta_{S_{m-1}}$ and $\beta_{\{i_m\}}$ to initialize π^0 when calculating the activation probability vector for S_m using Algorithm 2, $\pi^{(0)} = \beta_{S_{m-1} \cup \{i_m\}} = \beta_{S_{m-1}} \biguplus \beta_{\{i_m\}}$;

In this way, we get another seed set S. To get a better seed set, we do Incremental Search starting from the seed sets calculated from both greedy methods.

5.3 Incremental Search Strategy

After we get an initialization seed set by those two greedy methods introduced in last section, we want to keep digging more efficient seed set by replacing the nodes those have least contribution in the current seed set to the final influence score.

First, we give two important procedures, add k node to current seed set S_c and drop k node from S_c, which are listed in Algorithm 4 and Algorithm 5, respectively. For k, we normally choose $k = 1, 2, 3$.

Input: Current Seed Set: S_c, the size of S_c: $q = |S_c|$,
  Activation Probability Matrix: $\mathbf{B}$
for $i = 1 : C^k_{n-q}$ **do**
  Select nodes to add : $\{v_{i_1}, \cdots, v_{i_k}\}$,
  $S_n = S_c \cup \{v_{i_1}, \cdots, v_{i_k}\}$,
  $\pi^{(0)} = \beta_{S_c \cup \{i_1\} \cdots \cup \{i_k\}}$,
  $\pi \longleftarrow$ Call Algorithm 2(P, S_n, $\pi^{(0)}$),
  $\sigma(S_n) = \sum_v \pi_v$.
end
$\{v_{i_1}, \cdots, v_{i_k}\} \longleftarrow \arg\max(\sigma(S_n))$,
Output: $S_c = S_c \cup \{v_{i_1}, \cdots, v_{i_k}\}$, $\beta_{S_c} = \pi$,
  $\sigma(S_c) = \sigma(S_n)$

Algorithm 4: Add k node

Now, we are ready to present the incremental search strategy in Algorithm 6,

After using incremental search on seed set obtained from greedy methods, we get a seed set with larger influence sore.

6. EXPERIMENTS

To validate the performance of our Inclusion-Exclusion Theorem, we conduct experiments on real data sets to compare the results using Inclusion-Exclusion Theorem (Algorithm 2) and that of using Monte Carlo Simulation. We also conduct experiments on real world data sets to compare the influence spread of seed set get from our incremental search strategy with those of seed sets get from various algorithms.

6.1 Data Sets

We use two real world data sets - p2p-Gnutella08 and wiki-Vote, which are two directed networks, and downloaded from SNAP [1]. Table 1 lists the detailed information of these two data sets.

p2p-Gnutella08 is a snapshot of Gnutella peer-to-peer file sharing network from August 2002. Nodes represent hosts in the Gnutella network topology and edges represent connections between the Gnutella hosts. We call this network p2p in this paper.

Wiki-Vote contains the Wikipedia voting data from the inception of Wikipedia till January 2008. Nodes in the network represent wikipedia users and a directed edge from node i to node j represents that user i voted on user j.

We also generate one subgraph from p2p-Gnutella08, which has 223 nodes and 954 edges. We call this subgraph p2p-223.

6.2 Experiment Setup

There are three parts for our experiments. One is to verify that given a seed set S, the influence spread score $\sigma(S)$ using our Inclusion-Exclusion Theorem is almost the same with the average score of running Monte Carlo Simulation for thousands of times. The second part is to present the time needed to compute the influence spread given a set of seed nodes using Algorithm 2 based on Inclusion-Exclusion Theorem, comparing with that of Monte-Carlo Simulations [7]. The last part is to compare the social influence of seed sets selected by our incremental strategies with those of seed sets selected by other methods. We compare the following set of algorithms.

(1) Random selection: select m nodes randomly from V as the seed nodes.

(2) Degree selection: select m nodes with the largest out-degree as the seed nodes.

(3) Distance selection: select m nodes with smallest average shortest-path distances to all other nodes as the seed nodes.

(4) Incremental search 1: it's a combination of greedy method 1 and incremental search.

- Compute activation probability matrix **B** using Algorithm 3.
- Select m nodes with the largest influence score $\sigma(v_i)$.
- Apply incremental search strategy using Algorithm 6 on seed nodes selected by last step, with parameter $k = [1, 2, 3]$.

(5) Incremental search 2: it's a combination of greedy method 2 and incremental search.

- Select m nodes using greedy method 2 introduced in section 5.3.
- Apply incremental search strategy using Algorithm 6 on seed nodes selected by last step, with parameter $k = [1, 2, 3]$.

Note, for all the above 5 methods, we apply our Algorithm 2 to compute the influence spread for the entire network for a given seed set. And edge weight P remains fixed for the same data set. The experiments are run on a PC with a 3.0GHz Intel Core 2 Duo Processor and 12GB memory.

[1] http://snap.stanford.edu

Input: Current Seed Set: S_c, the size of S_c: $q = |S_c|$, Activation Probability Matrix: **B**
for $i = 1 : C_q^k$ **do**
  Select nodes to drop : $\{v_{i_1}, \cdots, v_{i_k}\}$,
  $S_n = S_c \backslash \{v_{i_1}, \cdots, v_{i_k}\}$,
  $\pi^{(0)} = \beta_{\cup\{j|j\in S_n\}}$,
  $\pi \longleftarrow$ Call Algorithm 2(P, S_n, $\pi^{(0)}$),
  $\sigma(S_n) = \sum_v \pi_v$.
end
$\{v_{i_1}, \cdots, v_{i_k}\} \longleftarrow \arg\max(\sigma(S_n))$,
Output: $S_c = S_c \backslash \{v_{i_1}, \cdots, v_{i_k}\}$, $\beta_{S_c} = \pi$, $\sigma(S_c) = \sigma(S_n)$

Algorithm 5: Drop k node

Input: Current Seed Set: S_c, the size of S_c: $q = |S_c|$, k
for $i = 1 : maxIter$ **do**
  $S_a, \sigma(S_a) \longleftarrow$ Add k nodes,
  $S_d, \sigma(S_d) \longleftarrow$ Drop k nodes,
  if $\|\sigma(S_a) - \sigma(S_d)\| < \delta$ **then**
    break;
  end
end
Output: $S_c = S_d$

Algorithm 6: Incremental search strategy

Table 1: Description of data sets

Name	# Nodes	# Edges
p2p-Gnutella08	6301	20,777
wiki-Vote	7115	103,689

6.3 Experiment Results

As discussed in last section, first, we present the influence spread (activation probability for each node) comparison between IC Monte Carlo Simulation and Inclusion-Exclusion Theorem (Algorithm 2), to verify the effectiveness of Inclusion-Exclusion Theorem in approximating the influence spread for entire network. For demonstration purpose, we first present the results for each node on p2p-223 subgraph. We randomly selected 10 nodes as seed nodes. The results from Monte-Carlo Simulations are the average of 20000 simulations/realizations. Edge weight P is fixed for both methods. The activation probabilities for each node from two methods are shown in Figure 6. There are 223 nodes and 954 edges on p2p-223 network, and we omit the nodes with activation probability 0, which left us less than 90 nodes presented in the figure. For convenient comparison, we sort the activation probabilities of Monte Carlo Simulations, and present their corresponding activation probabilities calculated by Inclusion-Exclusion Theorem. Apparently, the two

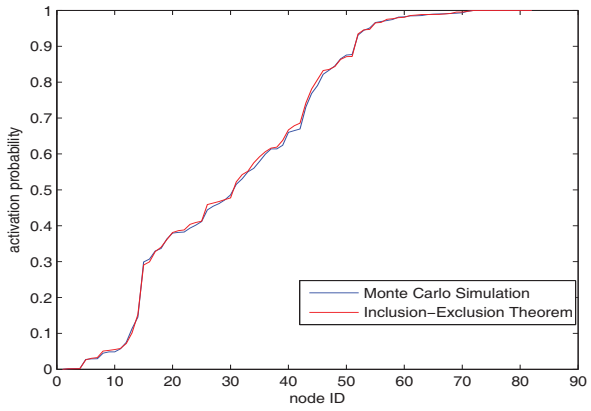


Figure 6: Activation probability comparison between IC Monte Carlo Simulation and Inclusion-Exclusion Theorem

Table 2: Mean squared errors (MSE) and mean absolute errors (MAE) between the two activation probability vectors achieved Inclusion-Exclusion Theorem and Monte Carlo simulation on p2p-223 network

	$m = 20$	$m = 30$	$m = 40$	$m = 50$
MSE	8.5×10^{-5}	7.9×10^{-5}	7.8×10^{-5}	7.1×10^{-5}
MAE	4.8×10^{-4}	4.7×10^{-4}	4.5×10^{-4}	3.8×10^{-4}

curves from these two methods almost coincide with each other.

We then present more comparisons between activation probabilities achieved by applying Monte Carlo simulation and those achieved by Inclusion-Exclusion Theorem. Figure 7 shows the influence spreads by those two methods at different sizes of the same seed set - $m = \{10, 20, 30, 40, 50\}$ on the three data sets - p2p-223, p2p and wiki-Vote. The influence spreads are almost the same on p2p-223 subgraph, and are very close on p2p and wiki-Vote data sets. As shown on the figures, the altitudes of the histogram at the same m are almost the same for those two methods. We also show the mean squared errors (MSE) and mean absolute errors (MAE) between two activation probability vectors achieved by those two methods. Table 2, 3 and 4 show the MSE and MAE results at $m = \{20, 30, 40, 50\}$ (results at $m = 10$ are omitted due space limit) on the three data sets mentioned above, which demonstrate that the two vectors are almost the same when given the same seed set, i.e., the approximate results achieved by Inclusion-Exclusion Theorem are effective for generic graphs.

Second, we compare the time needed to compute the influence spread given different sizes of seed sets. Table 5 and Table 6 list the time needed for both Inclusion-Exclusion Theorem and Monte Carlo Simulation on two data sets -

Table 3: Mean squared errors (MSE) and mean absolute errors (MAE) between the two activation probability vectors achieved Inclusion-Exclusion Theorem and Monte Carlo simulation on p2p network

	$m = 20$	$m = 30$	$m = 40$	$m = 50$
MSE	7.5×10^{-5}	8.5×10^{-5}	8.4×10^{-5}	8.3×10^{-5}
MAE	3.0×10^{-3}	3.8×10^{-3}	3.7×10^{-3}	3.3×10^{-3}

Table 4: Mean squared errors (MSE) and mean absolute errors (MAE) between the two activation probability vectors achieved Inclusion-Exclusion Theorem and Monte Carlo simulation on wiki-Vote network

	$m = 20$	$m = 30$	$m = 40$	$m = 50$
MSE	5.0×10^{-6}	5.1×10^{-6}	5.2×10^{-6}	5.1×10^{-6}
MAE	8.1×10^{-5}	8.1×10^{-5}	8.2×10^{-5}	8.0×10^{-5}

Table 5: Time (sec) needed to compute the influence spread given different sizes of seed sets m on p2p-223 network

Methods	$m = 10$	$m = 20$	$m = 30$
Inclusion-Exclusion	0.1079	0.02531	0.01925
Monte Carlo Simulation (20000 times)	95.3935	94.0613	93.1055

p2p-223 and p2p, at different m, where m denotes the number of seed nodes selected. And at the same m, the same seed set is selected for both methods. Let's look at Table 6, when $m = 50$, the time for Inclusion-Exclusion Theorem is just 2.8450 seconds, while Monte Carlo Simulation needs 36945.6 seconds for 20000 realizations. So the time for Inclusion-Exclusion Theorem is competitive with that of just one time Monte Carlo Simulation, however, Monte Carlo Simulations need thousands of times simulations to reach a steady solution. Therefore, our probabilistic solutions speed up the computation of influence spread, especially on large data sets, which make greedy methods to viral marketing scalable to large data sets.

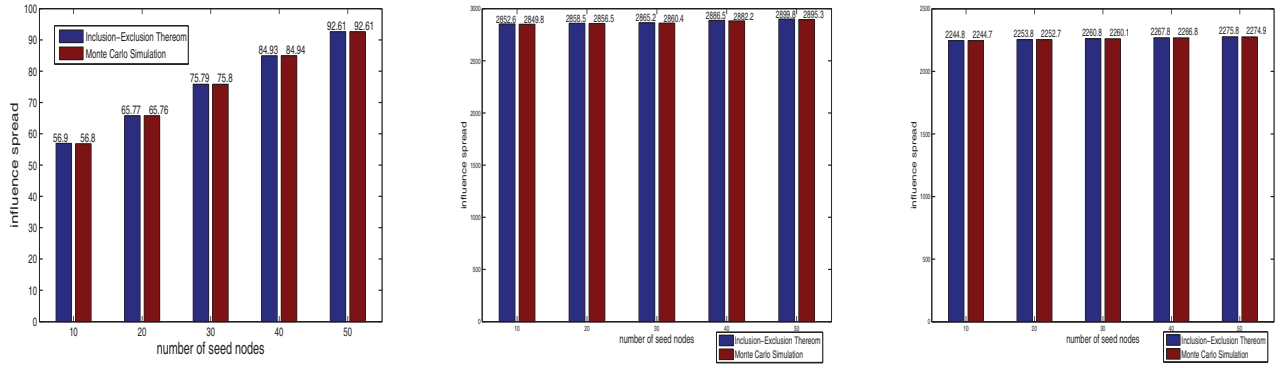
Last but not least, we apply our incremental search method 1 and incremental search method 2 to select the largest influential nodes (seed set), and compare the influence spread for the seed set selected by our methods with three simple heuristic methods. The results on 3 data sets are shown in Figure 8. Our methods outperform the other methods significantly. The point is that once we get the activation probability matrix $\mathbf{B}$, a lot of recalculation can be omitted by combining our proposed probabilistic additive strategy.

7. CONCLUSIONS

In this paper, we propose probabilistic solutions to the computation of influence spread under IC model. We show that our probabilistic solutions can significantly speed up the computation of influence spread. We also give a probabilistic additive based incremental search strategy to solve the influence maximization problem. Experiments on real data sets demonstrate the effectiveness of our probabilistic solutions and incremental search strategy. There are limitations for our probabilistic solutions, which are targeting at DAG network. But experiments on real data sets with non-DAG structures show that our method can provide good approx-

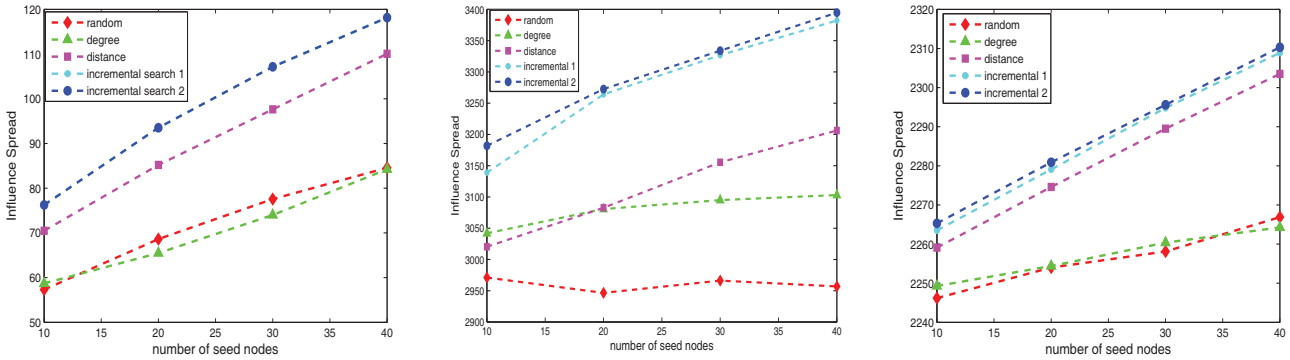
Table 6: Time (sec) needed to compute the influence spread given different sizes of seed sets m on p2p network

Methods	$m = 10$	$m = 30$	$m = 50$
Inclusion-Exclusion	3.2012	3.0126	2.8450
Monte Carlo Simulation (20000 times)	32345.4	35372.2	36945.6



(a) p2p-223 (b) p2p (c) wiki-Vote

Figure 7: Influence Spreads computed by Inclusion-Exclusion Theorem and Monte Carlo Simulation given different sizes of seed sets



(a) p2p-223 (b) p2p (c) wiki-Vote

Figure 8: Influence Spreads of seed sets selected by different methods. Note, the curves corresponding to incremental search 1 and incremental search 2 almost coincide with each other on p2p-223 network

imation to directed graph. We will develop more accurate probabilistic solutions for IC model in the future work.

Acknowledgements. This work is partially supported by U.S. NSF-CCF-0917274, NSF-DMS-0915228, and Research Innovation Project of Shanghai Municipal Educational Commission (NO. 12YZ190).

8. REFERENCES

[1] C. C. Aggarwal, A. Khan, and X. Yan. On flow authority discovery in social networks. In *SDM*, pages 522–533, 2011.

[2] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD*, pages 1029–1038, 2010.

[3] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *KDD*, pages 199–208, 2009.

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

[5] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan. Learning influence probabilities in social networks. In *WSDM*, pages 241–250, 2010.

[6] M. Granovetter. Threshold models of collective behavior. *The American Journal of Sociology*, 83:1420–1443, 1978.

[7] D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003.

[8] M. Kimura and K. Saito. Tractable models for information diffusion in social networks. In *PKDD*, pages 259–271, 2006.

[9] T. Liggett. *Interacting Particle Systems*. Classics in Mathematics. Springer, 1985.

[10] I. R. Misner. *The Worldąŕs best known marketing secret: Building your business with word-of-mouth marketing*. Bard Press, 2nd edition, 1999.

[11] J. Nail. The consumer advertising backlash. Technical report, Forrester Research and Intelliseek Market Research Report, May 2004.

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

[13] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *KDD*, pages 807–816, 2009.

[14] J. Tang, S. Wu, B. Gao, and Y. Wan. Topic-level social network search. In *KDD*, pages 769–772, 2011.

[15] Y. Yang, E. Chen, Q. Liu, B. Xiang, T. Xu, and S. A. Shad. On approximation of real-world influence spread. In *ECML/PKDD (2)*, pages 548–564, 2012.