Neurocomputing 338 (2019) 92–100

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Post and repost: A holistic view of budgeted influence maximization

Qihao Shi, Can Wang*, Jiawei Chen, Yan Feng, Chun Chen

ZJU-LianlianPay Joint Research Center College of Computer Science, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

Article history:
Received 22 May 2018
Revised 26 January 2019
Accepted 8 February 2019
Available online 12 February 2019

Communicated by Dr. Guan Ziyu

Keywords:
Influence maximization
Budget constraint
Seed&boost node

ABSTRACT

Existing studies on influence maximization (IM) mainly focus on activating a set of influential users (seed nodes). Originated from the seed nodes' promotion actions (e.g., posting an advertising tweet) on social networks, a large influence spread might be triggered. However, in practice it is usually very expensive to have influential users posting original tweets in a promotional event. In contrast, it will incur much lower costs to have influential users reposting tweets and have ordinary users posting original tweets. Inspired by these observations, in this paper, we consider the Holistic Budgeted Influence Maximization (HBIM) problem, which maximizes the influence spread by deploying the budget to select seed nodes (for posting) and boost nodes (for reposting). To tackle the NP-hardness and non-submodularity of the problem, we devise two efficient algorithms with the data-dependent approximation ratios. Extensive experiments on real social networks demonstrate the efficiency and effectiveness of our proposed algorithms.

1. Introduction

Social media marketing is drawing increasing attentions for industrial and research communities [1,2]. By selecting a group of influential users (seed nodes) to post specific tweets such as online comments, product reviews, etc., a large chain of product adoption might be triggered [3,4]. To make effective marketing strategies in social media, Influence Maximization has become a hot research topic [5,6]. Existing works mainly focus on selecting the optimal seed nodes to maximize the influence spread, with an underlying assumption that costs for involving different users are equal. In fact, this assumption seldom holds and it is usually more expensive to involve influential users in a promotion event than ordinary users. This difference in costs motivated the research on Budgeted Influence Maximization problem [7]. However, a random cost is used for each node in [7], disregarding the fact that selecting influential users as seed nodes will usually incur expensive cost in social media marketing.

More recently, Lin et al. [8] propose the influence boost model in which a set of nodes are "boosted" so that they are more susceptible to their friends' influence. By selecting appropriate boost nodes, the influence spread of a given set of initial seed nodes will be increased. In practice, such pattern does exist, e.g., reposting by influential users may boost the spread of a specific tweet.

However, the work of [8] only consider selecting boost nodes to increase the influence spread for a given set of seed nodes, with the equal cost assumption. Actually, the influence boost model can provide a more flexible mechanism for budget allocation with different cost, providing the fact that persuading a user for reposting a tweet usually incurs much lower cost than for posting an original one. Consequently, a better budget allocation can be achieved for influence maximization by involving both seed nodes and boost nodes in selection.

In this paper, we propose a new framework for influence maximization, named as Holistic Budgeted Influence Maximization (HBIM), to explicitly involve both seed and boost nodes in selection. Given the cost of seed/boost nodes, HBIM maximizes the expected influence spread in a social network with the optimal deployment of seed nodes (to post) and boost nodes (to repost) under the budget constraint. By involving both seed nodes and boost nodes in influence spread, HBIM offers more flexibility in budget-based influence maximization. As this is the case for most commercial promotions in social media, we expect our work to have good applicability in real world scenarios.

Nevertheless, the HBIM is NP-hard and computing the expected influence spread for a given budget deployment is #P-hard. Meanwhile, the influence spread in HBIM problem is not submodular, meaning that the greedy algorithm cannot ensure any performance guarantees. To address these problems, we develop two efficient algorithms IMD and IMD-LB for HBIM with data-dependent

* Corresponding author.
E-mail addresses: shiqihao321@zju.edu.cn (Q. Shi), wcan@zju.edu.cn (C. Wang), sleepyhunt@zju.edu.cn (J. Chen), fengyan@zju.edu.cn (Y. Feng), chenc@zju.edu.cn (C. Chen).

https://doi.org/10.1016/j.neucom.2019.02.010
0925-2312/

Table 1
Meanings of frequent used symbols.

Symbols	Meanings
n	Number of nodes in the network
m	Number of edges in the network
k	Total budget
S	Seed node set
B	Boost node set
$c_s(u)$	Cost of selecting node u as seed node
$c_b(u)$	Cost of selecting node u as boost node
$\langle S, B\rangle$	A deployment with seed set S and boost set B
$\sigma(\langle S, B\rangle)$	The expected influence spread of deployment $\langle S, B\rangle$

approximation ratios. Extensive experiments are conducted using real social networks. The experimental results show the efficiency and effectiveness of our proposed algorithms, and demonstrate the superiority of proposed algorithms over compared algorithms. It is worthwhile to summarize our major contributions as follows.

1. We propose a new framework of Holistic Budgeted Influence Maximization (HBIM), which explicitly involves both seed and boost nodes selection. This framework may offers more flexibility in real world scenarios.
2. We prove the HBIM is NP-hard and computing the expected influence spread for a given budget deployment is #P-hard. Meanwhile, the influence spread in HBIM problem is not submodular, meaning that the greedy algorithm cannot ensure any performance guarantees.
3. We develop two efficient algorithms IMD and IMD-LB for HBIM with provable data-dependent approximation ratios.
4. We conduct extensive experiments and the experimental results show the efficiency and effectiveness of our proposed algorithms, and demonstrate the superiority of proposed algorithms over compared algorithms.

The rest of this paper is organized as follows. In Section 2, we discuss the related works of this paper. After that, we formally define the HBIM problem and discuss its properties in Section 3. In Section 4, we develop two efficient algorithms for solving HBIM with data-dependent approximation ratios. Extensive experiments using real social networks are shown in Section 5. Conclusions are presented in Section 6. For conveniens, we list the most frequently used symbols in Table 1.

2. Related works

Domingos and Richardson [9,10] are the first to study influence maximization problem in social networks and they formulate the problem with a probabilistic framework. Kempe et al. [5] further formulate the problem as a discrete optimization problem, which is widely adopted by subsequent studies. They prove the problem is NP-hard and propose a greedy algorithm to approximately solve it by repeatedly selecting the node that brings the largest marginal influence increase. Following their work, a series of subsequent studies attempt to improve the empirical efficiency [11–15]. However, these works still suffer O($knmr$) computation time and cannot scale to large networks.

Recently, Borg et al. [16] make a theoretical breakthrough and present a near-linear time algorithm under the independent cascade (IC) model. With time complexity to be $O(k\ell^2(m+n)\log^2 n/\varepsilon^3)$, their algorithm returns a $(1-1/e-\varepsilon)$-approximate solution with at least $(1-1/n^\ell)$ probability. Based on the Reverse Reachable Sets (RR-sets) proposed in [16], recent subsequent works [17–20] further reduce the time complexity while retaining the same provable approximate ratio. Methods of TIM [17] and IMM [18] both decrease time complexity to $O((k+\ell)(m+n)\log^2 n/\varepsilon^2)$ while the latter one further reduce the unnecessary computational costs. Following these works, the study of [19] proposes a more tight sampling method which can achieve a sampling size with a constant factor to the optimal value. More recently, using a sampling method based on Bottom-K sketch, the work in [20] further speed up the practical running time with a provable approximate guarantee of $1-1/e-\varepsilon-\varepsilon'$.

Meanwhile, boosting influence spread of a given seed set also attracted research attentions. Some existing works increase influence spread by recommending connections in social networks [21–23]. More recently, the k-boosting problem is proposed in [8] by extending the IC model to the influence boosting model. Given a fixed seed set, it aims to find k boost nodes for increasing the influence spread. However, these works only consider selecting boost nodes or adding edges to increase the influence spread for a given set of seed nodes. Actually, we can explicitly involve both seed and boost nodes in selection, which may offer more flexibility in influence maximization.

Another line of research in IM is budgeted influence maximization [7,24], in which each seed node is assigned a cost and influence is maximized under certain budget constraint. Existing works vary in how the costs are derived. Singer [25] propose a mechanism to elicit the rational agents' true cost while a random cost is used in [7,24]. In practice, selecting influential users as seeds usually incur expensive cost. Actually, we can design a more flexible mechanism for budget allocation with different cost, providing the fact that persuading a user for reposting a tweet usually incurs much lower cost than for posting an original one. Consequently, a better budget allocation can be achieved for influence maximization by involving both seed nodes and boost nodes in selection. As this is the case for most commercial promotions in social media, we expect our work to have good applicability in real world scenarios.

Other extensions of influence maximization includes location-aware influence maximization [26,27], opinion-aware (positive or negative) influence maximization [28,29] and so on. Other diffusion-aware and topic model based social network researches are also actively explored [30–32]. Some recent works heuristically select influential nodes by utilizing label propagation methods [33] or considering the eigenvector centrality [34].

3. Problem definition

To present our problem definition, we will start with introducing the independent cascade (IC) model [5] and its extension of influence boosting model.

In the IC model, given a graph $G=(V,E)$, each edge $e_{uv}\in E$ is associated with a probability p_{uv} and each node $u\in V$ is initially inactive. During the diffusion process, a newly activated node only has one trial to activate its inactive neighboring nodes with probability p_{uv}. The Influence Maximization problem is to find a set $S\subset V$ of k seed nodes such that the expected influence spread $\sigma(S)$, i.e., the expected number of active nodes at the final state, is maximized as each seed in S is activated at the beginning.

Definition 1 (Influence Boosting Model [8])**.** Given a graph $G=(V,E)$, each edge $e_{uv}\in E$ is associated with a probability p_{uv} and a boost probability p'_{uv} with $p'_{uv}>p_{uv}$. During the influence diffusion process, if v is (is not) a boost node, each of its newly-activated in-neighbor u influences v with probability p'_{uv} (p_{uv}).

Given Definition 1, we define a *deployment* as a binary tuple which consists of two sets of nodes, i.e., $\langle S, B\rangle$, where S and B are the seed node set and boost node set respectively. We associate each node u with two costs $c_s(u)$ and $c_b(u)$ representing the costs for seed and boost nodes respectively. Given a *deployment* $\langle S, B\rangle$, we denote $C_s(S)=\sum_{u\in S}c_s(u)$ and $C_b(B)=\sum_{u\in B}c_b(u)$ as the total cost of set S and B respectively.

	a	b	c
cost(seed)	1	2	3
cost(boost)	0.1	0.3	0.5

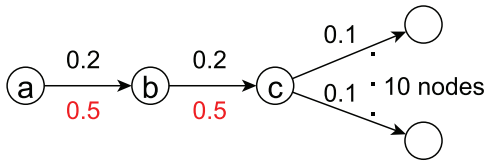


Fig. 1. Toy example of HBIM.

We denote $\sigma(\langle S, B\rangle)$ as the expected influence spread, i.e., the expected number of final active nodes following *deployment* $\langle S, B\rangle$. Note we assume that $c_s(\cdot)$ and $c_b(\cdot)$ are both positive and smaller than budget k. The Holistic Budgeted Influence Maximization problem is then formally stated as follows.

Definition 2 (HBIM)**.** Given a social network $G = (V, E)$ and a budget k, the HBIM problem is to find a deployment $\langle S, B\rangle^*$ which maximize $\sigma(\langle S, B\rangle)$:

$$\langle S, B\rangle^* = \arg\max_{\langle S,B\rangle} \sigma(\langle S, B\rangle)$$
$$s.t.\ C_s(S) + C_b(B) \le k.$$

We can reasonably assume that $c_b(\cdot)$ is much smaller than $c_s(\cdot)$ for each node. Otherwise, if we set $c_b(u) \ge c_s(u)$, then selecting u as a seed node takes smaller cost but brings larger increase on $\sigma(\cdot)$ than being selected as a boost node. Meanwhile, by further setting $c_s(u) = 1$ for each node u, we have the following result.

Theorem 1. *The HBIM problem is NP-hard, and given a deployment* $\langle S, B\rangle$*, computing* $\sigma(\langle S, B\rangle)$ *is #P-hard.*

Proof. If we set $c_b(u) \ge c_s(u)$ and $c_s(u) = 1$ for each node u, the HBIM problem is reduced to the traditional IM problem and $\sigma(\langle S, B\rangle)$ is exactly equal to $\sigma(S)$, since no boost node will be selected. It is known that the IM problem is NP-hard [5] and computing $\sigma(S)$ is #P-hard [12]. Therefore, the theorem is proved. □

To better illustrate HBIM problem, in Fig. 1 we construct a simple example. In Fig. 1, the propagation probabilities are labeled above the edges and the boost propagation probabilities are labeled below the edges (marked in red). The costs of seed and boost nodes are listed in the table. Following traditional IM problem and selecting one node to be seed, then node c is selected since the expected influence spread of node a, b, c are 1.28, 1.4, 2, respectively. However, under the HBIM setting, a better result exists. By selecting node a as seed node and b, c as boost nodes, the expected influence spread will be 2 (same as selecting c as seed), while the cost is 1.8 (less than selecting c as seed).

Non-submodular. It is easy to see that for any fixed B, $\sigma(\langle S, B\rangle)$ is monotone and submodular with S (It equals to traditional influence maximization process [5]). However, for any fixed S, $\sigma(\langle S, B\rangle)$ is monotone but not submodular for B. See Fig. 1 as a counter-example. Let $S = \{a\}$ and $B_1 = \emptyset$, $B_2 = \{c\}$. Then $\sigma(\langle S, B_1 \cup \{b\}\rangle) - \sigma(\langle S, B_1\rangle) = 0.42 < 0.6 = \sigma(\langle S, B_2 \cup \{b\}\rangle) - \sigma(\langle S, B_2\rangle)$ which violates submodularity.

4. Proposed algorithms

Given Theorem 1 and the non-submodularity of the problem, the classical greedy algorithm cannot achieve $1 - 1/e$ approximation. To tackle these problems, in this section, we propose two algorithms for solving HBIM problem with data-dependent approximation by utilizing the Potentially Reverse Reachable graphs (PRR-graph).

4.1. Potentially reverse reachable graphs

Given a network $G = (V, E)$, the generation process of a random PRR-graph [8] is presented as follows.

1. Denote each edge e_{uv} in E as "live" with probability of p_{uv}, "live-upon-boost" with probability of $p'_{uv} - p_{uv}$, and "blocked" with probability of $1 - p'_{uv}$.
2. Denote the residual graph as g with all blocked edges removed, and sample a random root node r from V.
3. Take the subgraph of g which contains all paths that can reach r as a random PRR-graph.

Let R be a random PRR-graph with root r. Then we say r is *reachable* from a node u if there is a path in R containing only *live* edges which starts at u and ends at r. Similarly, given a boost node set B, we say r is *reachable − upon − boosting* B from a node u if there is a path in R which starts at u and ends at r with every edge e_{uv} on it either *live* or *live − upon − boost* with $v \in B$. Now we define the concept of *cover*.

Definition 3 (Cover)**.** Given a deployment $\langle S, B\rangle$, we say a random PRR-graph for root node r is covered by $\langle S, B\rangle$ if r is either *reachable* from a node in S or *reachable − upon − boosting* B from a node in S.

Given $\mathcal{R}$ as a set of random PRR-graphs and $\mathrm{Cov}_{\mathcal{R}}(\langle S, B\rangle)$ as the set of PRR-graphs in $\mathcal{R}$ that covered by $\langle S, B\rangle$, we define $f_{\mathcal{R}}(\langle S, B\rangle) = \frac{n}{|\mathcal{R}|} \cdot |\mathrm{Cov}_{\mathcal{R}}(\langle S, B\rangle)|$ where $n = |V|$. Based on Chernoff bound, $f_{\mathcal{R}}(\langle S, B\rangle)$ can closely estimate $\sigma(\langle S, B\rangle)$ for any $\langle S, B\rangle$ if $|\mathcal{R}|$ is sufficiently large. Therefore, an intuitive approach for solving the HBIM problem is to greedily select seed/boost nodes that marginally maximize $f_{\mathcal{R}}(\cdot)$. However, since the greedy algorithm has no approximation guarantee as discussed above, we turn to optimize a submodular lower bound of the influence spread and utilize the Sandwich Approximation (SA) strategy [19] to approach the optimal solution.

4.2. IMD algorithm

Before detailed into the proposed algorithm, we first present the lower bound function $L(\langle S, B\rangle)$ of $\sigma(\langle S, B\rangle)$.

Lower bound function. Given a PRR-graph set $\mathcal{R}$ and a *deployment* $\langle S, B\rangle$, we define the lower bound function $L(\langle S, B\rangle) = \mathbb{E}[f^-_{\mathcal{R}}(\langle S, B\rangle)]$, where

$$f^-_{\mathcal{R}}(\langle S, B\rangle) = \frac{n}{|\mathcal{R}|} \cdot |\cup_{v \in B} \mathrm{Cov}_{\mathcal{R}}(\langle S, \{v\}\rangle)|.$$

Lemma 1. $L(\langle S, B\rangle) \le \sigma(\langle S, B\rangle)$ *holds for any* $\langle S, B\rangle$.

Proof. By definition, we have $\cup_{v \in B}\mathrm{Cov}_{\mathcal{R}}(S, v) \subseteq \mathrm{Cov}_{\mathcal{R}}(\langle S, B\rangle)$ which leads $f^-_{\mathcal{R}}(\langle S, B\rangle) \le f_{\mathcal{R}}(\langle S, B\rangle)$ hold for any $\langle S, B\rangle$. Meanwhile, by definition we have $\sigma(\langle S, B\rangle) = \mathbb{E}[f_{\mathcal{R}}(\langle S, B\rangle)]$ and $L(\langle S, B\rangle) = \mathbb{E}[f^-_{\mathcal{R}}(\langle S, B\rangle)]$. Thus we have $L(\langle S, B\rangle) \le \sigma(\langle S, B\rangle)$ which proves the lemma. □

Given the lower bound function, we present our algorithm Influence Maximization via Deployment (IMD) in Algorithm 1. It contains three building blocks, DynamicSampling (Line 1), DeploySelectionLB (Line 2) and DeploySelection (Line 3). The DynamicSampling algorithm derives from the D-SSA algorithm [19] which returns a set $\mathcal{R}$ of sufficient number of PRR-graphs. Then we greedily select two solutions which marginally maximize $f^-_{\mathcal{R}}$ and $f_{\mathcal{R}}$ by

Algorithm 1: IMD (G, k, ε).

$\mathcal{R} = \text{DynamicSampling}(G, k, \varepsilon)$;
$\langle S, B\rangle_L = \text{DeploySelectionLB}(G, k, \mathcal{R})$;
$\langle S, B\rangle_\sigma = \text{DeploySelection}(G, k, \mathcal{R})$;
$\langle S, B\rangle^* = \arg\max_{\langle S,B\rangle \in \{\langle S,B\rangle_L, \langle S,B\rangle_\sigma\}} f_{\mathcal{R}}(\langle S, B\rangle)$;
return $\langle S, B\rangle^*$

DeploySelectionLB and DeploySelection respectively. The final solution is selected between the above two greedy solutions with larger value of $f_{\mathcal{R}}$. In the following, we will explain the key steps of the three blocks, and the approximation guarantees and complexity analysis are left in Section 4.3.

4.2.1. DynamicSampling (Algorithm 2)

First, Algorithm 2 generates a set $\mathcal{R}$ with size Λ and invoke DeploySelectionLB function to obtain a current solution

Algorithm 2: DynamicSampling(G, k, ε).

1 $\Lambda \leftarrow (1+1/\varepsilon)^2(2+2\varepsilon/3)\ln 2n$;
2 $\mathcal{R} \leftarrow$ Generate Λ random PRR-graphs;
3 $\langle\langle S, B\rangle_L, f_1\rangle = \text{DeploySelectionLB}(G, k, \mathcal{R})$;
4 **while** $|\mathcal{R}| < (2+2\varepsilon/3)\cdot\ln(2n\Phi)\cdot n\varepsilon^{-2}/k$ **do**
5 $\mathcal{R}' \leftarrow$ Generate $|\mathcal{R}|$ random PRR-graphs;
6 $f_2 \leftarrow \text{Cov}_{\mathcal{R}'}(\langle S, B\rangle_L)$; $\varepsilon_1 \leftarrow f_1/f_2 - 1$; $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}'$;
7 **if** $\varepsilon_1 \le \varepsilon$ **then**
8 $\varepsilon_2 \leftarrow \frac{\varepsilon-\varepsilon_1}{2(1+\varepsilon_1)}$, $\varepsilon_3 \leftarrow \frac{\varepsilon-\varepsilon_1}{2(1-1/\sqrt{e})}$;
9 $\delta_1 \leftarrow \exp(-\frac{f_1\varepsilon_3^2}{(2+2\varepsilon_3/3)(1+\varepsilon_1)(1+\varepsilon_2)})$;
10 $\delta_2 \leftarrow \exp(-\frac{(f_2-1)\varepsilon_2^2}{(2+2\varepsilon_2/3)(1+\varepsilon_2)} + \ln(2\log_2 n))$;
11 **if** $\delta_1 + \delta_2 \le 1/n$ **then**
12 break;
13 $\langle\langle S, B\rangle_L, f_1\rangle = \text{DeploySelectionLB}(G, k, \mathcal{R})$;
14 **return** $\mathcal{R}$

(Line 1–3). Then with the size of $\mathcal{R}$ not exceeding a threshold (Line 4), the algorithm generate another set $\mathcal{R}'$ of $|\mathcal{R}|$ PRR-graphs to check the quality of current solution. If the stopping condition is satisfied (Line 11), it returns the PRR-graph set $\mathcal{R}$.

4.2.2. DeploySelectionLB (Algorithm 3)

Feeded with set $\mathcal{R}$, the DeploySelectionLB algorithm returns a deployment $\langle S, B\rangle_L$ which approximately maximizes $L(\cdot)$. To explain, we define $\Delta_S^-(v)/c_s(v)$ and $\Delta_B^-(v)/c_b(v)$ as gain-cost ratio where

$$\Delta_S^-(v) = f_{\mathcal{R}}^-(\langle S\cup\{v\}, B\rangle) - f_{\mathcal{R}}^-(\langle S, B\rangle),$$
$$\Delta_B^-(v) = f_{\mathcal{R}}^-(\langle S, B\cup\{v\}\rangle) - f_{\mathcal{R}}^-(\langle S, B\rangle).$$

As we can see, the return of DeploySelectionLB algorithm comes from two candidate deployments. $\langle S_1, B_1\rangle$ (obtained by the first *while* loop) is a deployment that contains one seed node with the largest $\Delta_S^-(\cdot)$ and several boost nodes which are greedily selected by the gain-cost ratio. Meanwhile, $\langle S_2, B_2\rangle$ (obtained by the second *while* loop) contains seed nodes and boost nodes which are selected by the gain-cost ratio. It can be proved that the final solution return by DeploySelectionLB ensures a $1-1/\sqrt{e}-\varepsilon$ approximation ratio under the budget constraint (See Section 4.3).

4.2.3. DeploySelection

After returning the deployment $\langle S, B\rangle_L$ which approximately maximizes $L(\cdot)$, the DeploySelection algorithm greedily selects a solution $\langle S, B\rangle_\sigma$ for maximizing $f_{\mathcal{R}}(\cdot)$. It can be implemented by the same process of DeploySelectionLB algorithm with all $\Delta^-(\cdot)$ replaced by $\Delta(\cdot)$, which takes the same formulation of $\Delta^-(\cdot)$ by replacing $f_{\mathcal{R}}^-$ with $f_{\mathcal{R}}$.

4.3. Approximation and complexity

4.3.1. Approximation of DeploySelectionLB

First, Algorithm 2 derives from Algorithm 4 in [19] with slight differences on the threshold (Line 4) and the updating of parameters (Lines 8–10), which ensures the set $\mathcal{R}$ returned is sizable enough if Algorithm 3 achieves an approximate solution $\langle S, B\rangle_L$.

Algorithm 3: DeploySelectionLB $(G, k, \mathcal{R})$.

1 Initialize $\langle S_1, B_1\rangle$ with $S_1 = \emptyset$ and $B_1 = \emptyset$; $V_s = V$;
2 $u_1 = \arg\max_{u\in V_s \cap c_s(u)\le k} \Delta_S(u)$;
3 $S_1 = S_1 \cup \{u_1\}$; $k = k - c_s(u_1)$; $V_s = V_s \backslash u_1$;
4 **while** $V_s \neq \emptyset$ **do**
5 Remove the nodes v with $C_b(B_1\cup\{v\}) > k$ from V.
 $v_1 = \arg\max_{v\in V_s} \Delta_B^-(v)/c_b(v)$;
6 $B_1 = B_1 \cup \{v_1\}$; $V_s = V_s \backslash v_1$;
7 Initialize $\langle S_2, B_2\rangle$ with $S_2 = \emptyset$ and $B_2 = \emptyset$; $V_s = V$;
8 **while** $V_s \neq \emptyset$ **do**
9 $v_1 = \arg\max_v \Delta_S^-(v)/c_s(v)$;
10 $v_2 = \arg\max_v \Delta_B^-(v)/c_b(v)$;
11 **if** $\Delta_S^-(v_1)/c_s(v_1) \ge \Delta_B^-(v_2)/c_b(v_2)$ **then**
12 **if** $C_s(S\cup\{v_1\}) + C_b(B) \le k$ **then**
13 $S_2 = S_2 \cup \{v_1\}$;
14 $V_s = V_s \backslash v_1$;
15 **else**
16 **if** $C_s(S) + C_b(B\cup\{v_2\}) \le k$ **then**
17 $B_2 = B_2 \cup \{v_2\}$;
18 $V_s = V_s \backslash v_2$;
19 $\langle S, B\rangle = \arg\max_{\langle S,B\rangle\in\{\langle S,B\rangle_1, \langle S,B\rangle_2\}} \text{Cov}_{\mathcal{R}}(\langle S, B\rangle)$;
20 **return** $\langle\langle S, B\rangle, \text{Cov}_{\mathcal{R}}(\langle S, B\rangle)\rangle$

Lemma 2 ([19]). *If DeploySelectionLB (Algorithm 3) returns a solution with $(1-1/\sqrt{e})$-approximate ratio for maximizing $f_{\mathcal{R}}^-$, then the set $\mathcal{R}$ returned by DynamicSampling contains no more than, to within a constant factor, the least number of PRR-graphs, which ensures Line 2 of Algorithm 1 can return a $1-1/\sqrt{e}-\varepsilon$-approximate solution for maximizing L with at least $(1-1/n)$ probability.*

The pre-condition of Lemma 2 is that Algorithm 3 ensures a $1-1/\sqrt{e}$ approximation ratio, as proved below.

Lemma 3. *Algorithm 3 returns a $(1-1/\sqrt{e})$ approximate solution for maximizing $f_{\mathcal{R}}^-(\cdot)$.*

Proof. Let $\langle S, B\rangle^\circ$ be the optimal deployment that maximize $f_{\mathcal{R}}^-(\cdot)$. Meanwhile, we denote c_i as the cost of adding the i^{th} node into the deployment in the *while* loop of GreedySelection in Algorithm 2, and $\langle S, B\rangle^i$ be the corresponding generated deployment.

Clearly, $f_{\mathcal{R}}^-(\langle S, B\rangle^\circ) - f_{\mathcal{R}}^-(\langle S, B\rangle^{i-1})$ is no more than the number of PRR-graphs covered by $\langle S, B\rangle^\circ$, but not covered by $\langle S, B\rangle^{i-1}$. For each seed node or boost node in $\langle S, B\rangle^\circ$ but not in $\langle S, B\rangle^{i-1}$, the gain-cost ratio is at most $(f_{\mathcal{R}}^-(\langle S, B\rangle^i) - f_{\mathcal{R}}^-(\langle S, B\rangle^{i-1}))/c_i$, since the greedy selection maximizes this ratio over all candidate nodes. Since the total cost of is bounded by k, we have

$$f_{\mathcal{R}}^{-}(\langle S,B\rangle^{\circ}) - f_{\mathcal{R}}^{-}(\langle S,B\rangle^{i-1}) \le \frac{k}{c_i}\cdot\left(f_{\mathcal{R}}^{-}(\langle S,B\rangle^{i}) - f_{\mathcal{R}}^{-}(\langle S,B\rangle^{i-1})\right).$$

Next we prove the following equation holds.

$$f_{\mathcal{R}}^{-}(\langle S,B\rangle^{i}) \ge \left[1-\prod_{j=1}^{i}\left(1-\frac{c_j}{k}\right)\right]\cdot f_{\mathcal{R}}^{-}(\langle S,B\rangle^{\circ}).$$

Note for $i=1$, it obviously holds (We fix $f_{\mathcal{R}}^{-}(\langle S,B\rangle^{0})=0$ since S, B are empty sets). By an induction which suppose the equation holds for $i-1$ with $i>2$, we have

$$\begin{aligned}
&f_{\mathcal{R}}^{-}(\langle S,B\rangle^{i})\\
&= f_{\mathcal{R}}^{-}(\langle S,B\rangle^{i-1}) + (f_{\mathcal{R}}^{-}(\langle S,B\rangle^{i}) - f_{\mathcal{R}}^{-}(\langle S,B\rangle^{i-1}))\\
&\ge f_{\mathcal{R}}^{-}(\langle S,B\rangle^{i-1}) + \frac{c_i}{k}\cdot(f_{\mathcal{R}}^{-}(\langle S,B\rangle^{\circ}) - f_{\mathcal{R}}^{-}(\langle S,B\rangle^{i-1})\\
&= (1-\frac{c_i}{k})\cdot f_{\mathcal{R}}^{-}(\langle S,B\rangle^{i-1}) + \frac{c_i}{k}\cdot f_{\mathcal{R}}^{-}(\langle S,B\rangle^{\circ})\\
&\ge (1-\frac{c_i}{k})\cdot\left(1-\prod_{j=1}^{i-1}\left(1-\frac{c_j}{k}\right)\right)\cdot f_{\mathcal{R}}^{-}(\langle S,B\rangle^{\circ})\\
&\quad+\frac{c_i}{k}\cdot f_{\mathcal{R}}^{-}(\langle S,B\rangle^{\circ})\\
&= \left(1-\prod_{j=1}^{i}\left(1-\frac{c_j}{k}\right)\right)\cdot f_{\mathcal{R}}^{-}(\langle S,B\rangle^{\circ}).
\end{aligned}$$

Based on all the results above, we new discuss three cases of the return by Algorithm 2.

Case 1: There exists a node s which leads $f_{\mathcal{R}}^{-}(\langle\{s\},\emptyset\rangle)\ge\frac{1}{2}f_{\mathcal{R}}^{-}(\langle S,B\rangle^{\circ})$. If such seed exists, it must be examined by the algorithm as a candidate solution, i.e., $\langle S,B\rangle_1$ returned by SingletonSelection (Line 1 in Algorithm 2) with a value of at least $\frac{1}{2}f_{\mathcal{R}}^{-}(\langle S,B\rangle^{\circ})$.

Case 2: There is no seed satisfying Case 1 and the return of GreedySelection, i.e., $\langle S,B\rangle_2$ in Line 2 of Algorithm 2, satisfies $C_s(S)+C_b(B)<\frac{1}{2}k$. Then for any v not in $S\cup B$, both $c_s(v)$ and $c_b(v)$ are larger than $\frac{1}{2}k$ (otherwise, it could be added to $\langle S,B\rangle_2$). Therefore, there must be only one seed node or one boost node that in $\langle S,B\rangle^{\circ}$ but not in $\langle S,B\rangle_2$. Suppose it is a seed node s. Since $f_{\mathcal{R}}^{-}(\langle\{s\},\emptyset\rangle)<\frac{1}{2}f_{\mathcal{R}}^{-}(\langle S,B\rangle^{\circ})$, it follows that $f_{\mathcal{R}}^{-}(\langle S,B\rangle^{\circ}\cap\langle S,B\rangle_2)\ge\frac{1}{2}f_{\mathcal{R}}^{-}(\langle S,B\rangle^{\circ})$ and thus $f_{\mathcal{R}}^{-}(\langle S,B\rangle_2)\ge\frac{1}{2}f_{\mathcal{R}}^{-}(\langle S,B\rangle^{\circ})$. See if it is a boost node s that in $\langle S,B\rangle^{\circ}$ but not in $\langle S,B\rangle_2$, it follows that $f_{\mathcal{R}}^{-}(\langle S',\{s\}\rangle)\le f_{\mathcal{R}}^{-}(\langle\{s\},\emptyset\rangle)$ for any seed set S' which also confirms the conclusion.

Case 3: There is no seed satisfying Case 1 and $\langle S,B\rangle_2$ satisfies $C_s(S)+C_b(B)\ge\frac{1}{2}k$. See for $a_1,\ldots,a_n\in\mathbb{R}^+$ such that $\sum_{i=1}^{n}a_i=\alpha A$, function $(1-\prod_{i=1}^{n}(1-\frac{a_i}{A}))$ achieves its minimum of $1-(1-\alpha/n)^n$ when $a_1=\cdots=a_n=\alpha A/n$, for $A,\alpha>0$. Let r be the number of nodes added into $\langle S,B\rangle_2$. Then we have

$$\begin{aligned}
f_{\mathcal{R}}^{-}(\langle S,B\rangle^{r}) &\ge \left[1-\prod_{j=1}^{r}\left(1-\frac{c_j}{k}\right)\right]\cdot f_{\mathcal{R}}^{-}(\langle S,B\rangle^{\circ})\\
&\ge \left[1-\left(1-\frac{1}{2r}\right)^{r}\right]\cdot f_{\mathcal{R}}^{-}(\langle S,B\rangle^{\circ})\\
&\ge \left(1-1/\sqrt{e}\right)\cdot f_{\mathcal{R}}^{-}(\langle S,B\rangle^{\circ}).
\end{aligned}$$

Thus, in each case, a value of the solution produced by the Algorithm 2 is at least $(1-1/\sqrt{e})\cdot f_{\mathcal{R}}^{-}(\langle S,B\rangle^{\circ})$, and the lemma follows. □

4.4. Sandwich approximation strategy and submodular lower bound

Combining Lemma 2 and 3, the candidate solution $\langle S,B\rangle_L$ returned in Line 2 of Algorithm 1 is a $(1-1/\sqrt{e}-\varepsilon)$-approximation of maximizing function L. Since the non-submodularity of the problem, we utilize the SA Strategy [23] to select the final solution $\sigma(\langle S,B\rangle^{*})$ between $\langle S,B\rangle_L$ and $\langle S,B\rangle_\sigma$. It is ensured that if $\langle S,B\rangle_L$ is a $(1-1/\sqrt{e}-\varepsilon)$-approximate solution for maximizing $L(\cdot)$, the solution $\sigma(\langle S,B\rangle^{*})$ returned by Algorithm 1 satisfies:

$$\sigma(\langle S,B\rangle^{*}) \ge \frac{L(\langle S,B\rangle^{\circ})}{\sigma(\langle S,B\rangle^{\circ})}\cdot(1-1/\sqrt{e}-\varepsilon)\cdot OPT,$$

where $\langle S,B\rangle^{\circ}$ is optimal solution and $OPT=\sigma(\langle S,B\rangle^{\circ})$.

Let $U(\cdot)$ be a submodular upper bound of the influence spread and $\langle S_u,B_u\rangle$ be the greedy solution with $(1-1/\sqrt{e}-\varepsilon)$ approximate ratio. Then the upper bound version of SA strategy is

$$\sigma(\langle S,B\rangle^{*}) \ge \frac{\sigma(\langle S_u,B_u\rangle^{\circ})}{U(\langle S_u,B_u\rangle^{\circ})}\cdot(1-1/\sqrt{e}-\varepsilon)\cdot OPT.$$

However, in this work, we only use the lower-bound side since L is significantly closer to $\sigma\cdot$ than any upper bound we have tested.

4.4.1. Complexity

By Lemma 2, the set $\mathcal{R}$ returned by DynamicSampling has a theoretically least size within a constant factor. As the worst case, the DynamicSampling never meet the stopping condition until $|\mathcal{R}|$ exceed the threshold in Line 4 of Algorithm 2. The threshold, with value $(2+\frac{2}{3}\varepsilon)\cdot\ln(2n\Phi)\cdot\frac{n}{\varepsilon^2 k}$, is derived from Tang et al. [18] with a slightly loose factor, where Φ is the number of possible *deployment* under the budget constraint. See Φ is bounded by $O(2^{\binom{n}{k/c_m}})$ where c_m is the minimum cost. Thus in the worst case, the size of $\mathcal{R}$ is $O((2+2\varepsilon/3)\cdot\ln(4n)\cdot n^{k+1}\varepsilon^{-2}/k)$. The DeploySelectionLB algorithm can be implemented by the greedy algorithm for maximum coverage and runs in time linear to the size of $\mathcal{R}$. In the DeploySelection algorithm, after we selecting a node, updating $\Delta_S(\cdot)$ and $\Delta_B(\cdot)$ for each node takes time linear to the size of $\mathcal{R}$. Therefore, the time complexity of Algorithm 1 in the worst case is $O((2+2\varepsilon/3)\cdot\ln(4n)\cdot n^{k+1}/\varepsilon^2)$. Combining all the above analysis, we have the following result.

Theorem 2. *With a probability of at least $1-1/n$, the IMD algorithm (Algorithm 1) returns a $(1-1/\sqrt{e}-\varepsilon)\cdot\frac{L(\langle S,B\rangle^{\circ})}{\sigma(\langle S,B\rangle^{\circ})}$-approximate solution, of which the worst time complexity is $O((2+2\varepsilon/3)\cdot\ln(4n)\cdot n^{k+1}/\varepsilon^2)$.*

Though the worst time complexity is expensive when n, k are large, in the experiments we find the algorithm meets the stopping condition very fast with common settings of n, k, which demonstrates that our algorithm is far more practical than the theoretical analysis. In addition, the approximation ratio given in Theorem 2 depends on the ratio of $\frac{L(\langle S,B\rangle^{\circ})}{\sigma(\langle S,B\rangle^{\circ})}$, which should be close to one if the lower bound function is close to the actual influence spread. In the experiments we confirm such closeness on real datasets. Actually, we can simply use $\langle S,B\rangle_L$ returned by Line 2 of Algorithm 1 as the final solution which retains the same approximate ratio but can reduce the running time. We name it as IMD-LB algorithm and compare it with IMD in the experiments.

5. Experiments

5.1. Experimental settings

5.1.1. Datasets

We use three real social networks[1], as listed in Table 2. Epinions is a who-trust-whom online social network of a a general consumer review site Epinions.com. Members of the site can decide whether to "trust" each other. Gowalla is a location-based social networking website where users share their locations by checking-in. The friendship network is undirected and was collected using their public API. Youtube is a video-sharing web site that includes a social network. Note in the experiments we change each undirected edge to bi-directed edge.

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

Table 2
Statistics of data sets.

Data Sets	Nodes	Edges	EdgeType	AverageDegree
Epinions	75K	508K	directed	6.77
Gowalla	197K	950K	undirected	9.64
Youtube	1.1M	2.9M	undirected	5.36

5.1.2. Comparison methods

As far as we know, no existing algorithm is applicable to the HBIM problem. Thus, we compare our proposed algorithms with the modified IM methods, as listed below. All compared algorithms are under the same budget constraint.

- *IMD.* This is Algorithm 1 proposed in this paper, which uses the SA strategy and takes the better one between the exact greedy solution and the greedy solution for maximizing the lower bound function.
- *IMD-LB.* This is an algorithm that directly takes $\langle S, B\rangle_L$ in Line 2 of IMD as solution, which is the greedy solution for maximizing the lower bound function.
- *IMD-Seed.* This method takes only the seed node set in the returned solution of IMD algorithm.
- *IMD-LB-Seed.* This method takes only seed node set in the returned solution of IMD-LB algorithm.
- *IM-N.* This method uses the state-of-the-art IM method D-SSA [19] to greedily select seed nodes, by decreasing order of the marginal gain of the number of nodes.
- *IN-R.* This method uses state-of-the-art IM method D-SSA [19] to greedily select seed nodes, by decreasing order of the marginal gain of the gain-cost ratio defined in Section 4.2.2.

5.1.3. Parameters setting

For the cost of seed/boost node, we set $c_s(v) = \alpha \cdot (1 - e^{-(d_v+1)})$, where d_v is the out-degree of node v and α is a random value sampled from Beta distribution, $\alpha \sim Be(5, 5)$. Meanwhile, we set $c_b(v) = c_s(v)/d_{avg}$ where d_{avg} is the average out-degree of the dataset.

Following [8], we set the boost propagation probability of edge e_{uv} as $p'_{uv} = 1 - (1 - p_{uv})^{\beta}$ $(\beta > 1)$ with $p_{uv} = 1/din_v$ is the propagation probability, where din_v is the in-degree of node v. β is the boost parameter and we set $\beta = 2$ unless otherwise specified. Intuitively, β indicates that every activated neighbor of a boost node v has β independent chances to activate v.

In addition, we set $\varepsilon = 0.1$ and evaluate the influence spread of each solution by 10,000 Monte-Carlo simulations. Each data point is averaged over 5 runs. All the code are implemented with C++. We run the experiments on a Linux server with 24 Core Intel E5 CPU and 256 GB RAM.

5.2. Experimental results

To show the effectiveness and efficiency of the proposed algorithms, we first vary the budget k from 5 to 25 and show the influence spread in Fig. 2(a)–(c). Fig. 2(d) is the ratio of budget for selecting boost nodes of IMD-LB. Meanwhile, we show the increased influence spread brought by the boost nodes (boosted influence spread) in Fig. 3(a)–(c). Fig. 3(d) is the corresponding number of boost nodes of IMD-LB.

5.2.1. Influence Spread

In Fig. 2(a)–(c), the proposed algorithms obviously outperform other methods, except when k is small (Fig. 2(b)). The results show that when the budget is limited we should concentrate the budget on the more expensive seed nodes, which is rather counter-intuitive. Gowalla is a more densely connected network and a small budget k limits the number of both seed nodes and boost nodes. Thus the increase from selecting boost nodes on Gowalla is rather limited. Under such circumstance, concentrating more of the limited budget on seed nodes is worthwhile, despite some boost nodes might have higher gain-cost ratio. For the same reason, the influence spread of IM algorithms (IM-N/IM-R) are higher than the dashed lines (the influence spread of only seed set) in Fig. 2(b), while in Fig. 2(a) and 2(c) they are very close.

Meanwhile, the proposed algorithms achieve much higher influence spread compared to the dashed lines, which shows the importance of selecting both seed nodes and boost nodes. With the help of boost nodes, we can achieve higher influence spread with less seed nodes. This also indicates by selecting both seed nodes and boost nodes the budget can be spent more effectively and flexibly.

In Fig. 2(d), the ratio of budget for selecting boost nodes (boost ratio) are close to a fixed value in all datasets, showing that it is not sensitive to the budget. It can be explained by the "small world" and "scale free" properties [35,36]. With these properties, the densely connnected influential nodes share similar influence spread capabilities and similar number of nearby boost nodes are selected to boost their influence spread. Intuitively, the boost ratio is related to specific boosting models which is confirmed by the following experiments of boost parameter β.

5.2.2. Boosted influence spread and running time

In Fig. 3(a)–(c), B-IMD-LB is IMD-LB minus IMD-LB-Seed, B-IMD is IMD minus IMD-Seed and B-IM is IMD minus max(IM-N,IM-R) correspondingly in Fig. 2(a)–(c). We can see the boosted influence spread, though occasionally have small decrease, exhibit stable increasing tendency in all the three datasets. The corresponding numbers of boost nodes are also increasing with the increase of budget k as shown in Fig. 2(d). The only exception is that in Fig. 2(b) when k is small, as for the same reason discussed in Fig. 2(b).

In addition, combining Figs. 2(d) and 3(d), we find that the lower average degree of a data set, the less number of boost nodes are selected. It is consistent to the intuition that a small amount of boost nodes is enough to transmit the influence if the seed nodes can get in touch with the rich body of social network through a handful of paths.

Meanwhile, we show the running time of the two proposed algorithms in Table 3. We can see IMD-LB always runs faster than IMD with a reduced ratio increasing with the budget k. The only exception is in Youtube. Since Youtube is the largest dataset, both IMD-LB and IMD need to generate larger numbers of PRR-graphs to estimate the influence spread when k is small. Running a Greedy selection on a larger set of PRR-graphs request a larger time cost for IMD, and the IMD-LB thus can achieve a larger time reduction. In addition, the time cost are step changing with budget k. The

Table 3
Running time with different budgets.

Data sets		Running time (s)				
		k = 5	k = 10	k = 15	k = 20	k = 25
Ep	IMD-LB	22.9	47.4	107.5	117.9	217.7
	IMD	24.0	52.1	118.1	130.2	244.0
	Reduce	4.6%	9.1%	9.0%	9.5%	10.8%
Gw	IMD-LB	170.0	337.2	354.9	761.2	1343.6
	IMD	192.0	382.2	419.0	920.7	1647.1
	Reduce	11.5%	11.8%	15.3%	17.3%	18.4%
Yt	IMD-LB	727.4	405.9	407.9	427.7	781.8
	IMD	841.0	434.9	445.4	450.5	870.3
	Reduce	13.5%	6.7%	8.4%	5.1%	10.2%

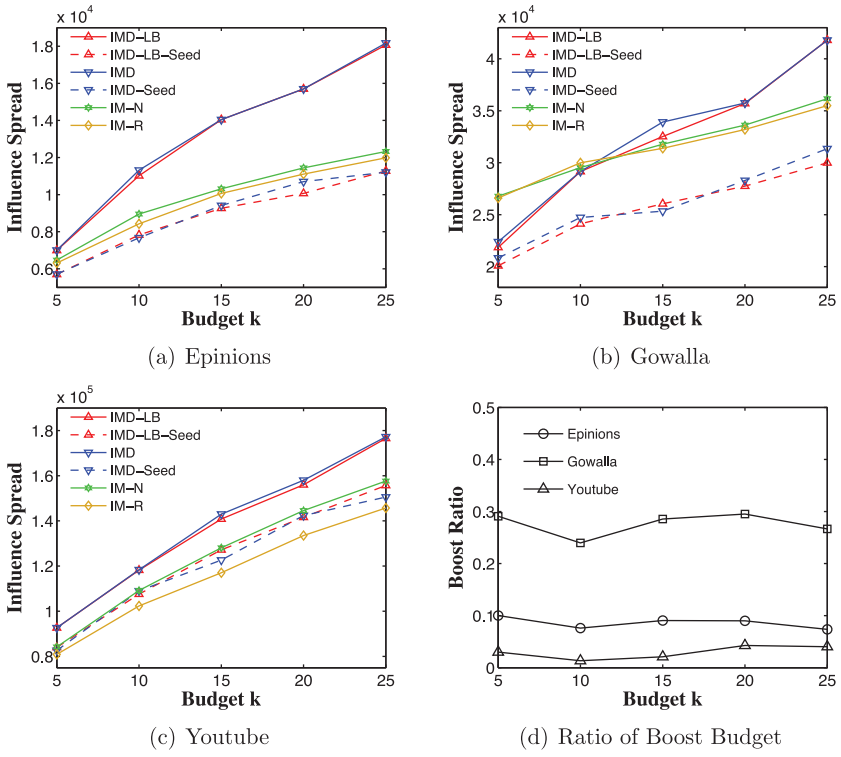


Fig. 2. Influence spread & ratio of budget for boost nodes with different budgets.

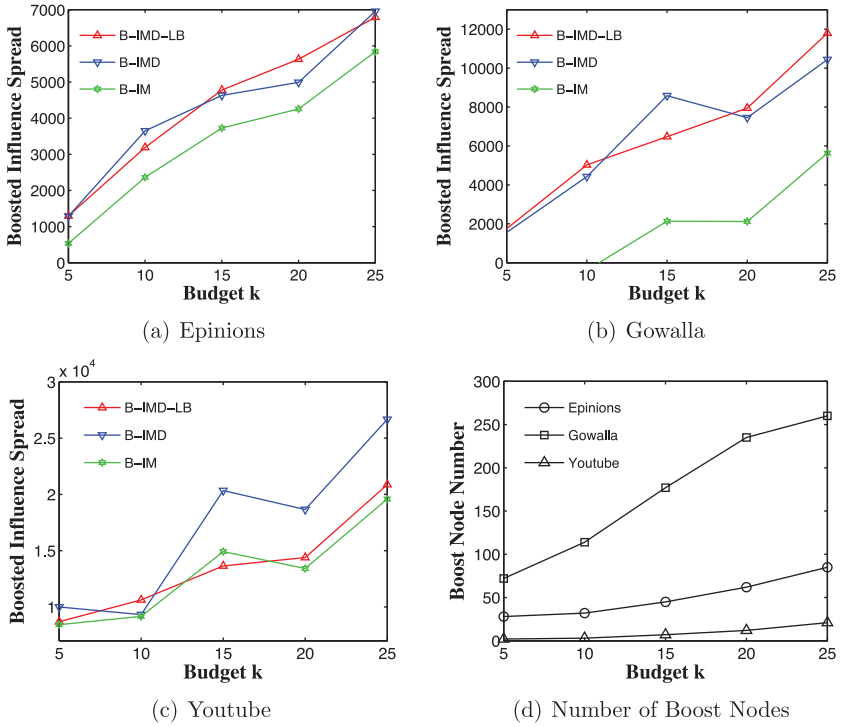


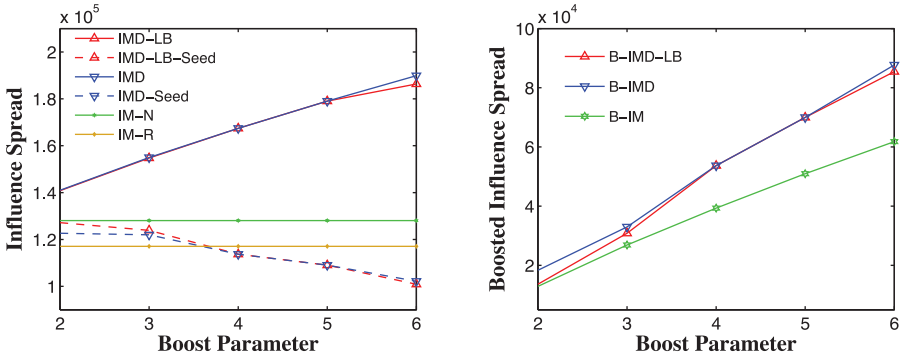
Fig. 3. Boosted influence spread & number of boost nodes with different budgets.

rational is that if the current solution does not match the stopping condition, we need to double the set of PRR-graphs.

To conclude, IMD-LB consistently runs faster than IMD and achieves comparable influence spread in all experiments.

5.2.3. Effects of boost parameter β

To further explore the effects of boost nodes, we fix the budget $k = 15$ and vary the boost parameter β from 2 to 6 in the largest dataset Youtube. Results are shown in Fig. 4(a) and (b).



(a) Influence Spread (b) Boosted Influence Spread

Fig. 4. Influence spread & boosted influence spread in youtube with budget $k = 15$ and different boost parameter β.

Table 4
Running time and boost results in youtube with $k = 15$.

β	Running Time (s)			Boost	Boost
	IMD-LB	IMD	Reduced ratio(%)	Ratio	Number
2	407.87	445.36	8.42	0.02	6
3	687.56	786.76	12.63	0.05	17
4	910.33	1087.05	16.26	0.21	53
5	1054.85	1333.10	20.87	0.25	65
6	2188.41	2945.65	25.71	0.43	97

With the increase of β, both the influence spread and the boosted influence spread of the proposed algorithms stably increase as excepted. Meanwhile, we interestingly find the influence spread of only seed set decrease with β being large. To explain, we list the detail of boost nodes in Table 4.

In Table 4, the ratio of budget used for selecting boost nodes and the number of boost nodes increase very fast with β being large. Generally, the cost of selecting one seed node can be used to select dozens of boost nodes. When β is large, which means boosting a node may bring large increase of influence spread, it deserves to use a large fraction of budget to select boost nodes. Actually, the boost parameter β (or other boost patterns), in a considerable degree, decides the boost ratio. Accordingly, by experiments or some prior knowledge of finding β, the boost ratio will be subsequently derived which can help finding the optimal deployment more efficiently. It is also interesting to learn other specific boost patterns from real information spread data. Exploring the relation between boost patterns and boost ratio may open new directions for further investigations of improving the influence spread in real world applications.

As for running time, compared to IMD, IMD-LB achieves a reduced ratio which is near proportional to β, and the step changing property is also observed. The results in Table 4 also confirm that IMD-LB is both efficient and effective.

6. Conclusion

In this work, we present a novel holistic budgeted influence maximization (HBIM) problem that maximizes the influence spread by finding the optimal deployment of seed&boost nodes. We develop two efficient approximation algorithms, IMD and IMD-LB, with data-dependent approximation ratios. Both algorithms are delicate integrations of Potentially Reverse Reachable Graphs, state-of-the-art IM method and greedy selection algorithm. Extensive experiments are conducted on real social networks and the experimental results have demonstrated the superiority of the proposed algorithms. Compared with IMD, IMD-LB returns solution with comparable quality but has lower computational costs. Specifically, in the experiments we find the boost pattern do affect the boost ratio, of which the inside relation deserves exploration. In addition, it is also interesting to learn the boost patterns from real information spread data.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Grant no: U1866602) and by a Discovery Grant from the National Science and Engineering Research Council of Canada. It is also partially supported by ByteDance.

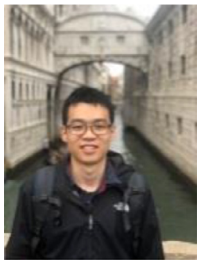
References

[1] R. Felix, P.A. Rauschnabel, C. Hinsch, Elements of strategic social media marketing: a holistic framework, J. Bus. Res. 70 (2017) 118–126.
[2] A.M. Shaltoni, E-marketing education in transition: an analysis of international courses and programs, Int. J. Manag. Educ. 14 (2) (2016) 212–218.
[3] R. Filieri, What makes online reviews helpful? A diagnosticity-adoption framework to explain informational and normative influences in e-WOM, J. Bus. Res. 68 (6) (2015) 1261–1270.
[4] B. Schivinski, D. Dabrowski, The effect of social media communication on consumer perceptions of brands, J. Market. Commun. 22 (2) (2016) 189–214.
[5] D. Kempe, J. Kleinberg, É. Tardos, Maximizing the spread of influence through a social network, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2003, pp. 137–146.
[6] W. Chen, C. Wang, Y. Wang, Scalable influence maximization for prevalent viral marketing in large-scale social networks, in: Proceedings of the Sixteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 1029–1038.
[7] N. Hallier, On budgeted influence maximization in social networks, IEEE J. Sel. Areas Commun. 31 (6) (2013) 1084–1094.
[8] Y. Lin, W. Chen, J.C.S. Lui, Boosting information spread: an algorithmic approach, in: Proceedings of the IEEE International Conference on Data Engineering, 2017.
[9] P. Domingos, M. Richardson, Mining the network value of customers, in: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2001, pp. 57–66.
[10] M. Richardson, P. Domingos, Mining knowledge-sharing sites for viral marketing, in: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2002, pp. 61–70.
[11] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, N. Glance, Cost-effective outbreak detection in networks, in: Proceedings of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2007, pp. 420–429.
[12] W. Chen, Y. Wang, S. Yang, Efficient influence maximization in social networks, in: Proceedings of the Fifteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2009, pp. 199–208.
[13] W. Chen, Y. Yuan, L. Zhang, Scalable influence maximization in social networks under the linear threshold model, in: Proceedings of the IEEE Tenth International Conference on Data Mining (ICDM), IEEE, 2010, pp. 88–97.
[14] A. Goyal, F. Bonchi, L.V. Lakshmanan, A data-based approach to social influence maximization, Proc. VLDB Endow. 5 (1) (2011) 73–84.

[15] J. Kim, S.-K. Kim, H. Yu, Scalable and parallelizable processing of influence maximization for large-scale social networks, in: Proceedings of the IEEE Twenty-Ninth International Conference on Data Engineering (ICDE), IEEE, 2013, pp. 266–277.
[16] C. Borgs, M. Brautbar, J. Chayes, B. Lucier, Maximizing social influence in nearly optimal time, in: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, 2014, pp. 946–957.
[17] Y. Tang, X. Xiao, Y. Shi, Influence maximization: near-optimal time complexity meets practical efficiency, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, ACM, 2014, pp. 75–86.
[18] Y. Tang, Y. Shi, X. Xiao, Influence maximization in near-linear time: a martingale approach, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, ACM, 2015, pp. 1539–1554.
[19] H.T. Nguyen, M.T. Thai, T.N. Dinh, Stop-and-stare: optimal sampling algorithms for viral marketing in billion-scale networks, in: Proceedings of the International Conference on Management of Data, ACM, 2016, pp. 695–710.
[20] X. Wang, Y. Zhang, W. Zhang, X. Lin, C. Chen, Bring order into the samples: a novel scalable method for influence maximization, IEEE Trans. Knowl. Data Eng. 29 (2) (2017) 243–256.
[21] D. Rafailidis, A. Nanopoulos, Crossing the boundaries of communities via limited link injection for information diffusion in social networks, in: Proceedings of the Twenty-Fourth International Conference on World Wide Web, ACM, 2015, pp. 97–98.
[22] D.-N. Yang, H.-J. Hung, W.-C. Lee, W. Chen, Maximizing acceptance probability for active friending in online social networks, in: Proceedings of the Nineteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2013, pp. 713–721.
[23] W. Lu, W. Chen, L.V. Lakshmanan, From competition to complementarity: comparative influence diffusion and maximization, Proc. VLDB Endowment 9 (2) (2015) 60–71.
[24] Y. Wang, W.J. Huang, L. Zong, T.J. Wang, D.Q. Yang, Influence maximization with limit cost in social network, Sci. China Inf. Sci. 56 (7) (2013) 1–14.
[25] Y. Singer, How to win friends and influence people, truthfully:influence maximization mechanisms for social networks, in: Proceedings of the ACM International Conference on Web Search and Data Mining, 2012, pp. 733–742.
[26] X. Li, X. Cheng, S. Su, C. Sun, Community-based seeds selection algorithm for location aware influence maximization, Neurocomputing 275 (2018) 1601–1613.
[27] X. Wang, Y. Zhang, W. Zhang, X. Lin, Distance-aware influence maximization in geo-social network., in: Proceedings of the ICDE, 2016, pp. 1–12.
[28] S. Galhotra, A. Arora, S. Roy, Holistic influence maximization: combining scalability and efficiency with opinion-aware models, in: Proceedings of the 2016 International Conference on Management of Data, ACM, 2016, pp. 743–758.
[29] D. Li, C. Wang, S. Zhang, G. Zhou, D. Chu, C. Wu, Positive influence maximization in signed social networks based on simulated annealing, Neurocomputing 260 (2017) 69–78.
[30] A. Khan, N. Li, X. Yan, Z. Guan, S. Chakraborty, S. Tao, Neighborhood based fast graph search in large networks, in: Proceedings of the ACM SIGMOD International Conference on Management of data, ACM, 2011, pp. 901–912.
[31] B. Wang, M. Ester, J. Bu, Y. Zhu, Z. Guan, D. Cai, Which to view: Personalized prioritization for broadcast emails, in: Proceedings of the Twenty-Fifth International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2016, pp. 1181–1190.
[32] S. Tan, Z. Guan, D. Cai, X. Qin, J. Bu, C. Chen, Mapping users across networks by manifold alignment on hypergraph., in: proceedings of the AAAI, 14, 2014, pp. 159–165.
[33] Y. Zhao, S. Li, F. Jin, Identification of influential nodes in social networks with community structure based on label propagation, Neurocomputing 210 (2016) 34–44.
[34] X. Li, Y. Liu, Y. Jiang, X. Liu, Identifying social influence in complex networks: a novel conductance eigenvector centrality model, Neurocomputing 210 (C) (2016) 141–154.
[35] A.L. Barabasi, R. Albert, H. Jeong, Scale-free characteristics of random networks: the topology of the world-wide web, Phys. A Stat. Mech. Appl. 281 (1) (2000) 69–77.
[36] A. Vazquez, R. Dobrin, D. Sergi, J.P. Eckmann, Z.N. Oltvai, A.L. Barabasi, The topological relationship between the large-scale attributes and local interaction patterns of complex networks, Proc. Natl. Acad. Sci. U.S.A 101 (52) (2004) 17940–17945.

Qihao Shi received his B.S. at Nanjing Normal University of China in 2014. He is currently a Ph.D. candidate in the college of computer science at Zhejiang University. His main research topics are social and information networks, algorithmic game theory and Internet economics.

Can Wang received the Ph.D. degree and M.S. degree in computer science and B.S. degree in economics from Zhejiang University, in 2009, 2003 and 1995 respectively. His research interests include data mining, machine learning and information retrieval.



Jiawei Chen received his B.S. at University of electronic and technology of China in 2014. he is currently a Ph.D. candidate in the college of computer science at Zhejiang University. His main research topics are recommendation, graphical model and deep learning.

Yan Feng received the Ph.D. degree in computer application from Zhejiang University in 2004. She is currently an associate professor in the College of Computer Science at Zhejiang University, China. Her research interests include database, data mining etc.

Chun Chen is a professor in the College of Computer Science, Zhejiang University. His research interests include data mining, computer vision, computer graphics and embedded technology.