Towards Profit Maximization for Online Social Network Providers

Jing Tang
School of Comp. Sci. & Eng.
Nanyang Technological University
Email: tang0311@ntu.edu.sg

Xueyan Tang
School of Comp. Sci. & Eng.
Nanyang Technological University
Email: asxytang@ntu.edu.sg

Junsong Yuan
School of EEE
Nanyang Technological University
Email: jsyuan@ntu.edu.sg

***Abstract*—Online Social Networks (OSNs) attract billions of users to share information and communicate where viral marketing has emerged as a new way to promote the sales of products. An OSN provider is often hired by an advertiser to conduct viral marketing campaigns. The OSN provider generates revenue from the commission paid by the advertiser which is determined by the spread of its product information. Meanwhile, to propagate influence, the activities performed by users such as viewing video ads normally induce diffusion cost to the OSN provider. In this paper, we aim to find a seed set to optimize a new profit metric that combines the benefit of influence spread with the cost of influence propagation for the OSN provider. Under many diffusion models, our profit metric is the difference between two submodular functions which is challenging to optimize as it is neither submodular nor monotone. We design a general two-phase framework to select seeds for profit maximization and develop several bounds to measure the quality of the seed set constructed. Experimental results with real OSN datasets show that our approach can achieve high approximation guarantees and significantly outperform the baseline algorithms, including state-of-the-art influence maximization algorithms.**

I. Introduction

Information can be disseminated widely and rapidly through Online Social Networks (OSNs) (such as Facebook, Twitter, Flickr, Google+, and LinkedIn) with "word-of-mouth" effects. Viral marketing is a typical application that leverages OSNs as the medium for information diffusion [9]. The market of OSN advertisement is growing explosively. For example, Fortune [25] reports that the online and digital advertisement spending for the 2016 election of the United States will reach 1.2 billion US dollars in which 49% is expected to go to social media.

Advertisers often delegate the operation of viral marketing campaigns to OSN providers who have the complete information of their social network structures [3]. Providing viral marketing services is a potential and promising approach that OSN providers can explore for monetization. When hiring an OSN provider to conduct the viral marketing campaign, the advertiser usually pays the OSN provider a commission for each user that adopts its product or shares its ads. Thus, the revenue of the OSN provider generated from the commission is determined by the spread of the product information. Meanwhile, to propagate influence, the activities performed by users such as viewing video ads normally induce diffusion cost to the OSN provider. For example, for a cloud-based OSN running viral video marketing [20], the OSN will be charged by the cloud for each video click due to the data traffic produced. Intuitively, the number of clicks for each video is dependent on the number of connections among users that are actually used during the viral marketing campaign. Therefore, to maximize its profit, the OSN provider needs to account for both the reward of influence spread and the expense of influence propagation, both of which are dependent on the seed users selected to initialize the viral marketing campaign. In this paper, we study the problem of finding a seed set to optimize such a new profit metric that combines the benefit of influence spread with the cost of influence propagation.

Our profit metric is challenging to optimize as it is significantly different from the influence metric that has been studied widely [6], [7], [12], [14], [15], [17], [22], [23], [24], [27], [28], [30], [33], [34], [35]. It is well known that the influence spread generated by a seed set is submodular and monotone under many diffusion models [1], [15], [26], which makes it easy to design influence maximization algorithms with strong approximation guarantees [21]. Some recent studies have addressed profit maximization from the advertiser's perspective by putting together the benefit of influence spread and the cost of seed selection [19], [29], [31], [36]. In these studies, the cost of seed selection is modeled by the incentives (e.g., free samples) provided to seed users. Since the seed selection cost is given by the sum of the costs of individual seeds which is modular, the resultant profit metric is still submodular. In contrast, in our problem, both the revenue and cost of the OSN provider are attached to the diffusion process. As a result, our profit metric can be viewed as the difference between two submodular functions, which is neither submodular nor monotone and is thus much more difficult to deal with.

In this paper, we propose a general two-phase framework to optimize profit for the OSN provider. Our framework is comprised of a pruning phase that iteratively narrows down the search space and a search phase that uses heuristics to select seed nodes within the reduced search space. We theoretically establish the advantages of our pruning technique in improving the effectiveness and efficiency of any algorithm used in the search phase. We further derive several bounds on the maximum achievable profit to evaluate the quality of the seed sets obtained by any algorithm. We conduct extensive experiments with several real OSN datasets. The results demonstrate the effectiveness and efficiency of our framework.

The rest of this paper is organized as follows. Section II reviews the related work. Section III defines the profit maximization problem for the OSN provider. Section IV elaborates the design of our two-phase framework. Section V develops the bounds for quality measurement. Section VI presents the experimental evaluation. Finally, Section VII concludes the paper.

II. Related Work

Influence Maximization. Kempe et al. [15] formulated an influence maximization problem for a given seed set size with two basic diffusion models, namely the Independent Cascade (IC) and Linear Threshold (LT) models. They showed that the influence spreads under these models are submodular and monotone. Thus, they proposed a simple hill-climbing greedy algorithm to address the problem, which can provide a $(1-1/e)$-approximation guarantee [21]. The follow-up studies have mainly concentrated on improving the efficiency of the algorithm implementation for large-scale OSNs [6], [7], [12], [14], [17], [22], [23], [24], [27], [28], [30], [33], [34], [35]. Different from the above studies, we target at maximizing the profit that accounts for both the benefit of influence spread and the cost of influence propagation in viral marketing.

Profit Maximization. Maximizing the influence spread alone has been shown to be ineffective for optimizing the profit return of viral marketing [29], [31]. This is because the number of seeds selected yields a tradeoff between the expense and reward of viral marketing. To avoid pre-setting the number of seeds to select, some recent work studied profit maximization from the advertiser's perspective [19], [29], [31], [36]. These studies considered the cost of seed selection which is modular and implies that their profit metric is still submodular. They investigated heuristics to select seeds under the assumption that the social network structures are available to the advertiser. In practice, only the OSN providers have the complete information of their social graphs and they often keep the formation secret for business and privacy reasons [3], [16]. Therefore, the OSN providers are able to run viral marketing campaigns more efficiently than the advertisers. Different from the above work, we formulate a new profit maximization problem from the OSN provider's perspective that takes into account the cost of information diffusion over the social network. As shall be shown later, our profit metric is the difference between two submodular functions, which is neither submodular nor monotone.

Submodular Optimization. Iyer and Bilmes [13] investigated optimizing the difference between two submodular functions. They showed that this problem is multiplicatively inapproximable unless P=NP and proposed several heuristic methods to address the problem. The greedy algorithm [15], [29] is another widely used heuristic for submodular optimization. We apply these heuristic methods to the search phase of our two-phase framework. More importantly, we develop an iterative pruning technique to reduce the search space which significantly improves these heuristics in terms of not only effectiveness but also efficiency. In addition, we also derive several upper bounds of the optimum to benchmark the output of any algorithm.

III. Problem Formulation

A. Preliminaries

Let $G = (V, E)$ be a directed graph modeling an OSN, where the nodes V represent users and the edges E represent the connections among users (e.g., friendships on Facebook, 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.

There are many diffusion models for the processes by which influence propagates in social networks. The diffusion model is normally a random process that starts with a set of seed nodes S. Initially, the seed nodes S are activated, while all the other nodes are not activated. When a node u becomes activated, it would attempt to further activate its neighbors which are not yet activated. The diffusion process terminates when no more node can be further activated. Let $g \sim G$ be a sample outcome of influence propagation by the diffusion process and let $V_g(S)$ be the set of nodes activated by the seed set S in the sample outcome g. The *influence spread* of the seed set S, denoted by $\sigma(S)$, is the expected number of nodes activated over all possible sample outcomes of influence propagation, i.e., $\sigma(S) = \mathbb{E}[|V_g(S)|]$.

B. The Profit Maximization Problem

As discussed, the influence spread is the benefit gained by the OSN provider and the cost of influence propagation is the price to pay for viral marketing. We assume that each node v in the social network is associated with a benefit weight $b(v)$, which represents the benefit offered by activating v (e.g., the commission received from the advertiser). The benefit $\beta(S)$ of influence spread generated by a seed set S is the total benefit brought by all the nodes activated:

$$\beta(S) = \mathbb{E}\Big[\sum_{v \in V_g(S)} b(v)\Big]. \tag{1}$$

To model the cost of influence propagation, we consider the diffusion process in the social network. Recall that when a node is activated, it attempts to further activate its neighbors through the connections to them in the social network. Without loss of generality, we assume that each node v is associated with a cost weight $c(v)$, which represents the cost of information diffusion incurred by v in attempting to activate its neighbors when it becomes activated (e.g., pushing video ads). The cost $\gamma(S)$ of influence propagation introduced by a seed set S is the total cost incurred by all the nodes activated:

$$\gamma(S) = \mathbb{E}\Big[\sum_{v \in V_g(S)} c(v)\Big]. \tag{2}$$

Then, we naturally define a *profit* metric from OSN provider's perspective as the benefit of influence spread less the cost of influence propagation, i.e., the profit $\phi(S)$ for running a viral marketing campaign with a seed set S is given by

$$\phi(S) = \beta(S) - \gamma(S), \tag{3}$$

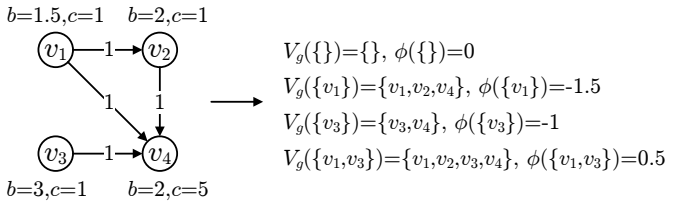


Fig. 1. An example showing the non-monotonicity and non-submodularity of the profit function $\phi(\cdot)$. Each node is associated with a benefit weight b and a cost weight c.

Our goal is to find a seed set S to maximize the profit $\phi(S)$.

By defining $w(v) = b(v) - c(v)$, we can rewrite the profit metric in (3) as

$$\phi(S) = \mathbb{E}\Big[\sum_{v\in V_g(S)} \big(b(v) - c(v)\big)\Big] \triangleq \mathbb{E}\Big[\sum_{v\in V_g(S)} w(v)\Big].$$

Here, $w(v)$ represents the profit gain of activating a node v. If the benefit offered by v outweighs its cost, $w(v)$ is positive. Otherwise, $w(v)$ is negative. Therefore, we can normalize the benefit and cost weights of each node v by setting $\bar{b}(v) = \max\{0, w(v)\}$ and $\bar{c}(v) = \max\{0, -w(v)\}$. Then, the benefit and cost metrics become

$$\bar{\beta}(S) = \mathbb{E}\Big[\sum_{v\in V_g(S)} \bar{b}(v)\Big], \quad \text{and} \quad \bar{\gamma}(S) = \mathbb{E}\Big[\sum_{v\in V_g(S)} \bar{c}(v)\Big],$$

respectively. We show later that this normalized form can make the pruning phase of our proposed framework more effective.

It has been proved that the influence function $\sigma(\cdot)$ is submodular under many commonly used diffusion models due to $V_g(S) \subseteq V_g(T)$ for any $S \subseteq T$ [15]. Similarly, it can be shown that the benefit and cost metrics defined above are both submodular under these diffusion models. That is, for any two seed sets S and T where $S \subseteq T$ and any node $v \notin T$, it holds that $\beta(S\cup\{v\}) - \beta(S) \geq \beta(T\cup\{v\}) - \beta(T)$ and $\gamma(S\cup\{v\}) - \gamma(S) \geq \gamma(T\cup\{v\}) - \gamma(T)$. Likewise, the normalized $\bar{\beta}(\cdot)$ and $\bar{\gamma}(\cdot)$ are also submodular. Although both the benefit and cost metrics are monotone and submodular, the profit metric is neither monotone nor submodular (an example is given in Fig. 1). Thus, the methods used for maximizing monotone submodular functions would perform poorly for our profit maximization problem as shall be shown in our experiments. In the following, we develop heuristic algorithms to address our profit maximization problem.

Example 1: Fig. 1 gives an example to illustrate that the profit function $\phi(\cdot)$ is neither monotone nor submodular. In this example, once a node is activated, it will activate all of its neighbors. It can be seen from the figure that $\phi(\emptyset) = 0$, $\phi(\{v_1\}) = -1.5$, $\phi(\{v_3\}) = -1$ and, $\phi(\{v_1, v_3\}) = 0.5$. Thus, $\phi(\emptyset) > \phi(\{v_1\})$ and $\phi(\{v_1\}) < \phi(\{v_1, v_3\})$, which indicates that $\phi(\cdot)$ is non-monotone. In addition, we can also get that $\phi(\{v_1\}) - \phi(\emptyset) = -1.5 < \phi(\{v_1, v_3\}) - \phi(\{v_3\}) = 1.5$, which indicates that $\phi(\cdot)$ is non-submodular.

IV. Two-Phase Framework

We propose a general two-phase framework to select seed nodes for optimizing the profit.

Algorithm 1: IterativePrune

start with $A_0 \leftarrow \emptyset$, $B_0 \leftarrow V$ and $t = 0$;
repeat
  $A_{t+1} \leftarrow A_t \cup \{v\colon \beta(v \mid B_t \setminus \{v\}) - \gamma(v \mid A_t) > 0$ and $v \in B_t \setminus A_t\}$;
  $B_{t+1} \leftarrow B_t \setminus \{v\colon \beta(v \mid A_t) - \gamma(v \mid B_t \setminus \{v\}) < 0$ and $v \in B_t \setminus A_t\}$;
  $t \leftarrow t + 1$;
until *converged, i.e.,* $A_t = A_{t-1}$ *and* $B_t = B_{t-1}$;
return A_t and B_t as A^* and B^*;

- **Pruning phase (Algorithm 1):** We develop an iterative pruning technique to cut the search space.
- **Search phase (Algorithms 2 and 3):** We use some heuristics to find the solution in the reduced search space.

A. Prune Search Space

To simplify the notations, we define the marginal gain of adding a node v to a seed set S for metrics ϕ, β and γ as

$$\phi(v \mid S) = \phi(S\cup\{v\}) - \phi(S),$$
$$\beta(v \mid S) = \beta(S\cup\{v\}) - \beta(S),$$
$$\gamma(v \mid S) = \gamma(S\cup\{v\}) - \gamma(S).$$

We start by proposing an iterative pruning approach that can dramatically reduce the search space from the power set of V to a smaller lattice for maximizing the profit function. Algorithm 1 shows the pseudo code of the pruning algorithm. Recall that by the submodularity of the benefit and cost metrics, their marginal gains for adding a new seed node decrease with the base seed set. Thus, the largest possible benefit (cost) gain is produced by adding a node into an empty seed set, whereas the smallest possible benefit (cost) gain is generated by adding the node into an almost universal set. Note that the marginal profit gain is bounded below by the smallest benefit gain less the largest cost gain. So, it is intuitive that if the latter is positive, the node must be selected in an optimal solution. Similarly, the marginal profit gain is bounded above by the largest benefit gain less the smallest cost gain. If the latter is negative, the node cannot be selected in an optimal solution. Algorithm 1 extends this idea in an iterative manner to reduce the search space. It is easy to verify that after each iteration, the newly generated lattice is a sublattice of that in the previous iteration, i.e., $A_t \subseteq A_{t+1} \subseteq B_{t+1} \subseteq B_t$. Furthermore, it can be proved that any seed set outside the resultant lattice $[A^*, B^*]$ delimited by the node sets A^* and B^* returned by Algorithm 1 can be transformed to a seed set in $[A^*, B^*]$ with higher profit. The formal proofs of all theoretical results are given in the extended version [32].

Theorem 1: For any node set S and $t \geq 0$, let $S_t = S \cap B_t \cup A_t$, it holds that $\phi(S_t) \leq \phi(S_{t+1})$, where the "=" holds if and only if $S_t = S_{t+1}$.

Theorem 1 shows that the pruning technique can only increase the profit value of the seed set $S_t = S \cap B_t \cup A_t$ at every iteration. The following corollary shows how our search space reduction improves the quality of any solution.

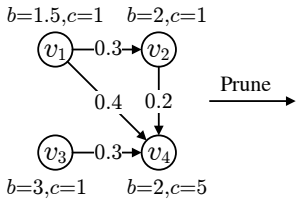


Fig. 2. An example of iterative pruning under the Independent Cascade diffusion model. Each node is associated with a benefit weight b and a cost weight c. Each edge has a propagation probability p.

Corollary 1: For any node set S, if $S \notin [A^*, B^*]$, then $\phi(S) < \phi(S \cap B^* \cup A^*)$.

By Corollary 1, the pruning approach can always improve the quality of any seed set S outside $[A^*, B^*]$ by transforming it to the seed set $S \cap B^* \cup A^*$ in $[A^*, B^*]$. Thus, the lattice $[A^*, B^*]$ retains all the optimal seed sets. As a result, we can reduce the search space from the lattice $[\emptyset, V]$ to $[A^*, B^*]$. Our pruning approach can be used prior to any seed selection algorithms to improve the solution quality.

Corollary 2: For any seed set S^* producing the maximum achievable profit, it holds that $A^* \subseteq S^* \subseteq B^*$.

Example 2: Fig. 2 gives an example to illustrate how the pruning algorithm works as well as the above theorem and corollaries. This example assumes the Independent Cascade (IC) diffusion model. The IC model is a representative and most widely-studied diffusion model for influence propagation [6], [7], [14], [15], [17], [22], [23], [24], [27], [28], [30], [33], [34], [35]. In the IC model, 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 their connection. In the diffusion process, when a node u first becomes activated, it has a chance to activate its neighbors who are not yet activated. Each such neighbor v would become activated with probability $p_{u,v}$. This process repeats until no more node can be activated. For example, in Fig. 2, when $\{v_1, v_3\}$ are selected as seeds, v_2 would be activated with probability 0.3. Meanwhile, v_4 would be activated by v_1 with probability 0.4, by v_2 with probability $0.3 \times 0.2 = 0.06$, and by v_3 with probability 0.3. Thus, overall, v_4 would be activated with probability $1 - (1 - 0.4) \times (1 - 0.06) \times (1 - 0.3) = 0.6052$.

To conduct iterative pruning, A_0 is initialized by $\emptyset$ and B_0 is initialized by $\{v_1, v_2, v_3, v_4\}$ respectively. To simplify the notations, let $\phi_t^-(v) = \beta(v \mid B_t \setminus \{v\}) - \gamma(v \mid A_t)$ and $\phi_t^+(v) = \beta(v \mid A_t) - \gamma(v \mid B_t \setminus \{v\})$ describe the calculations in Algorithm 1. At iteration 1, $\phi_1^-(v_1) = \beta(v_1 \mid \{v_2, v_3, v_4\}) - \gamma(v_1 \mid \emptyset) = 1 \times b(v_1) - \big(1 \times c(v_1) + 0.3 \times c(v_2) + (1 - (1 - 0.4) \times (1 - 0.3 \times 0.2)) \times c(v_4)\big) = 1.5 - (1 + 0.3 + 2.18) = 1.5 - 3.48 = -1.98 < 0$. Similarly, $\phi_1^-(v_2) = -0.6 < 0$, $\boldsymbol{\phi_1^-(v_3) = 0.5 > 0}$, $\phi_1^-(v_4) = -4.328 < 0$, $\phi_1^+(v_1) = 1.972 > 0$, $\phi_1^+(v_2) = 1.7 > 0$, $\phi_1^+(v_3) = 2.6 > 0$, and $\phi_1^+(v_4) = 0.32 > 0$. Thus, v_3 is added to A_1 so that $A_1 = \{v_3\}$ and $B_1 = \{v_1, v_2, v_3, v_4\}$. At iteration 2, $\phi_2^-(v_1) = -1.326 < 0$, $\phi_2^-(v_2) = -0.3 < 0$, $\phi_2^-(v_4) = -2.828 < 0$, $\phi_2^+(v_1) = 1.7104 > 0$, $\phi_2^+(v_2) = 1.58 > 0$, and $\boldsymbol{\phi_2^+(v_4) = -0.28 < 0}$. Thus, v_4 is removed from B_2 so that $A_2 = \{v_3\}$ and $B_2 = \{v_1, v_2, v_3\}$. At iteration 3, $\phi_3^-(v_1) = -0.878 < 0$, $\phi_3^-(v_2) = -0.1824 < 0$, $\phi_3^+(v_1) = 0.5904 > 0$, and $\phi_3^+(v_2) = 1.286 > 0$. Thus, both A_3 and B_3 remain the same as in the previous iteration. As a result, $A^* = \{v_3\}$ and $B^* = \{v_1, v_2, v_3\}$ are returned. For a seed set $S = \{v_2, v_4\} \notin [A^*, B^*]$, we have $S_1 = S \cap B_1 \cup A_1 = \{v_2, v_3, v_4\}$ and $S_2 = S_3 = S \cap B^* \cup A^* = \{v_2, v_3\}$. Then, it can be obtained that $\phi(S) = -2 < \phi(S_1) = 0 < \phi(S_2) = 1.68$, which demonstrates Theorem 1 and Corollary 1. Moreover, it is easy to verify that the optimal seed set $S^* = \{v_2, v_3\}$ belongs to $[A^*, B^*]$, which confirms Corollary 2.

Finally, it can be shown that normalizing the benefit and cost weights as described in Section III-B can only increase the amount of the search space cut by our pruning technique.

Theorem 2: Let $\bar{A}^*$ and $\bar{B}^*$ be the node sets returned by Algorithm 1 under the normalized form. Then, $A^* \subseteq \bar{A}^* \subseteq \bar{B}^* \subseteq B^*$.

We shall experimentally evaluate the additional reduction in the search space due to normalization in Section VI.

B. Heuristic Algorithms

We now present some heuristic methods to address the profit maximization problem since there does not exist any polynomial time algorithm with any polynomial time multiplicative approximation guarantees unless P=NP [13].

Greedy Algorithm: We apply a simple hill-climbing idea to optimize the profit function (3). Algorithm 2 describes the pseudo code. In each iteration, the greedy heuristic adds a new node u to S that has the largest marginal profit gain $\phi(u \mid S)$ until all the remaining nodes have negative marginal gains.

Algorithm 2: Greedy

```
1 initialize S ← A*;
2 while True do
3 |   find u ← arg max_{v ∈ B* \ S} {φ(v | S)};
4 |   if φ(u | S) ≤ 0 then return S;
5 |   S ← S ∪ {u};
```

Modular-Modular Algorithm: Iyer and Bilmes [13] introduced a modular-modular (ModMod) algorithm for optimizing the difference between submodular functions. Since we just need to search the lattice $[A^*, B^*]$ after pruning, we adapt the ModMod algorithm as shown in Algorithm 3. In line 4 of Algorithm 3, $h^{\pi}_{X^t}(Y; \beta)$ is a modular lower bound of $\beta(Y)$ that is tight at set X^t, i.e., $h^{\pi}_{X^t}(Y; \beta) \leq \beta(Y)$ for any $Y \subseteq V$ and $h^{\pi}_{X^t}(X^t; \beta) = \beta(X^t)$, while $m_{X^t}(X^t; \gamma)$ is a modular upper bound of $\gamma(Y)$ that is also tight at X^t, i.e., $m_{X^t}(Y; \gamma) \geq \gamma(Y)$ for any $Y \subseteq V$ and $m_{X^t}(X^t; \gamma) = \gamma(X^t)$. Thus, the difference $h^{\pi}_{X^t}(Y; \beta) - m_{X^t}(Y; \gamma)$ is a lower bound of the profit function $\phi(Y)$. The algorithm maximizes the lower bound in each iteration. Since the lower bound is tight at $Y = X^t$, it is guaranteed that $\phi(X^{t+1}) \geq h^{\pi}_{X^t}(X^{t+1}; \beta) - m_{X^t}(X^{t+1}; \gamma) \geq h^{\pi}_{X^t}(X^t; \beta) - m_{X^t}(X^t; \gamma) = \phi(X^t)$. This indicates that Algorithm 3 always increases the profit value at every iteration. Examples of the modular upper and lower bounds will be given in Section V-A.

Algorithm 3: Modular-Modular (ModMod)

initialize $X^0 \leftarrow A^*$ and $t \leftarrow 0$;
repeat
  choose the permutations of A^*, $X^t \setminus A^*$, $B^* \setminus X^t$ and concatenate them as π;
  $X^{t+1} \leftarrow \arg\max_{A^* \subseteq Y \subseteq B^*} h^{\pi}_{X^t}(Y;\beta) - m_{X^t}(Y;\gamma)$;
  $t \leftarrow t+1$;
until *converged, i.e.,* $X^t = X^{t-1}$;
return X^t;

C. Discussions

Time Complexity: Evaluating the profit metric involves estimating the influence spread given a seed set. Any existing influence estimation methods, such as Monte-Carlo simulation [15], [17], [24] and reverse influence sampling [2], [22], [23], [33], [34], can be used. Suppose the time complexity for computing the marginal profit gain of adding/removing a node into/from a seed set is $O(M)$. For the iterative pruning process (Algorithm 1), the size of the node set $B_t \setminus A_t$ to check reduces by at least 1 in each iteration. Therefore, it takes at most $O\big((|V| + |V| - 1 + \cdots + 1)M\big) = O(|V|^2 M)$ time to find A^* and B^*. After the reduction of the search space, there are $k_1 = |B^* \setminus A^*|$ nodes to be further examined. For the Greedy algorithm, it checks $k_1 - i + 1$ nodes in the ith iteration. Thus, it takes at most $O(k_1^2 M)$ time, which means the total time complexity of the Greedy algorithm is $O\big((|V|^2 + k_1^2)M\big)$. Each iteration of the ModMod algorithm has a time complexity of $O(k_1 M)$. Let k_2 denote the total number of iterations used for the ModMod algorithm. Then, the total time complexity of the ModMod algorithm is $O\big((|V|^2 + k_1 k_2)M\big)$.

Diffusion Models: Our analysis and algorithms are general frameworks that can be adapted to any diffusion models which are submodular, such as the Independent Cascade and Linear Threshold models, the triggering model [15], the continuous-time models [5], [10], and the topic-aware models [1], [4].

V. Performance Analysis

The challenges to evaluate the quality of the seed set constructed for the profit maximization problem are two-fold. First, optimizing the difference between two submodular functions is multiplicative inapproximability unless P=NP. Thus, it is difficult to measure the *gap* between the seed set obtained and an optimal seed set. Second, the random processes of many diffusion models are analytically intractable. For example, computing the exact influence spread under the IC diffusion model is #P-hard [6]. Thus, the benefit brought and the cost incurred by a seed set can only be estimated via some sampling approaches [15], [2]. As a result, the *sampling error* also affects the quality measurement of the seed set.

We propose techniques to analyze the aforementioned gap and sampling error, which enable us to evaluate the approximation guarantee of the seed set obtained by any algorithm on any given instance of the profit maximization problem. Specifically, let S^o be the seed set constructed for a problem instance. We develop an upper bound μ on the maximum achievable profit for the problem instance to characterize the gap between the real profit value $\phi(S^o)$ and the maximum achievable profit. Note that both $\phi(S^o)$ and μ are to be estimated by sampling. Let $\tilde{\phi}(S^o)$ and $\tilde{\mu}$ be their estimated values. We further study the sampling errors to bound the difference between $\tilde{\phi}(S^o)$ and $\phi(S^o)$ and the difference between $\tilde{\mu}$ and μ. In this way, we can obtain an approximation guarantee of S^o using the estimated values $\tilde{\phi}(S^o)$ and $\tilde{\mu}$.

A. Upper Bound of Maximum Achievable Profit

To derive our bounds on the maximum achievable profit, we first introduce two modular bounds for submodular functions.

Modular Upper Bounds: For any submodular set function $f(\cdot)$, we have the following two modular upper bounds m^1_X and m^2_X that are tight at a given set X [13]:

$$m^1_X(Y) \triangleq f(X) - \sum_{v \in X \setminus Y} f(v \mid V \setminus \{v\}) + \sum_{v \in Y \setminus X} f(v \mid X), \quad (4)$$

$$m^2_X(Y) \triangleq f(X) - \sum_{v \in X \setminus Y} f(v \mid X \setminus \{v\}) + \sum_{v \in Y \setminus X} f(v \mid \emptyset). \quad (5)$$

In the previous section, we have reduced the search space so that only the sets belonging to $[A^*, B^*]$ need to be considered for profit maximization. As a result, for any $A^* \subseteq X, Y \subseteq B^*$, the above two upper bounds can be improved to:

$$m^3_X(Y) \triangleq f(X) - \sum_{v \in X \setminus Y} f(v \mid B^* \setminus \{v\}) + \sum_{v \in Y \setminus X} f(v \mid X), \quad (6)$$

$$m^4_X(Y) \triangleq f(X) - \sum_{v \in X \setminus Y} f(v \mid X \setminus \{v\}) + \sum_{v \in Y \setminus X} f(v \mid A^*). \quad (7)$$

It is easy to show that the bounds $m^3_X(Y)$ and $m^4_X(Y)$ remain tight at X, i.e., $m^3_X(X) = m^4_X(X) = f(X)$, and they are tighter than $m^1_X(Y)$ and $m^2_X(Y)$ at other sets, i.e., $m^1_X(Y) \geq m^3_X(Y) \geq f(Y)$ and $m^2_X(Y) \geq m^4_X(Y) \geq f(Y)$ for any $A^* \subseteq Y \subseteq B^*$.

Modular Lower Bounds: For any submodular set function $f(\cdot)$, a modular lower bound h_X that is tight at a given set X can be obtained as follows [11]. Let π be any permutation of V that places all the nodes in X before the nodes in $V \setminus X$. Let $S^\pi_i = \{\pi(1), \pi(2), \cdots, \pi(i)\}$ be a chain formed by the permutation, where $S^\pi_0 = \emptyset$ and $S^\pi_{|X|} = X$. Define

$$h^\pi_X(\pi(i)) = f(S^\pi_i) - f(S^\pi_{i-1}). \quad (8)$$

Then, $h^\pi_X(Y) = \sum_{v \in Y} h^\pi_X(v)$ is a lower bound of $f(Y)$, which is tight at X, i.e., $h^\pi_X(Y) \leq f(Y)$ for any $Y \subseteq V$ and $h^\pi_X(X) = f(X)$. After the search space is reduced to $[A^*, B^*]$, we restrict π to any permutation of V in the order of A^*, $X \setminus A^*$ and $B^* \setminus X$.

Upper Bounds on Maximum Achievable Profit: Based on the above bounds, we can derive two series of upper bounds on the maximum value of the profit function $\phi(\cdot)$ as follows. For any set X where $A^* \subseteq X \subseteq B^*$, we define

$$\mu_i(X) \triangleq \max_{A^* \subseteq Y \subseteq B^*} m^i_X(Y;\beta) - h^\pi_X(Y;\gamma), \quad (9)$$

where $m^i_X(Y;\beta)$ $(i = 3, 4)$ denotes the modular upper bound on the benefit function β and $h^\pi_X(Y;\gamma)$ denotes the modular

lower bound on the cost function γ respectively. For briefness, we shall use $\mu(X)$ to refer to either bound with $i = 3$ or 4 in the rest of the paper.

Theorem 3: For any set X where $A^* \subseteq X \subseteq B^*$,

$$\mu(X) \geq \max_{S \subseteq V} \phi(S). \tag{10}$$

The upper bounds established above can be computed very fast since $m_X^i(Y;\beta)$ and $h_X^\pi(Y;\gamma)$ are both modular functions with respect to Y. It is much easier to find the maximum value for a modular function than that for a submodular function. We can obtain upper bounds by arbitrarily choosing the set X in $\mu(X)$. Given a seed set solution S^o obtained by any algorithm, we simply choose $X = S^o$. Then, the approximation guarantee of S^o can be estimated by $\phi(S^o)/\mu(S^o)$.

B. Sampling Error

To estimate $\phi(S^o)$ and $\mu(S^o)$, we make use of a state-of-the-art technique called reverse influence sampling [2], [22].

Definition 1 (RR Set for Weighted Graph): A random reverse reachable (RR) set R for a weighted graph G is generated by (1) first selecting a random node $v \in V$ with a probability distribution $p(\cdot)$ proportional to the node weights, (2) then sampling a graph g randomly from G according to the diffusion model, (3) finally taking the set of nodes in g that can reach v as R.

Taking our benefit metric as an example, the probability for choosing a node v in a random RR set is given by $p(v) = b(v)/\Upsilon_b$, where $\Upsilon_b = b(V) = \sum_{v \in V} b(v)$ is the total benefit weight of all nodes. For a seed set S, the relation between its benefit and a random RR set R is $\beta(S) = \Upsilon_b \cdot \Pr[S \cap R \neq \emptyset]$, where $\Pr[S \cap R \neq \emptyset]$ is the probability that R contains at least one node in S [22]. Similarly, we can estimate the diffusion cost $\gamma(S)$ by choosing each node in a random RR set with a probability proportional to the cost of the node, i.e., $p(v) = c(v)/\Upsilon_c$, where $\Upsilon_c = c(V) = \sum_{v \in V} c(v)$ is the total cost weight of all nodes. Therefore, we generate two groups of RR sets to estimate the benefit and cost respectively for a seed set.

Suppose that we generate a total of θ_β random RR sets to estimate the benefit brought by a seed set S. An RR set is said to be covered by S if it contains at least one node in S. Let $\Lambda_\beta(S)$ denote the number of RR sets covered by S among the θ_β random RR sets. Then, the benefit brought by S can be estimated as $\Lambda_\beta(S) \cdot \Upsilon_b/\theta_\beta$. To analyze the sampling error, we make use of the Chernoff-Hoeffding Theorem [8].

Lemma 1 (Chernoff-Hoeffding Theorem [8]): Let $Z_1, Z_2, \ldots, Z_\theta$ denote random variables that are independently and identically distributed according to Z in the interval $[0, 1]$ with mean $\mathbb{E}[Z]$. For any fixed $\theta > 0$ and $\varepsilon > 0$,

$$\Pr\left[\sum_{i=1}^{\theta} Z_i - \theta \cdot \mathbb{E}[Z] \geq \varepsilon\right] \leq \exp\left(-\frac{\varepsilon^2}{4(e-2)\theta\mathbb{E}[Z]}\right),$$

$$\Pr\left[\sum_{i=1}^{\theta} Z_i - \theta \cdot \mathbb{E}[Z] \leq -\varepsilon\right] \leq \exp\left(-\frac{\varepsilon^2}{4(e-2)\theta\mathbb{E}[Z]}\right).$$

Based on Lemma 1, we can establish the following relation between $\Lambda_\beta(S)$ and the real benefit $\beta(S)$.

Theorem 4: For θ_β random RR sets that are independent of S and any $\delta \in (0, 1)$, we have

$$\Pr\left[\beta(S) \geq \left(\sqrt{\Lambda_\beta(S) + 0.25a} - 0.5\sqrt{a}\right)^2 \cdot \frac{\Upsilon_b}{\theta_\beta}\right] \geq 1 - \frac{\delta}{2},$$

$$\Pr\left[\beta(S) \leq \left(\sqrt{\Lambda_\beta(S) + 0.25a} + 0.5\sqrt{a}\right)^2 \cdot \frac{\Upsilon_b}{\theta_\beta}\right] \geq 1 - \frac{\delta}{2},$$

where $a = 4(e-2)\ln(2/\delta)$.

According to Theorem 4, given $\Lambda_\beta(S)$, we can define a lower bound and an upper bound of $\beta(S)$ with a probability at least $1 - \delta/2$ as

$$\begin{cases} \beta_l(S) \triangleq \left(\sqrt{\Lambda_\beta(S) + 0.25a} - 0.5\sqrt{a}\right)^2 \cdot \Upsilon_b/\theta_\beta, \\ \beta_u(S) \triangleq \left(\sqrt{\Lambda_\beta(S) + 0.25a} + 0.5\sqrt{a}\right)^2 \cdot \Upsilon_b/\theta_\beta. \end{cases} \tag{11}$$

The above analysis can also be applied to the estimation of the cost. Let θ_γ denote the number of random RR sets generated to estimate the cost incurred by a seed set S. Let $\Lambda_\gamma(S)$ denote the number of RR sets covered by S among the θ_γ random RR sets. Then, the cost incurred by S can be estimated as $\Lambda_\gamma(S) \cdot \Upsilon_c/\theta_\gamma$. We can similarly define a lower bound $\gamma_l(S)$ and an upper bound $\gamma_u(S)$ on the real cost $\gamma(S)$ with a probability at least $1 - \delta/2$. Then, given a seed set solution S^o obtained by any algorithm, we can derive a lower bound $\beta_l(S^o) - \gamma_u(S^o)$ on its real profit $\phi(S^o)$ such that $\Pr[\phi(S^o) \geq \beta_l(S^o) - \gamma_u(S^o)] \geq \Pr\left[\big(\beta(S^o) \geq \beta_l(S^o)\big) \wedge \big(\gamma(S^o) \leq \gamma_u(S^o)\big)\right] = 1 - \Pr\left[\big(\beta(S^o) < \beta_l(S^o)\big) \vee \big(\gamma(S^o) > \gamma_u(S^o)\big)\right] \geq 1 - (\delta/2 + \delta/2) > 1 - \delta$.

The difficulty in analyzing the sampling error for the upper bound $\mu(S^o)$ of the maximum achievable profit lies in that we can never find a specific seed set to achieve the upper bound. From Theorem 4, we know that the profit of any seed set S is bounded above by $\beta_u(S) - \gamma_l(S)$ with a high probability. By definition, $\beta_u(S) - \gamma_l(S)$ is a function of $\Lambda_\beta(S)$ and $\Lambda_\gamma(S)$. Let $\tilde{\phi}(S)$ be the estimated profit of S on the RR sets generated, which is defined by $\tilde{\phi}(S) = \Lambda_\beta(S) \cdot \Upsilon_b/\theta_\beta - \Lambda_\gamma(S) \cdot \Upsilon_c/\theta_\gamma$. Then, $\beta_u(S) - \gamma_l(S)$ can be represented as a function η of $\Lambda_\beta(S)$ and $\tilde{\phi}(S)$, i.e., $\eta\big(\Lambda_\beta(S), \tilde{\phi}(S)\big)$. We can prove that $\eta\big(\Lambda_\beta(S), \tilde{\phi}(S)\big)$ is increasing with both $\Lambda_\beta(S)$ and $\tilde{\phi}(S)$. Naturally, $\Lambda_\beta(S)$ is bounded above by θ_β. On the other hand, thanks to the submodularity of the set coverage used to estimate the benefit and cost metrics in the reverse influence sampling approach, we can easily get the estimated upper bound $\tilde{\mu}(S^o)$ on the maximum achievable profit from the RR sets generated according to the analysis in Section V-A. As a result, for any seed set S, the upper bound $\eta\big(\Lambda_\beta(S), \tilde{\phi}(S)\big)$ of its profit is bounded above by $\eta\big(\theta_\beta, \tilde{\mu}(S^o)\big)$. In this way, we can obtain the sampling error for the upper bound $\mu(S^o)$.

Theorem 5: For any $\delta \in (0, 1)$, we have

$$\Pr\left[\max_{S \subseteq V} \phi(S) \leq \tilde{\mu}(S^o) + \varepsilon\big(\tilde{\mu}(S^o)\big)\right] \geq 1 - \delta, \tag{12}$$

where $\varepsilon\big(\tilde{\mu}(S^o)\big)$ is the sampling error for $\tilde{\mu}(S^o)$ such that $\varepsilon\big(\tilde{\mu}(S^o)\big) = \rho_\gamma\sqrt{a\Big(\big(\rho_\beta\theta_\beta - \tilde{\mu}(S^o)\big)/\rho_\gamma + 0.25a\Big)} + 0.5a(\rho_\beta - \rho_\gamma) + \rho_\beta\sqrt{a(\theta_\beta + 0.25a)}$, and $\rho_\beta = \Upsilon_b/\theta_\beta$ and $\rho_\gamma = \Upsilon_c/\theta_\gamma$.

By Theorems 4 and 5, we have the approximation guarantee that

$$\frac{\phi(S^o)}{\max_{S\subseteq V}\phi(S)} \geq \frac{\beta_l(S^o) - \gamma_u(S^o)}{\tilde{\mu}(S^o) + \varepsilon\big(\tilde{\mu}(S^o)\big)} \tag{13}$$

with a probability at least $1 - 2\delta$.

C. Reduce Sampling Error via Normalization

As discussed in Section III-B, we can normalize the benefit and cost weights by $\bar{b}(v)$ and $\bar{c}(v)$ for every node $v \in V$. Intuitively, the normalization can avoid unnecessary samples conducted by the weights $\sum_{v\in V}\min\{b(v), c(v)\}$ for the estimations of both the benefit and cost. Therefore, the normalization can reduce the sampling error by increasing the number of useful samples.

Theorem 6: For any seed set S and a fixed number of samples, let ε_ϕ be the sampling error limit that can provide a probability guarantee of $1-\delta$, i.e., $\Pr[-\varepsilon_\phi \leq \tilde{\phi}(S) - \phi(S) \leq \varepsilon_\phi] \geq 1-\delta$, and let $\bar{\varepsilon}_\phi$ be the sampling error limit under the normalized form. We have $\bar{\varepsilon}_\phi \leq \varepsilon_\phi$.

Theorem 6 indicates that the normalization can improve the solution quality which shall be demonstrated in the experiments.

VI. Evaluation

A. Experimental Setup

Datasets. We use several real social networks available at [18] to evaluate our proposed techniques. Due to space limitations, we report here the results for two representative datasets, Google+ (108K nodes, 14M edges), and LiveJournal (5M nodes, 69M edges).

Algorithms. Recall that the ModMod algorithm needs a modular lower bound of the benefit function $\beta(\cdot)$ and a modular upper bound of the cost function $\gamma(\cdot)$. In Section V-A, we have presented one such lower bound and two such upper bounds. We use ModMod-1 to refer to that using the upper bound $m^3_X(Y)$ defined in Eq. (6) and use ModMod-2 to refer to that using the upper bound $m^4_X(Y)$ defined in Eq. (7). We compare our two-phase methods with the following baselines.

- *Random*: It randomly selects k nodes. We run the algorithm 10 times and take their average as the expected profit.
- *HighDegree*: It selects k nodes with the highest degrees.
- *BenefitMax*: It makes use of the reverse influence sampling technique to find the top-k influential nodes for influence/benefit maximization [2], [22], [23], [33], [34].

The above baselines are executed on the entire social networks without applying any pruning technique. To explore different seed numbers, in each baseline, we iterate through $k = \frac{|V|}{2^i}$ for $i = 0, 1, \ldots, 10$ (where $|V|$ is the network size) and choose the k value producing the largest profit.

Parameter Settings. By default, we use the IC diffusion model (as described in Example 2 of Section IV-A), a uniform benefit distribution (where every node has a unit benefit to model the commission paid by the advertiser for each user activated), and a degree-proportional cost distribution (where the cost of each node is set proportional to its out-degree to emulate the diffusion cost for each activated user to push the product advertisement to all of his neighbors). In the IC model, we set the propagation probability $p_{u,v}$ of each edge $\langle u, v\rangle$ to the reciprocal of v's in-degree (the number of v's inverse neighbors) as widely adopted by other studies [6], [14], [22], [23], [33], [34].

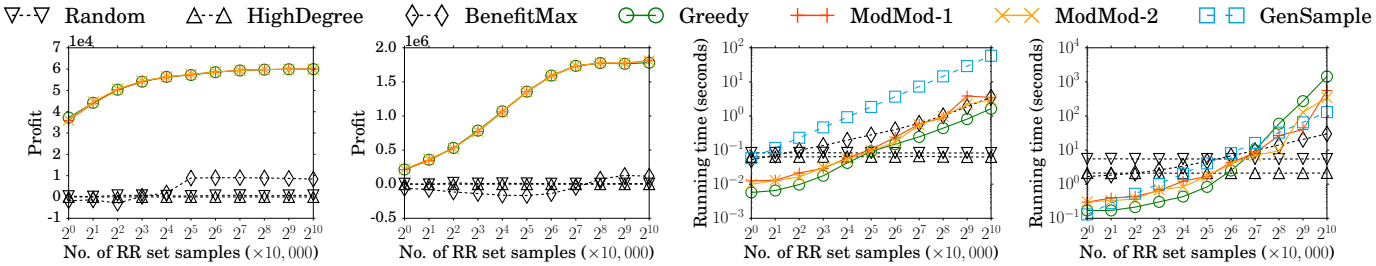
By default, we normalize the benefit and cost weights as described in Section III-B. We use a scale factor r to control the ratio between the total cost and total benefit of all nodes. A higher r implies a higher cost of influence propagation relative to the benefit of influence spread. The default value of r is set to 1. We have tested a wide range of r values and observed similar performance trends. In executing our two-phase methods and the baseline BenefitMax algorithm, we vary the number of RR sets generated to study the impact of benefit and cost estimations. To evaluate the profits of the seed sets returned by different algorithms, we generate a group of validation RR sets to keep the estimation errors within 1% with a high probability at least $1 - 10^{-6}$ according to Theorem 4.

B. Profits Produced by Different Algorithms

Fig. 3 shows the profits produced by different algorithms. Comparing the seed selection algorithms, our heuristic algorithms are more effective in optimizing the profit than the three baseline algorithms (Random, HighDegree and BenefitMax) on the datasets tested. This suggests that improving the influence spread or benefit alone is not effective for maximizing the profit. Our three heuristic algorithms perform quite close in terms of the profit produced. It can also be seen that with increasing number of RR sets generated for profit estimation in the seed selection process, the solution quality of our heuristics is improved due to lower estimation errors.

C. Running Times of Different Algorithms

Fig. 4 shows the running times of different algorithms. The algorithms are all implemented in C++ and the experiments are carried out on a machine with an Intel Xeon E5-1650 3.2GHz CPU and 16GB memory. The time spent for generating RR sets is common to all our heuristics as well as BenefitMax. We plot it as a separate curve and exclude it from the running times of all the algorithms in Fig. 4. It can be seen that generating the RR sets takes significant time. The Random and HighDegree algorithms do not need benefit and cost estimations. Thus, their running times are independent of the number of RR sets. The other four methods have running times increasing almost linearly with the number of RR sets (note that both axes are in logscale). In most cases, our heuristic methods complete execution within 100 seconds even for the LiveJournal dataset with millions of nodes. This shows the efficiency of our two-phase algorithms.

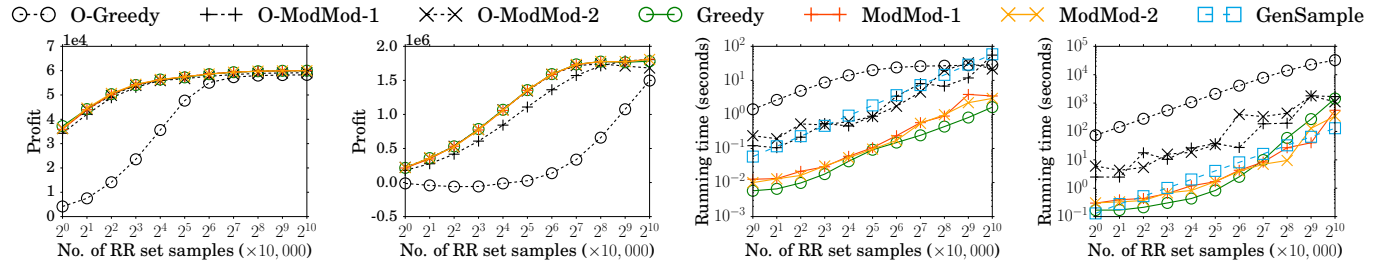


(a) Google+ (b) LiveJournal

Fig. 3. Profits produced by different algorithms.

(a) Google+ (b) LiveJournal

Fig. 4. Running times of different algorithms.



(a) Google+ (b) LiveJournal (c) Google+ (d) LiveJournal

Fig. 5. Impact of iterative pruning technique (algorithms with prefix "O-" do not use pruning technique).

TABLE I
SEARCH SPACE REDUCTION.

Dataset	$\|V\|$	$\|A^*\|$	$\|B^*\|$	$\|B^* \setminus A^*\|$	Reduction
Google+	108K	80.6K	82.4K	1.8K	98.3%
LiveJournal	5M	2.5M	3.3M	0.8M	83.7%

(a) With Normalization

Dataset	$\|V\|$	$\|A^*\|$	$\|B^*\|$	$\|B^* \setminus A^*\|$	Reduction
Google+	108K	62.2K	97.1K	34.9K	67.6%
LiveJournal	5M	1.3M	4.3M	3.0M	38.3%

(b) Without Normalization

TABLE II
APPROXIMATION GUARANTEE OF SEED SETS OBTAINED.

Dataset	Greedy	ModMod-1	ModMod-2
Google+	98.7%	**98.8%**	98.6%
LiveJournal	72.9%	**75.8%**	74.3%

D. Iterative Pruning Technique

Table I shows the amount of search space reduction by the iterative pruning technique proposed in Section IV-A. These results are for the experiments with $2^{10} \times 10,000$ RR sets generated. As can be seen, the pruning technique substantially reduces the number of nodes that need to be considered for seed selection. Specifically, the search space is reduced by 98.3% and 83.7% of the original network size for the Google+ and LiveJournal datasets respectively when the normalization is applied. On the other hand, the reduction is much less when the normalization is not applied (67.6% and 38.3% for the Google+ and LiveJournal datasets respectively). This observation confirms Theorem 2.

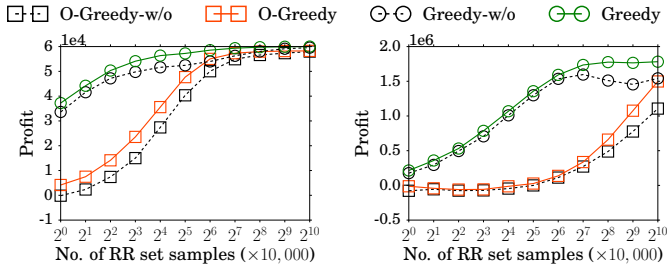
Fig. 5 shows the impact of the search space reduction on the profit produced and the running time of our heuristics, where O-X refers to heuristic X without the search space reduction. As claimed in Corollary 1, the pruning technique can help improve all the heuristic algorithms in terms of the profit produced. This is confirmed by the experimental results. For the LiveJournal dataset (Fig. 5(b)), the Greedy algorithm even produces negative profit without pruning the search space first, and the pruning technique can bring improvements up to 24.1% and 14.3% for the ModMod-1 and ModMod-2 algorithms respectively. Furthermore, the pruning technique can also help reduce the running time considerably. By pruning the search space first, our heuristic algorithms can run up to 3 orders faster. These observations demonstrate the effectiveness of our pruning technique.

E. Guarantee of Solution Quality

We evaluate the quality of the seed sets returned by different algorithms using the techniques presented in Section V. In our evaluation, we always choose the tighter bound between $\mu_3(X)$ and $\mu_4(X)$ as defined in (9). We set $\delta = 10^{-6}$ so that the approximation guarantees obtained by (13) have high confidence. Table II shows the approximation guarantees derived for the seed sets constructed with $2^{10} \times 10,000$ RR sets generated. As can be seen, the seed sets constructed by our algorithms have approximation guarantees above 98% and 70% for the Google+ and LiveJournal datasets respectively. This implies that (i) our proposed upper bounds on the maximum achievable profit are quite tight, and (ii) the heuristic algorithms perform rather close to the optimal.

F. Normalization

In Section VI-D, we have shown that the normalization can increase the amount of search space reduction by the pruning technique. Now, we further evaluate the impact of normalization on the profit. Fig. 6 shows the profit produced by the Greedy algorithm with and without the normalization. The



(a) Google+ (b) LiveJournal

Fig. 6. Impact of normalization (algorithms with prefix "O-" do not use pruning technique and with postfix "-w/o" are without normalization).

results for the ModMod algorithms are similar. As can be seen, no matter whether the pruning technique is used, the Greedy algorithm with normalization can always produce higher profit than that without normalization. This confirms Theorem 6 that the normalization can reduce the sampling error limit and thus, it can further improve the solution quality.

VII. Conclusion

In this paper, we have studied a profit maximization problem for OSN providers conducting viral marketing. The objective is to select initial seed nodes to maximize the total profit that accounts for both the benefit of influence spread and the cost of influence propagation. We have proposed a two-phase framework that first reduces the search space via an iterative pruning technique and then finds the solution via some heuristic algorithms. We have presented several bounds to measure the quality of the solution obtained by any algorithm. Experimental results with real OSN datasets demonstrate the effectiveness and efficiency of our techniques, and show the tightness of our derived upper bounds.

Acknowledgment

This research is supported 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.

References

[1] N. Barbieri, F. Bonchi, and G. Manco, "Topic-aware social influence propagation models," in *Proc. IEEE ICDM*, 2012, pp. 81–90.

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

[3] P. Chalermsook, A. Das Sarma, A. Lall, and D. Nanongkai, "Social network monetization via sponsored viral marketing," in *Proc. ACM SIGMETRICS*, 2015, pp. 259–270.

[4] S. Chen, J. Fan, G. Li, J. Feng, K.-L. Tan, and J. Tang, "Online topic-aware influence maximization," *Proc. VLDB Endowment*, vol. 8, no. 6, pp. 666–677, 2015.

[5] W. Chen, W. Lu, and N. Zhang, "Time-critical influence maximization in social networks with time-delayed diffusion process," in *Proc. AAAI*, 2012, pp. 592–598.

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

[7] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proc. ACM KDD*, 2009, pp. 199–208.

[8] P. Dagum, R. Karp, M. Luby, and S. Ross, "An optimal algorithm for monte carlo estimation," *SIAM Journal on Computing*, vol. 29, no. 5, pp. 1484–1496, 2000.

[9] P. Domingos and M. Richardson, "Mining the network value of customers," in *Proc. ACM KDD*, 2001, pp. 57–66.

[10] N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha, "Scalable influence estimation in continuous-time diffusion networks," in *Proc. NIPS*, 2013, pp. 3147–3155.

[11] S. Fujishige, *Submodular Functions and Optimization*. Elsevier Science, 2005, vol. 58.

[12] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan, "A data-based approach to social influence maximization," *Proc. VLDB Endowment*, vol. 5, no. 1, pp. 73–84, 2011.

[13] R. Iyer and J. Bilmes, "Algorithms for approximate minimization of the difference between submodular functions, with applications," in *Proc. UAI*, 2012, pp. 407–417.

[14] K. Jung, W. Heo, and W. Chen, "IRIE: Scalable and robust influence maximization in social networks," in *Proc. IEEE ICDM*, 2012, pp. 918–923.

[15] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proc. ACM KDD*, 2003, pp. 137–146.

[16] A. Khan, B. Zehnder, and D. Kossmann, "Revenue maximization by viral marketing: A social network host's perspective," in *Proc. IEEE ICDE*, 2016.

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

[18] J. Leskovec and A. Krevl, "SNAP Datasets: Stanford large network dataset collection," http://snap.stanford.edu/data, 2014.

[19] W. Lu and L. V. Lakshmanan, "Profit maximization over social networks," in *Proc. IEEE ICDM*, 2012, pp. 479–488.

[20] A. Meyer, "Viral video marketing: What's, why's & how's of going viral," http://www.marketergizmo.com/viral-video-marketing-cats-babies-and-your-company/, 2015.

[21] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher, "An analysis of approximations for maximizing submodular set functions-I," *Mathematical Programming*, vol. 14, no. 1, pp. 265–294, 1978.

[22] H. T. Nguyen, T. N. Dinh, and M. T. Thai, "Cost-aware targeted viral marketing in billion-scale networks," in *Proc. IEEE INFOCOM*, 2016.

[23] H. T. Nguyen, M. T. Thai, and T. N. Dinh, "Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks," in *Proc. ACM SIGMOD*, 2016, pp. 695–710.

[24] N. Ohsaka, T. Akiba, Y. Yoshida, and K. Kawarabayashi, "Fast and accurate influence maximization on large networks with pruned monte-carlo simulations," in *Proc. AAAI*, 2014, pp. 138–144.

[25] J. J. Roberts, "Facebook and Google are big winners as political ad money moves online," *Fortune*, 2016.

[26] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf, "Uncovering the temporal dynamics of diffusion networks," in *Proc. ICML*, 2011, pp. 561–568.

[27] G. Song, X. Zhou, Y. Wang, and K. Xie, "Influence maximization on large-scale mobile social network: A divide-and-conquer method," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1379–1392, 2015.

[28] J. Tang, X. Tang, X. Xiao, and J. Yuan, "Online processing algorithms for influence maximization," in *Proc. ACM SIGMOD*, 2018.

[29] J. Tang, X. Tang, and J. Yuan, "Profit maximization for viral marketing in online social networks," in *Proc. IEEE ICNP*, 2016, pp. 1–10.

[30] J. Tang, X. Tang, and J. Yuan, "Influence maximization meets efficiency and effectiveness: A hop-based approach," in *Proc. IEEE/ACM ASONAM*, 2017, pp. 64–71.

[31] J. Tang, X. Tang, and J. Yuan, "Profit maximization for viral marketing in online social networks: Algorithms and analysis," *IEEE Transactions on Knowledge and Data Engineering*, 2017.

[32] J. Tang, X. Tang, and J. Yuan, "Towards profit maximization for online social network providers," arXiv preprint, https://arxiv.org/abs/1712.08963, 2017.

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

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

[35] C. Zhou, P. Zhang, J. Guo, X. Zhu, and L. Guo, "UBLF: An upper bound based approach to discover influential nodes in social networks," in *Proc. IEEE ICDM*, 2013, pp. 907–916.

[36] Y. Zhu, Z. Lu, Y. Bi, W. Wu, Y. Jiang, and D. Li, "Influence and profit: Two sides of the coin," in *Proc. IEEE ICDM*, 2013, pp. 1301–1306.