Maximizing the Effect of Information Adoption: A General Framework

Tianyuan Jin Tong Xu Hui Zhong Enhong Chen Zhefeng Wang Qi Liu [*]

Abstract

With the development of social networking services, social influence analyses, as well as the influence maximization tasks, have attracted wide attention in both academia and industry. Traditional studies mainly focus on simulating process of influence spread. However, two basic functions of social spread, i.e., information propagation and information adoption have not been clearly distinguished. Usually, as information adoption could be even more significant for information publishers in application scenarios, more comprehensive analysis for effect of adoption is urgently required. To that end, in this paper, we propose a novel framework to generally describe social spread, in which information adoption process is separately formulated as random events. Along this line, when we apply this framework to the information adoption maximization task, with proving that the adoption maximization problem is NP-hard and submodular, we further design a polling-based algorithm to achieve an effective approximation. Extensive experiments on four real-world data sets demonstrate the effectiveness and efficiency of proposed algorithms, which validates that our approach could better summarize the complete social spread process, and further support the necessity of distinguishing information adoption from information propagation.

Keywords: Social network; Influence maximization; Information Adoption; Randomized algorithm

1 Introduction

Recent years have witnessed the booming of social network services, where individuals are now connected in cyber world to share news, ideas and comments. This new business model not only urges the traditional broadcasting way to be enriched by efficient social propagation, but also raises new challenges to manage and predict information flow. Traditionally, several models have been proposed to simulate the information propagation process, such as the Independent Cascade (IC) Model [8], Linear Threshold (LT) Model [15], in which each node could be simply summarized as "active" or "inactive". In these models, active nodes, i.e., the nodes who receive the information, can be viewed as adopting the new information simultaneously. Along this line, the *Influence Maximization* problem has been proposed to achieve the maximum information propagation with limited "seed nodes", and further support extensive applications in viral marketing [24], authority mining [20] and decision making [21], etc.

However, though extensive efforts have been made, these information diffusion models above, for some application scenarios may still fail to reflect the complete process of social spread. Specifically, people who **propagate** the information may not necessarily **adopt** it. The following example may intuitively explain this phenomenon in detail.

Motivating Example: *Cathy organized a charity party and tweeted the invitation. Her friend Irene retweeted this message, but she had no time to attend. On the contrary, another friend Abel, though not retweeted the message, finally participated the party.*

In this case above, Irene acted as the information channel without participation (*adoption*), while for Abel, participation happened without information spread (*propagation*). under these circumstances, the step of **information propagation** (e.g., retweeting), and **information adoption** (e.g., party attendance) should be treated separately. This phenomenon could be reasonable especially in interest-sensitive social network [22], where information propagation may not definitely lead to adoption if not attractive enough.

Unfortunately, current techniques may fail to distinguish these two basic functions of information spread. In the perspective of social spread in some traditional models, only the information propagators (e.g., Irene) will be treated as "activated", but not the information adopters (e.g., Abel). This might be unreasonable as participation could be more significant for event organizers. Thus, more comprehensive analysis is urgently required to describe complete social spread process.

To that end, in this paper, we propose a novel framework to generally describe information propagation and adoption process. Further, for the Information Adoption Maximization task, we design a polling-based algorithm: obtaining the nearly optimal approximation factor of $(1 - 1/e - \epsilon)$ with at least $(1 - 1/n)$ probability, in time $O(knm \log n \cdot OPT^{-1}\epsilon^{-2})$. Extensive experiments on four real-world data sets demonstrate the effectiveness and efficiency of our algorithm, which validates the potential of our framework in social spread simulation, and also support the necessity of distinguishing information adoption. Specially, the technical contributions can be summarized as follows:

- We distinguish two basic functions of social spread

[*] Author affiliations and emails: *Anhui Province Key Lab of Big Data Analysis and Application, University of Science and Technology of China*, {jty123, zhuiwin, zhfwang}@mail.ustc.edu.cn, {tongxu, cheneh, qiliuql}@ustc.edu.cn

i.e., information propagation and adoption, and then explore the value of adoption for better measuring the effect of information spread. Also, a novel framework is proposed to generally describe spread process.

- We propose the "Adoption Maximization" task based on our framework, and prove that this problem is NP-hard and submodular under the IC model. We also show the computation of information adoption under the IC model is #P-hard.

- We design a novel randomized algorithm to deal with the Adoption Maximization task, and theoretically prove the approximation within certain expected time. Experiments on real-world data sets verify the performance of proposed algorithm.

2 Related Work

Generally, researches on social spread analysis could be roughly divided into two classes, namely (1) social spread modeling task and (2) influence maximization task with its variants.

For the information diffusion modeling, large efforts have been made on simulating the spreading dynamics, such as Independent Cascade (IC) model, Linear Threshold (LT) model, continues time IC model [3], Linear model [13] and [23]. Usually, to ease the modeling, prior arts like [11] assume that once a user is activated, she will unconditionally adopt the information. Thus, process of adoption here is indeed equal to the influence spread. Correspondingly, some researches argued the function of social spread, which is essentially treated as "proxy" [1]. Along this line, [19] distinguished the information propagation and the information coverage, and then proposed a new problem called information coverage maximization. Different from prior arts, in this paper, we propose a novel concept called information adoption, and then propose the general framework to describe the complete process. To be specific, it will be shown that both influence maximization and information coverage maximization are special cases of our general framework.

For the influence maximization task, since Kempe et al. firstly proved that influence maximization problem is NP-hard in both IC model and LT model, and then proposed a greedy algorithm to approximate the result [11], lots of works have focused on this issue. For instance, Chen et al. proved that computing the influence spread is #P-hard in both IC and LT model [4, 6]. Also, Leskovec et al. applied the "lazy evaluation" strategy to the greedy algorithm and significantly reduced the estimation time [12]. Along this line, [5] and [9] further improved the performance with elaborate algorithms. At the same time, many heuristic algorithms are proposed, such as [7] and [10] which improved the efficiency with a compromise of the effectiveness. Based on these models, Borgs et al. proposed a polling-based algorithm, with the same approximation ratio as the greedy algorithm but with high efficiency [2]. Besides, Tang et al. further improved the algorithm and proposed two algorithms that are much faster than the other influence maximization algorithms [17, 16]. Different from prior arts, in this paper, we design a polling-based algorithm with a time complexity of $O(knm \log n \cdot OPT^{-1}\epsilon^{-2})$ for our problem.

3 A General Framework

In this section, we will first define the process of information propagation and adoption, as well as the adoption maximization task. Then, we will show some special cases of our framework. Finally, we discuss the computational complexity under IC model.

3.1 Framework & Maximization Task To study the complete process of social spread, first of all, we have a social network $G = (V, E)$, where V presents the set of nodes, and $E \subseteq V \times V$ is the set of propagation paths between nodes, i.e., weighted edges. Write $n = |V|$ for the number of nodes, and $m = |E|$ for the edges.

Then, a diffusion strategy is required for modeling the spread. To ease the modeling, the widely-used *Independent Cascade* (IC) model is introduced. What should be noted is that we propose a generic description of social spread process, thus the IC model could be replaced by any other diffusion model if needed.

Along this line, assuming that each edge e is associated with an activation probability $p(e)$, we propose the generic framework, in which two stages, i.e., *information propagation* and *information adoption* will be separately described as follows:

Propagation Stage. During the propagation stage, we will simulate the diffusion process according to the following steps:

- First, we have the source set S as the initial disseminators of the information.

- Second, each edge in G will be removed with probability $1 - p(e)$. Then, we have the resulting graph g called a "**live-edge graph**" [11], in which kept edges called "**live-edge**", while removed edges called "**dead-edge**".

- Finally, the node i will be activated, if i can be reached from S in g, i.e., there exists a directed path in g that starts from S and ends at i.

Then, we can get an active node set as **Active**(S).

Adoption Stage. Then, to simulate the adoption process, each node will get influenced from its activated

Table 1: Several important mathematical notations

Notations	Description
$G = (V, E)$	a social network
$n =\mid V \mid$	the number of nodes in G
$m =\mid E \mid$	the number of edges in G
G^T	the transpose graph of G, $(u, v) \in G$ iff$(v, u) \in G^T$
g	a live-edge graph instance of G
U_v	the union of a node v and its in-neighbors
A_v	the set of the active nodes in U_v
S	the seed set of social spread
$A_v \mid_S^g$	the active nodes in U_v, given g and S
$Q_g^u(S)$	the information adoption of node u, given g and S
$Y_g(S)$	the adoption spread in g when the seed set is S
$F(S)$	expected number of adopt nodes, given S
OPT	the maximum $F(S)$ for any seed set S of size k

neighbors, including itself, to adopt the information. Write $N^{in}(u)$ for in-neighbors of u and A_u for a set of activated nodes that can influence u, i.e., $A_u = u \cup N^{in}(u) \cap \mathbf{Active}(S)$, we have the adoption spread for given seed set S as:

$$F(S) = \mathbf{Adopt}(S) = \sum_{u \in V} [f_u(A_u)],$$

Here f_u: $2^{|u \cup N^{in}(u)|} \to [0, 1]$ is a metric function, which estimates the probability of adoption for node u. Along this line, finally, we formally define the information adoption maximization problem studied in this paper as follows:

Problem 1. ***Information Adoption Maximization***. *Given a graph $G = (V, E)$, a parameter k, and a set of adoption functions $\{f_v | v \in V\}$, the information adoption maximization problem aims to finding a seed set S^* of k nodes that maximizes the adoption spread, ie., $S^* = \underset{|S|=k}{arg\ max}\, F(S)$*

For better illustration, Table 1 lists frequently used mathematical notations.

3.2 Special Cases of the General Framework

We proposed a general framework of information adoption in previous section. Actually, the existing problem such as Influence Maximization problem [11] and Information Coverage Maximization problem [19] are special cases of the general framework. Due to the generality of adoption function f_v, we introduce another special case of the framework.

Influence maximization problem [11] aims to maximizing the expected number of active nodes. It assumes that adopt nodes is equal to the active nodes. The corresponding adopt function f_v is

$$f_v(A_v) = \begin{cases} 1 & \text{if } v \in A_v \\ 0 & \text{if } v \notin A_v. \end{cases} \tag{3.1}$$

Information coverage maximization problem [19] aims to maximizing the expected number of both active nodes and informed nodes. It assumes that both the active nodes and the informed nodes are adopt nodes. The corresponding adopt function f_v is

$$f_v(A_v) = \begin{cases} 1 & \text{if } A_v \neq \emptyset \\ 0 & \text{if } A_v = \emptyset. \end{cases} \tag{3.2}$$

In real life, the adoption, which usually happens with practical actions, e.g., buy the product or attend the party, could be more valuable to measure the effect of social spread. To calculate the adoption spread, we need a specific functional form of f_v. If we look the consist of U_v, we can find two types of nodes: v and its neighbors. Thus we can divide f_v into two parts: $\lambda(v)$ and $\rho(v)$, i.e., $f_v = \lambda(v) + \rho(v)$. Here, $\lambda(v)$ represents the contribution from v itself and $\rho(v)$ represents the contribution from v's in-neighbors. Specifically, for $\lambda(v)$, let $\lambda(v) = \theta \in [0, 1]$ if v is active, and $\lambda(v) = 0$ if v is inactive. For $\rho(v)$, we assume that neighbors v independently influence v and $\rho(v) = 1 - \lambda(v)$ when all in-neighbors of v are activated. We use α_v to denote the number of activated nodes among v's in-neighbors, τ_v to denote the number of nodes of v's in-neighbors and β_v to represent the contribution from a single active in-neighbor of v, i.e., the probability of v's adoption if v has an active in-neighbor. When all the nodes in U_v have been activated, we have $\lambda(v) = \theta$, $\rho(v) = 1 - (1 - \beta_v)^{\tau_v} = 1 - \theta$. Thus, $1 - \beta_v = \theta^{1/\tau_v}$ and $\rho(v) = 1 - (1 - \beta_v)^{\alpha_v} = 1 - \theta^{\alpha_v/\tau_v}$. To this end, we define a new information adoption function as

$$f_v(A_v) = \begin{cases} 1 + \theta - \theta^{\alpha_v/\tau_v} & \text{if } v \text{ is active} \\ 1 - \theta^{\alpha_v/\tau_v} & \text{if } v \text{ is inactive.} \end{cases} \tag{3.3}$$

3.3 Computational Complexity

In this part, we will show the properties of the framework. Let $U_v = v \cup N^{in}(v)$, we give an assumption for f_v.

Assumption 1. *If $N \subset M \subseteq U_v$, then $f_v(M) - f_v(N) > 0$.*

Explanation: If a node has more active in-neighbors, it will get more influence and the node will has higher probability adopt the information.

This assumption should be used as the requisite, since the adoption maximization problem is meaningless without it, e.g., $f_v(A_v) \equiv 1$.

Now, we are ready to prove the properties of the framework. We first prove that maximizing adoption spread is NP-hard.

Theorem 3.1. *In the IC model, the information adoption maximization problem is NP-hard.*

Proof. Consider the restricted class of instances of function f_v where

$$f_v(A_v) = \begin{cases} 1 & \text{if } v \in A_v \\ 0 & \text{if } v \notin A_v. \end{cases}$$

The problem of information adoption maximization in this case is equivalent to the classical problem of influence maximization defined in [11], which has been known as NP-hard. The theorem follows.

Luckily, as we will show that F is submodular and monotone, the greedy algorithm gives a $(1 - 1/e - \epsilon)$ approximation to the optimal solution.

Theorem 3.2. *In the IC model, F is monotone, and F is submodular iff f_v is sumodular.*

Proof. The monotonicity of F is straightforward. We focus on proving that if f_v is submodular, F is submodular. We denote the sum of corresponding adoption in g as $Y_g(S)$. Then we have

$$F(S) = \sum_{all\ \ possible\ \ g} Prob(g) Y_g(S). \tag{3.4}$$

Since a non-negative linear combination of submodular functions is also submodular, we only need to prove $Y_g(S)$ is submodular for all possible g.

For an arbitrary instance of g and $N \subseteq M \subseteq V$, let $A_v |^g_S$ be the active nodes in U_v given g and S. Define the set subtraction: $X - Y = \{x | x \in X \wedge x \notin Y\}$. Now, consider the set $A_v |^g_{N \bigcup u} - A_v |^g_N$, the elements from this set are the elements in $A_v |^g_u$ that are not ready in $A_v |^g_N$. Thus it must contain the elements in $A_v |^g_u$ that are not ready in $A_v |^g_M$. It follows that

$$\{A_v |^g_{M \cup u} - A_v |^g_M\} \subset \{A_v |^g_{N \cup u} - A_v |^g_N\}. \tag{3.5}$$

Sine f_v is submodular, for any $X \subseteq Y \subseteq V$, let $Z \subseteq X$, we have

$$f_v(X \cup Z) - f_v(X) \geq f_v(Y \cup Z) - f_v(Y). \tag{3.6}$$

Let $X = A_v |^g_{N \cup u} - [A_v |^g_{M \cup u} - A_v |^g_M]$, $Y = A_v |^g_M$, $Z = A_v |^g_{M \cup u} - A_v |^g_M$. Since $X \cup Z = A_v |^g_{N \cup u}$ and $Y \cup Z = A_v |^g_{M \cup u}$, from equation 3.6, we have

$$\begin{aligned} &f_v(A_v |^g_{N \cup u}) - f_v(A_v |^g_{N \cup u} - [A_v |^g_{M \cup u} - A_v |^g_M]) \\ \geq &f_v(A_v |^g_{M \cup u}) - f_v(A_v |^g_M). \end{aligned} \tag{3.7}$$

Combining Equation 3.5, 3.7, Assumption 1, we have

$$\begin{aligned} &f_v(A_v |^g_{M \cup u}) - f_v(A_v |^g_M) \\ \leq &f_v(A_v |^g_{N \cup u}) - f_v(A_v |^g_{N \cup u} - [A_v |^g_{M \cup u} - A_v |^g_M]) \\ \leq &f_v(A_v |^g_{N \cup u}) - f_v(A_v |^g_{N \cup u} - [A_v |^g_{N \cup u} - A_v |^g_N]) \\ = &f_v(A_v |^g_{N \cup u}) - f_v(A_v |^g_N). \end{aligned} \tag{3.8}$$

Equation 3.8 leads to

$$\begin{aligned} Y_g(M \cup u) - Y_g(M) &= \sum_{all\ \ v} [f_v(A_v |^g_{M \cup u}) - f_v(A_v |^g_M)] \\ &\leq \sum_{all\ \ v} [f_v(A_v |^g_{N \cup u}) - f_v(A_v |^g_N)] \\ &= Y_g(N \cup u) - Y_g(N). \end{aligned}$$

Thus $Y_g(S)$ is submodular for all g. Next we prove that if F is submodular, f_v is also submodular. For a given node v, assume that $U_v = \{x_1, \cdots, x_s, \cdots, x_t, x\}$ and let $N = \{x_1, \cdots, x_s\}$, $M = \{x_1, \cdots, x_t\}$. Now, we only have to prove that there exist a graph G having $f_v(M \cup x) - f_v(M) \leq f_v(N \cup x) - f_v(N)$.

We construct G with vertices set $U_v = N \cup M \cup \{x\}$ and $x \neq v$. For any node $u \neq v$, $u \in U_v$, let $p(u, v) = 0$. Then we have

$$F(M \bigcup x) - F(M) = f_v(M \cup x) - f_v(M) + f_x(x),$$

$$F(N \bigcup x) - F(N) = f_v(N \cup x) - f_v(N) + f_x(x).$$

Since F is submodular, we have $f_v(M \cup x) - f_v(M) \leq f_v(N \cup x) - f_v(N)$. Thus f_v is submodular.

It has been proved that computing the influence spread is #P-hard under the IC model [6], we will show that computing the adoption spread is #P-hard as well.

Theorem 3.3. *For any adopt function f_v, computing the information adoption F is #P-hard.*

Proof. We will prove the theorem by reducing from the #P-complete s-t connectedness problem [18]. Given a directed graph $G = (V, E)$ and two nodes s and t in the graph, we want to know the number of subgraphs of G in which s is connected to t. It has been proved that this problem is equivalent to computing the probability that s is connected to t when each edge in G is connected with probability of 1/2 [6].

Given an arbitrary instance of the s-t connectedness problem. Let $\{s\}{=}S$ and $p(e) = \frac{1}{2}$ for all $e \in E$, then let $F_G(S)$ denote the information adoption of seed set S in G. We compute $I_1 = F_G(S)$. Then we construct a new graph $G^{'}$ by adding a node t' and a directed edge $(t, t^{'})$ with the propagation probability $p_{t,t'} = 1$. Now let $p_G(S, t)$ denote the probability that node t is activated by S. Next, we compute $I_2 = F_{G'}(S)$. It is straightforward to see that $I_2 = F_G(S) + p_G(S, t) f_{t'}(t \cup t') + (1 - p_G(S, t)) f_{t'}(\emptyset)$. By Assumption 1, $f_{t'}(t \cup t') - f_{t'}(\emptyset) \neq 0$. Thus $p_G(S, t) = \frac{I_2 - I_1 - f_{t'}(\emptyset)}{f_{t'}(t \cup t') - f_{t'}(\emptyset)}$. This means that s-t connectedness problem must be solvable.

4 A Polling Based Algorithm For Adoption Maximization

In this section, we develop an efficient randomized algorithm to solve the information adoption maximization problem. Since we have proved that F is monotone and submodular, we can solve the problem with a greedy strategy and can approximate the optimal solution with a factor of $(1 - 1/e)$ [15]. However, as we have proved that computing F is #P-hard, we need to estimate F with Monte Carlo method in the greedy algorithm. The simulation process is very time consuming, since we need to run Monte Carlo simulations to estimate adoption for arbitrary seed set. Recently, a polling-based algorithm [2] was proposed and was shown to be the most efficient influence maximization algorithm so far.

Let us review the polling method for computing influence spread in the IC model. Given a graph G, a poll is conducted as follows: For each node v, we try to find out which nodes are likely to activate v. We run the Monte Carlo simulation from v in G^T, where G^T is a transpose graph of G. The set of nodes that is discovered by the simulation process is called a RR set. Since the edge (v, u) in G^T is the edge (u, v) in G, the reverse propagation process from v could be used for finding v's potential influencers. Here, assume that we randomly pick M nodes in G and generate M RR sets by the poll. Let $Cov_R(S)$ be the number of RR sets that contain at least one node in S. The key point that makes the polling method work well is that $Cov_R(S)/M$ is an unbiased estimation of $I(S)$, where $I(S)$ is the expected active nodes in G.

Inspired by the polling method, we design a polling based algorithm for our task in a non-trivial process.

4.1 An Unbiased Estimation of F(S) In our problem, to estimate the adoption of a node v, for any sampled graph g, we need to know the active nodes in U_v. In Algorithm 1, we first randomly pick v, then for each node $u \in U_v$, we generate a RR set by the poll. We call all the generated RR sets by U_v a $\mathcal{R}$, i.e., set of RR sets. More details could be found in Algorithm 1.

Algorithm 1 RR sampling

```
1: Initialize $\mathcal{L}$={$\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_M$ }.
2: for $\lambda = 1$ to $M$ do
3:     Choose a node $v$ from $G$ uniformly at random.
4:     Let $U_v = \{v_1, \cdots, v_j\}$.
5:     for $i = 1$ to $j$ do
6:         Simulate information spread, starting from $v_i$
           in $G^T$ and keep the edge results (live edges and
           dead edges) for the next simulations.
7:         Let $R_v(v_i)$ be the set of nodes discovered in the
           simulation process.
8:         Add $R_v(v_i)$ to the $\mathcal{R}_\lambda$.
9:     end for
10: end for
11: return $\mathcal{L}$.
```

To better illustrate, we briefly explain Line 6 to Line 8. Given $\mathcal{R}$, we want to find out nodes that are likely to exert the influence on v. Since a $\mathcal{R}$ is generated in a single live-edge graph, we need to guarantee the simulation accordance. In other words, when generate a $\mathcal{R}$, we can only flip a single coin for an edge. In line 6, if we toss a coin and get a live-edge e (that is, $e \in g$) at the i-th simulations, we keep it in the next simulations (that is, $i + 1, \cdots, j$). Similarly, if we get a dead edge e' (that is, $e' \notin g$), we will not include it in the next simulations. This means that we can generate a $\mathcal{R}$ in a single instance of g^T.

As we will show in Observation 1, we can identify the active nodes in U_v for any seed set S.

OBSERVATION 1. *For any seed set S and $\mathcal{R}$. $A_v = \{v_1, \cdots, v_i\}$ iff $R_v(v_1), \cdots, R_v(v_i)$ are all the RR sets in $\mathcal{R}$ that contain at least one node in S.*

Proof. Suppose that $\mathcal{R}$ can be generated in g^T. If a node v_i is active for given S and g, then there exists a live-path (a simple path consist of live-edges) from $x \in S$ to v_i in g, and also a live-path from v_i to x in g^T. This means that $R_v(v_i)$ contains at least one node in S.

Algorithm 2 RR sampling for triggering adoption functions

```
Initialize $\mathcal{L}$={$\mathcal{R}_1, \mathcal{R}_2, \cdots, \mathcal{R}_M$ }.
for $\lambda = 1$ to $M$ do
    Choose a node $v$ from $G$ uniformly at random.
    Sample a triggering set $T_v$ for node $v$.
    Simulate information spread, starting from $T_v$ in
    $G^T$.
    Let $\mathcal{R}_\lambda$ be the set of nodes discovered in the
    simulation process.
end for
return $\mathcal{L}$.
```

From observation 1, for a given $\mathcal{R}$, we can identify the activated nodes in U_v. Now, we can find an estimation of $F(S)$.

THEOREM 4.1. *Given $\mathcal{L}$ returned by Algorithm 1. For any seed set S,*

$$W(S) = \frac{n \sum_{\mathcal{R} \in \mathcal{L}} f_v(v_1 \cup \cdots \cup v_i)}{M}, \tag{4.9}$$

is an unbiased estimator of $F(S)$, where $R_v(v_1), \cdots, R_v(v_i)$ are the RR sets in $\mathcal{R}$ that contains at least one of the nodes in S.

Proof. Given a $\mathcal{R}$ and seed set S, from observation 1, we have $A_v = \{v_1, \cdots, v_i\}$. Let $Q_g^v(S)$ be the information adoption of v in g. We have

$$Q_g^v(S) = f_v(v_1 \cup \cdots \cup v_i),$$

$$\mathbb{E}[f_v(v_1 \cup \cdots \cup v_i)] = \sum_{all\, possible\, g} Prob(g) Q_g^v(S).$$

In Algorithm 1, we choose node v from G uniformly at random. Thus, we have

$$\begin{aligned}\mathbb{E}[W(S)] &= n \cdot \mathbb{E}[\sum_{\lambda=1}^{M} f_v(v_1 \cup \cdots \cup v_i)]\\ &= n \sum_{all\ v \in G} Prob(v) \sum_{all\ possible\ g} Prob(g) Q_g^v(S)\\ &= \sum_{all\ possible\ g} Prob(g) \sum_{all\ v \in G} Q_g^v(S)\\ &= \sum_{all\ possible\ g} Prob(g) Y_g(S) = F(s).\end{aligned}$$

It follows that $W(S)$ is an unbiased estimator of $F(S)$.

Theorem 4.1 gives the unbiased estimation of $F(S)$ which holds for any adoption function. However, for some special adoption functions, we have a better idea. We called these special adoption functions: triggering adoption functions and formally define it as follows.

- **The triggering adoption function** Each node v independently choose a random "triggering set" T_v according to some distribution over the set U_v. In each sampled graph g, if $A_v \cap T_v \neq \emptyset$, then $f_v = 1$, i.e., the node v will adopt the information.

Theorem 4.2. *Assume that all the adoption functions are triggering adoption functions. Then for a given graph G, and $\mathcal{L}$ returned by Algorithm 2,*

$$W(S) = \frac{n \sum_{\mathcal{R}_i \in \mathcal{L}} min\{|\mathcal{R}_i \cap S|, 1\}}{M}, \tag{4.10}$$

is an unbiased estimator of $F(S)$.

Proof. We can utilize the same technique to prove this theorem in a similar way as the proof of Theorem 4.1.

4.2 Seed Set Selection With the estimation of $F(S)$, we can choose the seed set greedily. In each iteration, we add the node that has the largest marginal gain of adoption. More details can be found in Algorithm 3.

Algorithm 3 optimal seed set selection

```
Initialize a set $S = \emptyset$
for $j = 1$ to $k$ do
    find the node $u$ such that
    $u = \underset{u \in (V \setminus S)}{arg\ max}\, W(S \cup u)$
    add $u$ into $S$.
end for
return $S$
```

In summary, we first use Algorithm 1 or 2 to generate M $\mathcal{R}$s and then feeds the $\mathcal{L}$ to Algorithm 3, we finally will get a $1 - 1/e - \epsilon$ approximate solution to adoption maximization problem. Next, we will show the time complexity of our algorithm.

Theorem 4.3. *In the IC model, if we can compute f_v in $O(1)$ time, then we can get a $(1 - 1/e - \epsilon)$ approximate solution with at least $(1 - 1/n)$ probability in $O(knm \log n / OPT\epsilon^2)$ time, where OPT is maximum adoption for any k size seed set.*

Proof. We get a random variable $x_\lambda = f_v(v_1 \cup \cdots \cup v_i)$ for each $\mathcal{R}$ in Algorithm 1. Thus x_λ is a i.i.d random variables with a distribution on [0,1]. Now, we apply the Chernoff bound [14] to prove the theorem. Applying the conclusion of [17], we can get that M in Algorithm 1 should be no less than

$$\frac{2n(1 - \frac{1}{e})(\log \binom{n}{k} + \log n + \log 2)}{OPT\epsilon^2} \tag{4.11}$$

for achieving a $(1-1/e-\epsilon)$ approximation of the optimal solution with at least $(1 - 1/n)$ probability. Since the time cost of each iteration is $O(m)$, the total time complexity is $O(knm \log n / OPT\epsilon^2)$.

Theorem 4.3 shows the time complexity which holds for any adoption function. However, for triggering adoption functions, we have a better result.

Theorem 4.4. *In the IC model, if for any node v, f_v is the triggering adoption function. We can get a $(1-1/e-\epsilon)$ approximate solution with at least $(1-1/n)$ probability in $O(k(n + m) \log n/\epsilon^2)$ time.*

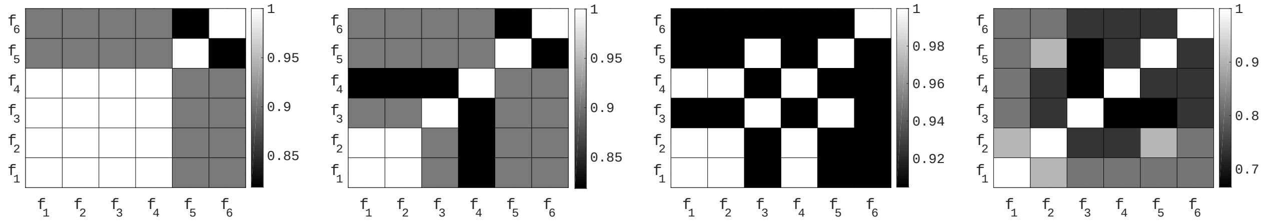
Proof. For the triggering adoption functions, let EPT be the expected number of coin tosses in one iteration, similar to [17], we have $\frac{n}{m} EPT \leq OPT$. Combining Eq 4.11, the time complexity is $O(k(n + m) \log n/\epsilon^2)$.

Note that the adoption function Eq 3.1 and Eq 3.2 are triggering adoption functions, which means we only need $O(k(n + m) \log n/\epsilon^2)$ time to solve the problem.

5 Experiment

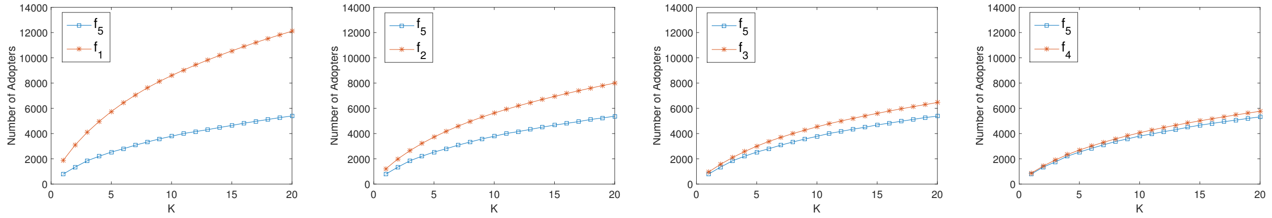
In this section, we first explore the differences among different adoption functions. Then, we verify the correctness of the proposed algorithm. Last, we show the performance of our algorithm.

5.1 Experiment Setup We conduct our experiments on four real network data sets: **Wiki-Vote**, **soc-Epinions**, **soc-Slashdot0922** and **Weibo**. The first three data sets are publicly available in **SNAP** platform. The last one (Weibo) is crawled from Weibo.com, which is a Chinese microblogging website like Twitter. Table 2 shows the details of the data sets.



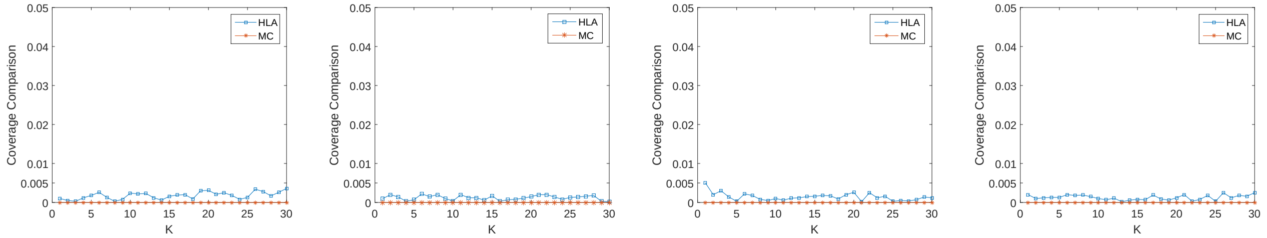
(a) Wiki (b) Epinions (c) Slashdot (d) Weibo

Figure 1: The seed set comparison on four data sets.



(a) $f_{5,1}$ (b) $f_{5,2}$ (c) $f_{5,3}$ (d) $f_{5,4}$

Figure 2: The adoption spread comparison with different approaches on Weibo data set.



(a) Wiki (b) Epinions (c) Slashdot (d) Weibo

Figure 3: The coverage comparison with different compute methods on four data sets.

Table 2: The statistics of the data set

Data set	Nodes	Edges	Average degree
Wiki	7,115	103,689	14.57
Epinions	75,879	508,837	6.71
Slashdot	82,168	948,464	11.54
Weibo	76,491	9,572,897	125.15

In the experiments, the most popular settings of the IC model is adopted. The propagation probability of an edge (u, v) is set to $\frac{\alpha}{indegree(v)}$, where $\alpha = 1$.

Algorithms for Comparison : The algorithms used in the experiments includes:

- *HLA* is the algorithm proposed in section 4.
- *LFG* is a greedy algorithm for the adoption maximization problem with the lazy forward strategy [12], which is utilized in [19].

Adoption Function f : We compare the differences among several adoption functions. The adoption functions we used in the experiments include $f_1 - f_6$ $(f_1, f_2, \cdots f_6)$. $f_1 - f_4$ correspond to Eq. 3.3 with $\theta = 0.1, 0.3, 0.5, 0.7$ respectively, f_5 corresponds to the influence maximization problem in Eq. 3.1 and f_6 corresponds to the information coverage maximization problem in Eq. 3.2.

Iteration times M: For the algorithms proposed in Section 4, we need to set the value of parameter M. Borgs et.al [2] pointed out that we could get a good enough experimental result by setting $M = n \ln n$. For Weibo date set, we set $M = 1,000,000$ while $f = f_5$. Since the number of OPT in f_6 is much larger than in f_5, according to Eq. 4.11, we set $M = 200,000$. More details about can be found in Table 3.

Evaluation Metrics: With the output of our algorithm, we use it as the seed nodes to compute the information adoption with Monte Carlo simulations. In the simulation process, we run MC simulation 10,000 times to get a good estimation of the adoption.

All algorithms are implemented in Java and run on a Linux server with two 2.0GHz Six-Core Intel Xeon E5-2620 and 128G memory.

Table 3: Iteration times M

Data set	Adoption function	M
Wiki,Epinions,Slashdot	f_{1-6}	1,000,000
Weibo	f_{1-4}	70,000
	f_5	1,000,000
	f_6	200,000

5.2 Seed Set Similarity We run the experiments to obtain the seed set whose size is 20. To compare the seed sets obtained by different adoption functions, we compute the Jaccard similarity coefficient of the seed sets. The Jaccard similarity coefficient of set A and B is defined as $\frac{|A \bigcap B|}{|A \bigcup B|}$. The results are shown in Fig 1.

As shown in Fig 1(a), we can see that the seed sets selected with $f_{1,2}$ are similar to each other, the difference between seed sets selected by f_5, f_6 and $f_1 - f_4$ is significant. This phenomenon shows the difference between influence maximization(f_5), information coverage maximization(f_6) and adoption maximization($f_1 - f_4$). For influence maximization problem, we can see that there is almost no other functions selected the same seed sets as f_5, which proves the difference between influence maximization and other adoption maximization once more. For influence maximization(f_5), it focuses on the message propagation(active nodes), while other adoption function focus on the influence that the message receiver get from the network. Thus, there exist significate differences among them. In Fig 1(d), the seed set selected by $f_{3,4}, f_5$ have much more similarly than $f_{1,2}, f_5$. This phenomenon shows that as θ becomes large, the active node plays more important role and the adoption maximization is more similar to the influence maximization.

5.3 Adoption comparison To show the difference between influence spread and information adoption, we report the adoption of the seeds selected by optimizing the proposed adoption function and influence spread in a real world application.

In weibo, users can express their feelings about a message by clicking the "like" button, which is an orthogonal action of retweeting. It can be viewed as a kind of adoption, as the probability of a node "likes" a message is dependent on its active neighbors. In the experiment, we use the adoption function Eq. 3.3 to simulate the "like" spread. Here, the "like" probability functions we used include $f_1 - f_4$.

In Weibo date set, for a given seed set, we compare the coverage of "like" functions($f_1 - f_4$) with traditional influence spread function(f_5). Here, the seed sets are selected by HLA using "like" functions with the size of seed set ranging from 1 to 20. As shown in Figure 2(a)(b), there is a huge gap between the coverage prediction of $f_{1,2}$ and the influence spread function, f_5. In the meantime, the differences in Figure 2(c)(d) are relatively smaller. This phenomenon shows that as θ become larger, the active nodes occupy a large proportion in "like" nodes. As shown in Figure 2(a), it is possible to achieve the coverage of "like" nodes even twice than the classical prediction(f_5).

5.4 Effectiveness validation We run tests on four social networks to obtain information adoption. The size of seed set ranges from 1 to 30. For the purpose of demonstrating the correctness of our algorithm, we use the nodes which were selected by HLA as the seed nodes and run the Monte Carlo simulations(MC) to get the approximation of $F(S)$, denoted it by $F'(S)$. As shown in Figure 3, we compare the difference between $F'(S)$ and $W(S)$(Eq. 4.10). From the figure, we can find that the largest difference is only 5‰. It means that the estimation used in HLA is indeed unbiased, which guarantees the correctness of the algorithm.

Table 4: Efficiency with $f = f_3$(in seconds)

Data set	Wiki	Epinions	Slashdot	Weibo
HLA	586	742	1,822	11,641
LFG	15,334	20,567	49,568	/

Table 5: Efficiency with $f = f_6$(in seconds)

Data set	Wiki	Epinions	Slashdot	Weibo
HLA	13	174	204	6700
LFG	881	940	3200	/

5.5 Efficiency Comparison The running times of different functions ($f_1 - f_4$) are almost the same, since different parameter(θ) settings do not affect the polling based algorithm. Without loss of generality, we only report the running time of f_3. Table 4 shows the running time of different algorithms when the size of the seed set is 20. For triggering adoption functions, we choose f_6, i.e., information converge maximization functions. Table 4 shows the experimental results. It is too time-consuming to run the LFG algorithm, since we need to run full Monte Carlo simulations to estimate the adoption for arbitrary seed set. To improve the efficiency of the algorithm, we utilize Java multi-thread techniques(20 threads). Even so, we can only run the LFG algorithm on Wiki, Epinions and Slashdot date sets. From the table, we can see that although we use multi-thread for LFG, it is much slower than HLA. This result shows that the proposed algorithm indeed outperform the greedy strategy with lazy forward evaluation.

6 Conclusion

In this paper, we discuss about the two basic functions of social spread, i.e., information propagation and information adoption, which are usually confused as equal in prior arts. Specially, we propose a novel framework to generally describe the complete social spread process, in which information adoption process is separately formulated as random events. Based on the general framework, a new problem called information adoption maximization is proposed. Along this line, we first prove that this problem is NP-hard and the computation of information adoption is #P-hard in the IC model, and then design a novel polling-based randomized algorithm to solve the information adoption maximization problem. Finally, we prove the proposed algorithm can approximate the optimal result within certain expected time. Extensive experiments on four real-world data sets demonstrate the effectiveness and efficiency of proposed algorithms, which validates the potential of our framework in social spread simulation.

7 Acknowledgement

This research was supported by grants from the National Natural Science Foundation of China (Grant No. U1605251 and 61703386), the Anhui Provincial Natural Science Foundation (Grant No. 1708085QF140), and the Fundamental Research Funds for the Central Universities (Grant No. WK2150110006).

References

[1] S. Bhagat, A. Goyal, and L. V. Lakshmanan, *Maximizing product adoption in social networks*, in WSDM, 2012, pp. 603–612.

[2] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, *Maximizing social influence in nearly optimal time*, in SODA, 2014, pp. 946–957.

[3] W. Chen, W. Lu, and N. Zhang, *Time-critical influence maximization in social networks with time-delayed diffusion process*, arXiv preprint arXiv:1204.3074, (2012).

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

[5] W. Chen, Y. Wang, and S. Yang, *Efficient influence maximization in social networks*, in SIGKDD, 2009, pp. 199–208.

[6] W. Chen, Y. Yuan, and L. Zhang, *Scalable influence maximization in social networks under the linear threshold model*, in ICDM, 2010, pp. 88–97.

[7] S. Cheng, H. Shen, J. Huang, W. Chen, and X. Cheng, *Imrank: influence maximization via finding self-consistent ranking*, in SIGIR, 2014, pp. 475–484.

[8] J. Goldenberg, B. Libai, and E. Muller, *Talk of the network: A complex systems look at the underlying process of word-of-mouth*, Marketing letters, 12 (2001), pp. 211–223.

[9] A. Goyal, W. Lu, and L. V. Lakshmanan, *Celf++: optimizing the greedy algorithm for influence maximization in social networks*, in WWW, 2011, pp. 47–48.

[10] L. W. Goyal, Amit and Lakshmanan, *Simpath: An efficient algorithm for influence maximization under the linear threshold model*, in ICDM, 2011, pp. 211–220.

[11] D. Kempe, J. Kleinberg, and É. Tardos, *Maximizing the spread of influence through a social network*, in SIGKDD, 2003, pp. 137–146.

[12] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, *Cost-effective outbreak detection in networks*, in SIGKDD, 2007, pp. 420–429.

[13] Q. Liu, B. Xiang, N. J. Yuan, E. Chen, H. Xiong, Y. Zheng, and Y. Yang, *An influence propagation view of pagerank*, TKDD, 11 (2017), p. 30.

[14] R. Motwani and P. Raghavan, *Randomized algorithms*, 2010.

[15] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, *An analysis of approximations for maximizing submodular set functionsi*, Mathematical Programming, 14 (1978), pp. 265–294.

[16] Y. Tang, Y. Shi, and X. Xiao, *Influence maximization in near-linear time: A martingale approach*, in SIGMOD, 2015, pp. 1539–1554.

[17] Y. Tang, X. Xiao, and Y. Shi, *Influence maximization: Near-optimal time complexity meets practical efficiency*, in SIGMOD, 2014, pp. 75–86.

[18] L. G. Valiant, *The complexity of enumeration and reliability problems*, SIAM Journal on Computing, 8 (1979), pp. 410–421.

[19] Z. Wang, E. Chen, Q. Liu, Y. Yang, Y. Ge, and B. Chang, *Maximizing the coverage of information propagation in social networks*, in IJCAI, 2015, pp. 2104–2110.

[20] B. Xiang, Q. Liu, E. Chen, H. Xiong, Y. Zheng, and Y. Yang, *Pagerank with priors: An influence propagation perspective.*, in IJCAI, 2013.

[21] T. Xu, H. Zhong, H. Zhu, H. Xiong, E. Chen, and G. Liu, *Exploring the impact of dynamic mutual influence on social event participation*, in SDM, 2015, pp. 262–270.

[22] T. Xu, H. Zhu, E. Chen, B. Huai, H. Xiong, and J. Tian, *Learning to annotate via social interaction analytics*, Knowledge and information systems, 41 (2014), pp. 251–276.

[23] Y. Yang, E. Chen, Q. Liu, B. Xiang, T. Xu, and S. A. Shad, *On approximation of real-world influence spread*, in ECML-PKDD, Springer, 2012, pp. 548–564.

[24] Y. Yang, X. Mao, J. Pei, and X. He, *Continuous influence maximization: What discounts should we offer to social network users?*, in SIGMOD, 2016, pp. 727–741.