On the Upper Bounds of Spread for Greedy Algorithms in Social Network Influence Maximization

Chuan Zhou, Peng Zhang, Wenyu Zang, and Li Guo

Abstract—Influence maximization, defined as finding a small subset of nodes that maximizes spread of influence in social networks, is NP-hard under both Independent Cascade (IC) and Linear Threshold (LT) models, where many greedy-based algorithms have been proposed with the best approximation guarantee. However, existing greedy-based algorithms are inefficient on large networks, as it demands heavy Monte-Carlo simulations of the spread functions for each node at the initial step [7]. In this paper, we establish new upper bounds to significantly reduce the number of Monte-Carlo simulations in greedy-based algorithms, especially at the initial step. We theoretically prove that the bound is tight and convergent when the summation of weights towards (or from) each node is less than 1. Based on the bound, we propose a new *Upper Bound based Lazy Forward* algorithm (**UBLF** in short) for discovering the top-k influential nodes in social networks. We test and compare UBLF with prior greedy algorithms, especially CELF [30]. Experimental results show that UBLF reduces more than 95% Monte-Carlo simulations of CELF and achieves about $2-10$ times speedup when the seed set is small.

Index Terms—Influence maximization, social networks, greedy algorithms, upper bounds.

1 Introduction

Recent years witness the increasing popularity of online social networks, where users are connected by various social relationships. Online social networks enable convenient information dissemination and marketing campaign by allowing ideas and behaviors to fast propagate along social connections in the word-of-mouth manner. Many companies have made efforts to popularize or promote their brands or products over online social networks by launching campaigns akin to viral marketing [10]. The success of viral marketing is rooted in the interpersonal influence, which has been empirically studied in various contexts [13], [26], [35].

Influence maximization is a fundamental problem for viral marketing. The seminal work by Kempe et al. [25] first formulated 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, influence maximization aims to find k nodes in the graph, such that by activating these nodes, the expected spread of the influence can be maximized. Two commonly-used influence spread models are the *independent cascade* (IC) and *linear threshold* (LT) models. Influence maximization under both IC and LT models is NP-hard [25]. Exploiting the monotone and submodular properties of the spread function, Kempe et al. [25] presented a greedy algorithm (GREEDY in short) which repeatedly chooses the node with the maximum marginal gain and adds it to the seed set, until the budget k is reached. Computing exact marginal gain (or exact expected spread) with either model is $\#P$-hard [8], [9]. In practice, it is usually estimated by running Monte-Carlo (MC in short) simulations. GREEDY can approximate the optimal solution within a factor of $(1-1/e-\epsilon)$ for any $\epsilon > 0$. According to Feige's inapproximation result [15], this is the best approximation guarantee one can hope for.

GREEDY is a popular benchmark in influence maximization. However, GREEDY suffers two sources of inefficiency. First, the MC simulations that run sufficiently many times (generally over 10,000 times) to obtain an accurate estimate of spread has been proved computationally expensive in a large network. Second, it demands $O(kN)$ iterations at the spread estimation step, where k is the size of an initially picked seed set, and N is the number of nodes. When N is large, the efficiency drops severely.

Considerable work has been conducted to tackle the inefficiency problem in influence maximization. For example, many heuristic algorithms have been proposed to improve the efficiency of seed selection, such as the DegreeDiscount [7], MIA [8], DAG [9], SIMPATH [20], ShortestPath [26] and SPIN [32] algorithms. These heuristic algorithms can reduce compu-

- *C. Zhou, W. Zang, and L. Guo are with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, 100093, China. E-mail: {zhouchuan, zangwenyu, guoli}@iie.ac.cn*
- *P. Zhang is with the University of Technology, Sydney, NSW 2007, Australia. E-mail: Peng.Zhang@uts.edu.au*
- *A preliminary version [42] of this work has been published as a regular paper in the 13^{th} IEEE International Conference on Data Mining (ICDM-13).*
- *C. Zhou and P. Zhang are corresponding authors.*

tational cost in orders of magnitude with competitive results of the influence spread level. However, none of these algorithms has a theoretical guarantee on the reliability of the results and they often use Greedy as the benchmark for performance testing. Alternatively, the *bond percolation* method is used to estimate influence spread instead of the MC simulations [10], [27], [34]. However, bond percolation improves Greedy at the expense of high memory consumption and low estimation accuracy, as it needs to store numerous snapshots and estimate the spread from the non-independent samples.

Based on the above observation, improving Greedy by pruning its MC simulation calls plays a critical role in influence maximization. Along this direction, some sophisticated greedy algorithms were proposed with fewer MC simulations of the spread, such as the CELF [30], CELF++ [19], CGA [40], NewGreedy and MixedGreedy [7] algorithms. The principle behind these algorithms is that the marginal gain of a node in the current iteration cannot be more than that in previous iterations (i.e., the submodular property), and thus the number of MC simulation calls can be greatly reduced. Such algorithms are collectively called SubmodularGreedy.

SubmodularGreedy, albeit efficient, needs to establish an initial upper bound of marginal return for each node by using MC simulations, which incurs N times of MC simulations at the initial step. Such a limitation leads to a fundamental question that, *can we derive an initial upper bound of spreads which can be used to prune unnecessary spread MC simulations in* SubmodularGreedy?

In this paper, we explore new upper bounds to significantly reduce the number of MC simulations in SubmodularGreedy, especially at the initial step. We theoretically prove that the bound is tight and convergent when the summation of weights towards (from) each node is less than 1. Based on the bound, we propose a new *Upper Bound based Lazy Forward* algorithm (**UBLF** in short) to discover the top-k influential nodes in social networks. We test and compare UBLF with CELF. Experimental results demonstrate that UBLF reduces more than 95% MC simulations of CELF and achieves about $2-10$ times speedup.

The **contributions** of the paper are threefold:

1) We derive upper bounds for spread functions $\sigma_I(S)$ and $\sigma_L(S)$ under the IC model and LT model respectively.
2) Based on the upper bounds, a new UBLF algorithm is proposed which outperforms all the SubmodularGreedy algorithms in influential nodes discovery in social networks.
3) Theoretical studies, examples, and extensive experiments on synthetic and real data demonstrate the performance of the proposed bounds and the UBLF algorithm.

TABLE 1
Major variables used 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
$Chi(u)$	set of children of node u
$\mathbb{P}^S$	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. (8)
PP	N by N propagation probability matrix
W	N by N weight matrix
$\mathbf{1}$	column vector with all elements being 1

The derived bounds can be also used in other network mining problems where the submodular property stands. For example, one can combine the upper bounds and submodularity to enhance the efficiency of outbreak detection. Besides, the derived bounds can be employed to approximately estimate the real value of influence spread as in Eq. (17).

The rest of the paper is organized as follows. Section 2 briefly reviews the IC model and Greedy. Sections 3 and 4 introduce the upper bound for the spread function $\sigma_I(S)$ and the UBLF algorithm, under the IC model. Section 5 discusses in detail the upper bound for the spread function $\sigma_L(S)$ under the LT model and other analytic properties. Section 6 validates the performance of the proposed algorithm through experiments. Section 7 surveys the related work. Section 8 concludes the paper. Table 1 outlines major symbols and variables used in the paper.

2 Preliminaries

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.*,

$$Par(v) := \big\{u \in V,\ (u, v) \in E\big\}. \tag{1}$$

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

$$1 - \prod_{u \in S_t \cap Par(v)} \big(1 - pp(u, v)\big) \tag{2}$$

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, 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 by $\sigma_I(S)$ as follow,

$$\sigma_I(S) := \mathbb{E}^S\Big[\big|\bigcup_{t=0}^{N-|S|} S_t\big|\Big] \tag{3}$$

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 set of nodes activated in all $N-|S|+1$ steps.

*Eq. (3) converts the **global** influence function, $\sigma_I(S)$, as a summation of influence from **locally** activated node sets S_t ($0 \leq t \leq N-|S|$), which will be used in Proposition 1 in Section 3.1.*

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

$$S^* = \arg\max_{|S|=k, S\subseteq V} \sigma_I(S) \tag{4}$$

where k is a given parameter. The problem, as proved in the previous work [25], is NP-hard, and a constant-ratio approximation algorithm is feasible.

In the work [25], [31], it is shown that the objective function $\sigma_I(S)$ in Eq. (4) has the submodular and monotone properties with $\sigma_I(\varnothing) = 0$. Thus, the problem in Eq. (4) can be approximated by the greedy algorithm as shown in Algorithm 1, where $f := \sigma_I$. 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, to be included into the seed set S, until k seeds are selected. It is shown in [33] 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)

```
initial $S = \varnothing$
for $i = 1$ to $k$ do
    select $u = \arg\max_{w\in V\setminus S}\big(f(S\cup\{w\}) - f(S)\big)$
    $S = S \cup \{u\}$
end for
output $S$
```

In Greedy(k,σ_I), a critical issue is that there is no efficient way to compute $\sigma_I(S)$ given a set S. Kempe et al. [25] 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. [8] 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 Greedy(k,σ_I), one can either reduce the number of times calling Monte-Carlo simulations in computing $\sigma_I(S)$, or develop advanced heuristic algorithms which reduce the number of iterations without accuracy guarantees.

3 Analysis on Upper Bound of $\sigma_I(S)$

In this part, we aim to derive an upper bound for $\sigma_I(S)$, as the exact computation of $\sigma_I(S)$ is #P-hard [8]. Before introducing the upper bound in Theorem 1 and Corollary 1, we introduce two preparations first.

3.1 Preparations

Let $\mathbb{P}^S(v \in S_t)$ denote the probability that node v becomes activated at step t under the seed S, we have the first preparation as follows,

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

$$\sigma_I(S) = \sum_{t=0}^{N-|S|} \sum_{v\in V} \mathbb{P}^S(v \in S_t). \tag{5}$$

Proof: Note that the random sets $S_0, S_1, \ldots, S_{N-|S|}$ are pairwise disjoint, we have

$$\begin{aligned}
\sigma_I(S) &= \mathbb{E}^S\Big[\big|\bigcup_{t=0}^{N-|S|} S_t\big|\Big] = \mathbb{E}^S\Big[\sum_{t=0}^{N-|S|} |S_t|\Big] \\
&= \sum_{t=0}^{N-|S|} \mathbb{E}^S\Big[|S_t|\Big] = \sum_{t=0}^{N-|S|} \mathbb{E}^S\Big[\sum_{v\in V} I_{S_t}(v)\Big] \\
&= \sum_{t=0}^{N-|S|} \sum_{v\in V} \mathbb{E}^S\big[I_{S_t}(v)\big] \\
&= \sum_{t=0}^{N-|S|} \sum_{v\in V} \mathbb{P}^S(v \in S_t)
\end{aligned}$$

where $I_{S_t}(v)$ is a binary indicative function, if $v \in S_t$, $I_{S_t}(v) = 1$; otherwise, $I_{S_t}(v) = 0$. □

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

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

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

and

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

By following the classical Markov processes theory [11], if the process $(S_t)_{t=0}^{N-|S|}$ is Markovian, the series $\{\mathbb{P}^S(v\in S_t):v\in V\}_{t=0}^{N-|S|}$ should meet the well-known Chapman-Kolmogorov equation:

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

which is an identity relating to the joint probability distributions of different sets of coordinates.

Since whether a node v can be activated at step $t+1$ is conditioned by its activation results at prior steps $0,1,\ldots,t$ in the IC model, the process $(S_t)_{t=0}^{N-|S|}$ is not Markovian. But we still have the following Proposition 2 to reveal the recurrence relations among $\{\mathbb{P}^S(v\in S_t):v\in V\}_{t=0}^{N-|S|}$.

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

$$\mathbb{P}^S(v\in S_t)\le\sum_{u\in V}\mathbb{P}^S\big(u\in S_{t-1}\big)pp(u,v). \qquad (6)$$

Proof: For $t=1,2,\ldots,N-|S|$, by the definition of conditional expectation and IC model, it follows that

$$\begin{aligned}
&\mathbb{P}^S(v\in S_t)\\
=\;&\mathbb{E}^S\big[\mathbb{P}^S(v\in S_t|S_0,\cdots,S_{t-1})\big]\\
=\;&\mathbb{E}^S\Big[I_{\{v\notin\cup_{i=0}^{t-1}S_i\}}\cdot\big(1-\prod_{u\in S_{t-1}}(1-pp(u,v))\big)\Big]\\
\le\;&\mathbb{E}^S\Big[I_{\{v\notin\cup_{r=0}^{t-1}S_r\}}\cdot\big(\sum_{u\in S_{t-1}}pp(u,v)\big)\Big]\\
=\;&\mathbb{E}^S\Big[I_{\{v\notin\cup_{r=0}^{t-1}S_r\}}\cdot\big(\sum_{u\in V}I_{\{u\in S_{t-1}\}}pp(u,v)\big)\Big]\\
=\;&\mathbb{E}^S\Big[\sum_{u\in V}I_{\{v\notin\cup_{r=0}^{t-1}S_r,u\in S_{t-1}\}}\cdot pp(u,v)\Big]\\
=\;&\sum_{u\in V}\mathbb{P}^S\big(v\notin\cup_{r=0}^{t-1}S_r,u\in S_{t-1}\big)pp(u,v)\\
=\;&\sum_{u\in V}\mathbb{P}^S\big(v\notin\cup_{r=0}^{t-1}S_r\big|u\in S_{t-1}\big)\mathbb{P}^S\big(u\in S_{t-1}\big)pp(u,v)\\
\le\;&\sum_{u\in V}\mathbb{P}^S\big(u\in S_{t-1}\big)pp(u,v).
\end{aligned}$$

In the above derivation, $\{v\notin\cup_{i=0}^{t-1}S_i\}$ means that node v does not belong to any set of $S_0,\cdots,S_{t-1}$. The first '=' stems from the conditional expectation, the second '=' is from the assumption of IC model, the first '$\le$' comes from the fact that

$$1-\prod_{i=1}^{n}(1-x_i)\le\sum_{i=1}^{n}x_i$$

(for proof, see Lemma 3 in Appendix A), and the second '$\le$' is due to $\mathbb{P}^S\big(v\notin\cup_{r=0}^{t-1}S_r\big|u\in S_{t-1}\big)\le 1$. Hence Eq. (6) is obtained. □

Proposition 2 describes the ordering relationship between two adjacent elements in the series $\{\mathbb{P}^S(v\in S_t):v\in V\}_{t=0}^{N-|S|}$.

Now we simplify the results in Propositions 1 and 2 using the form of matrix. Let PP be the propagation probability matrix with the element at position (u,v) denoted by $pp(u,v)$. For $t=0,1,2,\ldots,N-|S|$, we use a row vector

$$\Pi_t^S=(\pi_t^S(v)) \qquad (7)$$

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

$$\pi_t^S(v):=\mathbb{P}^S(v\in S_t). \qquad (8)$$

Then, Proposition 1 can be rewritten as

$$\sigma_I(S)=\sum_{t=0}^{N-|S|}\Pi_t^S\cdot\mathbf{1} \qquad (9)$$

where $\mathbf{1}$ is a column vector with all elements being 1 and the notation '$\cdot$' is matrix multiplication. Likewise, Proposition 2 can be rewritten as

$$\Pi_t^S\le\Pi_{t-1}^S\cdot PP \qquad (10)$$

where PP is the propagation probability matrix as mentioned before.

3.2 The Upper Bound of $\sigma_I(S)$

With the above preparations, now we present the results as follows,

Theorem 1: The upper bound of spread $\sigma_I(S)$ is

$$\sigma_I(S)\le\sum_{t=0}^{N-|S|}\Pi_0^S\cdot PP^t\cdot\mathbf{1}. \qquad (11)$$

Proof: Employing the iteration in Eq. (10), we first have

$$\Pi_t^S\le\Pi_0^S\cdot PP^t \qquad (12)$$

where

$$\Pi_0^S=(\pi_0^S(v))=\begin{cases}1, & \text{if } v\in S\\ 0, & \text{if } v\notin S\end{cases}$$

according to the definition in Eq. (8). By incorporating Eq. (12) into Eq. (9), we have the following inequation,

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2419659, IEEE Transactions on Knowledge and Data Engineering

Note that here $\Pi_0^S \cdot PP^t \cdot \mathbf{1}$ is a real number after matrix calculation. □

Theorem 1 reveals that we can calculate the upper bound of spread $\sigma_I(S)$ using Eq. (11). Note that Eq. (11) implies that the expected value of diffusion range (i.e. spread) is less than the upper bound $\sigma_I(S)$, rather than that every possible diffusion range is less than the bound.

Based on Eq. (11) in Theorem 1, one may easily raise the following three 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 upper bound to $\sum_{t=0}^{\infty} \Pi_0^S \cdot PP^t \cdot \mathbf{1}$, then in what condition the latter series will be convergent?
- If the relaxed upper bound is convergent, what's the limit of convergence, *i.e.*, $\sum_{t=0}^{\infty} \Pi_0^S \cdot PP^t \cdot \mathbf{1} = ?$
- How large is the gap between $\sum_{t=0}^{N-|S|} \Pi_0^S \cdot PP^t \cdot \mathbf{1}$ and the limit $\sum_{t=0}^{\infty} \Pi_0^S \cdot PP^t \cdot \mathbf{1}$?

We expect that the relaxed upper bound can take a simple and tractable form, which will reduce the summation calculations in Eq. (11).

Also, we expect that the gap is small enough to utilize the relaxed one. In the sequel, we derive Corollary 1 and Proposition 3 to answer the above questions.

Corollary 1: If the propagation probability satisfies the condition

$$\max_v \sum_u pp(u,v) < 1 \quad \text{or} \quad \max_u \sum_v pp(u,v) < 1,[1] \qquad (13)$$

then the series $\sum_{t=0}^{\infty} \Pi_0^S \cdot PP^t \cdot \mathbf{1}$ is convergent with the limit $\Pi_0^S \cdot (E - PP)^{-1} \cdot \mathbf{1}$. In other words, under the condition (13), the upper bound of $\sigma_I(S)$ can be relaxed to be

$$\sigma_I(S) \le \Pi_0^S \cdot (E - PP)^{-1} \cdot \mathbf{1}, \qquad (14)$$

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

Proof: By matrix analysis [23], condition (13) implies

$$\| PP \|_1 < 1 \text{ or } \| PP \|_\infty < 1$$

where $\| PP \|_1 := \max_v \sum_u pp(u,v)$ and $\| PP \|_\infty := \max_u \sum_v pp(u,v)$, which ensures the convergence of matrix series $\sum_{t=0}^{\infty} PP^t$ with the limit $(E - PP)^{-1}$. Hence we have

$$\begin{aligned} \sigma_I(S) &\le \sum_{t=0}^{N-|S|} \Pi_0^S \cdot PP^t \cdot \mathbf{1} \le \sum_{t=0}^{\infty} \Pi_0^S \cdot PP^t \cdot \mathbf{1} \\ &= \Pi_0^S \cdot \Big(\sum_{t=0}^{\infty} PP^t \Big) \cdot \mathbf{1} = \Pi_0^S \cdot (E - PP)^{-1} \cdot \mathbf{1} \end{aligned}$$

where the first '$\le$' is from Theorem 1. □

Proposition 3: Given the condition in Eq. (13), the gap between the two upper bounds satisfies

$$\begin{aligned} &\Big(\Pi_0^S \cdot (E - PP)^{-1} \cdot \mathbf{1} \Big) - \Big(\sum_{t=0}^{N-|S|} \Pi_0^S \cdot PP^t \cdot \mathbf{1} \Big) \\ \le\ & \frac{\| \Pi_0^S \|_r \cdot \| PP \|_r^{N-|S|+1} \cdot \| \mathbf{1} \|_r}{1 - \| PP \|_r} \end{aligned}$$

where $r = 1$ or $r = \infty$ according to Eq. (13), $\| PP \|_r$ is defined in the proof of Corollary 1, $\| \Pi_0^S \|_1 := 1$, $\| \Pi_0^S \|_\infty := |S|$, $\| \mathbf{1} \|_1 := N$, and $\| \mathbf{1} \|_\infty := 1$.

Proof: See Appendix A. [2] □

Corollary 2: Given Eq. (13), we obtain

$$\Big(\Pi_0^S \cdot (E - PP)^{-1} \cdot \mathbf{1} \Big) - \Big(\sum_{t=0}^{N-|S|} \Pi_0^S \cdot PP^t \cdot \mathbf{1} \Big) \to 0$$

as $N - |S|$ is large enough.

Proof: See Appendix A. □

Corollary 1, Proposition 3 and Corollary 2 tell that, given the condition in Eq. (13), we can obtain a simple and tractable upper bound as in Eq. (14). Also, the gap between the two bounds is negligible when $N - |S|$ is large enough.

3.3 An example

In this part, we use a simple example to explain how to calculate the upper bound.

Example 1: Given a graph G, as shown in Fig. 1, with propagation probability matrix as in Eq. (15), which meets the condition (13) [3],

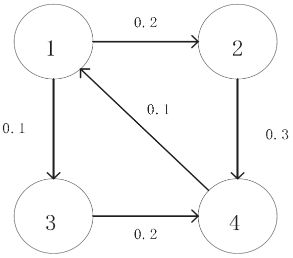


Fig. 1. A simple graph for the bound calculation.

$$PP = \begin{pmatrix} 0 & 0.2 & 0.1 & 0 \\ 0 & 0 & 0 & 0.3 \\ 0 & 0 & 0 & 0.2 \\ 0.1 & 0 & 0 & 0 \end{pmatrix}. \qquad (15)$$

we have

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

[1] The condition means that either the total influence to a node is less than 1 or the total influence diffused by a node is less than 1. In social networks, the propagation probability is often very small and the condition commonly stands. Also, one can refer to the data-driven work [17], [38] to confirm the small propagation probability.

[2] We move some proofs to Appendix A due to space.

[3] The sum of each row in PP is less than 1.

$$= \begin{pmatrix} 1.3911 \\ 1.3417 \\ 1.2278 \\ 1.1391 \end{pmatrix}.$$

Based on Corollary 1, the upper bound of spread $\sigma_I(S)$ with, say, the seed set $S = \{\textcircled{2}, \textcircled{4}\}$ can be calculated as follows,

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

To the same token, we can obtain the upper bound of $\sigma_I(S)$ with any seed set $S \subseteq \{\textcircled{1}, \textcircled{2}, \textcircled{3}, \textcircled{4}\}$. □

3.4 The calculation of the upper bound

The matrix calculation used in the upper bound is expensive, because the inverse $(E-PP)^{-1}$ is intractable when the network size is large. To overcome the difficulty, we adopt the following iterative procedure to calculate $(E - PP)^{-1} \cdot \mathbf{1}$.

For $n \geq 0$, let $\mathbf{b}_n := \sum_{t=0}^{n} PP^t \cdot \mathbf{1}$, then it follows that,

$$(E - PP)^{-1} \cdot \mathbf{1} = \sum_{t=0}^{\infty} PP^t \cdot \mathbf{1} = \lim_{n \to \infty} \mathbf{b}_n, \tag{16}$$

Based on the iteration structure $\mathbf{b}_n = PP \cdot \mathbf{b}_{n-1} + \mathbf{1}$, to obtain $(E - PP)^{-1} \cdot \mathbf{1}$, we only need to calculate $\mathbf{b}_n$ until some gap of $\mathbf{b}_n - \mathbf{b}_{n-1}$ with L_1-norm less than 10^{-3}. This transformation saves memory cost by storing vectors instead of matrixes.

In addition, an alternative way to calculate $(E - PP)^{-1} \cdot \mathbf{1}$ is to solve linear equations $(E - PP) \cdot \mathbf{x} = \mathbf{1}$, which avoids the matrix inversion.

4 UBLF Algorithm

Based on the upper bound, we propose a new UBLF algorithm for influence maximization.

Now we explain the difference between UBLF and CELF. The latter is one of the most popular greedy algorithms in the influence maximization field.

The CELF algorithm, proposed by Leskovec *et al.* [30], exploits 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 contrast, the proposed Upper Bound based Lazy Forward (UBLF) algorithm uses the derived new bound to further reduce the number of spread estimations, especial in the initialization step. This way, the nodes are ranked by their upper bound scores which can be used to prune unnecessary calculations in the CELF algorithm. We use Example 2 for illustration.

Example 2: We still use the network in Fig. 1 for example. The goal is to find the top-1 most influential node in the network. For a specific node ①, according to Corollary 1, its upper bound is calculated as follow,

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

To the same token, we obtain

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

Obviously, the upper bound of $\sigma_I(\textcircled{1})$ is 1.3911, which is the largest in the graph. Thus, we use Monte-Carlo simulation (or do precise calculation due to the simple structure) to estimate $\sigma_I(\textcircled{1})$, and obtain

$$\sigma_I(\textcircled{1}) = 1.3788$$

We can observe that 1.3788 is already larger than the upper bounds of the remaining nodes, e.g., $\sigma_I(\textcircled{2})$, $\sigma_I(\textcircled{3})$ and $\sigma_I(\textcircled{4})$. Thus, it is unnecessary to estimate the remaining nodes, and the node ① is the most influential one in the network.

By introducing the upper bounds, UBLF can significantly reduce the number of Monte-Carlo simulation calls. In this example, UBLF needs only one Monte-Carlo simulation call, while CELF [30] needs four Monte-Carlo simulation calls. □

We summarize the UBLF algorithm in Algorithm 2.

We explain the key step 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.* $S = \emptyset$), UBLF estimates an upper bound for each node by following Theorem 1 (or Corollary 1 when condition in Eq. (13) is met).

In the algorithm, $MC(S \cup \{u\})$ denotes that the Monte-Carlo simulation is used to estimate $\sigma_I(S \cup \{u\})$ for the initial seed set $S \cup \{u\}$, $I(v) = 0$ denotes that the Monte-Carlo simulation has not been used to estimate $\sigma_I(S \cup \{v\})$ yet in the current iteration, and $I(v) = 1$ means the Monte-Carlo simulation has already been computed to estimate $\sigma_I(S \cup \{v\})$.

5 Discussions on the Upper Bound

Regarding the upper bound for the spread function $\sigma_I(S)$, one may have the following questions:

1) Under what condition, the estimated upper bound can approximate the real value of $\sigma_I(S)$?

Algorithm 2: UBLF

```
Input: the propagation probability matrix $PP$
        of a graph $G = (V, E)$, a budget $k$
Output: The most influential set $S$ with $k$
        nodes
initial $S \leftarrow \varnothing$, $\sigma_I(S) \leftarrow 0$, and
    $\delta \leftarrow \sum_{t=0}^{N-|S|} \cdot PP^t \cdot \mathbf{1}$
    (or $\delta \leftarrow (E - PP)^{-1} \cdot \mathbf{1}$ if condition (13) is met)
for $i = 1$ to $k$ do
    set $I(v) \leftarrow 0$ for $v \in V \backslash S$
    while TRUE do
    { $u \leftarrow \arg\max_{v \in V \backslash S} \delta_v$
        if $I(u) = 0$
            $\delta_u \leftarrow MC(S \cup \{u\}) - \sigma_I(S)$
            $I(u) \leftarrow 1$
        end if
        if $\delta_u \geq \max_{v \in V \backslash (S \cup \{u\})} \delta_v$
            $\sigma_I(S \cup \{u\}) \leftarrow \sigma_I(S) + \delta_u$
            $S \leftarrow S \cup \{u\}$
            break
        end if }
end for
output $S$
```

2) Why does the upper bound take the form as $\sum_{t=0}^{N-|S|} \Pi_0^S \cdot PP^t \cdot \mathbf{1}$? What does this value mean?
3) How to obtain the upper bound for the spread function $\sigma_L(S)$ under the LT model?

This section is organized in three parts to answer the three questions respectively.

5.1 In what conditions the upper bound asymptotically approximates the real value of $\sigma_I(S)$

This part explains that under two conditions,

- Condition (I): the propagation probabilities $\{pp(u, v)\}$ are relatively small, and
- Condition (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 convert Eq. (11) to Eq. (17) as follows,

$$\sigma_I(S) \lessapprox \sum_{t=0}^{N-|S|} \Pi_0^S \cdot PP^t \cdot \mathbf{1} \tag{17}$$

where $A \lessapprox B$ means that A approximates B though $A \leq B$.

In the sequel, we 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 1: For small positive numbers $x_1, x_2, \ldots, x_n$, we have

$$1 - \prod_{i=1}^{n}(1 - x_i) \approx \sum_{i=1}^{n} x_i. \tag{18}$$

Proof: Note that

$$\prod_{i=1}^{n}(1 - x_i) = 1 - \sum_{i=1}^{n} x_i + o(\sum_{1 \leq i < j \leq n} x_i x_j)$$

when $x_1, \ldots, x_n$ are relatively small, thus we obtain Eq. (18). □

We use Example 3 to explain Lemma 1.

Example 3: 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 $t + 1$ under the IC model is

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

and we can observe that the probability almost equals to the value

$$0.1 + 0.2 = 0.3.$$

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

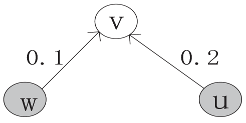


Fig. 2. An example of activation under IC model

Based on Lemma 1 above and Lemma 3 in Appendix A, we have

$$\begin{aligned} & \mathbb{E}^S\Big[I_{\{v \notin \cup_{i=0}^{t-1} S_i\}} \cdot \big(1 - \prod_{u \in S_{t-1}} (1 - pp(u, v))\big)\Big] \\ \lessapprox \;& \mathbb{E}^S\Big[I_{\{v \notin \cup_{r=0}^{t-1} S_r\}} \cdot \big(\sum_{u \in S_{t-1}} pp(u, v)\big)\Big] \end{aligned}$$

in the proof of Proposition 2, when the propagation probabilities $\{pp(u, v)\}$ are relatively small.

Lemma 2: If Conditions (I) and (II) are satisfied, we have

$$\mathbb{P}^S\big(v \notin \cup_{r=0}^{t-1} S_r \big| u \in S_{t-1}\big) \lessapprox 1 \tag{19}$$

for majority $v \in V$, regardless of any $u \in V$.

Proof: If Conditions (I) and (II) stand, there will be only few nodes in the set $\cup_{r=0}^{t-1} S_r$, and the majority nodes can not be activated in finite steps, no matter weather $u \in S_{t-1}$ is given as a priori knowledge for any $u \in V$. Hence Eq. (19) is obtained for majority $v \in V$. □

We incorporate the above two lemmas into the proof of Proposition 2, and obtain an approximate version of Eq. (6) as follows,

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

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

$$\Pi_t^S \lessapprox \Pi_{t-1}^S \cdot PP \tag{20}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2419659, IEEE Transactions on Knowledge and Data Engineering

Incorporating the above approximation into the proof of Theorem 1, we obtain the final result in Eq. (17). Furthermore, under the condition in Eq. (13), we have the approximate version of Eq. (14) as follows,

$$\sigma_I(S) \lessapprox \Pi_0^S \cdot (E - PP)^{-1} \cdot \mathbf{1} \tag{21}$$

with the gap guarantee given in Proposition 3.

To sum up, when the two given conditions are met, the upper bound approximates the spread function $\sigma_I(S)$. Hence, we have high accuracy guarantee to use the bound, $\sum_{t=0}^{N-|S|} \Pi_0^S \cdot PP^t \cdot \mathbf{1}$ or $(E - PP)^{-1} \cdot \mathbf{1}$, as the selecting criterion.

Specifically, we can choose k nodes with the highest values in the column vector $\sum_{t=0}^{N-|S|} \Pi_0^S \cdot PP^t \cdot \mathbf{1}$ or $(E - PP)^{-1} \cdot \mathbf{1}$ as the initial seed set. For instance, in Example 1,

$$(E - PP)^{-1} \cdot \mathbf{1} = \begin{pmatrix} 1.3911 \\ 1.3417 \\ 1.2278 \\ 1.1391 \end{pmatrix}$$

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. The UBound algorithm is summarized in Algorithm 3.

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: initial $\delta \leftarrow \sum_{t=0}^{N-|S|} PP^t \cdot \mathbf{1}$
      (or $\delta \leftarrow (E - PP)^{-1} \cdot \mathbf{1}$ if condition (13) is met)
4: Select the biggest $k$ nodes in $\delta$ as the output $S$
```

5.2 An alternative view of the upper bounds in Eq. (11) and Eq. (14)

The upper bound in Eq. (11) and Eq. (14) equals to the integral influence under the voter model. The voter model, proposed in [12], is probably one of the most basic and natural probability models to represent opinion diffusions in which people may switch opinions back and forth from time to time due to the interactions with other people in the network.

Here we introduce a variant of the common *voter model* from the single-item-based point of view.

Consider a directed graph $G = (V, E)$ with edge labels propagation probability $pp : E \to [0, 1]$. Let $pp(u, v) = 0$ if $(u, v) \notin E$. For $v \in V$, the set of parents $Par(v)$ of v is also defined as in Eq. (1). For propagation probability pp, we assume that

$$\sum_{u \in Par(v)} pp(u, v) \leq 1$$

for each $v \in V$. Given a seed set $S \subseteq V$, the voter model works as follows. Let $S_t \subseteq V$ be the set of nodes that are activated at step $t \geq 0$ with $S_0 = S$. At step $t + 1$, each node $v \in V$ can be activated by its newly activated neighbors with probability $\sum_{u \in Par(v) \cap S_t} pp(u, v)$. If v is activated successfully, then it is put into the set S_{t+1}. The process ends at a step τ with $S_\tau = \varnothing$. For mathematical tractability, we still denote $S_t = \varnothing$ for $t > \tau$. Obviously the process $(S_t)_{t \geq 0}$ is Markovian due to its memoryless.

The *integral influence* over time span T triggered by S, i.e., the expected value of the total activation numbers from the initial propagation time to the time point T, can be denoted as $\sigma_V^T(S)$, *i.e.*,

$$\sigma_V^T(S) := \mathbb{E}^S \Big[\sum_{t=0}^{T} |S_t| \Big].$$

Theorem 2: The integral influence $\sigma_V^T(S)$ over time span T can be calculated as follows

$$\sigma_V^T(S) = \sum_{t=0}^{T} \Pi_0^S \cdot PP^t \cdot \mathbf{1} \tag{22}$$

where $PP = \{pp(u, v)\}$ is the propagation probability matrix in the voter model.

Proof: By the Markov property of the voter model, it follows that $\Pi_t^S = \Pi_0^S \cdot PP^t$, where Π_t^S is a row vector with element $\pi_t^S(v) := \mathbb{P}^S(v \in S_t)$. Hence, we obtain that, $\sigma_V^T(S) = \sum_{t=0}^{T} \Pi_t^S \cdot \mathbf{1} = \sum_{t=0}^{T} \Pi_0^S \cdot PP^t \cdot \mathbf{1}$, where $\mathbf{1}$ is a column vector with all elements being 1. Note that $\Pi_0^S \cdot PP^t \cdot \mathbf{1}$ is a real number after matrix calculations. □

Regarding the convergence of the integral influence $\sigma_V^\infty(S)$ in the long term, we have

Corollary 3: If the propagation probability matrix PP further satisfies the condition

$$\max_v \sum_{u \in Par(v)} pp(u, v) < 1,$$

then the series $\sum_{t=0}^{\infty} \Pi_0^S \cdot PP^t \cdot \mathbf{1}$ is convergent, and the limit of convergence exists as

$$\sigma_V^\infty(S) = \Pi_0^S \cdot (E - PP)^{-1} \cdot \mathbf{1} \tag{23}$$

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

Proof: The proof is similar to that of Corollary 1. □

To sum up, the upper bound of spread $\sigma_I(S)$ in Theorem 1 and Corollary 1 can be rewritten as

$$\sigma_I(S) \leq \sigma_V^{N-|S|}(S) \leq \sigma_V^\infty(S),$$

if the condition $\max_v \sum_{u \in Par(v)} pp(u, v) < 1$ is satisfied under the IC model and the voter model.

5.3 How to derive the upper bound of $\sigma_L(S)$ under the LT model

We can also derive the upper bound for the spread $\sigma_L(S)$ under the LT model, though the exact computation of $\sigma_L(S)$ is #P-hard [9].

Following the definition in [25], we define the LT model as follows. Consider a directed graph $G = (V, E)$ with N nodes in V and edges labeled a weight function $w : E \to [0, 1]$, such that for each $v \in V$,

$$\sum_{u \in V} w(u, v) \leq 1$$

where $w(u, v)$ is defined to be 0 if $(u, v) \notin E$.

Given a seed set $S \subseteq V$, the LT model works as follows. First, every vertex $v \in V \backslash S$ independently selects a threshold θ_v uniformly at random in range $[0, 1]$, which reflects our lack of knowledge of users' true thresholds as pointed out in [25]. Next, let $S_t \subseteq V$ be the set of nodes that are activated at step $t \geq 0$ with $S_0 = S$. At step $t + 1$, a vertex $v \in V \backslash \cup_{i=0}^{t} S_i$ is activated (and thus is in S_{t+1}) if the total weight from its activated neighbors reaches its threshold, i.e.,

$$\sum_{u \in \cup_{i=0}^{t} S_i} w(u, v) \geq \theta_v$$

The process stops at a step τ when $S_\tau = \varnothing$. Obviously, the propagation process has $N - |S|$ steps at most, as there are at most $N - |S|$ nodes outside the seed set S. For convenience, let $S_{\tau+1} = \varnothing, \cdots, S_{N-|S|} = \varnothing$, if $\tau < N - |S|$. Let $\sigma_L(S)$ denote the expected number of activated nodes given the seed set S, i.e.,

$$\sigma_L(S) := \mathbb{E}^S \Big[\big| \bigcup_{t=0}^{N-|S|} S_t \big| \Big], \quad (24)$$

where the expectation operator $\mathbb{E}^S$ with seed set S is taken among all θ_v values from their uniform distributions, and the subscript $'L'$ denotes the LT model. We call $\sigma_L(S)$ the influence spread of seed set S in the graph G under the LT model.

As shown in [25], the LT model defined above is equivalent to the reachability in the following random graphs, called live-edge graphs: Given a graph $G = (V, E)$ with weight function w, for every $v \in V$, select at most one of its incoming edges at random, such that edge (u, v) is selected with probability $w(u, v)$, and no edge is selected with probability $1 - \sum_u w(u, v)$. The selected edges are called live and all other edges are called blocked. Let R_G denote the random graph generated from G, which includes all vertices in V and all live edges selected. Thus, we have

Proposition 4: (Claim 2.6 of [25]) Given a graph G and a seed set S, the distribution of the set of active nodes in G with seed set S under the LT model is the same as the distribution of the set of nodes reachable from S in the random graph R_G.

Proof: For proof, see Claim 2.6 of [25]. □

Let $\mathcal{P}$ denote the set of all simple paths with the starting node in S. By the equivalence given in Proposition 4, we have

$$\sigma_L(S) = \sum_{\pi \in \mathcal{P}} \prod_{e \in \pi} w(e). \quad (25)$$

Based on this, Theorem 3 is followed.

Theorem 3: The upper bound of spread $\sigma_L(S)$ is

$$\sigma_L(S) \leq \sum_{t=0}^{N-|S|} \Pi_0^S \cdot W^t \cdot \mathbf{1} \quad (26)$$

where $W = (w_{ij})$ is the weight matrix.

Proof: For $t = 0, 1, \ldots, N - |S|$, let B_t be the set of simple paths of length t in $\mathcal{P}$, and C_t be the set of all paths of length t with the starting node in S (in a path belonging to C_t, the nodes are allowed to reappear). In fact, we have

$$\begin{aligned} \sigma_L(S) &= \sum_{t=0}^{N-|S|} \sum_{\pi \in B_t} \prod_{e \in \pi} w(e) \\ &\leq \sum_{t=0}^{N-|S|} \sum_{\pi \in C_t} \prod_{e \in \pi} w(e) \\ &= \sum_{t=0}^{N-|S|} \Pi_0^S \cdot W^t \cdot \mathbf{1} \end{aligned}$$

where the first $'='$ is from Eq. (25) and the definition of B_t, the first $'\leq'$ is due to $B_t \subseteq C_t$, and the second $'='$ stems from the classical graph theory (for relevant reference, see e.g. [41]). □

Parallel to Corollary 1, we have the following Corollary 4, which presents a tractable upper bound under certain condition.

Corollary 4: If the weight matrix W of the graph G satisfies the condition

$$\max_v \sum_u w(u, v) < 1, \quad (27)$$

then the upper bound of $\sigma_L(S)$ can be relaxed as

$$\sigma_L(S) \leq \Pi_0^S \cdot (E - W)^{-1} \cdot \mathbf{1} \quad (28)$$

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

Proof: We omit it again, as it is completely similar with that of Corollary 1. □

Actually, based on the upper bounds (Theorem 3 and Corollary 4), we can derive new greedy algorithms, similar to UBLF, to enhance the efficiency of the greedy algorithms in the LT model.

6 Experiments

We conduct experiments on three real-world and one synthetic data sets to evaluate the upper bound and the **UBLF** algorithm. We implement the algorithms using C++ with the Standard Template Library (STL). All experiments are run on a Linux (Ubuntu 11.10) machine with six-core 1400 MHz AMD CPU and 32 GB memory.

6.1 Data Sets

We use three real and one synthetic **data sets** for testing and comparisons.

The **Digger** data set is available at *http://arnetminer.org/heterinf*. Digger is a heterogeneous network, including Digg stories, user actions (submit, digg, comment and reply) with respect to the stories, and friendship relations among users. The **Twitter** data set can be obtained from *http://snap.stanford.edu/data/*. The **Epinions** data can be obtained from *http://snap.stanford.edu/data/*. Epinions is a general consumer review site where visitors can read reviews about a variety of items to help them decide a purchase. The synthetic **Small-world** data set is the type of graphs in which each node can be reached by a small number of hops. For small-world model we set the parameter of the nearest neighbors $k = 15$ and the rewiring probability $p = 0.1$.

The above networks are representative ones, covering a variety of networks with different types of relations and sizes. The details of the data sets are listed in Table 2 where degree means out-degree. In our experiments, an undirected graph is regarded as a bidirectional graph.

TABLE 2
Statistics of the four real-world networks.

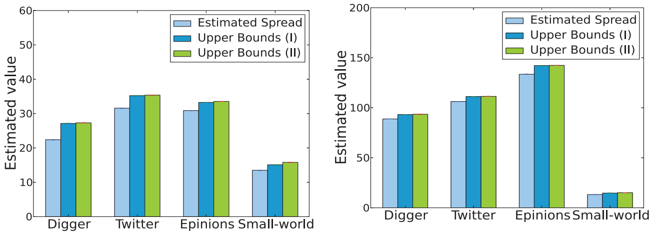
Dataset	Digger	Twitter	Epinions	Small-world
#Node	8,194	32,986	51,783	200,000
#Edge	56,440	763,713	476,491	2,999,999
Average Degree	6.9	23.2	9.2	15.0
Maximal Degree	850	674	190	29

6.2 Benchmark methods

We compare the UBLF in Algorithm 2 and the UBound heuristic in Algorithm 3 with both the greedy algorithms and heuristics.

- CELF [30]. The state-of-the-art greedy algorithm.
- StaticGreedy [10]. A bond percolation based algorithm. We set $R = 10,000$, i.e., 10,000 snapshots in the whole process for any network.
- Degree [25]. A heuristic algorithm based on ”degree centrality”, with high-degree nodes as influential ones. The seeds are the nodes with the k highest out-degrees.
- (Generalized)DegreeDiscount [7]. A degree discount heuristic algorithm, which can be used in directed graph with weighted propagation probabilities. It generalizes the original degree discount in [7]. The details are given in Appendix B.
- MIA [8]. A heuristic algorithm based on the maximum influence arborescence model.
- SP1M [26]. A heuristic algorithm based on the Shortest-Path model.

Greedy is not compared because many works have reported that CELF has the same influence spread result and less running time than Greedy. We test the NewGreedy and MixedGreedy algorithms [7]. The results are similar to the CELF algorithm. Thus, we also omit these two greedy algorithms. The CELF++ [19] algorithm is not compared because its performance is similar to CELF based on the reports in their original work. Since the MIA heuristic is the state-of-the-art [8], we do not implement heuristics such as PageRank [5], distance centrality and betweenness centrality-based heuristics.



(a) Under the set of pp_1

(b) Under the set of pp_2

Fig. 3. Evaluations of the upper bounds.

6.3 Parameter settings

To obtain the influence spread of the heuristic algorithms for each seed set, we run MC simulations on the networks $10,000$ times and take the average results as the influence spreads.

We assign the propagation probability of each directed link $(u, v) \in E$ in the network for the IC Model in two ways.

1). Weighted distribution

$$pp_1(u, v) = \min\{\frac{1}{|Par(v)| + 1}, 0.01\}, \text{ for } u \in Par(v)$$

where $Par(v)$ is defined as Eq. (1) and $|Par(v)|$ denotes the number of parents of node v.

2). Weighted distribution

$$pp_2(u, v) = \min\{\frac{1}{|Chi(u)| + 1}, 0.01\}, \text{ for } v \in Chi(u)$$

where $Chi(u) := \{w \in V | (u, w) \in E\}$ is defined as the children of node u, and $|Chi(u)|$ denotes the number of children of node u. The more children u have, the less likely that u meets a certain individual at one time unit.

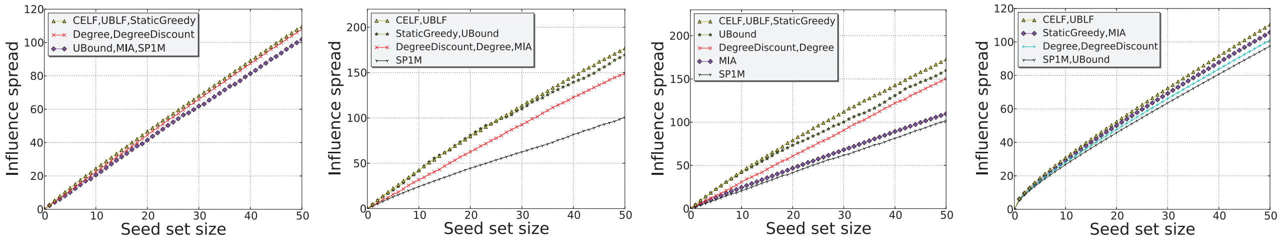
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2419659, IEEE Transactions on Knowledge and Data Engineering

TABLE 3
The number of Monte-Carlo simulation calls in the first ten iterations and their sums under pp_1

Datasets	Algorithms	1	2	3	4	5	6	7	8	9	10	Sum(1:10)	Sum(1:50)
Digger	CELF	**8,194**	4	12	12	95	28	9	58	7	8	**8,427**	**8,669**
	UBLF	**67**	5	18	7	41	18	12	28	7	10	**213**	**429**
Twitter	CELF	**32,986**	43	31	58	8	72	56	35	108	82	**33,479**	**34,733**
	UBLF	**98**	33	27	23	12	52	36	25	34	25	**365**	**1,672**
Epinions	CELF	**51,783**	26	37	117	139	46	59	15	12	175	**52,409**	**54,523**
	UBLF	**229**	3	7	27	167	16	62	4	50	113	**678**	**2,634**
Small-world	CELF	**200,000**	125	67	34	63	213	78	89	312	60	**201,041**	**204,571**
	UBLF	**356**	87	39	120	278	162	321	61	92	17	**1,533**	**5,683**

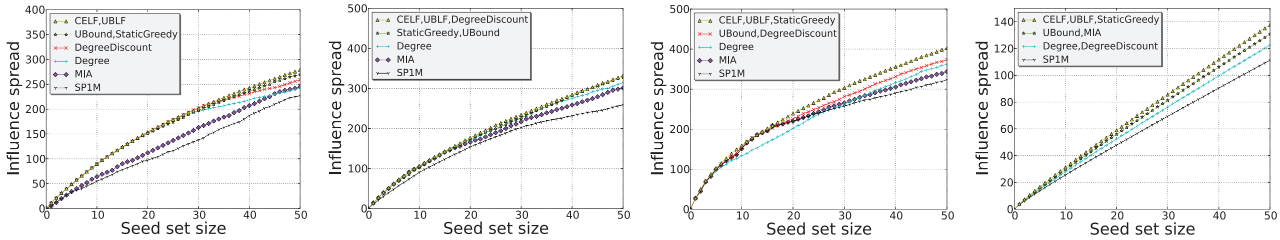
TABLE 4
The number of Monte-Carlo simulation calls in the first ten iterations and their sums under pp_2

Datasets	Algorithms	1	2	3	4	5	6	7	8	9	10	Sum(1:10)	Sum(1:50)
Digger	CELF	**8,194**	2	5	4	6	6	2	2	14	3	**8,238**	**8,512**
	UBLF	**45**	3	3	5	4	3	8	2	5	3	**81**	**377**
Twitter	CELF	**32,986**	2	2	3	2	3	2	9	9	2	**33,020**	**33,419**
	UBLF	**260**	8	2	3	6	8	3	2	6	2	**300**	**763**
Epinions	CELF	**51,783**	2	2	2	2	4	2	2	3	3	**51,805**	**52,096**
	UBLF	**42**	2	3	2	4	3	3	2	3	4	**68**	**418**
Small-world	CELF	**200,000**	12	27	32	51	34	52	9	27	14	**200,258**	**200,560**
	UBLF	**168**	18	65	12	21	34	78	112	73	45	**626**	**1,221**



(a) Digger (b) Twitter (c) Epinions (d) Small-world

Fig. 4. Influence spread results *w.r.t.* seed size k on the four data sets under pp_1.



(a) Digger (b) Twitter (c) Epinions (d) Small-world

Fig. 5. Influence spread results *w.r.t.* seed size k on the four data sets under pp_2.

6.4 Experimental Results

Evaluations of the upper bounds. Fig. 3 shows the gap between the real value of spread $\sigma_I(S)$ and its upper bounds in Eq. (11) and Eq. (14). Here we select ten nodes with the highest out-degrees in each network as the initial seed set. The gray bars (Estimated Spread) shows the real value of spread $\sigma_I(S)$ by MC estimations. The blue bar (Upper Bound (I)) shows the upper bound value presented in Eq. (11), and the green bar (Upper Bound (II)) shows the upper bound value presented in Eq. (14). The experimental results reveal that the exact value is close to the bound in Eq. (11), and bounds in Eq. (11) and Eq. (14) are almost identical on all the four data sets. These experimental results verify the analysis in Section 3.2 and Section 5.1.

Number of Monte-Carlo calls. We compare the number of MC calls between CELF and UBLF. In Table 3 and Table 4, we list the number of MC calls at the first 10 iterations on the four data sets. From the results, we can observe that the number of MC calls in UBLF is significantly reduced, compared to that in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TKDE.2015.2419659, IEEE Transactions on Knowledge and Data Engineering

CELF, especially at the first round.

One may notice that in Table 3 and Table 4, CELF occasionally defeats our method, but the total call number of UBLF is much less than CELF. As listed in the right of the second column, the total call number of the first 10 iterations of UBLF, compared to CELF, is reduced at a rate of 97.5%, 98.9%, 98.7%, 99.2% on the four data sets under pp_1. The corresponding rates are 99.0%, 99.1%, 99.9%, 99.7%,under pp_2. Similarly, at least 95% reduction of Monte-Carlo calls of CELF can be observed in the first 50 iterations. From these results, we can further come to the conclusion that *the larger the network is, the more efficient our UBLF is*.

Influence spread. Influence spread is an important measure for performance comparison. We run tests on the four data sets to obtain influence spread results *w.r.t.* parameter k (the seed set size), where k increases from 1 to 50. We list the results in Fig. 4 and Fig. 5. In the figures, the legend ranks the algorithms top-down based on their average influence spread values. If two curves are close, we group them together.

From the results, we can observe that UBLF has competitive spread results on the four data sets and UBound also performs well on average. An important observation is that the spreads of UBLF and CELF are completely the same in the four figures, which explains again that UBLF and CELF share the same results in node selection. The only difference between UBLF and CELF is the number of MC calls.

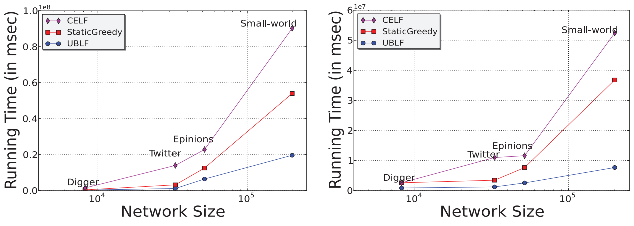
Now, we explain the difference among the results in Fig. 4 and Fig. 5. **First**, the ranges of the y-axis are very different. For example, the value of y is between 0 and 120 in Fig. 4 (a), but between 0 and 200 in Fig. 4 (b). **Second**, the algorithms often show consistent results but not completely insensitive to different data sets. For example, in Fig. 4, UBound performs almost the same as UBLF on the Twitter and Epinions data sets, but it is inferior to UBLF on the Digger and Small-world data sets. Another example, Degree performs better than MIA on the Digger, Twitter and Epinions data sets, but the results get reversed on the Small-world data set in Fig. 5. **Third**, the bigger scale of the ”small world” actually impacts the results. As shown in Fig. 5 (d), the spread increases almost linearly with the seed size; in contrast, the curves on the other three data sets are logarithmic in Fig. 5 (a)-(c).

Time cost. Fig. 6 and Fig. 7 show time costs of selecting seeds. From the results, we can observe that the heuristic algorithms, Degree and DegreeDiscount, are very fast in selecting candidate nodes. The UBound algorithm is slightly slower than the above two algorithms. Considering UBLF reduces a clear majority of MC calls, as shown in Table 3 and Table 4, the runtime is acceptable. From Fig. 6, we can observe that UBLF is **2-10** times faster than CELF.

One may question that such a low improvement of UBLF can be neglected in large networks. In fact, UBLF scales well to large networks (see Fig. 6). Note that with the networks increase, MC simulations take heavier time, and UBLF will achieve better performance gain by pruning unnecessary MC simulations.

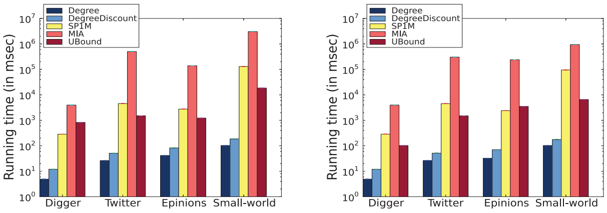
Sensitivity analysis. We run tests on Twitter data set and obtain the running time *w.r.t.* parameter pp (the propagation probability). In the experiments, pp increases from 0.01 to 0.1. We assign a uniform propagation probability pp to each directed link under the IC Model. From the results in Fig. 8, we can conclude that the smaller the propagation probabilities are, the more improvement UBLF can obtain. When the propagation probability $pp > 0.08$, UBLF loses the superiority to StaticGreedy.

In fact, the propagation probabilities are often set to be very small in the literature (*e.g.*, $pp = 0.01$ in references [7] and [8]). This is because the adoption events in social networks are often rare, which has been verified by the previous work [17] and [38]. Therefore, UBLF is a practically useful algorithm.



(a) Under the set of pp_1 (b) Under the set of pp_2

Fig. 6. Runtime of different greedy algorithms $w.r.t.$ network size. The seed size $k = 50$.



(a) Under the set of pp_1 (b) Under the set of pp_2

Fig. 7. Runtime of different heuristic algorithms with seed size $k = 50$.

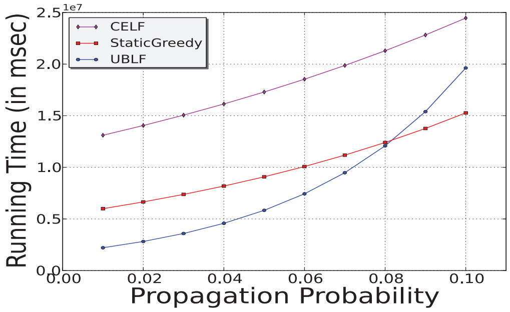


Fig. 8. Runtime of different greedy algorithms $w.r.t.$ network propagation probability with seed size $k = 50$.

7 Related Work

Domingos and Richardson [13], [35] first formulated the influence maximization problem as an algorithmic problem in probabilistic methods. Later, Kempe et al. [25] first modeled the problem as the discrete optimization problem, as described in Section 2.

A common limitation of existing approaches is computational inefficiency on large networks. Thus, two major types of solutions have been proposed.

First, a line of heuristic algorithms have been proposed. Chen et al. [7] proposed *degree discount heuristics* under the uniform IC model, in which all edge probabilities are the same. Experimental results indicate that the new heuristics are efficient with satisfactory influence spread. Following the idea, Chen et al. [8] proposed a Maximum Influence Arborescence (MIA) model to evaluate the influence spread of a user. In [26], Kimura and Saito proposed the *shortest-path* based influence cascade models and efficient algorithms to compute influence spread under modified IC models.

For the LT model, Chen et al. [9] observed that the spread can be computed in linear time on directed acyclic graphs (DAGs). And they experimentally showed that this heuristic is significantly faster than the greedy algorithm and achieves high quality seed set selection, measured in terms of the spread achieved. Similarly, Goyal et al. in [20] proposed SIMPATH, an efficient and effective algorithm for influence maximization under the LT model that addresses these drawbacks by incorporating several clever optimizations. In [32], Narayanam and Narahari proposed a Shapley value based heuristic SPIN for the LT model. However, SPIN only relies on the evaluation of influence spreads of seed sets, and thus does not use specific features of the LT model. Moreover, SPIN is not scalable, with running time comparable (as shown in [32]) or slower than the optimized greedy algorithm.

Second, several optimized greedy algorithms have been proposed. Lescovec et al. [30] proposed the CELF algorithm, which achieves 700 times speed improvement with respect to the simple greedy algorithm. Later, Goyal et al. [19], proposed CELF++, an extension to CELF that further reduces the number of spread estimation calls, which achieves approximately $35\% - 55\%$ speedup compared with CELF. Besides, Chen et al. [7] proposed the NewGreedy and MixedGreedy algorithms in the IC model with uniform probabilities. However, their performance are non-steady, sometimes even worse than CELF. Wang et al. [40] discussed the influence maximization from the view of social community and proposed a new community-based greedy algorithm for mining top-k influential nodes.

Besides these two types of solutions, Jiang et al. [24] proposed a totally different approach based on Simulated Annealing (SA) for the influence maximization problem. Guo et al. [21] investigated the personalized influence maximization problem. Rodriguez et al. [37] studied the influence maximization problem in continuous time diffusion networks. The competitive influence maximization problem in social networks was studied in the works [2], [4]. The problem of minimizing the spread of misinformation in social networks was discussed in the works of [6], [22].

Recently, **data-driven methods** were proposed to solve influence maximization problem by maximizing likelihood functions *w.r.t.* real propagation cascades. Data-driven methods are based on the assumption that the pairwise transmission probabilities are unknown in a network. In contrast, simulation-based methods are based on the assumption that the pairwise transmission probabilities are known a prior. In data-driven methods, Barbieri et al. [1] studied social influence from the topic modeling perspective. Bonchi [3] discussed what to learn and how to learn from trace data of past propagations to identify of influential users. Goyal et al. [18] proposed to directly use the past available data instead of assuming influence probabilities are given as input. The complementary problem of learning influence probabilities from the available data was studied in the works [17], [36], [38], and [39].

8 Conclusion

In this paper, we derived an upper bound for the spread functions in solving the influence maximization problem in social networks. Based on the bound, we proposed a new algorithm *Upper Bound based Lazy Forward* (**UBLF** in short) to discover the top-k influential nodes in social networks. Compared with the popular CELF algorithm, UBLF significantly reduces the number of Monte-Carlo simulations, *e.g.*, more than **95% reduction of Monte-Carlo calls** of CELF was observed in our experiments when the seed size was less than 50. The experimental results also verified that UBLF can enhance the efficiency of CELF by about **2-10 times** when the seed set is small.

The work focuses on the *upper bounds* of influence spread, how to derive *lower bounds* of influence spread is an interesting problem in the future.

Acknowledgment

This work was supported by the NSFC (No. 61370025), the Strategic Leading Science and Technology Projects of CAS (No.XDA06030200), 973 project (No. 2013CB329605) and Australia ARC Discovery Project (DP140102206).

References

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

[2] S. Bharathi, D. Kempe, and M. Salek, "Competitive influence maximization in social networks," in WINE 2007.

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

[4] A. Borodin, Y. Filmus, and J. Oren, "Threshold models for competitive influence in social networks," in WINE 2010.

[5] 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.

[6] C. Budak, D. Agrawal, and A. E. Abbadi, "Limiting the spread of misinformation in social networks," in WWW 2011.

[7] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in KDD 2009.

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

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

[10] S. Cheng, H. Shen, J. Huang, G. Zhang, and X. Cheng, "StaticGreedy: solving the scalability-accuracy dilemma in influence maximization," in CIKM 2013.

[11] K. L. Chung, "Lectures from Markov processes to Brownian motion," Springer-Verlag, New YorkCBerlin, 1982.

[12] P. Clifford and A. Sudbury, "A model for spatial conflict," Biometrika, 60(3):581-588, 1973.

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

[14] E. Even-Dar and A. Shapira, "A note on maximizing the spread of influence in social networks," in Internet and Network Economics, pages 281-286. Springer, 2007.

[15] U. Feige, "A threshold of ln n for approximating set cover," Journal of the ACM (JACM), 45(4):634-652, 1998.

[16] S. Ge, U. Hou, N. Mamoulis, and D. W. Cheung, "Efficient All Top-k Computation - A Unified Solution for All Top-k, Reverse Top-k and Top-m Influential Queries," in TKDE 2013.

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

[18] A. Goyal, F. Bonchi, and L. V. Lakshmanan, "A data-based approach to social influence maximization," in PVLDB 2012.

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

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

[21] J. Guo, P. Zhang, C. Zhou, Y. Cao, and L. Guo, "Personalized influence maximization on social networks," in CIKM 2013.

[22] X. He, G. Song, W. Chen, and Q. Jiang, "Influence blocking maximization in social networks under the competitive linear threshold model," in SDM 2012.

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

[24] Q. Jiang, G. Song, G. Cong, Y. Wang, W. Si, and K. Xie, "Simulated Annealing Based Influence Maximization in Social Networks," in AAAI 2012.

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

[26] M. Kimura and K. Saito, "Tractable models for information diffusion in social networks," in PKDD 2006.

[27] M. Kimura, K. Saito, R. Nakano, and H. Motoda, "Extracting influential nodes on a social network for information diffusion," in Data Mining and Knowledge Discovery, 2010, 20(1): 70-97.

[28] Y. Li, W. Chen, Y. Wang, and Z.-L. Zhang, "Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships," in WSDM, 2013.

[29] B. Liu, G. Cong, Y. Zeng, D. Xu, and Y. Chee, "Influence Spreading Path and its Application to the Time Constrained Social Influence Maximization Problem and Beyond," in TKDE, 2013.

[30] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in KDD 2007.

[31] E. Mossel and S. Rich, "On the Submodularity of Influence in Social Networks," in STOC 2007.

[32] R. Narayanam and Y. Narahari, "A shapley value based approach to discover influential nodes in social networks," in TASAE 2010.

[33] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of the approximations for maximizing submodular set functions," Mathematical Programming, 14: 265-294, 1978.

[34] N. Ohsaka, T. Akiba, Y. Yoshida, and K. I. Kawarabayashi, "Fast and Accurate Influence Maximization on Large Networks with Pruned Monte-Carlo Simulations," in AAAI 2014.

[35] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in KDD 2002.

[36] M. G. Rodriguez, J. Leskovec, and A. Krause, "Inferring networks of diffusion and influence," in KDD 2010.

[37] M. G. Rodriguez and B. Schölkopf, "Influence maximization in continuous time diffusion networks," in ICML 2012.

[38] K. Saito, R. Nakano, and M. Kimura, "Prediction of information diffusion probabilities for independent cascade model," in KES 2008.

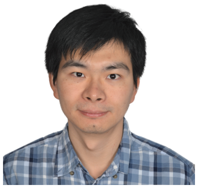
[39] J. Tang, J. Sun, C. Wang, and Z. Yang, "Social influence analysis in large-scale networks," in KDD 2009.

[40] Y. Wang, G. Cong, G. Song, and K. Xie, "Community-based greedy algorithm for mining top-k influential nodes in mobile social networks," in KDD 2010.

[41] D. B. West, "Introduction to graph theory," Prentice hall Englewood Cliffs, 2001.

[42] 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 ICDM 2013.

Chuan Zhou obtained Bachelor degree from Sichuan University in 2008, and Ph.D. degree from Chinese Academy of Sciences in 2013, both in mathematics. Currently, he is an Assistant Professor at the Institute of Information Engineering, Chinese Academy of Sciences. His research interests include stochastic analysis, data mining and social networks.



Peng Zhang received his PhD degree in July 2009 from the Chinese Academy of Sciences. Since then, Dr. Zhang has worked with two universities in the USA and one national laboratory at the Chinese Academy of Sciences. He won the Best Paper Award at ICCS-14 (ERA Rank A) held in Queensland, Australia. In January 2014, he joined the QCIS research center, University of Technology Sydney (UTS) as a Lecturer.

Dr. Zhang is an Associate Editor in two Springer journals, Journal of Big Data, and Annals of Data Science. To date, Dr. Zhang has published over 100 papers, including IEEE TKDE, KDD, ICDM, SDM, AAAI, CIKM, WWW, PAKDD, etc. He has been serving as a PC member (reviewer) in IEEE TKDE, ACM TKDD, KDD, ICDM, IJCAI, etc.

Wengyu Zang is a Ph.D. student at the Institute of Information Engineering, Chinese Academy of Sciences. She obtained bachelor degree at the Department of Computer Science, Beijing University of Posts and Telecommunications. Her research focuses on data stream mining and social network mining.

Li Guo is a Professor at the Institute of Information Engineering, Chinese Academy of Sciences. She is also the chairman of the Intelligent Information Processing Research Center, Institute of Information Engineering, Chinese Academy of Sciences. Her research interests include data stream management systems and information security.