Social Network Analysis and Mining (2018) 8:10
https://doi.org/10.1007/s13278-018-0489-y

ORIGINAL ARTICLE

An efficient and effective hop-based approach for influence maximization in social networks

Jing Tang[1] · **Xueyan Tang**[1] · **Junsong Yuan**[2]

Received: 30 September 2017 / Revised: 10 January 2018 / Accepted: 31 January 2018

Abstract
Influence maximization in social networks is a classic and extensively studied problem that targets at selecting a set of initial seed nodes to spread the influence as widely as possible. However, it remains an open challenge to design fast and accurate algorithms to find solutions in large-scale social networks. Prior Monte Carlo simulation-based methods are slow and not scalable, while other heuristic algorithms do not have any theoretical guarantee and they have been shown to produce poor solutions for quite some cases. In this paper, we propose hop-based algorithms that can be easily applied to billion-scale networks under the commonly used Independent Cascade and Linear Threshold influence diffusion models. Moreover, we provide provable data-dependent approximation guarantees for our proposed hop-based approaches. Experimental evaluations with real social network datasets demonstrate the efficiency and effectiveness of our algorithms.

Keywords Influence maximization · Social networks · Hop-based influence estimation · Submodular

1 Introduction

Information can be disseminated widely and rapidly through social networks with "word-of-mouth" effects. Viral marketing is such a typical application in which new products or activities are advertised by some influential users in the social network to other users in a cascading manner (Domingos and Richardson 2001). A large amount of recent work (Kempe et al. 2003; Leskovec et al. 2007b; Chen et al. 2009, 2010a, b; Goyal et al. 2011a, b, c; Jung et al. 20112; Cheng et al. 2013, 2014; Zhou et al. 2013, 2014; Borgs et al. 2014; Cohen et al. 2014; Lee and Chung 2014; Ohsaka et al. 2014, 2017; Tang et al. 2014, 2015; Song et al. 2015; Galhotra et al. 2016; Nguyen et al. 2016a, b; Arora et al. 2017; Tang et al. 2018a) has been focusing on *influence maximization* in viral marketing, which targets at selecting a set of initial seed nodes in the social network to spread the influence as widely as possible. The influence maximization problem was formulated by (Kempe et al. 2003) with two basic diffusion models, namely the *Independent Cascade* (IC) and *Linear Threshold* (LT) models. Although finding the optimal seed set is NP-hard (Kempe et al. 2003), a simple greedy hill-climbing algorithm has a $(1 - 1/e)$-approximation guarantee due to the submodularity and monotone properties of the influence spread under these models (Nemhauser et al. 1978). Follow-up studies have mostly concentrated on efficient implementation of the hill-climbing algorithm for large-scale social networks. The key difference among various methods lies in how to estimate the influence spread of a seed set.

It is known that computing the exact influence spread on general graphs is #P-hard for both the IC and LT models (Chen et al. 2010a, b). Thus, some Monte Carlo simulation-based methods (Kempe et al. 2003; Leskovec et al. 2007b; Goyal et al. 2011b; Zhou et al. 2013, 2014; Ohsaka et al. 2014) estimate the influence spread by reachability tests, while some *reverse influence sampling* methods carry out seed selection (Borgs et al. 2014; Tang et al. 2014, 2015; Nguyen et al. 2016a, b) by leveraging the concept of reverse reachability. These sampling-based methods can provide theoretical guarantees up to $(1 - 1/e - \varepsilon)$-approximation.

✉ Jing Tang
tang0311@ntu.edu.sg

Xueyan Tang
asxytang@ntu.edu.sg

Junsong Yuan
jsyuan@buffalo.edu

[1] School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798, Singapore

[2] Computer Science and Engineering Department, State University of New York at Buffalo, Buffalo 602000, New York, USA

Published online: 10 February 2018

However, the sampling-based methods can encounter the efficiency problem even for the dramatically improved versions (Borgs et al. 2014; Ohsaka et al. 2014; Tang et al. 2014, 2015; Nguyen et al. 2016a, b) as they may consume a lot of time/memory to obtain/store one sample. Other heuristic methods (Chen et al. 2009, 2010a, b; Goyal et al. 2011a, c; Jung et al. 20112; Cheng et al. 2013, 2014; Cohen et al. 2014; Lee and Chung 2014; Song et al. 2015; Galhotra et al. 2016) conduct rough estimation of the influence spread either by exploiting some related features (such as node degrees) or extracting subgraphs where the influence spread is easier to compute. However, these heuristics do not have any theoretical guarantee and they have been shown to suffer from the effectiveness problem.

To achieve both efficiency and effectiveness, our conference paper (Tang et al. 2017a) proposes a new hop-based approach for the influence maximization problem under the IC model. Unlike other heuristic methods, we provide provable data-dependent approximation guarantees for our methods. Experimental results show that our hop-based methods outperform the existing heuristics and perform as well as the sampling-based methods in terms of the influence spread produced. Meanwhile, our hop-based methods run much faster than the sampling-based methods. For a large social network with billions of edges (i.e., Twitter), only our hop-based methods and some ineffective heuristics can work with acceptable time and memory usage for various distributions of propagation probabilities. In this paper, we considerably extend our conference paper (Tang et al. 2017a) from three aspects: (1) we formally present the extension of our hop-based approach to the LT model, (2) we enhance the theoretical analysis by leveraging the concepts of curvature and data-dependent approximation, and (3) more experiments are carried out to demonstrate that our hop-based methods perform well under the IC model even when the propagation probabilities are large (e.g, all the propagation probabilities are set to 0.1) and under the LT model, and verify the theoretical guarantees of our proposed methods under both the IC and LT models.

Our contributions are summarized as follows.

1. We propose hop-based influence estimation algorithms for efficiently selecting seed nodes to maximize the influence spread under both the IC and LT models.
2. We develop an upper bounding approach on the influence generated by a singleton seed set to further speed up seed selection.
3. We carry out theoretical analysis for our hop-based algorithms and derive data-dependent approximation guarantees.
4. We conduct extensive experiments with several real social network datasets. The results demonstrate the efficiency and effectiveness of our hop-based algorithms.

The rest of this paper is organized as follows. Section 2 introduces the influence maximization problem and the greedy hill-climbing algorithm. Section 3 elaborates our algorithm design under both the IC and LT models. Section 4 analyzes the theoretical guarantees and time/space complexities. Section 5 presents the experimental study. Section 6 reviews the related work. Finally, Sect. 7 concludes the paper.

2 Preliminaries

2.1 Problem definition

Let $G = (V, E)$ be a directed graph modeling an social network, where the nodes V represent users and the edges E represent the connections among users (e.g., followships on Twitter). For each directed edge $\langle u, v\rangle \in E$, we refer to v as a *neighbor* of u, and refer to u as an *inverse neighbor* of v.

We shall focus on the Independent Cascade (IC) and Linear Threshold (LT) models (Kempe et al. 2003)—two representative and most widely studied diffusion models for influence propagation. In both models, a propagation probability $p_{u,v}$ is associated with each edge $\langle u, v\rangle$, representing the probability for v to be activated by u through the edge. Let N_u denote the set of node u's neighbors, i.e., $N_u = \{v : v \in V, \langle u, v\rangle \in E\}$. Given a set of seed nodes S, the diffusion process proceeds as follows. Initially, the seed nodes S are activated, while all the other nodes are not activated. When a node u first becomes activated, it attempts to further activate its neighbors who are not yet activated. This process repeats until no more node can be activated. The difference between the IC and LT models lies in the details of node activation:

- *IC model* When a node u first becomes activated, it has a *single* chance to activate each inactive neighbor v with a probability $p_{u,v}$. After that, u stops activating any other nodes.
- *LT model* It requires the propagation probabilities to satisfy $\sum_{u\in I_v} p_{u,v} \le 1, \quad \text{where} \quad I_v = \{w : w \in V, \langle w, v\rangle \in E\}$ denotes the set of node v's inverse neighbors. Each node v uniformly and randomly selects a threshold θ_v from the interval [0, 1]. An inactive node v becomes activated only if $\sum_{u\in I_v^A} p_{u,v} \ge \theta_v$, where $I_v^A \subseteq I_v$ denotes the set of v's inverse neighbors that are activated.

The *influence spread* of the seed set S, denoted by $\sigma(S)$, is the expected number of nodes activated by the above process. The *influence maximization* problem (Kempe et al. 2003) is to find a set S of k nodes to maximize $\sigma(S)$, where k is a given parameter. Formally,

$$\arg\max_{|S|=k} \sigma(S). \tag{1}$$

2.2 Greedy heuristic

The influence function $\sigma(\cdot)$ has been proved to be submodular and monotone under the IC and LT models (Kempe et al. 2003). Thus, a simple greedy hill-climbing algorithm that provides $(1-1/e)$-approximation (Nemhauser et al. 1978) was proposed for influence maximization as described in Algorithm 1. It starts with an empty seed set $S=\emptyset$. In each iteration, the greedy heuristic chooses a new seed u from the non-seed nodes $V \setminus S$ with the largest marginal influence gain $\sigma(S\cup\{u\})-\sigma(S)$ and adds u to S. The algorithm stops after selecting k seeds.

Algorithm 1: Greedy(G,σ)

```
initialize S ← ∅;
while the size of S is smaller than k do
    find u ← arg max_{v∈V\S} {σ(S ∪ {v}) − σ(S)};
    S ← S ∪ {u};
return S;
```

A CELF technique (Leskovec et al. 2007b) can be used to enhance the efficiency of the greedy algorithm due to the submodularity of influence spread. Specifically, it may not be necessary to evaluate the marginal influence gain for every node of $V\backslash S$ in each iteration. Due to the submodularity, the marginal gains can only decrease over iterations. Thus, the marginal gains obtained in the previous iterations can be used as upper bounds for a new iteration. In the new iteration, the nodes can be evaluated in decreasing order of these upper bounds. Once the largest marginal gain evaluated is greater than the upper bound of the next node to evaluate, the evaluation can stop as none of the remaining nodes would be able to produce a larger marginal gain.

3 Hop-based approaches

We start by studying some empirical results of influence spread tested on a real social network graph—Twitter (41.7 million nodes and 1.5 billion edges) (Kwak et al. 2010). Figures 1 and 2 plot the influence spread within different numbers of hops of propagation under the IC and LT models (the data points of ∞ hops represent the actual influence spread without any hop limit, and the seed set size is represented as a permillage of the entire node set in the social network). We use the Monte Carlo simulation via Breadth-first search (Kempe et al. 2003) to sample the number of nodes activated by different hops of propagation. We take the average of 10,000 Monte Carlo simulations as the estimated influence

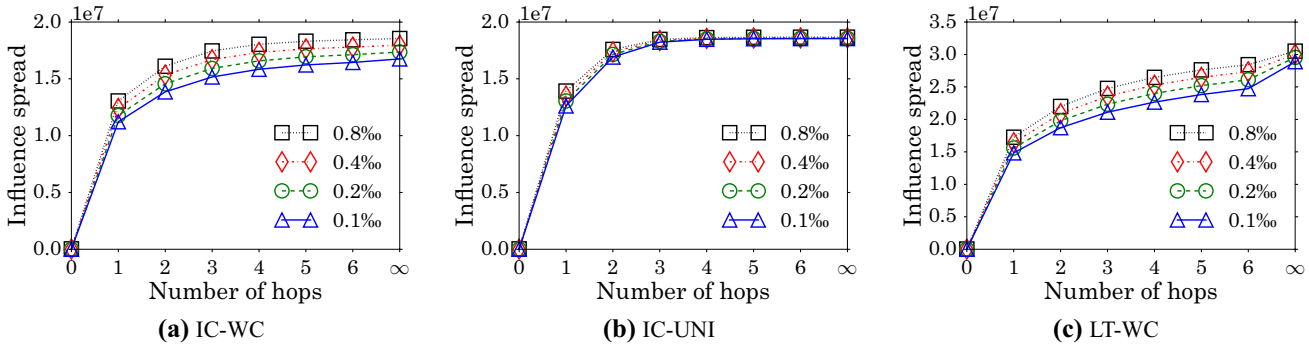


Fig. 1 Influence spread of high degree nodes for different hops of propagation on Twitter

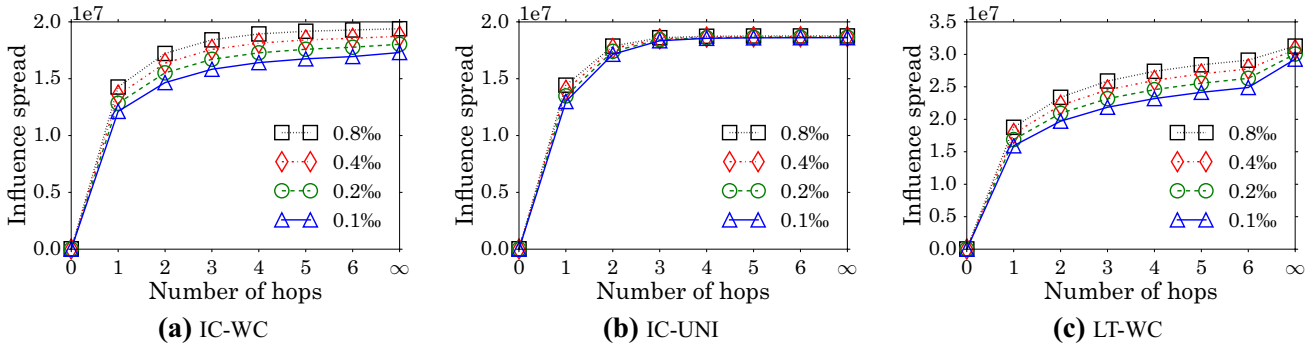


Fig. 2 Influence spread of top influential nodes for different hops of propagation on Twitter

spread of different hops. Two propagation probability settings are used: (1) under the WC setting, the propagation probability $p_{u,v}$ of each edge $\langle u, v\rangle$ in the social network is set to the reciprocal of v's in-degree, and (2) under the UNI setting, $p_{u,v}$ of each edge $\langle u, v\rangle$ is set to 0.1. We test two typical seed sets for influence maximization: selecting the nodes with highest degrees (Fig. 1) and selecting the top influential nodes (Fig. 2) using a greedy hill-climbing heuristic (Kempe et al. 2003) as described in Sect. 2.2. As seen from Figs. 1 and 2, the increase in the influence spread for considering each additional hop of propagation generally decreases with increasing number of hops. The majority of influence spread is produced within the first few hops of propagation. Similar trends have also been observed by several measurement-driven studies on real social networks (Leskovec et al. 2007a; Cha et al. 2009; Goel et al. 2012). For example, Goel et al. (2012) showed that less than 10% of the cascades in the diffusion are more than 2 hops away from the seed. These observations motivate us to design hop-based algorithms to efficiently capture the major influence propagation, especially for the first two hops of propagation.

3.1 Hop-based influence estimation under IC model

A hop-based algorithm focuses on the influence propagation up to a given number of h hops starting from the initial seed set. For $h = 1$ and $h = 2$, we can efficiently calculate the *exact* influence spread within h hops of propagation under the IC model and maintain it incrementally when the seed set expands. Let $\pi_h^S(v)$ denote the probability for a node v to be activated within h hops of propagation from a seed set S, and let $\sigma_h(S)$ denote the influence spread produced within h hops of propagation from S.

One Hop of Propagation We first model one hop of propagation. Obviously, for all the seed nodes $v \in S$, we have $\pi_1^S(v) = 1$. With one hop of propagation, for all the non-seed nodes $v \notin S$, v can only be activated directly by its inverse neighbors I_v who are seed nodes in S. Since each of such v's inverse neighbors activates v independently, the probability for all of them to fail to activate v is $\prod_{w\in I_v\cap S}(1 - p_{w,v})$. Consequently, the probability for v to be activated is $1 - \prod_{w\in I_v\cap S}(1 - p_{w,v})$. Thus, for any node $v \in V$, its one-hop activation probability is given by

$$\pi_1^S(v) = \begin{cases} 1, & \text{if } v \in S, \\ 1 - \prod_{w\in I_v\cap S}(1 - p_{w,v}), & \text{otherwise.} \end{cases} \tag{2}$$

Now, we show how to maintain $\pi_1^S(v)$ when the seed set changes. Suppose that $\pi_1^S(v)$ is known for every node $v \in V$. If a new seed node u is added to a seed set S, it is clear that the activation probability of the new seed becomes 1, i.e., $\pi_1^{S\cup\{u\}}(u) = 1$. In addition to u, only the one-hop activation probabilities of its neighbors are affected. So, we can compute $\pi_1^{S\cup\{u\}}(v)$ based on $\pi_1^S(v)$ for each node $v \in N_u$ (see Fig. 3). The new activation probability $\pi_1^{S\cup\{u\}}(v)$ is given by

$$\begin{aligned} \pi_1^{S\cup\{u\}}(v) &= 1 - \prod_{w\in(I_v\cap S)\cup\{u\}} (1 - p_{w,v}) \\ &= 1 - \left(1 - \pi_1^S(v)\right) \cdot (1 - p_{u,v}). \end{aligned} \tag{3}$$

Algorithm 2 calculates the increment of one-hop influence spread $\sigma_1(S \cup \{u\}) - \sigma_1(S)$ efficiently by maintaining $\pi_1^S(v)$ for every node v based on the above equation.

Algorithm 2: OneHopIncIC(G, S, u)

```
1 π_1^{S∪{u}}(u) ← 1;
2 for each node v ∈ N_u \ S do
3     π_1^{S∪{u}}(v) ← 1 − (1 − π_1^S(v)) · (1 − p_{u,v});
4 return Σ_{v∈{u}∪(N_u\S)} (π_1^{S∪{u}}(v) − π_1^S(v));
```

Two Hops of Propagation To better approximate $\sigma(S)$, we next model two hops of propagation. As illustrated in Fig. 4a, with two hops of propagation, a non-seed node v may be activated directly by a seed node u_i or indirectly via a neighbor w_j of a seed node u_i. In the former case, the probability for v to be activated by u_i is $p_{u_i,v}$, which can be rewritten as $p_{u_i,v} \cdot \pi_1^S(u_i)$ since $\pi_1^S(u_i) = 1$. In the latter case, the probability for v to be activated by w_j is $p_{w_j,v} \cdot \pi_1^S(w_j)$.

Since the activation probability of each seed node $u_i \in S$ is 1, Fig. 4a is equivalent to Fig. 4b in which v is activated independently by all of its inverse neighbors. As a result, the probability for v to be activated is given by $1 - \prod_{w\in I_v}(1 - p_{w,v} \cdot \pi_1^S(w))$. Thus, for any node $v \in V$, its two-hop activation probability is given by

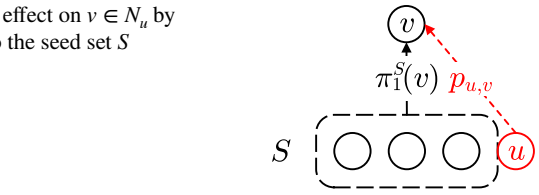


Fig. 3 The effect on $v \in N_u$ by adding u to the seed set S

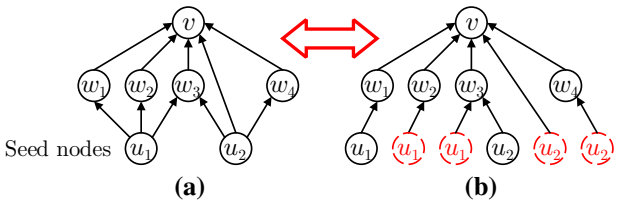


Fig. 4 An example of how a non-seed node v is activated

$$\pi_2^S(v) = \begin{cases} 1, & \text{if } v \in S, \\ 1 - \prod_{w \in I_v} \left(1 - p_{w,v} \cdot \pi_1^S(w)\right), & \text{otherwise.} \end{cases} \tag{4}$$

According to the above equation, we can obtain $\pi_2^S(v)$ based on $\pi_1^S(w)$'s of its inverse neighbors. When a new seed node u is added, only the nodes within two hops of u are affected. Let $N_u^2 = N_u \cup \left(\bigcup_{w \in N_u} N_w\right) \setminus \{u\}$ denote the set of nodes within two hops of u. Figure 5 illustrates the effect of adding u to S on the activation of a node v in N_u^2. Due to the independence among all two-hop activation paths as discussed above, we can incrementally update the two-hop activation probability by considering the outgoing edges from u one at a time.

Theorem 1 *After adding a seed u to S, the new two-hop activation probability $\pi_2^{S\cup\{u\}}(v)$ can be computed by*

$$\pi_2^{S\cup\{u\}}(v) = 1 - \left(1 - \pi_2^S(v)\right) \cdot \prod_{w \in (M_{u,v} \cup \{u\})} \rho(S, u, v, w), \tag{5}$$

where $\rho(S,u,v,w) \triangleq \frac{1-p_{w,v}\cdot\pi_1^{S\cup\{u\}}(w)}{1-p_{w,v}\cdot\pi_1^S(w)}$ and $M_{u,v}$ denotes the set of intermediate nodes connecting u and v, i.e., $M_{u,v} = \{w : \langle u, w\rangle \in E \text{ and } \langle w, v\rangle \in E\}$.

Algorithm 3: TwoHopsIncIC(G, S, u)

```
1  π_1^{S∪{u}}(u) ← 1;
2  π_2^{S∪{u}}(u) ← 1;
3  for each node v ∈ N_u^2 \ S do
4      π_2^{S∪{u}}(v) ← π_2^S(v);
5  for each node w ∈ N_u \ S do
6      π_1^{S∪{u}}(w) ← 1 − (1 − π_1^S(w)) · (1 − p_{u,w});
7      π_2^{S∪{u}}(w) ← 1 − (1 − π_2^{S∪{u}}(w)) · ρ(S, u, w, u);
8      for each node v ∈ N_w \ S do
9          π_2^{S∪{u}}(v) ← 1 − (1 − π_2^{S∪{u}}(v)) · ρ(S,u,v,w);
10 return Σ_{v∈{u}∪(N_u^2\S)} (π_2^{S∪{u}}(v) − π_2^S(v));
```

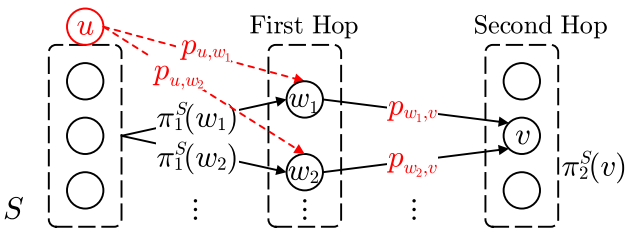


Fig. 5 The effect on $v \in N_u^2$ by adding u to the seed set S

We leave the formal proofs of all theoretical results to the appendix. Algorithm 3 performs the updates on the activation probabilities of the nodes within two hops of u when u is added as a new seed to S. Lines 1–2 set the one-hop and two-hop activation probabilities of the new seed node u to 1. Lines 3–4 initialize $\pi_2^{S\cup\{u\}}(v)$ for all the nodes within two hops of u. Line 6 computes the new one-hop activation probabilities for all of u's neighbors as explained earlier. For each node $v \in N_u^2 \setminus S$, lines 7–9 calculate the new two-hop activation probability $\pi_2^{S\cup\{u\}}(v)$ in an iterative manner according to Theorem 1. In this way, we can save a huge amount of space for storing the intermediate nodes $M_{u,v}$ for every pair of nodes u and v. Finally, the algorithm returns the total increment of two-hop influence spread $\sigma_2(S \cup \{u\}) - \sigma_2(S)$.

3.2 Hop-based influence estimation under LT model

Similar to the IC model, we can easily compute the exact influence spread within one and two hops of propagation under the LT model.

One Hop of Propagation For all the seed nodes $v \in S$, it is obvious that $\pi_1^S(v) = 1$. Meanwhile, for all the non-seed nodes $v \notin S$, the activation probability can be computed by $\pi_1^S(v) = \sum_{u \in I_v \cap S} p_{u,v}$. Thus, for any node $v \in V$, its one-hop activation probability is given by

$$\pi_1^S(v) = \begin{cases} 1, & \text{if } v \in S, \\ \sum_{u \in I_v \cap S} p_{u,v}, & \text{otherwise.} \end{cases} \tag{6}$$

When adding a new seed u to a seed set S, the new activation probability of u is 1, i.e., $\pi_1^{S\cup\{u\}}(u) = 1$. Meanwhile, for each non-seed neighbor v of u, i.e., $v \in N_u \setminus S$, the new activation probability of v is given by

$$\pi_1^{S\cup\{u\}}(v) = \sum_{w \in I_v \cap (S\cup\{u\})} p_{w,v} = \pi_1^S(v) + p_{u,v}. \tag{7}$$

Thus, the increment of v's activation probability is $p_{u,v}$, which is independent of $\pi_1^S(v)$. Therefore, under the LT model, the total increment of one-hop influence spread $\sigma_1(S \cup \{u\}) - \sigma_1(S)$ by adding u to S is easy to compute as shown in Algorithm 4.

Algorithm 4: OneHopIncLT(G, S, u)

```
1  Δ ← 1 − π_1^S(u);
2  for each node v ∈ N_u \ S do
3      Δ ← Δ + p_{u,v};
4  return Δ;
```

Algorithm 5: TwoHopsIncLT(G, S, u)

```
1 Δ ← 1 − π_2^S(u);
2 for each node w ∈ N_u \ S do
3     Δ ← Δ + (1 − π_1^S(u)) · p_{u,w};
4     for each node v ∈ N_w \ (S ∪ {u}) do
5         Δ ← Δ + p_{u,w} · p_{w,v};
6 return Δ;
```

Two Hops of Propagation Since all the propagation paths from seed nodes to a non-seed node within two hops are acyclic, for any non-seed node $v \notin S$, the activation probability of v is $\pi_2^S(v) = \sum_{w \in I_v} \left(p_{w,v} \cdot \pi_1^S(w)\right)$. Thus, for any node $v \in V$, its two-hop activation probability is given by

$$\pi_2^S(v) = \begin{cases} 1, & \text{if } v \in S, \\ \sum_{w \in I_v} \left(p_{w,v} \cdot \pi_1^S(w)\right), & \text{otherwise.} \end{cases} \tag{8}$$

When adding a new seed u to a seed set S, for any node $v \in N_u^2 \setminus S$, the new activation probability of v is given by

$$\begin{aligned} \pi_2^{S\cup\{u\}}(v) &= \sum_{w \in I_v} \left(p_{w,v} \cdot \pi_1^{S\cup\{u\}}(w)\right) \\ &= \sum_{w \in I_v} \left(p_{w,v} \cdot \pi_1^S(w)\right) \\ &\quad + \sum_{w \in I_v} \left(p_{w,v} \cdot \left(\pi_1^{S\cup\{u\}}(w) - \pi_1^S(w)\right)\right) \\ &= \pi_2^S(v) + \left(1 - \pi_1^S(u)\right) \cdot p_{u,v} \\ &\quad + \sum_{w \in M_{u,v} \setminus S} \left(p_{u,w} \cdot p_{w,v}\right). \end{aligned} \tag{9}$$

The last equality is because for any node $w \in I_v$, $\pi_1^{S\cup\{u\}}(w) - \pi_1^S(w) = 1 - \pi_1^S(u)$ if $w = u$, and $\pi_1^{S\cup\{u\}}(w) - \pi_1^S(w) = p_{u,w}$ if $w \in M_{u,v} \setminus S$, and $\pi_1^{S\cup\{u\}}(w) - \pi_1^S(w) = 0$ if $w \in I_v \setminus (M_{u,v} \cup \{u\})$ or $w \in M_{u,v} \cap S$. Analogous to the analysis for incrementally updating the two-hop activation probability under the IC model, we can compute the increment of influence spread by adding a new seed u under the LT model. Algorithm 5 describes the detail procedure. Line 1 calculates the increment of activation probability on u itself. For each node $v \in N_u^2 \setminus S$, lines 3–5 calculate the increment of activation probability on v according (9).

Discussions The above analysis shows that the one-hop and two-hop activation probabilities of nodes under both the IC and LT models can be computed exactly in a convenient manner so that the one-hop and two-hop influence spreads can be efficiently updated when the seed set changes. We remark that this is due to the independence among the activations of the inverse neighbors of each node. For one hop of propagation, all the propagation paths from seed nodes to a non-seed node are disjoint. Thus, their activations on the non-seed node are independent. For two hops of propagation, the propagation paths from seed nodes to a non-seed node may share common nodes. For instance, in Fig. 4a, the paths $u_1 \to w_3 \to v$ and $u_2 \to w_3 \to v$ share w_3, and the paths $u_1 \to w_1 \to v$ and $u_1 \to w_2 \to v$ share u_1. In the former case, since u_1 and u_2 are both seed nodes, their activations (with probabilities of 1) can be considered independent. Thus, we can compute w_3's one-hop activation probability conveniently. In the latter case, again, since u_1 is a seed node with activation probability of 1, the propagations along the two paths can be considered independent. Unfortunately, such an independence does not remain when three or more hops of propagation are considered. As mentioned, computing the exact influence spread without any hop limit under both the IC and LT models is #P-hard (Chen et al. 2010a, b). We leave it for future work to study efficient computation or approximation of the influence spread for more hops.

3.3 Further improvement on efficiency

Note that selecting the first seed in Algorithm 1 requires calculating the influence spread $\sigma(\{v\})$ for every node v even when the CELF technique (Leskovec et al. 2007b) is adopted. To avoid such computation, we develop an upper bound on $\sigma(\{v\})$ when hop-based influence estimation is applied.

Theorem 2 *Under both the IC and LT models, for each node $v \in V$, the h-hop influence spread $\sigma_h(\{v\})$ satisfies*

$$\sigma_h(\{v\}) \le 1 + \sum_{w \in N_v} \left(p_{v,w} \cdot \sigma_{h-1}(\{w\})\right). \tag{10}$$

Furthermore, let $\hat{\sigma}_0(\{v\}) = \sigma_0(\{v\}) = 1$ and $\hat{\sigma}_h(\{v\}) = 1 + \sum_{w \in N_v} \left(p_{v,w} \cdot \hat{\sigma}_{h-1}(\{w\})\right)$, then

$$\sigma_h(\{v\}) \le \hat{\sigma}_h(\{v\}). \tag{11}$$

Note that when $h = 1$, the upper bound $\hat{\sigma}_1(\{v\}) = 1 + \sum_{w \in N_v} p_{v,w}$ is the exact 1-hop influence spread of a single seed $\{v\}$ for the IC/LT model, i.e., $\hat{\sigma}_1(\{v\}) = \sigma_1(\{v\})$. Computing $\hat{\sigma}_1(\{v\})$ for a node v has a time complexity of $O(1 + |N_v|)$. Thus, it takes a time complexity of $O(\sum_{v \in V}(1 + |N_v|)) = O(|V| + |E|)$ to calculate $\hat{\sigma}_1(\{v\})$ for all nodes $v \in V$. The time complexity for computing the upper bound $\hat{\sigma}_2(\{v\})$ given in Theorem 2 is $O(1 + |N_v|)$ after obtaining $\hat{\sigma}_1(\{w\})$ for all nodes $w \in V$. Thus, the total time complexity for calculating $\hat{\sigma}_2(\{v\})$ for all nodes $v \in V$ is $O(|V| + |E|) + O(\sum_{v \in V}(1 + |N_v|)) = O(|V| + |E|) + O(|V| + |E|) = O(|V| + |E|)$, which is much lower than the time

complexity for computing the exact two-hop influence spread $\sigma_2(\{v\})$ for all nodes $v \in V$ using Algorithm 3 (respectively, Algorithm 5) for the IC model (respectively, the LT model) which is $O\big(|V| + |E| + \sum_{w \in V}(|I_w| \cdot |N_w|)\big)$ as shall be discussed in Sect. 4.2. On obtaining the upper bounds $\hat{\sigma}_h(\{v\})$, the CELF technique described in Sect. 2.2 can then be applied to the first iteration of Algorithm 1 so that only the influence spreads of a subset of nodes in V need to be calculated for selecting the first seed. We shall show in Sect. 5.2 that the upper bounding approach can dramatically reduce the running time of the two-hop method.

4 Theoretical analysis

In this section, we carry out theoretical analysis, including the approximation and complexity, for our hop-based algorithms.

4.1 Data-dependent approximation

We first show that the influence spread within h hops of propagation is submodular and monotone.

Theorem 3 *For any $h \geq 1$, the influence spread produced within h hops of propagation is submodular and monotone under both the IC and LT models.*

The theoretical guarantees for our hop-based algorithms to the influence maximization problem depend on the curvature of submodular functions.

Definition 1 (*Curvature* (*Conforti and Cornuéjols* 1984)) Given a submodular function f, the total curvature κ_f is defined as

$$\kappa_f = 1 - \min_{v \in V} \frac{f(V) - f(V \setminus \{v\})}{f(\{v\})}, \tag{12}$$

and the curvature $\kappa_f(S)$ with respect to a set $S \subseteq V$ is defined as

$$\kappa_f(S) = 1 - \min_{v \in S} \frac{f(S) - f(S \setminus \{v\})}{f(\{v\})}. \tag{13}$$

By the definition of curvature, it is easy to obtain that $0 \leq \kappa_f(S) \leq \kappa_f(V) = \kappa_f \leq 1$ for any $S \subseteq V$. The curvature κ_f measures how much f deviates from *modularity*. That is, the curvature κ_f of a modular function f is 0 and the curvature κ_f is larger if f is further away from modularity. Similarly, the curvature $\kappa_f(S)$ of S measures how much f deviates from modularity under the given context S. Let S^* denote the optimal seed set for maximizing the actual influence spread without any hop limit, i.e., $\sigma(S^*) = \max_{|S|=k} \sigma(S)$. Then, we can derive the following guarantees for our hop-based methods.

Theorem 4 *Under both the IC and LT models, let $\alpha \triangleq \sigma_h(S^*)/\sigma(S^*)$, the solution S_h returned by the greedy heuristic (Algorithm 1) with hop-based influence estimation satisfies*

$$\sigma(S_h) \geq \left(\frac{1}{\kappa_{\sigma_h}} (1 - e^{-\kappa_{\sigma_h}}) \alpha \right) \cdot \sigma(S^*). \tag{14}$$

Theorem 4 indicates that the hop-based methods can provide a multiplicative guarantee of $\frac{1}{\kappa_{\sigma_h}}(1 - e^{-\kappa_{\sigma_h}})\alpha$. Note that this approximation factor is dependent on the data but not the worst-case constant factor. Some previous work (Lu et al. 2015; Lin et al. 2017; Wang et al. 2017) has adopted the data-dependent approximation to indicate the theoretical guarantee. The data-dependent approximation is usually obtained from the sandwich approximation algorithm (Lu et al. 2015) that finds an approximate solution to the lower/upper bound of the original function. In our hop-based approaches, it is easy to see that $\sigma_h(S)$ is a lower bound of $\sigma(S)$ for any $h \geq 1$ and $S \subseteq V$. Thus, our hop-based approaches belong to the sandwich approximation scheme. On the other hand, thanks to the curvature κ_{σ_h}, the factor of $\frac{1}{\kappa_{\sigma_h}}(1 - e^{-\kappa_{\sigma_h}})$ slightly improves the usual approximation factor of $(1 - 1/e)$ for monotone submodular maximization subject to a cardinality constraint.

Next, we derive a lower bound on the ratio $\sigma_h(S^*)/\sigma(S^*)$ under the IC model in the class of scale-free random graphs which are commonly used to model social networks (Barabási and Albert 1999; Li et al. 2012). The degree distribution of a scale-free (undirected) graph follows a power law. That is, the probability of a node having degree d is $P_0(d) = \frac{d^{-\gamma}}{\sum_{d_i=1}^{\infty} d_i^{-\gamma}}$, where γ is a given power scale parameter whose typical value is in the range of $2 \leq \gamma \leq 3$. We first analyze the expected number of nodes activated within one hop of propagation, which gives a lower bound on $\sigma_h(S)$ for any $h \geq 1$ since $\sigma_h(S)$ increases with h. Next, we derive an upper bound on the expected number of nodes activated $\sigma(S)$. Using the lower bound on $\sigma_h(S)$ and upper bound on $\sigma(S)$, we can derive a lower bound α on the ratio $\sigma_h(S)/\sigma(S)$.

Theorem 5 *For scale-free random graphs with propagation probability $p_{u,v} = p$ for every edge $\langle u, v \rangle \in E$ and any seed set S and any hop of $h \geq 1$, we have*

$$\frac{\mathbb{E}[\sigma_h(S)]}{\mathbb{E}[\sigma(S)]} \geq \frac{1-(1-k/|V|)(1-pk/|V|)}{1-(1-k/|V|)P_0(1)(1-pA)}, \quad (15)$$

where $A = 1 - \left(1 - \frac{k}{|V|}\right)P_1(1)$ *and* $P_1(d) = \frac{d^{1-\gamma}}{\sum_{d_i=1}^{\infty} d_i^{1-\gamma}}$.

Figure 6 shows the lower bound derived in (15) when varying the propagation probability p from 0 to 0.1 and the seed ratio k / $|V|$ from 0 to 0.5. We can see that the lower bound generally increases with both the seed ratio and propagation probability.

4.2 Complexity

We shall refer to the greedy heuristic (Algorithm 1) as the OneHop and TwoHop algorithms, respectively, when the influence spread is approximated by one hop and two hops of propagation.

Time Complexity The time complexity of Algorithm 2 (respectively, Algorithm 4) for the IC model (respectively, the LT model) is $O(1 + |N_u|)$. Thus, the time complexity of selecting one seed in the OneHop algorithm for the IC/LT model is $O\left(\sum_{u\in V}(1 + |N_u|)\right) = O(|V| + |E|)$. Therefore, the total time complexity of OneHop for the IC/LT model is $O(k(|V| + |E|))$. The time complexity of Algorithm 3 (respectively, Algorithm 5) for the IC model (respectively. the LT model) is $O\left(1 + |N_u| + \sum_{w\in N_u} |N_w|\right)$. Therefore, the time complexity of selecting one seed in the TwoHop algorithm for the IC/LT model is $O\left(\sum_{u\in V}(1 + |N_u| + \sum_{w\in N_u} |N_w|)\right)$ $= O\left(|V| + |E| + \sum_{u\in V}\sum_{w\in N_u} |N_w|\right) = O\left(|V| + |E| + \sum_{w\in V}(|I_w| \cdot |N_w|)\right)$. Thus, the total time complexity of TwoHop for the IC/LT model is $O\left(k\left(|V| + |E| + \sum_{w\in V}(|I_w| \cdot |N_w|)\right)\right)$.

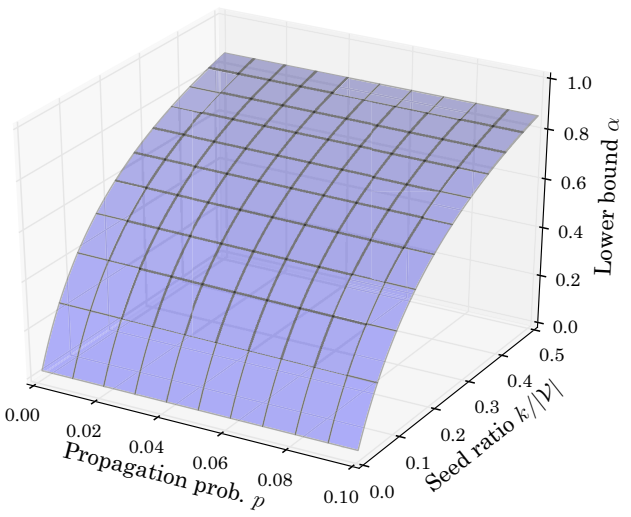


Fig. 6 Lower bound α for different propagation probabilities and seed ratios ($\gamma = 3$)

Space Complexity Besides the space used to store the graph, the OneHop algorithm only requires $O(|V|)$ space to store the one-hop activation probability $\pi_1^S(v)$ for every $v \in V$ before and after a new seed is added. Similarly, the TwoHop algorithm requires $O(|V|)$ space to store the one-hop and two-hop activation probabilities $\pi_1^S(v)$ and $\pi_2^S(v)$ for computing the influence increment. Thus, the space complexities of the OneHop and TwoHop algorithms for both the IC and LT models are $O(|V|)$.

5 Evaluation

This section experimentally evaluates our hop-based algorithms against the state of the art. All experiments are carried out on a machine with an Intel Xeon 2.4 GHz CPU and 64 GB memory.

5.1 Experimental setup

Datasets We use several real social network datasets in our experiments. We present the results for three representative datasets: NetHEPT (Chen 2009), LiveJournal (Leskovec and Krevl 2014), and Twitter (Kwak et al. 2010). Table 1 shows the statistics of these three datasets.

Algorithms We compare our hop-based algorithms, including OneHop, TwoHop and TwoHop-O (without the upper bounding technique described in Sect. 3.3), with the following state-of-the-art algorithms.

- HighDegree: Select the k nodes with highest degrees (Kempe et al. 2003).
- DegreeDiscount: The degree discount heuristic was developed by (Chen et al. 2009).
- IRIE: IRIE (Jung et al. 20112) is a state-of-the-art heuristic for the IC model. We set its algorithm parameters $\alpha = 0.7$ and $\theta = 1/320$, respectively, as suggested in (Jung et al. 20112).
- SIMPATH: SIMPATH (Goyal et al. 2011c) is a state-of-the-art heuristic for the LT model. We set its algorithm parameters $\eta = 0.001$ (the pruning threshold) and $l = 4$ (the lookahead value) as recommended in (Goyal et al. 2011c).

Table 1 Statistics of social network datasets

Dataset	$\|V\|$	$\|E\|$	Type	Avg. degree
NetHEPT	15.2K	31.4K	Undirected	4.1
LiveJournal	4.8M	69.0M	Directed	28.5
Twitter	41.7M	1.5B	Directed	70.5

- PMC: PMC (Ohsaka et al. 2014) is a state-of-the-art Monte Carlo simulation-based method originally proposed for the IC model. PMC reuses the samples generated to estimate the influence of nodes by averaging over all samples. Same as (Ohsaka et al. 2014), we generate 200 samples for influence estimation.
- IMM: IMM (Tang et al. 2015) is one of the most advanced sampling-based methods that can provide a $(1 - 1/e - \varepsilon)$-approximation guarantee with probability at least $1 - \delta$.
- D-SSA: D-SSA (Nguyen et al. 2016b) aims to further reduce the number of samples generated compared to IMM while providing the same approximation guarantee. We set $\varepsilon = 0.1$ and $\delta = 1/|V|$ for both IMM and D-SSA according to the default setting in (Nguyen et al. 2016b).

We adopt the C++ implementations of HighDegree, DegreeDiscount, IRIE, SIMPATH, PMC, IMM and D-SSA provided by their respective inventors, and we also implement our proposed hop-based algorithms in C++.

Parameter Settings We set the propagation probability via the following three widely adopted settings.

- Weighted Cascade (WC): $p_{u,v}$ of each edge $\langle u, v\rangle$ is set to the reciprocal of v's in-degree, i.e., $p_{u,v} = 1/|I_v|$.
- Trivalency (TRI): $p_{u,v}$ of each edge $\langle u, v\rangle$ is set by choosing a probability from the set $\{0.1, 0.01, 0.001\}$ at random.
- Uniform (UNI): $p_{u,v}$ of each edge $\langle u, v\rangle$ is set to 0.1.

To evaluate the seed sets returned by different algorithms, we estimate the influence spread of each seed set by taking the average measurement of 10,000 Monte Carlo simulations. We limit the running time of each algorithm up to 100 h (3.6×10^5 s).

5.2 Results under the IC Model

In the first set of experiments, we compare the algorithms under the IC model.

Influence Spread Tables 2, 3 and 4 show the influence spread produced by different algorithms on various graphs when the size of seed set is set to $k = 1, 10, 100, 1000$ under the IC model with the WC, TRI and UNI settings, respectively (referred to as IC-WC, IC-TRI and IC-UNI). Due to out-of-memory reasons and prohibitively long computation times, IRIE and PMC *failed* to produce results on the Twitter dataset under all the IC-WC, IC-TRI and IC-UNI settings, while both IMM and D-SSA *failed* on the LiveJournal and Twitter datasets under the IC-TRI and IC-UNI settings. From the results obtained, we can make the following observations. Our OneHop, TwoHop and TwoHop-O methods usually generate influence spread as high as that by the sampling-based methods PMC, IMM and D-SSA, where IMM and D-SSA can provide the state-of-the-art $(1 - 1/e - \varepsilon)$ -approximation guarantee. Our methods remarkably outperform both the HighDegree and DegreeDiscount heuristics (by up to 40%) on the NetHEPT dataset when selecting 100 and 1000 seeds under all the IC-WC, IC-TRI and IC-UNI settings and on the LiveJournal dataset when selecting 1000 seeds under the IC-WC setting. These observations demonstrate the effectiveness of our hop-based methods.

Running Time Figs. 7, 8 and 9 show the running times of different algorithms. The OneHop and DegreeDiscount methods run almost at the same speed which can find the top 1000 influential nodes on the Twitter dataset (with billions of edges) within 30 s. They are just slightly slower than the HighDegree method and run several orders faster than other methods, including IRIE, PMC, IMM, D-SSA, TwoHop and TwoHop-O (note that the y-axis is in logscale). This shows a trade-off between efficiency and effectiveness for the hop-based methods. Estimating the influence spread with a higher hop limit takes more time but can improve the quality of the seed set chosen. For example, on the LiveJournal

Table 2 Influence spread on various graphs under the IC-WC setting

Method	NetHEPT				LiveJournal				Twitter			
	$k = 1$	10	100	1000	$k = 1$	10	100	1000	$k = 1$	10	100	1000
HighDegree	44.53	292.34	1.13e3	4.15e3	7.17e3	4.93e4	1.47e5	3.27e5	1.19e6	4.93e6	1.05e7	1.54e7
DegreeDiscount	43.40	297.84	1.08e3	4.12e3	5.41e3	4.81e4	1.47e5	3.26e5	1.20e6	4.95e6	1.06e7	1.54e7
IRIE	35.28	307.08	1.49e3	5.74e3	5.80e3	5.02e4	1.72e5	3.92e5	–	–	–	–
PMC	44.68	316.13	1.52e3	5.84e3	9.65e3	5.40e4	1.77e5	4.02e5	–	–	–	–
IMM	42.45	308.99	1.50e3	5.79e3	9.68e3	5.37e4	1.75e5	3.99e5	1.22e6	4.95e6	1.09e7	1.58e7
D-SSA	43.56	311.98	1.49e3	5.80e3	9.60e3	5.39e4	1.78e5	4.01e5	1.23e6	4.96e6	1.06e7	1.57e7
OneHop	37.96	231.10	1.41e3	5.61e3	5.79e3	5.07e4	1.48e5	3.64e5	9.41e5	4.93e6	1.07e7	1.57e7
TwoHop/TwoHop-O	37.14	302.32	1.50e3	5.79e3	6.07e3	5.12e4	1.65e5	3.89e5	1.01e6	4.94e6	1.08e7	1.58e7

The field with "–" means that the method cannot run under the setting

Table 3 Influence spread on various graphs under the IC-TRI setting

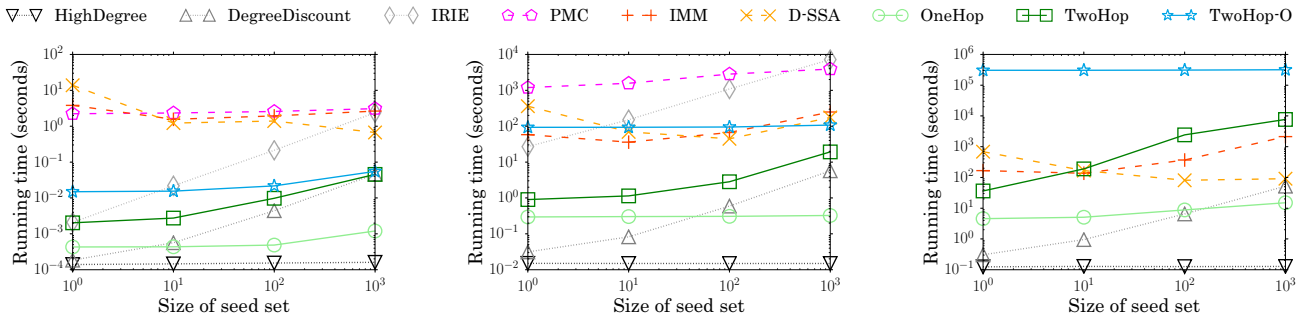
Method	NetHEPT				LiveJournal				Twitter			
	$k = 1$	10	100	1000	$k = 1$	10	100	1000	$k = 1$	10	100	1000
HighDegree	9.03	53.57	254.79	1.34e3	6.62e5	6.63e5	6.66e5	6.69e5	9.89e6	1.00e7	1.03e7	1.09e7
DegreeDiscount	8.79	54.70	245.07	1.35e3	6.62e5	6.64e5	6.66e5	6.70e5	9.89e6	1.00e7	1.03e7	1.09e7
IRIE	8.79	53.68	288.90	1.52e3	6.62e5	6.64e5	6.66e5	6.71e5	–	–	–	–
PMC	9.57	63.62	312.15	1.66e3	6.63e5	6.63e5	6.80e5	6.83e5	–	–	–	–
IMM	9.74	65.75	310.10	1.65e3	–	–	–	–	–	–	–	–
D-SSA	9.49	66.52	318.36	1.59e3	–	–	–	–	–	–	–	–
OneHop	7.82	57.22	309.20	1.60e3	6.62e5	6.64e5	6.66e5	6.70e5	9.82e6	9.99e6	1.03e7	1.09e7
TwoHop/TwoHop-O	9.71	65.55	318.38	1.64e3	6.62e5	6.64e5	6.66e5	6.71e5	9.89e6	1.01e7	1.03e7	1.09e7

The field with "–" means that the method cannot run under the setting

Table 4 Influence spread on various graphs under the IC-UNI setting

Method	NetHEPT				LiveJournal				Twitter			
	$k = 1$	10	100	1000	$k = 1$	10	100	1000	$k = 1$	10	100	1000
HighDegree	647.74	687.20	875.07	2.01e3	1.67e6	1.68e6	1.69e6	1.71e6	1.78e7	1.81e7	1.82e7	1.83e7
DegreeDiscount	647.74	684.63	865.44	2.02e3	1.68e6	1.68e6	1.69e6	1.71e6	1.78e7	1.81e7	1.82e7	1.83e7
IRIE	647.74	677.77	863.48	2.31e3	1.67e6	1.69e6	1.69e6	1.69e6	–	–	–	–
PMC	626.97	776.70	1.14e3	2.76e3	1.70e6	1.70e6	1.70e6	1.73e6	–	–	–	–
IMM	651.70	765.65	1.14e3	2.75e3	–	–	–	–	–	–	–	–
D-SSA	643.21	776.68	1.13e3	2.73e3	–	–	–	–	–	–	–	–
OneHop	647.74	721.22	1.02e3	2.57e3	1.68e6	1.68e6	1.68e6	1.69e6	1.78e7	1.81e7	1.82e7	1.84e7
TwoHop/TwoHop-O	647.74	720.05	1.04e3	2.67e3	1.69e6	1.71e6	1.71e6	1.71e6	1.79e7	1.82e7	1.82e7	1.84e7

The field with "–" means that the method cannot run under the setting



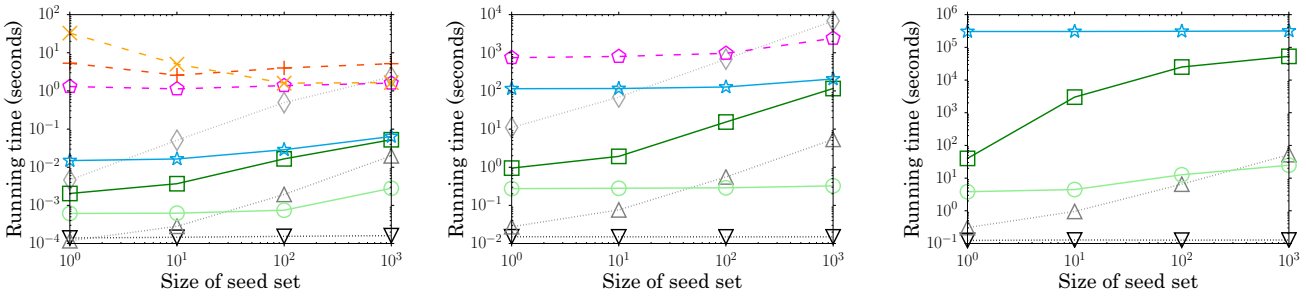
(a) NetHEPT **(b)** LiveJournal **(c)** Twitter (IRIE and PMC cannot run)

Fig. 7 Running time on various graphs under the IC-WC setting

dataset under the IC-WC setting (Table 2), the TwoHop method performs notably better than the OneHop method. If the application is highly time-sensitive, the OneHop method could be preferable to the TwoHop method. Otherwise, the TwoHop method is favoured since its running time is quite acceptable even for very large networks.

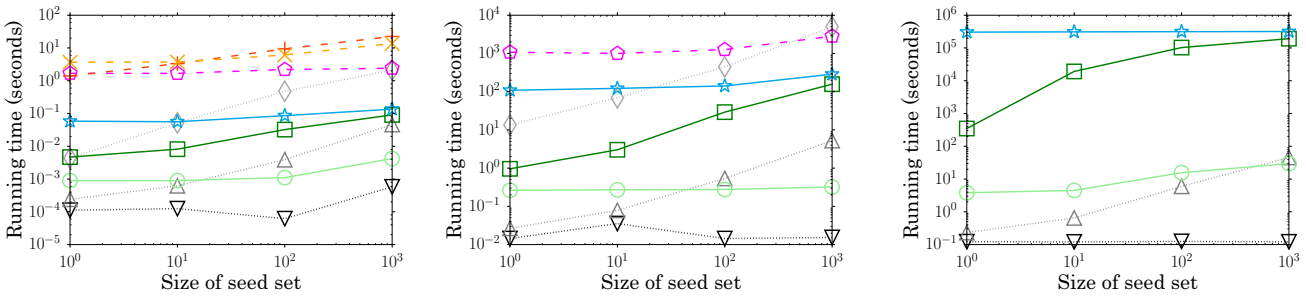
We also observe that while the TwoHop and TwoHop-O methods always produce the same seed set solution, the former runs significantly faster than the latter (by up to 4 orders of magnitude). This is because TwoHop-O consumes too much time on computing the influence spread of all the single seed sets. This demonstrates the efficiency of our upper bounding approach.

Moreover, we observe that the OneHop and TwoHop methods generally run much faster than the state-of-the-art IRIE, PMC, IMM and D-SSA methods. This demonstrates



(a) NetHEPT **(b)** LiveJournal (IMM and D-SSA cannot run) **(c)** Twitter (IRIE, PMC, IMM and D-SSA cannot run)

Fig. 8 Running time on various graphs under the IC-TRI setting



(a) NetHEPT **(b)** LiveJournal (IMM and D-SSA cannot run) **(c)** Twitter (IRIE, PMC, IMM and D-SSA cannot run)

Fig. 9 Running time on various graphs under the IC-UNI setting

the efficiency of our hop-based methods. Since IMM and D-SSA are reverse influence sampling methods, their running times heavily depend on the sizes of the reverse reachable sets sampled. It is known that when the social network has high propagation probabilities (e.g, IC-TRI and IC-UNI), IMM and D-SSA require significant memory and time to compute as the samples of reverse reachable sets are large, which makes IMM and D-SSA intractable on the LiveJournal and Twitter datasets. On the other hand, under the IC-WC model, for any node, the expected number of its direct reverse reachable neighbors is 1 because the edge propagation probabilities are given by the reciprocal of the node's in-degree. This significantly limits the sizes of reverse reachable sets and favors IMM and D-SSA. However, the running times of IMM and D-SSA are still very sensitive to the propagation probabilities. Table 5 shows the trends when we scale the propagation probabilities under the WC model by a small factor up to 1.5 (for selecting 1000 seeds in the Twitter graph). It can be seen that the running times of IMM and D-SSA increase rapidly with the scale factor and far exceed our hop-based methods even at a minor factor of 1.2. In contrast, our hop-based methods are much less sensitive to the propagation probabilities. Meanwhile, although IRIE and PMC are also less sensitive to the propagation probabilities, they are not scalable to large datasets such as Twitter.

Table 5 Running time (seconds) for selecting 1000 seeds in the Twitter graph under the IC-WC setting and different propagation probability scale factors f

Method	$f = 1.0$	1.1	1.2	1.3	1.4	1.5
HighDegree	0.20	0.20	0.21	0.21	0.20	0.21
DegreeDiscount	52.23	51.79	51.71	52.14	52.70	51.53
IRIE	–	–	–	–	–	–
PMC	–	–	–	–	–	–
IMM	2.13e3	3.52e4	–	–	–	–
D-SSA	91.21	3.76e3	2.44e4	–	–	–
OneHop	15.07	16.08	17.70	17.18	17.33	17.85
TwoHop	7.70e3	7.89e3	8.11e3	8.25e3	7.94e3	8.32e3

The field with "–" means that the method cannot run under the setting

Remark on Memory Usage Recall that our hop-based algorithms just need $O(|V|)$ space which is negligible compared to the space $O(|V| + |E|)$ required for storing the

social network graph. On the other hand, the IRIE, PMC, IMM and D-SSA methods have significantly higher space complexities, e.g., those of IMM and D-SSA are both $O\big((k \ln |V| + \ln(1/\delta)) \cdot (|V| + |E|) \cdot \varepsilon^{-2}\big)$ (Tang et al. 2015; Nguyen et al. 2016b). Thus, they fail to produce results on very large datasets. Our hop-based algorithms never face the out-of-memory problems as long as the memory is large enough to store the social network graph.

5.3 Results under the LT model

In the second set of experiments, we evaluate the algorithms under the LT model. Recall that the LT model requires the propagation probabilities to satisfy $\sum_{u \in I_v} p_{u,v} \leq 1$. The TRI and UNI settings may violate this requirement. Thus, we only consider the WC setting for the LT model, i.e., LT-WC. Table 6 and Fig. 10 show the influence spread and running time by different algorithms on various graphs. Similar to the case of the IC model, the results show that (1) our hop-based algorithms produce remarkably higher influence spread than the HighDegree and DegreeDiscount algorithms, and are on par with IMM and D-SSA, (2) our hop-based methods run faster than SIMPATH, IMM and D-SSA by several orders of magnitude, and (3) our upper bounding approach significantly improves the efficiency of the TwoHop method. These results demonstrate the robustness of our hop-based approaches with respect to different information diffusion models.

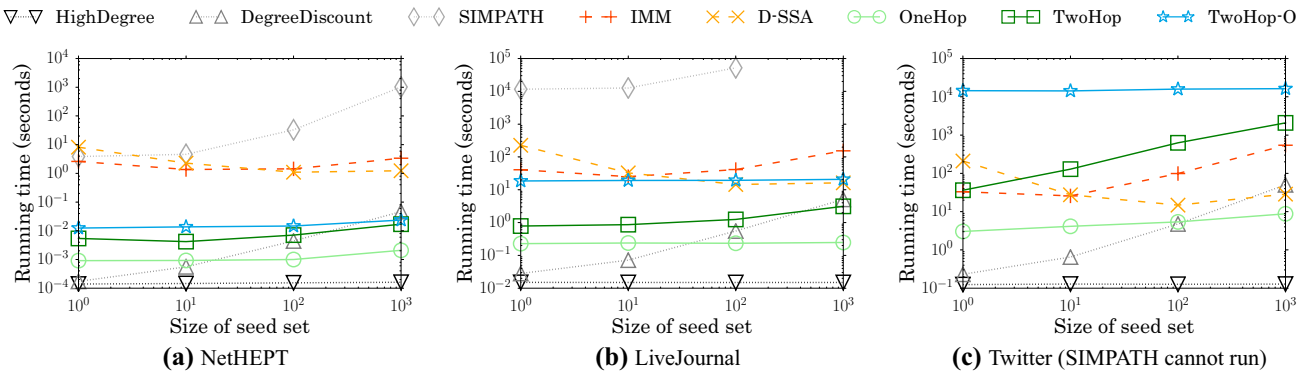
5.4 Data-dependent approximation guarantee

Recall that the approximate guarantee of our hop-based methods depends on the ratio $\sigma_h(S^*)/\sigma(S^*)$, where S^* is the optimal seed set that maximizes the influence spread (without any hop limit). The closer to one the ratio is, the better the approximation guarantee is. However, due to the NP-hardness of the problem, it is difficult to obtain the exact S^*. We estimate the ratio $\sigma_h(S^*)/\sigma(S^*)$ via two series of S (Lin et al. 2017). In the first case, we use the solution S_h returned by our hop-based algorithms as $\sigma(S_h)$ is close to $\sigma(S^*)$, and thus, S_h shall be also close to S^*. In the second case, we randomly generate 1000 sets S_k of k nodes and take the average ratio of $\sigma_h(S_k)/\sigma(S_k)$ to represent $\sigma_h(S^*)/\sigma(S^*)$. Figures 11, 12, 13 and 14 show the

Table 6 Influence spread on various graphs under the LT-WC setting

Method	NetHEPT				LiveJournal				Twitter			
	$k = 1$	10	100	1000	$k = 1$	10	100	1000	$k = 1$	10	100	1000
HighDegree	52.77	381.92	1.59e3	5.16e3	8.52e3	6.77e4	2.56e5	6.06e5	1.55e6	8.93e6	1.97e7	2.72e7
DegreeDiscount	52.77	393.42	1.50e3	5.09e3	8.52e3	6.77e4	2.56e5	6.03e5	1.55e6	8.93e6	1.97e7	2.72e7
SIMPATH	42.90	391.12	2.03e3	7.15e3	7.99e3	5.35e4	2.46e5	–	–	–	–	–
IMM	51.84	398.63	1.99e3	7.16e3	1.10e4	7.07e4	2.88e5	7.36e5	1.83e6	8.95e6	2.04e7	2.79e7
D-SSA	52.78	398.72	2.01e3	7.10e3	1.11e4	7.13e4	2.78e5	7.23e5	1.78e6	8.99e6	2.04e7	2.79e7
OneHop	40.28	371.03	1.82e3	6.91e3	6.93e3	4.96e4	2.68e5	6.31e5	1.22e6	8.77e6	2.00e7	2.77e7
TwoHop	41.25	379.48	2.01e3	7.12e3	6.93e3	6.12e4	2.63e5	6.87e5	1.55e6	8.91e6	2.01e7	2.78e7

The field with "–" means that the method cannot run under the setting



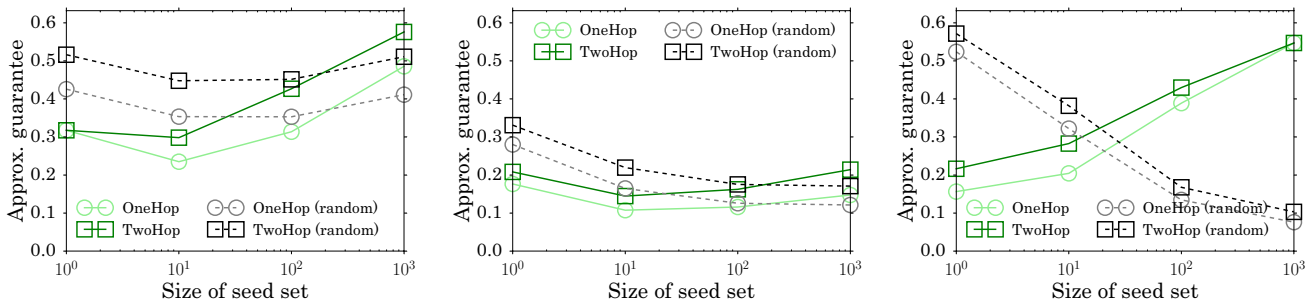
(a) NetHEPT **(b)** LiveJournal **(c)** Twitter (SIMPATH cannot run)

Fig. 10 Running time on various graphs under the LT-WC setting

data-dependent approximation guarantees of our OneHop and TwoHop methods under the IC-WC, IC-TRI, IC-UNI and LT-WC settings. As can be seen, in general, our hop-based methods can provide good approximation guarantees for all the cases tested. Moreover, sometimes the approximation guarantees of our hop-based methods are even higher than 0.55 (Figs. 11a, c, 12a and 13a), which is tighter than $(1 - 1/e - \varepsilon)$ provided by IMM and D-SSA when ε is set to 0.1. We also observe that our methods provide low approximation guarantees under high propagation probabilities and small seed size (e.g., IC-UNI with 1 seed). The reason is that under such scenarios, the information can propagate through long paths. As a result, hop-limited influence spread just takes a low fraction of the entire influence spread.

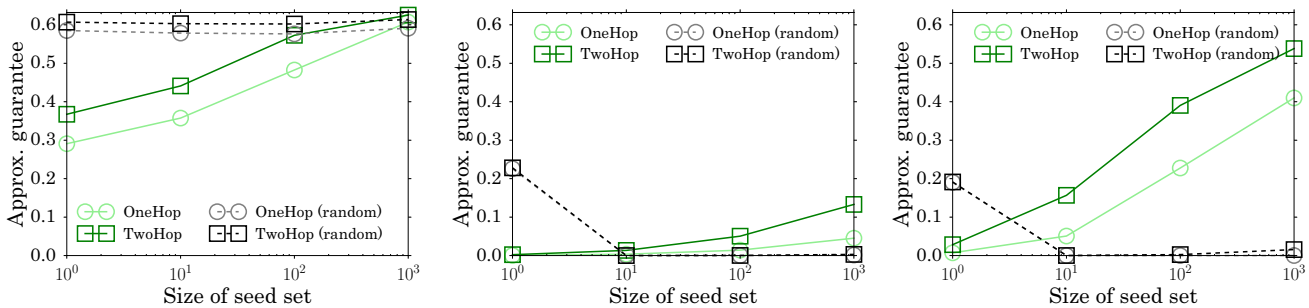
6 Related work

Since the $(1 - 1/e - \epsilon)$-approximation greedy algorithm was proposed by Kempe et al. (2003) for influence maximization, there has been considerable research on improving the efficiency of the greedy algorithm by using heuristics to trade the accuracy of influence estimation for computational efficiency (Chen et al. 2009, 2010a, b; Goyal et al. 2011a, c; Jung et al. 20112; Cheng et al. 2013, 2014; Cohen



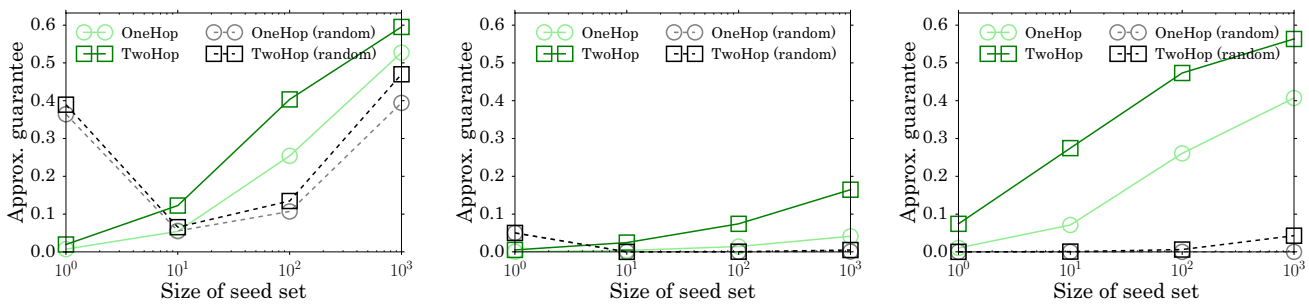
(a) NetHEPT **(b)** LiveJournal **(c)** Twitter

Fig. 11 Data-dependent approximation guarantee on various graphs under the IC-WC setting



(a) NetHEPT **(b)** LiveJournal **(c)** Twitter

Fig. 12 Data-dependent approximation guarantee on various graphs under the IC-TRI setting



(a) NetHEPT **(b)** LiveJournal **(c)** Twitter

Fig. 13 Data-dependent approximation guarantee on various graphs under the IC-UNI setting

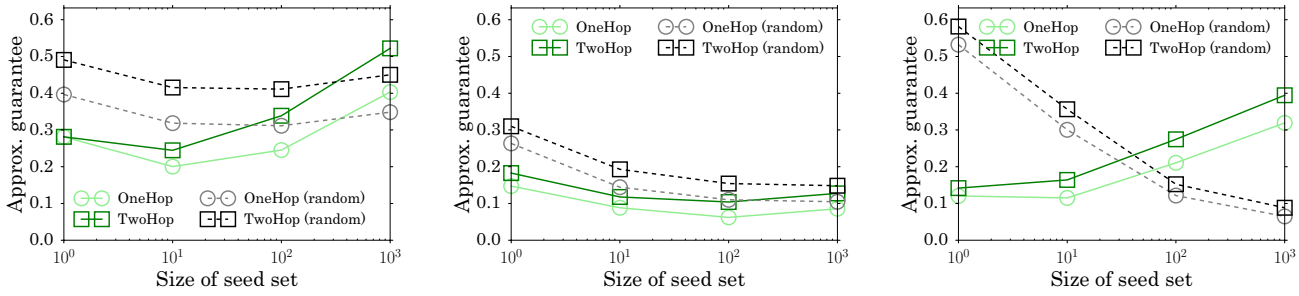


Fig. 14 Data-dependent approximation guarantee on various graphs under the LT-WC setting

et al. 2014; Lee and Chung 2014; Song et al. 2015; Galhotra et al. 2016), or optimizing the Monte Carlo simulations for influence estimation (Leskovec et al. 2007b; Goyal et al. 2011b; Zhou et al. 2013, 2014; Borgs et al. 2014; Ohsaka et al. 2014; Tang et al. 2014, 2015; Nguyen et al. 2016a, b; Tang et al. 2018a). Among them, the DegreeDiscount heuristic (Chen et al. 2009) roughly estimated influence spread within one-hop neighborhood. (Chen et al. 2010a, b) further proposed PMIA and LDAG heuristics that use independent propagation paths to construct arborescences for rough influence estimation under the IC and LT models. PMIA and LDAG were further improved by the follow-up IRIE (Jung et al. 20112) and SIMPATH (Goyal et al. 2011c) algorithms, respectively. DegreeDiscount, IRIE and SIMPATH as well as advanced sampling-based methods are all included in our experimental comparison. Leveraging the independency among propagation paths, we have developed efficient hop-based methods to compute the *exact* influence spread within a certain number of hops. In addition, compared with the state-of-the-art heuristics like IRIE and SIMPATH, and the advanced sampling-based methods like PMA, IMM and D-SSA, our hop-based methods are the most memory efficient technique. We notice that the memory consumption of EaSyIM (Galhotra et al. 2016) is also low. However, their benchmark work (Arora et al. 2017) shows that EaSyIM runs orders of magnitude slower than our hop-based methods (see Fig. 7 and Table 3 in Arora et al. 2017). For example, EaSyIM takes tens of seconds to find 100 seeds on the NetHEPT dataset under the IC-WC, IC-UNI and LT-WC settings (see Fig. 7 in Arora et al. 2017), while our hop-based methods use less than 0.1 s. Moreover, EaSyIM cannot run on the Twitter dataset under the IC-WC, IC-UNI and LT-WC settings (see Table 3 in Arora et al. 2017).

Our hop-based influence estimation is in spirit similar to the time-constrained Independent Cascade model studied in (Chen et al. 2012; Liu et al. 2012; Dinh et al. 2014) by concentrating on the diffusion within a fixed number of hops. We make new technological advances by inventing algorithms to compute the exact influence spreads of one-hop and two-hop propagations, which could only be approximately estimated in previous work where the approximation guarantee is difficult to analyze (Lee and Chung 2014). Our algorithms enable very efficient evaluation of the change in influence spread when a new seed node is added. We also derive an upper bound on the influence spread to further speed up our hop-based algorithms. Our hop-based approaches can be easily applied to many influence-based applications, such as topic-aware influence maximization (Zhang et al. 2013), profit maximization combining the benefit of influence spread with the cost of seed selection or information propagation (Tang et al. 2016, 2017b, 2018b), community detection via influence maximization (Jiang et al. 2014), influence maximization under a variant of the IC model (Xu et al. 2014) and seed minimization with a given amount of influence spread to achieve (Goyal et al. 2013).

7 Conclusion

In this paper, we have proposed lightweight hop-based methods to address the problem of influence maximization in social networks. We have also developed an upper bounding technique to further speed up the seed selection algorithm. Through analysis, we show that our methods can provide certain theoretical guarantees. Experiments are conducted with real social network datasets to compare the efficiency and effectiveness of our algorithms with state-of-the-art ones. In terms of solution quality, our hop-based methods are on par with the most advanced IMM and D-SSA methods which can provide the best $(1 - 1/e - \epsilon)$-approximation guarantee and remarkably outperform the HighDegree and DegreeDiscount heuristics for quite some cases. In terms of efficiency, our hop-based methods run much faster than the IRIE, SIMPATH, PMC, IMM and D-SSA methods for most cases tested. Furthermore, while all these existing methods fail to run on some test cases, our hop-based methods can always execute and find solutions. To summarize,

HighDegree and DegreeDiscount are not effective in influence maximization. IRIE, SIMPATH and PMC cannot scale to large networks like Twitter. IMM and D-SSA are unable to handle networks with high propagation probabilities, e.g., IC-TRI and IC-UNI. Our hop-based methods are designed to tackle these issues.

Acknowledgements This research is supported by the National Research Foundation, Prime Minister's Office, Singapore, under its IDM Futures Funding Initiative, and by Singapore Ministry of Education Academic Research Fund Tier 1 under Grant 2017-T1-002-024 and Tier 2 under Grant MOE2015-T2-2-114.

Appendix

Proof of Theorem 1 To consider the outgoing edges from u one at a time, we first disable all the edges from u to its neighbors except for one edge $\langle u, w_1\rangle$. Then, for each neighbor v of w_1, all of v's inverse neighbors other than w_1 have their one-hop activation probabilities unchanged by adding $\langle u, w_1\rangle$. Let $\pi_2^{S\cup\{u\}}(v|w_1)$ denote the new two-hop activation probability of v. Then, we have

$$\frac{1-\pi_2^{S\cup\{u\}}(v|w_1)}{1-\pi_2^{S}(v)} = \rho(S,u,v,w_1), \tag{16}$$

where $\rho(S,u,v,w) = \frac{1-p_{w,v}\cdot\pi_1^{S\cup\{u\}}(w)}{1-p_{w,v}\cdot\pi_1^{S}(w)}$. Next, we enable the second edge $\langle u, w_2\rangle$. Let $\pi_2^{S\cup\{u\}}(v|w_1,w_2)$ denote the new two-hop activation probability of v. Following similar arguments, for each neighbor v of w_2, we have

$$\frac{1-\pi_2^{S\cup\{u\}}(v|w_1,w_2)}{1-\pi_2^{S\cup\{u\}}(v|w_1)} = \rho(S,u,v,w_2). \tag{17}$$

We continue to enable the outgoing edges of u sequentially. In general, when an edge $\langle u, w_i\rangle$ is enabled after edges $\langle u, w_1\rangle, \langle u, w_2\rangle, \ldots, \langle u, w_{i-1}\rangle$, for each neighbor v of w_i, we have

$$\frac{1-\pi_2^{S\cup\{u\}}(v|w_1,\ldots,w_i)}{1-\pi_2^{S\cup\{u\}}(v|w_1,\ldots,w_{i-1})} = \rho(S,u,v,w_i). \tag{18}$$

Therefore, we can initialize $\pi_2^{S\cup\{u\}}(v)$ with $\pi_2^{S}(v)$ and iteratively update $\pi_2^{S\cup\{u\}}(v)$ with

$$1-\left(1-\pi_2^{S\cup\{u\}}(v)\right)\cdot\rho(S,u,v,w), \tag{19}$$

for all the nodes $w \in N_u \setminus S$ and $v \in N_w \setminus S$. Moreover, for the direct neighbors of u, their two-hop activation probabilities also need to be adjusted because u's one-hop activation probability has changed from $\pi_1^S(u)$ to 1. For each neighbor v of u, the adjustment can be made in a similar way by updating $\pi_2^{S\cup\{u\}}(v)$ with

$$1-\left(1-\pi_2^{S\cup\{u\}}(v)\right)\cdot\rho(S,u,v,u). \tag{20}$$

Then, the final two-hop activation probability $\pi_2^{S\cup\{u\}}(v)$ by the iterative updates (19) and (20) is

$$\pi_2^{S\cup\{u\}}(v) = 1-\left(1-\pi_2^{S}(v)\right)\cdot\prod_{w\in(M_{u,v}\cup\{u\})}\rho(S,u,v,w). \tag{21}$$

Hence, the theorem is proven. □

Proof of Theorem 2 Consider a single seed $\{u\}$. Let $A_u \subseteq N_u$ denote a subset of a node u's neighbors. Let $p(A_u)$ denote the probability that all the nodes in A_u are activated directly by u under the IC and LT models, while all the nodes in $N_u \setminus A_u$ are not directly activated by u (they may not even be activated eventually). Since each of u's neighbors is activated by u independently, we have

$$p(A_u) = \left(\prod_{v\in A_u} p_{u,v}\right)\cdot\left(\prod_{v\in N_u\setminus A_u}(1-p_{u,v})\right). \tag{22}$$

Furthermore, with h hops of propagation, for each node $w \in V \setminus \{u\}$, w can only be activated by a propagation path starting from a node $v \in A_u$ whose path length is no longer than $h-1$ hops. In other words, the probability for w to be activated by A_u is $\pi_{h-1}^{A_u}(w)$. Considering all the possible node sets A_u activated directly by u, we have

$$
\begin{aligned}
&\sigma_h(\{u\}) \\
&= 1 + \sum_{A_u \subseteq N_u} \left(p(A_u) \cdot \sum_{w \in V \setminus \{u\}} \pi_{h-1}^{A_u}(w) \right) \\
&\leq 1 + \sum_{A_u \subseteq N_u} \left(p(A_u) \cdot \sum_{w \in V} \pi_{h-1}^{A_u}(w) \right) \\
&= 1 + \sum_{A_u \subseteq N_u} \left(p(A_u) \cdot \sigma_{h-1}(A_u) \right) \\
&\leq 1 + \sum_{A_u \subseteq N_u} \left(p(A_u) \cdot \sum_{v \in A_u} \sigma_{h-1}(\{v\}) \right) \\
&= 1 + \sum_{A_u \subseteq N_u} \left(p(A_u) \cdot \sum_{v \in N_u} \left(\sigma_{h-1}(\{v\}) \cdot p(v \in A_u) \right) \right) \\
&= 1 + \sum_{A_u \subseteq N_u} \left(\sum_{v \in N_u} \left(p(A_u) \cdot \sigma_{h-1}(\{v\}) \cdot p(v \in A_u) \right) \right) \\
&= 1 + \sum_{v \in N_u} \left(\sum_{A_u \subseteq N_u} \left(p(A_u) \cdot \sigma_{h-1}(\{v\}) \cdot p(v \in A_u) \right) \right) \\
&= 1 + \sum_{v \in N_u} \left(\sigma_{h-1}(\{v\}) \cdot \sum_{A_u \subseteq N_u} \left(p(A_u) \cdot p(v \in A_u) \right) \right).
\end{aligned}
\tag{23}
$$

The second "$\leq$" is due to the submodularity of $\sigma_h(\cdot)$ (see Theorem 3) such that $\sigma_{h-1}(A_u) \leq \sum_{v \in A_u} \sigma_{h-1}(\{v\})$. In the third "=", $p(v \in A_u)$ is such a binary value that $p(v \in A_u) = 1$ if and only if $v \in A_u$. Meanwhile, we have

$$
\begin{aligned}
&\sum_{A_u \subseteq N_u} \left(p(A_u) \cdot p(v \in A_u) \right) \\
&= \sum_{A_u \subseteq N_u \setminus \{v\}} \left(p(A_u) \cdot p(v \in A_u) \right) \\
&\quad + \sum_{A_u \subseteq N_u \setminus \{v\}} \left(p(A_u \cup \{v\}) \cdot p(v \in A_u \cup \{v\}) \right) \\
&= \sum_{A_u \subseteq N_u \setminus \{v\}} p(A_u \cup \{v\}).
\end{aligned}
\tag{24}
$$

The last "=" follows the fact that $p(v \in A_u) = 0$ since $v \notin A_u \subseteq N_u \setminus \{v\}$ and $p(v \in A_u \cup \{v\}) = 1$ since $v \in A_u \cup \{v\}$. Therefore, from (23) and (24), we have

$$
\sigma_h(\{u\}) \leq 1 + \sum_{v \in N_u} \left(\sigma_{h-1}(\{v\}) \cdot \sum_{A_u \subseteq N_u \setminus \{v\}} p(A_u \cup \{v\}) \right).
\tag{25}
$$

Furthermore, by definition,

$$
\begin{aligned}
&\sum_{A_u \subseteq N_u \setminus \{v\}} p(A_u \cup \{v\}) \\
&= \sum_{A_u \subseteq N_u \setminus \{v\}} \left(\left(\prod_{w \in A_u \cup \{v\}} p_{u,w} \right) \cdot \left(\prod_{w \in N_u \setminus (A_u \cup \{v\})} (1 - p_{u,w}) \right) \right) \\
&= \sum_{A_u \subseteq N_u \setminus \{v\}} \left(p_{u,v} \cdot \left(\prod_{w \in A_u} p_{u,w} \right) \cdot \left(\prod_{w \in N_u \setminus (A_u \cup \{v\})} (1 - p_{u,w}) \right) \right) \\
&= p_{u,v} \cdot \sum_{A_u \subseteq N_u \setminus \{v\}} \left(\left(\prod_{w \in A_u} p_{u,w} \right) \cdot \left(\prod_{w \in N_u \setminus (A_u \cup \{v\})} (1 - p_{u,w}) \right) \right) \\
&= p_{u,v} \cdot 1 \\
&= p_{u,v}.
\end{aligned}
\tag{26}
$$

Thus, by (25) and (26), it holds that

$$
\sigma_h(\{u\}) \leq 1 + \sum_{v \in N_u} \left(\sigma_{h-1}(\{v\}) \cdot p_{u,v} \right).
\tag{27}
$$

Inequality (11) can be proved by induction. When $h = 1$, the inequality follows directly from Inequality (10). Suppose that it holds for $h-1$ hops of propagation, i.e., $\sigma_{h-1}(\{u\}) \leq \hat{\sigma}_{h-1}(\{u\})$. Then, for h hops of propagation, we have

$$
\begin{aligned}
\sigma_h(\{u\}) &\leq 1 + \sum_{v \in N_u} \left(p_{u,v} \cdot \sigma_{h-1}(\{v\}) \right) \\
&\leq 1 + \sum_{v \in N_u} \left(p_{u,v} \cdot \hat{\sigma}_{h-1}(\{v\}) \right) \\
&= \hat{\sigma}_h(\{u\}).
\end{aligned}
\tag{28}
$$

Therefore, for any $h \geq 0$, we have $\sigma_h(\{u\}) \leq \hat{\sigma}_h(\{u\})$. □

Proof of Theorem 3 This can be proved using the *live edge* approach (Kempe et al. 2003).

- Under the IC model, for each edge $\langle u, v \rangle \in E$, we independently flip a coin of bias $p_{u,v}$ to decide whether the edge $\langle u, v \rangle$ is *live* or *blocked* to generate a sample influence propagation outcome X.
- Under the LT model, for each node $v \in V$, it picks at most one of its incoming edge at random—selecting the edge from an inverse neighbor u with probability $p_{u,v}$ and not selecting any incoming edge with probability $1 - \sum_{u \in I_v} p_{u,v}$.

We use $p(X)$ to denote the probability of a specific outcome X in the sample space. Let $V_h^X(v)$ denote the node set

that can be reached from a node v within h hops in the sample outcome X. Then, the number of nodes that can be reached from a seed set S within h hops in the outcome X is given by $\sigma_h^X(S) = \left|\bigcup_{v\in S} V_h^X(v)\right|$. Thus,

$$\sigma_h(S) = \sum_X \left(p(X)\cdot \sigma_h^X(S)\right), \tag{29}$$

where the monotonicity of $\sigma_h(S)$ holds since $\sigma_h^X(S)$ increases as S expands.

The marginal influence gain

$$\sigma_h^X(S\cup\{u\}) - \sigma_h^X(S) = \left|V_h^X(u) \setminus \bigcup_{v\in S} V_h^X(v)\right| \tag{30}$$

is the number of nodes that are reachable from a node u within h hops but are not reachable from any node in a seed set S within h hops in a sample outcome X. For any two node sets S and T where $S \subseteq T$, we have $\bigcup_{v\in S} V_h^X(v) \subseteq \bigcup_{v\in T} V_h^X(v)$. Thus, $V_h^X(u) \setminus \bigcup_{v\in S} V_h^X(v) \supseteq V_h^X(u) \setminus \bigcup_{v\in T} V_h^X(v)$, which implies that

$$\sigma_h^X(S\cup\{u\}) - \sigma_h^X(S) \geq \sigma_h^X(T\cup\{u\}) - \sigma_h^X(T). \tag{31}$$

Since $p(X) \geq 0$ for any X, taking the linear combination, we have

$$\sigma_h(S\cup\{u\}) - \sigma_h(S) \geq \sigma_h(T\cup\{u\}) - \sigma_h(T). \tag{32}$$

Thus, $\sigma_h(\cdot)$ is submodular. □

Proof of Theorem 4 Let S_h^* denote the optimal seed set for maximizing the influence spread within h hops of propagation, i.e., $\sigma_h(S_h^*) = \max_{|S|=k} \sigma_h(S)$. We have

$$\begin{aligned}
\sigma(S_h) &\geq \sigma_h(S_h)\\
&\geq \left(\frac{1}{\kappa_{\sigma_h}}(1-e^{-\kappa_{\sigma_h}})\right)\sigma_h(S_h^*)\\
&\geq \left(\frac{1}{\kappa_{\sigma_h}}(1-e^{-\kappa_{\sigma_h}})\right)\sigma_h(S^*)\\
&= \left(\frac{1}{\kappa_{\sigma_h}}(1-e^{-\kappa_{\sigma_h}})\alpha\right)\sigma(S^*)
\end{aligned} \tag{33}$$

The first inequality follows from the fact that the exact influence spread is equal to the influence spread without any hop limitation of propagation. The second inequality is because that the greedy algorithm can achieve $\left(\frac{1}{\kappa_f}(1-e^{-\kappa_f})\right)$-approximation for maximizing a monotone submodular function f with a cardinality constraint (Conforti and Cornuéjols 1984), where the submodularity and monotonicity of $\sigma_h(\cdot)$ is given by Theorem 3. The third inequality is because S_h^* is the optimal solution for maximizing $\sigma_h(\cdot)$. □

We first introduce some lemmas used to prove Theorem 5.

Lemma 1 *For scale-free random graphs with propagation probability $p_{u,v} = p$ for every edge $\langle u, v\rangle \in E$, the expected influence spread produced within one hop of propagation from a random seed set S satisfies*

$$\mathbb{E}[\sigma_1(S)] \geq (p+1)k - pk^2/|V|. \tag{34}$$

Proof of Lemma 1 With one hop of propagation, for a randomly selected node v, it is not activated if and only if v is not a seed and v is not activated by any of its inverse neighbors. The probability for v to be a non-seed node is $1-\frac{k}{|V|}$. The probability for an inverse neighbor of v to be a seed is $\frac{k}{|V|}$, and thus, the probability for it to activate v is $p\cdot\frac{k}{|V|}$. Therefore, the probability for all of v's inverse neighbors to fail to activate v is

$$\prod_{u\in I_v}\left(1-p\cdot\frac{k}{|V|}\right) = \left(1-\frac{pk}{|V|}\right)^{|I_v|}. \tag{35}$$

Note that if v is selected as a seed, it must be activated. Hence, the overall activation probability of v is

$$\pi_1^S(v) = 1-\left(1-\frac{k}{|V|}\right)\cdot\left(1-\frac{pk}{|V|}\right)^{|I_v|}. \tag{36}$$

As a result, the expectation of the activation probability of a random node v is given by

$$\begin{aligned}
\mathbb{E}[\pi_1^S(v)] &= \mathbb{E}\left[1-\left(1-\frac{k}{|V|}\right)\cdot\left(1-\frac{pk}{|V|}\right)^{|I_v|}\right]\\
&= 1-\left(1-\frac{k}{|V|}\right)\cdot\sum_{|I_v|=1}^{\infty}\left(P_0(|I_v|)\cdot\left(1-\frac{pk}{|V|}\right)^{|I_v|}\right)\\
&\geq 1-\left(1-\frac{k}{|V|}\right)\cdot\left(1-\frac{pk}{|V|}\right)\cdot\sum_{|I_v|=1}^{\infty}P_0(|I_v|)\\
&= 1-\left(1-\frac{k}{|V|}\right)\cdot\left(1-\frac{pk}{|V|}\right)\\
&= \frac{(1+p)k}{|V|} - \frac{pk^2}{|V|^2}.
\end{aligned} \tag{37}$$

Therefore, it holds that $\mathbb{E}[\sigma_1(S)] = |V| \cdot \mathbb{E}[\pi_1^S(v)] \geq (p+1)k - pk^2/|V|$. This completes the proof. □

Lemma 2 (Li et al. 2012) *For an infinite random power law graph, the expected fraction of nodes activated* $\phi(S) = \mathbb{E}[\sigma(S)]/|V|$ *can be computed by*

$$\begin{cases} 1-\varphi(S) = \left(1-\frac{k}{|V|}\right)\sum_{d=0}^{\infty} P_1(d+1)(1-p\varphi(S))^d, \\ 1-\phi(S) = \left(1-\frac{k}{|V|}\right)\sum_{d=1}^{\infty} P_0(d)(1-p\varphi(S))^d, \end{cases} \tag{38}$$

where $P_1(d) = \frac{d^{1-\gamma}}{\sum_{d=1}^{\infty} d^{1-\gamma}}$ *is the probability of a node connecting to a neighbor whose degree is d, and* $\varphi(S)$ *is an instrumental variable.*

Lemma 3 *The expected fraction of nodes activated* $\phi(S)$ *is bounded by*

$$\mathbb{E}[\sigma(S)] \leq |V| \cdot \left(1-\left(1-\frac{k}{|V|}\right)P_0(1)(1-pA)\right), \tag{39}$$

where $A = 1-\left(1-\frac{k}{|V|}\right)P_1(1)$.

Proof of Lemma 3 From (38) in Lemma 2, we have

$$1-\varphi(S) \geq \left(1-\frac{k}{|V|}\right)P_1(1)(1-p\varphi(S))^0 = 1-A, \tag{40}$$

and

$$1-\phi(S) \geq \left(1-\frac{k}{|V|}\right)P_0(1)(1-p\varphi(S)). \tag{41}$$

Hence, by (40) and (41), the lemma follows. □

Proof of Theorem 5 Lemma 1 indicates that

$$\mathbb{E}[\sigma_h(S)] \geq \mathbb{E}[\sigma_1(S)] \geq (p+1)k - pk^2/|V|. \tag{42}$$

Lemma 3 indicates that

$$E[\sigma(S)] \leq |V| \cdot \left(1-\left(1-\frac{k}{|V|}\right)P_0(1)(1-pA)\right). \tag{43}$$

Putting (42) and (43) together, the theorem follows. □

References

Arora A, Galhotra S, Ranu S (2017) Debunking the myths of influence maximization: an in-depth benchmarking study. In: Proceedings of ACM SIGMOD, pp 651–666

Barabási AL, Albert R (1999) Emergence of scaling in random networks. Science 286(5439):509–512

Borgs C, Brautbar M, Chayes J, Lucier B (2014) Maximizing social influence in nearly optimal time. In: Proceedings of SODA, pp 946–957

Cha M, Mislove A, Gummadi KP (2009) A measurement-driven analysis of information propagation in the Flickr social network. In: Proceedings WWW, pp 721–730

Chen W (2009) NetHEPT dataset. http://research.microsoft.com/en-us/people/weic/

Cheng S, Shen H, Huang J, Chen W, Cheng X (2014) IMRank: influence maximization via finding self-consistent ranking. In: Proceedings ACM SIGIR, pp 475–484

Cheng S, Shen H, Huang J, Zhang G, Cheng X (2013) Staticgreedy: solving the scalability-accuracy dilemma in influence maximization. In: Proceedings ACM CIKM, pp 509–518

Chen W, Lu W, Zhang N (2012) Time-critical influence maximization in social networks with time-delayed diffusion process. In: Proceedings of AAAI, pp 592–598

Chen W, Wang C, Wang Y (2010a) Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of ACM KDD, pp 1029–1038

Chen W, Wang Y, Yang S (2009) Efficient influence maximization in social networks. In: Proceedings of ACM KDD, pp 199–208

Chen W, Yuan Y, Zhang L (2010b) Scalable influence maximization in social networks under the linear threshold model. In: Proceedings of IEEE ICDM, pp. 88–97

Cohen E, Delling D, Pajor T, Werneck RF (2014) Sketch-based influence maximization and computation: scaling up with guarantees. In: Proceedings ACM CIKM, pp 629–638

Conforti M, Cornuéjols G (1984) Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the rado-edmonds theorem. Discrete Appl Math 7(3):251–274

Dinh TN, Zhang H, Nguyen DT, Thai MT (2014) Cost-effective viral marketing for time-critical campaigns in large-scale social networks. IEEE ACM Trans Netw 22(6):2001–2011

Domingos P, Richardson M (2001) Mining the network value of customers. In: Proceedings ACM KDD, pp 57–66

Galhotra S, Arora A, Roy S (2016) Holistic influence maximization: Combining scalability and efficiency with opinion-aware models. In: Proceedings ACM SIGMOD, pp 743–758

Goel S, Watts DJ, Goldstein DG (2012) The structure of online diffusion networks. In: Proceedings ACM EC, pp 623–638

Goyal A, Bonchi F, Lakshmanan LVS (2011a) A data-based approach to social influence maximization. Proc VLDB Endow 5(1):73–84

Goyal A, Bonchi F, Lakshmanan L, Venkatasubramanian S (2013) On minimizing budget and time in influence propagation over social networks. Social Netw Anal Min 3(2):179–192

Goyal A, Lu W, Lakshmanan LV (2011b) Celf++: Optimizing the greedy algorithm for influence maximization in social networks. In: Proceedings WWW Companion, pp 47–48

Goyal A, Lu W, Lakshmanan LVS (2011c) Simpath: An efficient algorithm for influence maximization under the linear threshold model. In: Proceedings IEEE ICDM, pp 211–220

Jiang F, Jin S, Wu Y, Xu J (2014) A uniform framework for community detection via influence maximization in social networks. In: Proceedings IEEE/ACM ASONAM, pp 27–32

Jung K, Heo W, Chen W (2012) IRIE: scalable and robust influence maximization in social networks. In: Proceedings IEEE ICDM, pp 918–923

Kempe D, Kleinberg J, Tardos E (2003) Maximizing the spread of influence through a social network. In: Proceedings ACM KDD, pp 137–146

Kwak H, Lee C, Park H, Moon S (2010) What is Twitter, a social network or a news media? In: Proceedings of WWW, pp 591–600

Lee JR, Chung CW (2014) A fast approximation for influence maximization in large social networks. In: WWW Companion, pp 1157–1162

Leskovec J, Adamic LA, Huberman BA (2007a) The dynamics of viral marketing. ACM Trans Web 1(1):5:1–5:39

Leskovec J, Krause A, Guestrin C, Faloutsos C, VanBriesen J, Glance N (2007b) Cost-effective outbreak detection in networks. In: Proceedings of ACM KDD, pp 420–429

Leskovec J, Krevl A (2014) SNAP datasets: stanford large network dataset collection. http://snap.stanford.edu/data

Li Y, Zhao BQ, Lui JCS (2012) On modeling product advertisement in large-scale online social networks. IEEE ACM Trans Netw 20(5):1412–1425

Lin Y, Chen W, Lui JC (2017) Boosting information spread: an algorithmic approach. In: Proceedings of IEEE ICDE, pp 883–894

Liu B, Cong G, Xu D, Zeng Y (2012) Time constrained influence maximization in social networks. In: Proceedings of IEEE ICDM, pp 439–448

Lu W, Chen W, Lakshmanan LV (2015) From competition to complementarity: comparative influence diffusion and maximization. Proc VLDB Endow 9(2):60–71

Nemhauser GL, Wolsey LA, Fisher ML (1978) An analysis of approximations for maximizing submodular set functions-I. Math Program 14(1):265–294

Nguyen HT, Dinh TN, Thai MT (2016a) Cost-aware targeted viral marketing in billion-scale networks. In: Proceedings of IEEE INFOCOM

Nguyen HT, Thai MT, Dinh TN (2016b) Stop-and-stare: optimal sampling algorithms for viral marketing in billion-scale networks. In: Proceedings of ACM SIGMOD, pp 695–710

Ohsaka N, Akiba T, Yoshida Y, Kawarabayashi K (2014) Fast and accurate influence maximization on large networks with pruned Monte-Carlo simulations. In: Proceedings of AAAI, pp 138–144

Ohsaka N, Sonobe T, Fujita S, Kawarabayashi Ki (2017) Coarsening massive influence networks for scalable diffusion analysis. In: Proceedings of ACM SIGMOD, pp 635–650

Song G, Zhou X, Wang Y, Xie K (2015) Influence maximization on large-scale mobile social network: a divide-and-conquer method. IEEE Trans Parallel Distrib Syst 26(5):1379–1392

Tang Y, Shi Y, Xiao X (2015) Influence maximization in near-linear time: A martingale approach. In: Proceedings of ACM SIGMOD, pp 1539–1554

Tang J, Tang X, Xiao X, Yuan J (2018a) Online processing algorithms for influence maximization. In: Proceedings of ACM SIGMOD

Tang J, Tang X, Yuan J (2016) Profit maximization for viral marketing in online social networks. In: Proceedings of IEEE ICNP, pp 1–10

Tang J, Tang X, Yuan J (2017a) Influence maximization meets efficiency and effectiveness: a hop-based approach. In: Proceedings of IEEE/ACM ASONAM, pp 64–71

Tang J, Tang X, Yuan J (2017b) Profit maximization for viral marketing in online social networks: algorithms and analysis. IEEE Trans Knowl Data Eng (**Preprint**)

Tang J, Tang X, Yuan J (2018b) Towards profit maximization for online social network providers. In: Proceedings of IEEE INFOCOM

Tang Y, Xiao X, Shi Y (2014) Influence maximization: Near-optimal time complexity meets practical efficiency. In: Proceedings of ACM SIGMOD, pp 75–86

Wang Z, Yang Y, Pei J, Chu L, Chen E (2017) Activity maximization by effective information diffusion in social networks. IEEE Trans Knowl Data Eng 29(11):2374–2387

Xu W, Lu Z, Wu W, Chen Z (2014) A novel approach to online social influence maximization. Social Netw Anal Min 4(1):153

Zhang C, Sun J, Wang K (2013) Information propagation in microblog networks. In: Proceedings of IEEE/ACM ASONAM, pp 190–196

Zhou C, Zhang P, Guo J, Guo L (2014) An upper bound based greedy algorithm for mining top-k influential nodes in social networks. In: Proceedings of WWW Companion, pp 421–422

Zhou C, Zhang P, Guo J, Zhu X, Guo L (2013) UBLF: an upper bound based approach to discover influential nodes in social networks. In: Proceedings of IEEE ICDM, pp 907–916