Efficient Influence Maximization in Social Networks

Chuan Zhou and Peng Zhang

Abstract. In social networks, influence maximization, defined as finding a small subset of nodes that maximizes spread of influence, is NP-hard under both Linear Threshold (LT) and Independent Cascade (IC) models, where a line of heuristic algorithms have been proposed. The simple greedy algorithm achieves an approximation ratio of $1-1/e$. The advanced CELF algorithm, by exploiting the submodular property of the spread function, runs 700 times faster than the simple greedy algorithm on average. However, CELF is still inefficient, as the first iteration calls for N times of spread estimations (N is the number of nodes in networks), which is computationally expensive, especially on large networks. To this end, in this paper we derive an upper bound function for the spread function. The bound can be used to reduce the number of Monte-Carlo simulation calls in greedy algorithms, especially in the first iteration of initialization. Based on the upper bound, we propose an efficient *Upper Bound based Lazy Forward* algorithm (**UBLF** in short), by incorporating the bound into the CELF algorithm. We test and compare our algorithm with prior algorithms on real-world data sets. Experimental results demonstrate that UBLF, compared with CELF, reduces more than 95% Monte-Carlo simulations and achieves at least $2-5$ times speed-raising.

Influence maximization, social networks, Independent Cascade model, greedy algorithms.

1. Introduction

Social networks, such as Facebook, Flickr, Twitter, have become important mediums, with rapidly increasing users over the past few years. Through the powerful effect of word-of-mouth in social networks, social influence plays a critical role in affecting people's emotions, opinions and behaviors.

Much attention has been attracted to the study of social influence propagation in social networks, among which one of the fundamental problems is influence maximization [1, 2]. The seminal work, by Kempe, Kleinberg and Tardos [13], first formulates influence maximization as a discrete optimization problem: Given a directed social graph with users as nodes, edge weights reflecting influence between users and a budget/threshold number k, finding k nodes in the graph, such that by activating these nodes, the expected spread of the influence can be maximized, based on a given stochastic influence propagation model.

Two popularly used stochastic influence propagation models are the *Independent Cascade* (IC) and *Linear Threshold* (LT) models [13]. In both models, at any time step, a user is represented as a binary variable with either active (an adopter of the product) or inactive, and influence propagates until no more users can become active. The major difference between the two models is, in the IC model when an inactive user becomes active at a time step t, it gets exactly one chance to independently activate its currently inactive neighbors at the next time step $t+1$; while in the LT model, the sum of incoming edge weights on any node is assumed to be at most 1, every user chooses an activation threshold uniformly at random from $[0, 1]$, and at any time step, a node becomes activated if the sum of incoming edge weights from the active neighbors exceeds the threshold.

M. Kosta and T. Sturm, eds. Proceedings of the MACIS 2013, Nanning, China, December 11-13, 2013.

Influence maximization under both IC and LT models is NP-hard, and the spread function is monotone and submodular [13]. A set function $f: 2^U \to \mathbb{R}^+$ is monotone, if $f(S) \leq f(T)$ whenever $S \subseteq T \subseteq U$. The set function is submodular, if $f(S \cup \{w\}) - f(S) \geq f(T \cup \{w\}) - f(T)$ for all $S \subseteq T$ and $w \in U \backslash T$. Intuitively, submodularity indicates that f has dimishing margin returns when adding more nodes into the set.

Exploiting these two properties, Kempe et al. [13] presented a simple greedy algorithm which repeatedly chooses the node with the maximum marginal gain and adds it to the seed set, until the budget k is reached. However, computing exact marginal gain (or exact expected spread) under both the IC and LT models is $\#P$-hard [5, 6]. Hence, it is usually estimated by running Monte Carlo (MC) simulations. The simple greedy algorithm can approximate the solution within a factor of $(1 - 1/e - \epsilon)$ for any $\epsilon > 0$.

Unfortunately, the simple greedy algorithm suffers from two major sources of inefficiency. (I) The MC simulations that run sufficiently many times (typically 10,000) to obtain an accurate estimate of spread, has been proved very expensive, especially when the network is large. (II) The greedy algorithm calls for $O(kN)$ iterations at the spread estimation step, where k is the size of initially picked seed set, and N is the number of nodes. When N is large, the algorithm has low efficiency.

Considerable work has been conducted to tackle the above two limitations. To address the first limitation, many heuristic solutions have been proposed to improve the efficiency of seed selection, *e.g.* [4, 5, 6, 11]. In these work, the heuristic algorithms can reduce computational cost in orders of magnitude, with competitive results of influence spread level. However, none of them has a theoretical guarantee with reliable results. In other words, it is unknown how far these heuristic solutions approximate the optimal solution. One can only borrow the simple greedy algorithm as the benchmark for performance testing.

To tackle the second limitation, a representative work exploited the submodular property of the objective function, and proposed a Cost-Effective Lazy Forward selection (CELF) algorithm. The algorithm can significantly reduce the number of MC simulation calls in spread estimations. The principle behind is that the marginal gain of a node in the current iteration cannot be more than that in previous iterations, and thus the number of spread estimation calls can be greatly pruned, with report that CELF improves the running time of the simple greedy algorithm by up to 700 times.

Although CELF significantly improves the running time of the simple greedy algorithm, it is still quite slow on large networks [4]. In particular, in the initialization step, CELF needs to estimate the spread using Monte-Carlo for each node in a network, resulting in N times of Monte-Carlo calls (N is the total number of nodes in the network), which is time-consuming, especially when the network is very large. The limitation leads to a question that, *can we derive an upper bound of spreads which can be used to prune unnecessary spread estimations (Monte-Carlo calls) in the CELF algorithm*? To the best of our knowledge, there is no work in the literature that mathematically discuss the upper bound properties of the spread function.

Motivated by the above question, in this paper we derive an upper bound for greedy algorithms in influence maximization problem. Based on the bound, we propose a new greedy algorithm *Upper Bound based Lazy Forward* (UBLF for short), which outperforms the original CELF algorithm. We summarize the contributions of the paper as follows:

1. We derive an upper bound for spread $\sigma_I(S)$ whose exact expected estimation under the IC model is $\#P$-hard.
2. We propose, based on the upper bound, an efficient UBLF algorithm to discover the influential nodes in social networks.
3. We conduct extensive experiments on real-world data sets to demonstrate the performance of the proposed UBLF algorithm.

TABLE 1. Major variables in the paper

Variables	Descriptions
$G = (V, E)$	social network G with node set V edge set E
N	number of nodes in the network G
S	initial seed set
S_t	set of activated nodes at step t
$\|S\|$	number of nodes in S
k	number of seeds to be selected
$Par(v)$	set of parents of node v
$\mathbb{P}^S$	probability measure with the seed set S
$\mathbb{E}^S$	expectation operator with the seed set S
Π_t^S	row vector with probabilities as in Eq. (7)
PP	N by N propagation probability matrix
$\mathbf{1}$	column vector with all elements being 1

2. IC Model and Greedy Algorithm

Consider a directed graph $G = (V, E)$ with N nodes in V and edge labels $pp : E \to [0, 1]$. For each edge $(u, v) \in E$, $pp(u, v)$ denotes the propagation probability that v is activated by u through the edge. If $(u, v) \notin E$, $pp(u, v) = 0$. Let $Par(v)$ be the set of parent nodes of v, *i.e.*,

$$Par(v) := \big\{u \in V,\ (u, v) \in E\big\}.$$

Given an initially activated set $S \subseteq V$, the independent cascade (IC) model works as follows. Let $S_t \subseteq V$ be the set of nodes that are activated at step $t \geq 0$, with $S_0 = S$. Then, at step $t + 1$, each node $u \in S_t$ may activate its out-neighbors $v \in V \setminus \cup_{0 \leq i \leq t} S_i$ with an independent probability of $pp(u, v)$, where $\cup_{0 \leq i \leq t} S_i := S_0 \cup S_1 \cup \cdots \cup S_t$. Thus, a node $v \in V \setminus \cup_{0 \leq i \leq t} S_i$ is activated at step $t+1$ with the probability

$$1 - \prod_{u \in S_t \cap Par(v)} \big(1 - pp(u, v)\big) \tag{1}$$

where the subscript $u \in S_t \cap Par(v)$ means that node u, a parent node of v, is activated at step t. If node v is successfully activated, then it is added into the set S_{t+1}. The process ends at a step τ with $S_\tau = \varnothing$. Obviously, the propagation process has $N - |S|$ steps at most, as there are at most $N - |S|$ nodes outside the seed set S. Let $S_{\tau+1} = \varnothing, \cdots, S_{N-|S|} = \varnothing$, if $\tau < N - |S|$. Note that each activated node only has one chance to activate its out-neighbors at the step right after itself is activated, and each node stays activated once it is activated by others.

In the IC model, the influence spread of a seed set S, which is the expected number of activated nodes by S, is denoted as $\sigma_I(S)$ as follow,

$$\sigma_I(S) := \mathbb{E}^S\Big[\Big|\bigcup_{t=0}^{N-|S|} S_t\Big|\Big] \tag{2}$$

where $\mathbb{E}^S$ is the expectation operator with set S, the subscript 'I' denotes the IC model, $\bigcup_{t=0}^{N-|S|} S_t := S_0 \cup \cdots \cup S_{N-|S|}$ is the sets of nodes activated in all $N - |S| + 1$ steps.

The above equation provides us convenience to treat the global influence function, $\sigma_I(S)$, as a summation of locally activated node sets S_t $(1 \leq t \leq N - |S|)$, as we will see in *Proposition 1* in the next section.

The influence maximization problem, under the IC model, is to find a subset $S^* \subseteq V$ such that $|S^*| = k$ and $\sigma_I(S^*) = \max\big\{\sigma_I(S) \ \big|\ |S| = k, S \subseteq V\big\}$, *i.e.*,

$$S^* = \arg \max_{|S|=k, S \subseteq V} \sigma_I(S) \tag{3}$$

where k is a given parameter. The problem, as proved in the work [13], is NP-hard, and a constant-ratio approximation algorithm is feasible.

In the work [13], it is shown that the objective function $\sigma_I(S)$ in Eq.(3) has the submodular and monotone properties [13] with $\sigma_I(\varnothing) = 0$. Thus, the problem in Eq.(3) can be approximated by the greedy algorithm as shown in Algorithm 1. Theoretically, a non-negative real valued function f on subsets of V is submodular, if $f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$ for all $v \in V$ and $S \subseteq T \subseteq V$. Thus, f has diminishing marginal return. Moreover, f is monotone, if $f(S) \leq f(T)$ for all $S \subseteq T$. For any submodular and monotone function f with $f(\varnothing) = 0$, the problem of finding a set S of size k that maximizes $f(S)$ can be approximated by the greedy algorithm in Algorithm 1. The algorithm iteratively selects a new seed u that maximizes the incremental change of f, into the seed set S, until k seeds are selected. It is shown that the algorithm guarantees the approximation ratio $f(S)/f(S^*) \geq 1 - 1/e$, where S is the output of the greedy algorithm and S^* is the optimal solution.

Algorithm 1: Greedy(k,f)

1: initial $S = \varnothing$
2: **for** $i = 1$ to k **do**
3: select $u = \arg\max_{w \in V \setminus S}(\sigma_I(S \cup \{w\}) - \sigma_I(S))$
4: $S = S \cup \{u\}$
5: **end for**
6: output S

In Algorithm 1, an important issue is that there is no efficient way to compute $\sigma_I(S)$ given a set S. Kempe et al. [13] run Monte-Carlo simulations of the propagation model for $10,000$ trials to obtain an accurate estimate of the expected spread, leading to very expensive computation cost. Chen et al. [5] pointed out that computing $\sigma_I(S)$ is actually #P-hard, by showing a reduction from the counting problem of s-t connectness in a graph.

Based on the above observations, in order to improve the efficiency of Algorithm 1, one can either reduce the call times of Monte-Carlo simulations in computing $\sigma_I(S)$, or develop advanced heuristic algorithms which reduce the number of iterations without accuracy guarantees.

3. Analysis and Approaches

In this part, we aim to derive an upper bound of $\sigma_I(S)$, as the exact computation of $\sigma_I(S)$ is #P-hard [5]. To the best of our knowledge, we are the first to discuss the upper bound of the influence spread in the literature. The upper bound provides us a new view to design efficient algorithms in the field of influence maximization.

3.1. Upper bound of $\sigma_I(S)$

Before introducing the bound in Theorem 3 and Corollary 4, we introduce two preparations first. Let $\mathbb{P}^S(v \in S_t)$ denote the probability that node v gets activated at step t under the seed S, we have the first preparation as follow,

Proposition 1. *For $S \subseteq V$, the spread $\sigma_I(S)$ under IC model can be calculated as*

$$\sigma_I(S) = \sum_{t=0}^{N-|S|} \sum_{v \in V} \mathbb{P}^S(v \in S_t). \tag{4}$$

Proposition 1 *reveals that we can treat the global influence measure $\sigma_I(S)$ as a summation of all $N - |S|$ propagation steps of local probabilities $\{\mathbb{P}^S(v \in S_t) : t \geq 0, v \in V\}$.*

Based on Proposition 1, a following question is, *what is the relationship between two sets,*

$$\{\mathbb{P}^S(v \in S_t) : v \in V\}$$

and

$$\big\{\mathbb{P}^S(v \in S_{t-1}) : v \in V\big\}.$$

Proposition 2. *For $t = 1, 2, \ldots, N - |S|$ and $v \in V$, we have the following inequation*

$$\mathbb{P}^S(v \in S_t) \leq \sum_{u \in V} \mathbb{P}^S\big(u \in S_{t-1}\big) pp(u, v). \tag{5}$$

Proposition 2 *clearly identifies the ordering relationship between two adjacent elements in the series $\mathbb{P}^S(v \in S_0), \cdots, \mathbb{P}^S(v \in S_t), \cdots, \mathbb{P}^S(v \in S_{N-|S|})$.*

Now we simplify the results in Propositions 1 and 2 into the form of matrix. Let PP be the propagation probabilities matrix with the (u, v) position's element being $pp(u, v)$. For $t = 0, 1, 2, \ldots, N-|S|$, denote the row vector

$$\Pi_t^S = (\pi_t^S(v)) \tag{6}$$

as the probabilities of nodes being activated at step t, *i.e.*,

$$\pi_t^S(v) := \mathbb{P}^S(v \in S_t). \tag{7}$$

Then, Proposition 1 can be rewritten as

$$\sigma_I(S) = \sum_{t=0}^{N-|S|} \Pi_t^S \cdot \mathbf{1} \tag{8}$$

where $\mathbf{1}$ is a column vector with all elements being 1, and Proposition 2 can be rewritten as

$$\Pi_t^S \leq \Pi_{t-1}^S \cdot PP \tag{9}$$

where PP denotes the propagation probability matrix. Then we have Theorem 1 as follow,

Theorem 3. *The upper bound of $\sigma_I(S)$ is*

$$\sigma_I(S) \leq \sum_{t=0}^{N-|S|} \Pi_0^S \cdot PP^t \cdot \mathbf{1}. \tag{10}$$

Based on Eq.(10) in Theorem 1, one may easily raise the following two questions,

- The function $\sigma_I(S)$ is bounded by a summation of series $\sum_{t=0}^{N-|S|} \Pi_0^S \cdot PP^t \cdot \mathbf{1}$, if we relax the series to $\sum_{t=0}^{\infty} \Pi_0^S \cdot PP^t \cdot \mathbf{1}$, then in what condition the series will be convergent?
- If the relaxed series is convergent, what's the limit of convergence, i.e., $\sum_{t=0}^{\infty} \Pi_0^S \cdot PP^t \cdot \mathbf{1} =?$.

In the sequel, we derive Corollary 1 to answer the two questions.

Corollary 4. *If the propagation probability satisfies the conditions*

$$\max_v \sum_u pp(u, v) < 1 \ \textit{or} \ \max_u \sum_v pp(u, v) < 1, \tag{11}$$

*then the series in Eq.(*10*) is convergent, and the limit of convergence exists,*

$$\sigma_I(S) \leq \Pi_0^S \cdot (E - PP)^{-1} \cdot \mathbf{1} \tag{12}$$

where E is an unit matrix and $(E - PP)^{-1}$ is the inverse of $(E - PP)$.

As the inverse $(E - PP)^{-1}$ may be intractable when the size of network is enormous, we adopt the following method to calculate $(E-PP)^{-1} \cdot \mathbf{1}$. For $t \geq 0$, we denote the column vector $\mathbf{a}_t := PP^t \cdot \mathbf{1}$, so we have

$$(E - PP)^{-1} \cdot \mathbf{1} = \sum_{t=0}^{\infty} PP^t \cdot \mathbf{1} = \sum_{t=0}^{\infty} \mathbf{a}_t$$

where $\mathbf{a}_{t+1} = PP \cdot \mathbf{a}_t$. With this iteration, we sum up $\mathbf{a}_0, \mathbf{a}_1, \cdots$ until some $\mathbf{a}_n$ with L_1-norm less than 10^{-6}. This transformation saves memory space during calculation, as it stores vectors instead of matrixes in the memory.

Based on Eq.(11), we can observe that the matrix series converges on condition that either the total influence to any node is less than 1, or the total influence diffused by any node is less than 1. In real-world social networks, the propagation probability is often very small. Thus, Condition (11) usually stands.

Now we use an example to explain the bound calculation.

Example 5. Given a graph G, as shown in Fig. 5, with propagation probability matrix in Eq. (13),

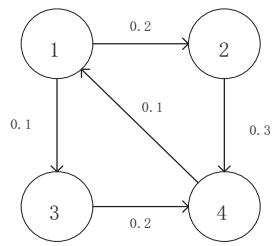

FIGURE 1. An illustration of the upper bound calculation.

$$PP = \begin{pmatrix} 0 & 0.2 & 0.1 & 0 \\ 0 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 0.2 \\ 0.1 & 0 & 0 & 0 \end{pmatrix} \tag{13}$$

we have

$$\begin{aligned} & (E - PP)^{-1} \cdot \mathbf{1} \\ = & \begin{pmatrix} 1 & -0.2 & -0.1 & 0 \\ 0 & 1 & 0 & -0.3 \\ 0 & 0 & 1 & -0.2 \\ -0.1 & 0 & 0 & 1 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \\ = & \begin{pmatrix} 1.3911 \\ 1.3417 \\ 1.2278 \\ 1.1391 \end{pmatrix} \end{aligned}$$

Based on Corollary 4, the upper bound of spread $\sigma_I(S)$ with the seed set $S = \{②, ④\}$ can be calculated as follow,

$$\begin{aligned}\sigma_I(②, ④) &\leq \Pi_0^{(②,④)} \cdot (E - PP)^{-1} \cdot \mathbf{1} \\ &= (0\ 1\ 0\ 1) \cdot \begin{pmatrix} 1.3911 \\ 1.3417 \\ 1.2278 \\ 1.1391 \end{pmatrix} = 2.4808\end{aligned}$$

□

3.2. UBLF algorithm

The Cost-Effective Lazy Forward (CELF) algorithm exploited the submodular property to improve the simple greedy algorithm. The idea is that the marginal gain of a node in the current iteration cannot be more than that in previous iterations, and thus the number of spread estimations can be significantly reduced.

However, CELF demands N spread estimations to establish the initial bounds of marginal increments, which is time expensive on large graphs. In our Upper Bound based Lazy Forward (UBLF) algorithm, we use the derived upper bound to further reduce the number of spread estimations in the initialization step. In doing so, the nodes will be all ranked by their upper bound scores, which can potentially reduce the computation cost of the original CELF algorithm. We use Example 2 for illustration.

Example 6. We still use the network in Fig. 5 for explanation. **The goal here is to find the top-1 node with maximal influence.** For a specific node ①, according to Corollary 1, we have its upper bound as follow,

$$\begin{aligned}\sigma_I(①) &\leq \Pi_0^{①} \cdot (E - PP)^{-1} \cdot \mathbf{1} \\ &= (1\ 0\ 0\ 0) \cdot \begin{pmatrix} 1.3911 \\ 1.3417 \\ 1.2278 \\ 1.1391 \end{pmatrix} = 1.3911\end{aligned}$$

By the same logic, we can obtain

$$\sigma_I(②) \leq 1.3417,\ \sigma_I(③) \leq 1.2278,\ \sigma_I(④) \leq 1.1391$$

Obviously, the upper bound of $\sigma_I(①)$, 1.3911 , is the largest in the graph. Thus, we use Monte-Carlo simulation to estimate $\sigma_I(①)$ (or explicitly calculate it due to the simple structure), and get

$$\sigma_I(①) = 1.3788$$

Now, we can observe that 1.3788 is already larger than the upper bounds of $\sigma_I(②)$, $\sigma_I(③)$ and $\sigma_I(④)$. Thus, we don't need extra Monte-Carlo simulations to estimate the other three nodes. Hence, we obtain the node ① which has the maximal influence in the graph.

We can observe that, by introducing the upper bounds, we can greatly reduce the number of Monte-Carlo simulation calls. In Example 2, we use only one Monte-Carlo simulation call, while in the CELF algorithm, we need four Monte-Carlo simulation calls. □

We summarize the UBLF algorithm in Algorithm 2.

In Algorithm 2, the column vector, $\delta = \{\delta_u\}$, denotes upper bounds of marginal increments under the current seed set S, i.e.,

$$\delta_u \geq \sigma_I(S \cup \{u\}) - \sigma_I(S).$$

Before searching for the first node (i.e. $S = \emptyset$), we estimate an upper bound for each node by Corollary 4. Then, the algorithm proceeds similar to CELF. Note that by the properties of submodular,

Algorithm 2: UBLF

01: Input: the propagation probability matrix PP of a graph $G = (V, E)$, a budget k
02: Output: The most influential set S with k nodes
03: initial $S \leftarrow \varnothing$ and $\delta \leftarrow (E - PP)^{-1} \cdot \mathbf{1}$
04: **for** $i = 1$ to k **do**
05: set $I(v) \leftarrow 0$ for $v \in V \backslash S$
06: **while** TRUE **do**
07: {
08: $u \leftarrow \arg\max_{v \in V \backslash S} \delta_v$
09: **if** $I(u) = 0$
10: $\delta_u \leftarrow MC(S \cup \{u\}) - MC(S)$
11: $I(u) \leftarrow 1$
12: **end if**
13: **if** $\delta_u \geq \max_{v \in V \backslash (S \cup \{u\})} \delta_v$
14: $S \leftarrow S \cup \{u\}$
15: break
16: **end if**
17: }
18: **end for**
19: output S

these upper bounds of marginal increments can be dynamically adjusted by MC simulations, which becomes smaller with the algorithm carrying on.

In the algorithm, $MC(S)$ denotes that we employ the Monte-Carlo simulation to estimate $\sigma_I(S)$ for the initial set S, $I(v) = 0$ denotes that the Monte-Carlo has not been used on the node v yet in the current iteration, $I(v) = 1$ means the Monte-Carlo simulation has already been computed on the node v.

3.3. Discussions on the upper bound

We have derived the upper bound for the spread function $\sigma_I(S)$, and developed a new UBLF algorithm. One may have the following concern: *How large is the gap between the estimated upper bound and the real value of $\sigma_I(S)$?*

In this part, we aim to explain that, under Conditions **(I)** the propagation probability $\{pp(u, v)\}$ is relatively small, and **(II)** the number of nodes N is large enough, the upper bound asymptotically approximates the real value of $\sigma_I(S)$.

Formally, if the two conditions are met, we can relax Eq.(12) to Eq.(14) as follow,

$$\sigma_I(S) \approx \Pi_0^S \cdot (E - PP)^{-1} \cdot \mathbf{1} \tag{14}$$

In the sequel, we will explain why Eq.(17) holds under the two given conditions. We first present two lemmas, based on which we derive the result in Eq.(17).

Lemma 7. *For small positive numbers $x_1, x_2, \ldots, x_n$, it follows that*

$$1 - \prod_{i=1}^{n}(1 - x_i) \approx \sum_{i=1}^{n} x_i. \tag{15}$$

We use Example 3 to explain Lemma 1.

Example 8. In Fig.2, nodes w and u are newly activated at step t, and they are both parents of v, with $pp(w, v) = 0.1$ and $pp(u, v) = 0.2$. Then, the probability of node v being activated at step $t + 1$

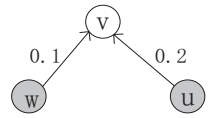

FIGURE 2. An example of 1D intervals

under the IC model is

$$1 - (1 - 0.1)(1 - 0.2) = 0.28,$$

and we can observe that the probability is also roughly equivalent to the value

$$0.1 + 0.2 = 0.3.$$

The two values are closer if the propagation probabilities $pp(w, v)$ and $pp(u, v)$ become smaller. □

Based on Lemma 1, we have

$$1 - \prod_{u \in S_{t-1}} (1 - pp(u, v)) \approx \sum_{u \in S_{t-1}} pp(u, v) \tag{16}$$

Lemma 9. *If the number of node N is large enough, we have*

$$\mathbb{P}^S\big(v \notin \cup_{r=0}^{t-1} S_r \big| u \in S_{t-1}\big) \approx 1 \tag{17}$$

We incorporate the two lemmas into the proof of Proposition 2, and obtain a relaxed version of Eq.(5) as follow,

$$\mathbb{P}^S(v \in S_t) \approx \sum_{u \in V} \mathbb{P}^S\big(u \in S_{t-1}\big) pp(u, v).$$

By using the matrix form, we rewrite the above approximation as follows,

$$\Pi_t^S \approx \Pi_{t-1}^S \cdot PP \tag{18}$$

Incorporating the above approximation into the proofs of Theorem 1 and Corollary 1, we obtain the final result in Eq.(14).

To sum up, when the two given conditions are satisfied, the upper bound well approximates the spread function $\sigma_I(S)$. Hence, we have high accuracy guarantee to use the bound, $(E - PP)^{-1} \cdot \mathbf{1}$, as the pruning criterion. Specifically, we can choose k nodes with the highest values in the column vector $(E - PP)^{-1} \cdot \mathbf{1}$ as the initial seed set. For instance, in Example 5, we have

$$(E - PP)^{-1} \cdot \mathbf{1} = \begin{pmatrix} 1.3965 \\ 1.3684 \\ 1.2279 \\ 1.1396 \end{pmatrix}$$

If $k = 1$, we can simply choose node ① as the most influential seed node. If $k = 2$, we choose nodes ① and ② as the most influential seed set. We call this approach as the **Upper Bound based algorithm** (UBound in short). The algorithm is summarized below.

Algorithm 3: UBound

1: Input: the propagation probability matrix PP of a graph $G = (V, E)$, a budget k
2: Output: The most influential set S with k nodes
3: **Score** $\leftarrow (E - PP)^{-1} \cdot \mathbf{1}$
4: Select the biggest k nodes in **Score** as the output S

4. Conclusion

In this paper, we derived an upper bound for the spread function in solving influence maximization problem in social networks. Based on the bound, we proposed a new Upper Bound based Lazy Forward algorithm (**UBLF** in short). Compared with CELF, UBLF significantly reduces the number of Monte-Carlo calls, e.g., more than **95% reduction of Monte-Carlo calls** than CELF in our experiments. The experimental results also verify that UBLF can enhance CELF's efficiency by **2-5 times** at least.

Acknowledgment

This work was supported by the NSFC (No. 61003167), IIE Chinese Academy of Sciences (No. Y3Z0062101), 863 projects (No. 2011AA010703 and 2012AA012502) and the Strategic Leading Science and Technology Projects of Chinese Academy of Sciences(No.XDA06030200)

References

[1] N. Barbieri, F. Bonchi, and G. Manco, "Topic-aware Social Influence Propagation Models," in ICDM 2012.

[2] F. Bonchi, "Influence propagation in social networks: A data mining perspective," IEEE Intelligent Informatics Bulletin, Vol.12, No.1, 2011.

[3] S. Brin, and L. Page, "The anatomy of a large-scale hypertextual web search engine," Computer Networks, vol. 30, no. 1-7, pp. 107-117, 1998.

[4] J. Guo, P. Zhang, C. Zhou, Y.Cao and L. Guo, "Personalized Influence Maximization on Social Networks," in CIKM 2013.

[5] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in KDD 2010.

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

[7] P. Domingos and M. Richardson, "Mining the network value of customers," in KDD 2001.

[8] A. Goyal, F. Bonchi, and L. V. Lakshmanan, "Learning influence probabilities in social networks," in WSDM 2010.

[9] C. Zhou, P. Zhang, J. Guo, X. Zhu and L. Guo, "UBLF: An upper bound based approach to discover influential nodes in social networks" in IEEE ICDM 2013.

[10] A. Goyal, W. Lu, and L. V. S. Lakshmanan, "CELF++: Optimiz-ing the Greedy Algorithm for Influence Maximization in Social Networks," in WWW 2011.

[11] A. Goyal, W. Lu, and L. V. S. Lakshmanan, "SIMPATH: An Efficient Algorithm for Influence Maximization under the Linear Threshold Model," in ICDM 2011.

[12] R. A. Horn and C. R. Johnson, "Matrix analysis," Cambridge university press, 1990.

[13] D. Kempe, J. M. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in KDD 2003.

Chuan Zhou
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
e-mail: zhouchuan@iie.ac.cn

Peng Zhang
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
e-mail: zhangpeng@iie.ac.cn