Pricing Influential Nodes in Online Social Networks

Yuqing Zhu
School of Comp. Sci. & Engg.
Nanyang Technological Uni.
yuqing002@e.ntu.edu.sg

Jing Tang*
Department of ISEM
National Uni. of Singapore
isejtang@nus.edu.sg

Xueyan Tang
School of Comp. Sci. & Engg.
Nanyang Technological Uni.
asxytang@ntu.edu.sg

ABSTRACT

Influential nodes with rich connections in online social networks (OSNs) are of great values to initiate marketing campaigns. However, the potential influence spread that can be generated by these influential nodes is hidden behind the structures of OSNs, which are often held by OSN providers and unavailable to advertisers for privacy concerns. A social advertising model known as influencer marketing is to have OSN providers offer and price candidate nodes for advertisers to purchase for seeding marketing campaigns. In this setting, a reasonable price profile for the candidate nodes should effectively reflect the expected influence gain they can bring in a marketing campaign.

In this paper, we study the problem of pricing the influential nodes based on their expected influence spread to help advertisers select the initiators of marketing campaigns without the knowledge of OSN structures. We design a function characterizing the divergence between the price and the expected influence of the initiator sets. We formulate the problem to minimize the divergence and derive an optimal price profile. An advanced algorithm is developed to estimate the price profile with accuracy guarantees. Experiments with real OSN datasets show that our pricing algorithm can significantly outperform other baselines.

PVLDB Reference Format:
Yuqing Zhu, Jing Tang, and Xueyan Tang. Pricing Influential Nodes in Online Social Networks. *PVLDB*, 13(10): 1614–1627, 2020.
DOI: https://doi.org/10.14778/3401960.3401961

1. INTRODUCTION

Online Social Networks (OSNs) attract billions of users to share information and bring new approaches to promote product sales or activity engagement. Real-world examples of web-based social networks include Facebook, Twitter, Orkut, etc. According to Facebook's official statistics [17], it has 2.13 billion monthly active users as of December 31, 2017. Given the tremendous number of active users, information can be propagated widely and rapidly through OSNs with the *word of mouth* effects. The interpersonal connections between individuals can strongly impact their decisions and behaviors. Applications like social advertising naturally emerge to make use of OSNs for information diffusion [12]. Nowadays, the advertisement market in OSNs is growing at an amazing speed. For example, eMarketer [13] estimates that advertisers are expected to spend \$35.98 billion on social media to promote their products. Fortune [30] claims that the expenditure of advertisement on social media will exceed traditional newspapers by 2020, which will be over \$50 billion.

In online social advertising, some influencers accept free products as rewards for running a marketing campaign. The well-known influence maximization problem emerges from giving a limited number of free samples to a subset of individuals to trigger a cascade of influence [23]. Meanwhile, some influencers charge a certain amount of money from advertisers. According to a survey conducted by an influencer platform named Klear, brands make an average payment of \$114 per video post on Instagram to nano-influencers who have between 500 and 5,000 followers, and \$775 to power users with followings between 30,000 and 50,000 for an Instagram video [15]. The rising cost for doing online advertising attracts investment in influencers (called influencer marketing), which often involves buying a list of influencer contracts and paying them to promote a product [14]. In 2016, an online celebrity named Papi Jiang with 10 million fans on Weibo, a Twitter-like micro blogging site, was valued at around 42 million dollars (300 million RMB) and received 2 million dollars investment for her potential market value without selling anything yet. According to a survey conducted by Influencer Marketing Hub, the majority of advertisers see influencer marketing as a direction and plan to increase their influencer marketing budget [22].

In practice, the social graph is normally possessed by OSN providers and kept secret for privacy reasons. Hence, it is difficult for advertisers to infer the values of influencers—only 39% of US marketers feel confident in identifying right influencers according to a Cision and PRWeek survey [16]. It remains an open question how to set reasonable prices on the influencers for their market values. Different from the advertisers, the OSN providers hold the structure of the networks and can identify reliable influencers and leverage data to set the marketing price properly. In other words, the OSN providers can offer the prices of the influencers to the advertisers. In fact, the OSN providers have started

*Corresponding author: Jing Tang.

Proceedings of the VLDB Endowment, Vol. 13, No. 10
ISSN 2150-8097.
DOI: https://doi.org/10.14778/3401960.3401961

setting up platforms to facilitate the influencer search and selection process, as well as making the system more transparent and easier for both advertisers and influencers [22]. Recently, YouTube offers access to the set of influencers on the platform FameBit [18]. On average, hiring an influencer costs $20 a video per 1,000 subscribers.

Intuitively, the price of seeding any set of users should effectively reflect the expected influence spread that these users can generate in the campaigns. In this way, the advertisers can hold a clear view over the influence potential of the seeds selected and make more sensible business decisions. There are several intuitive ways to price users or nodes in OSNs. A simple strategy is to set the price of each node based on its degree in the OSN. This strategy is rather primitive since the degree of a node is not necessarily proportional to its actual influence spread. Another intuitive strategy is to set the price of each node according to its expected influence spread when selected as the only seed. However, when multiple nodes are selected to seed a campaign, the influence spreads generated by different nodes may overlap substantially. Thus, the influence spreads of singleton seeds may not effectively reflect their influence contributions when they jointly initiate a campaign. A straightforward solution to precisely describe the influence contributions of the nodes in various seed sets is to derive a separate price of each node for including in each possible seed set. This method is unfortunately computationally expensive to implement due to the huge numbers of nodes and possible seed sets in real social networks.

In this paper, we propose a new pricing strategy that can effectively reflect the value of the nodes in any seed set. We define a function to measure the difference of the price of a randomly chosen seed set from its expected marketing value and formulate an optimization problem of pricing the nodes to minimize the difference. The optimization problem is challenging to solve in several aspects. First, in order to narrow down the divergence between the price and the expected influence, we need to calculate the expected influence spread of a seed set, which is #P-hard even for simple diffusion models [5, 7]. Second, as the number of possible seed sets grows exponentially with the number of candidate nodes offered by the OSN provider, it is computationally intractable to compute the expected influence spread for all possible seed sets. Furthermore, the nodes may have different contributions to the influence spreads of different seed sets, which makes it even more difficult to set a reasonable price for each node.

To tackle the pricing problem, we make the following major contributions in this paper:

- We propose a novel problem domain of pricing the nodes based on their expected influence spread to help advertisers select the initiators of marketing campaigns.
- We design a function to characterize the divergence between the price and the expected influence of seed sets and formulate and solve an optimization problem to minimize the divergence.
- We devise an efficient algorithm based on random reverse reachable sets [3] to compute the prices for the nodes. An advanced estimation algorithm is also developed to ensure that the estimated prices have accuracy guarantees.
- Extensive experiments based on real OSN datasets confirm that our pricing algorithm can yield high quality solutions and significantly outperform other baselines.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 introduces the preliminaries. Section 4 presents the pricing problem and the solution. Section 5 elaborates the algorithm to estimate the node prices. Section 6 describes the experimental evaluation. Finally, Section 7 concludes the paper.

2. RELATED WORK

Domingos and Richardson [31] were the first to study viral marketing as an algorithmic problem. They proposed approximation algorithms to determine the influential users and demonstrated that different sets of seed users in a marketing campaign can produce substantially different influence spreads. Kempe et al. [23] showed that the optimization problem of selecting the most influential seed set of a given size is NP-hard. They showed that the influence function is a submodular function under the Independent Cascade and Linear Threshold diffusion models [23]. They proposed a $(1 - 1/e)$-approximation greedy algorithm utilizing Monte-Carlo simulations. Follow-up work has mostly focused on improving the efficiency of the algorithm implementation for large-scale OSNs based on the submodularity property or heuristics [3, 5–7, 10, 26, 29, 33, 35, 36, 39, 40]. There was also work studying profit maximization in OSNs to optimize the profit return of viral marketing. Lu et al. [27] extended the classical Linear Threshold model to incorporate product prices and user valuations, and factor them into the user's decision process of adopting a product. They greedily chose the seeds with the greatest profit potential. Tang et al. [34, 37] combined the benefit of influence spread with the cost of seed selection for profit maximization. Furthermore, some recent work utilized adaptive algorithms to improve the performance for variants of influence based optimization problems, including adaptive influence maximization [19, 20], adaptive seed minimization [32], and adaptive profit maximization [21], etc. All the above seed selection methods were designed based on the premise that complete social network structures are available to advertisers. However, such information is normally kept secret by OSN providers for business and privacy reasons [4, 24]. Recently, Tang et al. [38] studied the profit maximization problem from the OSN provider's perspective by taking the cost of information diffusion over the social network into account. In addition, Aslay et al. proposed and tackled other practical problems of regret minimization [2] and revenue maximization [1] in online social advertising. Nevertheless, they did not provide any value-based seed pricing solution for advertisers. Different from all the above studies, in this paper, we aim to tighten the relationship between the price setting and the seed's influence spread.

3. PRELIMINARIES

3.1 Influence Spread

An OSN can be modeled as a directed graph $G = (V, E)$ with a set V of nodes and a set E of edges. Users are represented by the nodes and connections between users are represented by edges. For each edge $(u, v) \in E$, we say that v is an *out-neighbor* of u and u is an *in-neighbor* of v.

Many models have been proposed to capture the diffusion process in the OSN. Our problem definition and solution are general and can be used for various diffusion models. For the purpose of illustrating basic concepts, we briefly introduce a widely used diffusion model known as *Independent Cascade* [23]. In this model, each edge (u, v) is associated with a propagation probability $p_{u,v}$ denoting the probability that v will be influenced by u. Initially, a set of seed nodes S are activated while all the other nodes are inactive. When a node u first becomes active, it is given a single chance to activate each inactive out-neighbor v with a probability $p_{u,v}$. The diffusion process stops when no more activation can be made.

Let $\sigma(S)$ denote the expected number of nodes activated by the diffusion process starting with a set of seed nodes S. $\sigma(S)$ is known as the *influence spread* of the seed set S.

3.2 Influence Spread Estimation

The computational complexity of the exact influence spread $\sigma(S)$ for a seed set S is proved to be #P-hard for several diffusion models including *Independent Cascade* [5, 7]. As a result, various sampling methods have been proposed for unbiased estimation of the influence spread. The RIS method proposed by Borgs et al. [3] substantially improved the efficiency to estimate the influence spread compared to the naive Monte-Carlo simulation method. Thus, we adopt RIS for influence spread estimation.

Definition 1 ([3]). *A random reverse reachable (RR) set R for a graph G is generated by the following steps:*

1. *Select a random node $v \in V$.*
2. *Sample a random graph g from G according to the diffusion model.*
3. *Take the set of nodes in g that can reach v as R.*

For example, under the Independent Cascade model, a random RR set R on G can be constructed as follows:

1. Select a node $v \in V$ uniformly at random.
2. Starting from v, perform a stochastic breadth first search (BFS) following the incoming edges of each node. Specifically, for each node u encountered in the BFS, we examine the in-neighbors of u. For each in-neighbor w, we allow the BFS to traverse to w from u with probability $p_{w,u}$ (if w has not been traversed before).
3. Insert all the nodes traversed during the stochastic BFS into the RR set R.

Random RR sets have the following property [3].

Lemma 1 ([3]). *Given a seed set $S \subseteq V$, for a random RR set R, we have*

$$\sigma(S) = n \cdot \Pr[S \cap R \neq \emptyset], \tag{1}$$

where $n = |V|$ is the total number of nodes in the graph G.

According to Lemma 1, the influence spread of a seed set S is proportional to the probability that S intersects with a random RR set R. Thus, to estimate influence spread, we can generate a large number of RR sets $\mathcal{R}$. Given any seed set S, we can compute the number of RR sets in $\mathcal{R}$ that intersect with S (denoted by $\Lambda(\mathcal{R}, S)$) and estimate the influence spread of S by $\frac{n}{|\mathcal{R}|} \cdot \Lambda(\mathcal{R}, S)$.

4. DIVERGENCE FUNCTION & OPTIMAL PRICE PROFILE

In this section, we define a divergence function to measure the effectiveness of a pricing profile and derive an optimal pricing profile in terms of the function value.

4.1 Divergence Function

The seed users generate revenue from the seed purchase of the advertiser for initiating the campaigns. The influence spread, on the other hand, is the reward gained by the advertiser in the campaigns. Thus, it is important to make sure that the influence spread is worth the cost of seed purchase. In this way, the prices set for seed purchase can not only minimize the regret in deriving the revenue for seeds but also give the advertiser a more predictable return for its purchase. Therefore, our objective of pricing is to match the price of any seed set with the expected marketing value of the seed set as closely as possible.

Consider a candidate node set C consisting of n_c nodes $\{s_1, s_2, \ldots, s_{n_c}\}$ offered by the OSN provider for the advertisers to choose seeds. Let p_i be the price of the node s_i in C. We refer to $\langle p_1, p_2, \ldots, p_{n_c}\rangle$ as the *price profile*. For any seed set $S \subseteq C$, the total price of the nodes in S is $\sum_{s_i \in S} p_i$, and the influence spread of S is $\sigma(S)$. Let c be a constant representing the expected revenue to derive from influencing a node. Then, $c \cdot \sigma(S)$ is the expected market value for the seed set S. Thus, the divergence between the price and the influence spread can be characterized by $(c \cdot \sigma(S) - \sum_{s_i \in S} p_i)^2$. Since the advertisers can choose any subset of the nodes in C to initiate campaigns based on their preferences, we assume that all the subsets of C are equally likely to be chosen as the seed set. Therefore, the expected divergence between the price and the influence spread of a randomly chosen seed set is given by

$$\frac{1}{2^{n_c}} \sum_{S \subseteq C} \Big(c \cdot \sigma(S) - \sum_{s_i \in S} p_i\Big)^2. \tag{2}$$

We aim to find a price profile to minimize the divergence function such that the total price of all nodes in the candidate set C is equal to a given value b, i.e.,

$$\begin{aligned} \min \quad & \frac{1}{2^{n_c}} \sum_{S \subseteq C} \Big(c \cdot \sigma(S) - \sum_{s_i \in S} p_i\Big)^2, \\ \text{s.t.} \quad & \sum_{i=1}^{n_c} p_i = b, \ \text{and } \forall i\colon p_i \geq 0. \end{aligned}$$

This problem is equivalent to

$$\min \ \frac{1}{2^{n_c}} \sum_{S \subseteq C} \Big(\sigma(S) - \sum_{s_i \in S} \frac{p_i}{c}\Big)^2 \ \text{subject to} \ \sum_{i=1}^{n_c} \frac{p_i}{c} = \frac{b}{c}.$$

Here, $\frac{p_i}{c}$ and $\frac{b}{c}$ can be understood as the individual node price and the total node price when $c = 1$. To simplify the presentation and derivation, in the rest of this paper, we shall focus on addressing the following optimization problem assuming $c = 1$:

$$\begin{aligned} \min \quad & f(p_1, p_2, \ldots, p_{n_c}) := \frac{1}{2^{n_c}} \sum_{S \subseteq C} \Big(\sigma(S) - \sum_{s_i \in S} p_i\Big)^2, \\ \text{s.t.} \quad & \sum_{i=1}^{n_c} p_i = b, \ \text{and } \forall i\colon p_i \geq 0. \end{aligned}$$

It is easy to verify that the optimal price profile for a general c value can be obtained by simply scaling the solution to the above problem.

4.2 Optimal Price Profile

To solve the pricing problem, we first reformulate the divergence function.

LEMMA 2. *Let*

$$g(p_i) := \frac{p_i^2 + p_i b}{4} - \frac{p_i}{2^{n_c - 1}} \sum_{S \subseteq C \setminus \{s_i\}} \sigma(S \cup \{s_i\}). \quad (3)$$

Then, we have

$$f(p_1, p_2, \ldots, p_{n_c}) = \frac{1}{2^{n_c}} \sum_{S \subseteq C} \sigma(S)^2 + \sum_{i=1}^{n_c} g(p_i). \quad (4)$$

PROOF. We prove it by induction. When $n_c = 1$, we have

$$g(p_1) = \frac{p_1^2 + p_1 b}{4} - p_1 \sigma(\{s_1\}) = \frac{p_i^2}{2} - p_1 \sigma(\{s_1\}).$$

Meanwhile, by definition, we have

$$f(p_1) = \frac{(\sigma(\{s_1\}) - p_1)^2}{2} = \frac{\sigma(\{s_1\})^2}{2} + \frac{p_i^2}{2} - p_1 \sigma(\{s_1\}).$$

Thus, $f(p_1) = \frac{\sigma(\{s_1\})^2}{2} + g(p_1)$, which indicates that (4) holds when $n_c = 1$.

Suppose that (4) holds when $n_c = N$ for an integer $N \geq 1$. In what follows, we will show that (4) holds when $n_c = N+1$.

For any $i \in [2, n_c]$, let $\bar{b} = b - p_1$, $\bar{\sigma}(S) = \sigma(S \cup \{s_1\}) - p_1$,

$$\bar{g}(p_i) := \frac{p_i^2 + p_i \bar{b}}{4} - \frac{p_i}{2^{N-1}} \sum_{S \subseteq C \setminus \{s_1, s_i\}} \bar{\sigma}(S \cup \{s_i\}),$$

$$\text{and} \quad \hat{g}(p_i) := \frac{p_i^2 + p_i \bar{b}}{4} - \frac{p_i}{2^{N-1}} \sum_{S \subseteq C \setminus \{s_1, s_i\}} \sigma(S \cup \{s_i\}).$$

In addition, let

$$\bar{f} := \frac{1}{2^N} \sum_{S \subseteq C \setminus \{s_1\}} \left(\Big(\bar{\sigma}(S) - \sum_{s_i \in S} p_i \Big)^2 + \Big(\sigma(S) - \sum_{s_i \in S} p_i \Big)^2 \right).$$

For any node set S, the node s_1 satisfies either $s_1 \in S$ or $s_1 \notin S$. Thus, we have $\bar{f} = 2f(p_1, p_2, \ldots, p_{n_c})$.

We observe that $\sum_{i=2}^{n_c} p_i = \bar{b}$ and $|C \setminus \{s_1\}| = n_c - 1 = N$. According to the hypothesis, we have

$$\bar{f} = \frac{1}{2^N} \sum_{S \subseteq C \setminus \{s_1\}} \left(\bar{\sigma}(S)^2 + \sigma(S)^2 \right) + \sum_{i=2}^{n_c} \left(\bar{g}(p_i) + \hat{g}(p_i) \right).$$

For the first part, we have

$$\begin{aligned}
&\frac{1}{2^N} \sum_{S \subseteq C \setminus \{s_1\}} \left(\bar{\sigma}(S)^2 + \sigma(S)^2 \right) \\
&= \frac{1}{2^N} \sum_{S \subseteq C \setminus \{s_1\}} \Big(\left(\sigma(S \cup \{s_1\}) - p_1 \right)^2 + \sigma(S)^2 \Big) \\
&= \frac{1}{2^N} \sum_{S \subseteq C} \sigma(S)^2 + p_1^2 - \frac{p_1}{2^{N-1}} \sum_{S \subseteq C \setminus \{s_1\}} \sigma(S \cup \{s_1\}) \\
&= \frac{1}{2^N} \sum_{S \subseteq C} \sigma(S)^2 + 2g(p_1) + \frac{p_1^2 - p_1 b}{2}.
\end{aligned}$$

For the second part, we have

$$\begin{aligned}
&\bar{g}(p_i) + \hat{g}(p_i) \\
&= \frac{p_i^2 + p_i \bar{b}}{2} - \frac{p_i}{2^{N-1}} \sum_{S \subseteq C \setminus \{s_1, s_i\}} \Big(\bar{\sigma}(S \cup \{s_i\}) + \sigma(S \cup \{s_i\}) \Big) \\
&= \frac{p_i^2 + p_i \bar{b}}{2} - \frac{p_i}{2^{N-1}} \sum_{S \subseteq C \setminus \{s_i\}} \sigma(S \cup \{s_i\}) + p_1 p_i \\
&= \frac{p_i^2 + p_i b}{2} - \frac{p_i}{2^{N-1}} \sum_{S \subseteq C \setminus \{s_i\}} \sigma(S \cup \{s_i\}) + \frac{p_1 p_i}{2} \\
&= 2g(p_i) + \frac{p_1 p_i}{2}.
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
\bar{f} &= \frac{1}{2^N} \sum_{S \subseteq C} \sigma(S)^2 + 2g(p_1) + \frac{p_1^2 - p_1 b}{2} \\
&\quad + \sum_{i=2}^{n_c} \Big(2g(p_i) + \frac{p_1 p_i}{2} \Big) \\
&= \frac{1}{2^N} \sum_{S \subseteq C} \sigma(S)^2 + 2 \sum_{i=1}^{n_c} g(p_i).
\end{aligned}$$

This completes the proof. □

The following theorem gives the optimal price profile.

THEOREM 1. *Let*

$$c_i := \frac{b}{2} - \frac{\sum_{S \subseteq C \setminus \{s_i\}} \sigma(S \cup \{s_i\})}{2^{n_c - 2}}, \ \forall 1 \leq i \leq n_c, \quad (5)$$

$$\lambda \text{ be the root of } \sum_{i=1}^{n_c} \max\{0, \lambda - c_i\} = b, \quad (6)$$

$$\text{and } p_i^* = \max\{0, \lambda - c_i\}, \ \forall 1 \leq i \leq n_c. \quad (7)$$

Then, the price profile $\langle p_1^*, p_2^*, \ldots, p_{n_c}^* \rangle$ *minimizes the divergence function.*

PROOF. According to Karush-Kuhn-Tucker conditions, the optimal solution $\langle p_1^*, p_2^*, \ldots, p_{n_c}^* \rangle$ satisfies that

$$\begin{aligned}
\forall i \colon \ &\frac{p_i^*}{2} + \frac{b}{4} - \frac{\sum_{S \subseteq C \setminus \{s_i\}} \sigma(S \cup \{s_i\})}{2^{n_c - 1}} - \frac{\lambda}{2} - \lambda_i \\
&= \frac{p_i^*}{2} + \frac{c_i}{2} - \frac{\lambda}{2} - \lambda_i = 0, \quad (8)
\end{aligned}$$

$$\forall i \colon \lambda_i p_i^* = 0, \quad (9)$$

$$\sum_{i=1}^{n_c} p_i^* = b, \quad (10)$$

$$\forall i \colon p_i^* \geq 0, \quad (11)$$

$$\forall i \colon \lambda_i \geq 0. \quad (12)$$

In the above, (8) represents stationarity, (9) shows complementary slackness, (10) and (11) ensure primal feasibility, and (12) ensures dual feasibility.

If $\lambda \leq c_i$, by (8), we have $\frac{p_i^*}{2} \leq \lambda_i$. By (9), (11), and (12), we have $p_i^* = 0$. Similarly, if $\lambda > c_i$, we have $\lambda_i = 0$, which indicates that $p_i^* = \lambda - c_i$. Therefore,

$$p_i^* = \max\{0, \lambda - c_i\}. \quad (13)$$

Then, by (10), λ is the solution for $\sum_{i=1}^{n_c} \max\{0, \lambda - c_i\} = b$, which completes the proof. □

Theorem 1 states the optimal solution, where λ can be obtained via water-filling. Specifically, without loss of generality, we assume that $c_0 = +\infty > c_1 \geq c_2 \geq \cdots \geq c_{n_c}$. Then, we can find a unique $j \in [0, n_c - 1]$ such that $\sum_{i=j+1}^{n_c}(c_j - c_i) > b$ and $\sum_{i=j+2}^{n_c}(c_{j+1} - c_i) \leq b$. Then, λ is the root of $\sum_{i=j+1}^{n_c}(\lambda - c_i) = b$.

Corollary 1. *Let*

$$p_i^* = \frac{b}{n_c} + \frac{\sum_{S\subseteq C}\sigma(S)\big(\mathbf{1}_{\{s_i\in S\}} - \frac{|S|}{n_c}\big)}{2^{n_c-2}}, \ \forall 1 \leq i \leq n_c, \quad (14)$$

where $\mathbf{1}_{\{s_i\in S\}}$ is a binary value such that $\mathbf{1}_{\{s_i\in S\}} = 1$ if $s_i \in S$ and $\mathbf{1}_{\{s_i\in S\}} = 0$ otherwise. Let $\sigma(C)$ denote the influence spread of the candidate set C. If $b \geq \sigma(C)$, the price profile $\langle p_1^*, p_2^*, \ldots, p_{n_c}^*\rangle$ minimizes the divergence function.

Proof. Let λ be the value such that

$$\sum_{i=1}^{n_c}(\lambda - c_i) = b.$$

Then,

$$\lambda = \frac{b}{n_c} + \frac{1}{n_c}\sum_{i=1}^{n_c} c_i = \frac{b}{n_c} + \frac{b}{2} - \frac{\sum_{S\subseteq C}\big(\sigma(S)\cdot|S|\big)}{n_c \cdot 2^{n_c-2}},$$

and

$$\lambda - c_i = \frac{b}{n_c} + \frac{\sum_{S\subseteq C}\sigma(S)\big(\mathbf{1}_{\{s_i\in S\}} - \frac{|S|}{n_c}\big)}{2^{n_c-2}}.$$

Next, we utilize the RIS method [3] to show that $\lambda - c_i \geq 0$ to ensure that p_i^* is non-negative ($1 \leq i \leq n_c$) when $b \geq \sigma(C)$.

Denote $A_i = \lambda - c_i - \frac{b}{n_c} = \frac{\sum_{S\subseteq C}\sigma(S)\big(\mathbf{1}_{\{s_i\in S\}} - \frac{|S|}{n_c}\big)}{2^{n_c-2}}$ and $B_i = A_i + \frac{b}{n_c} = \lambda - c_i$. Let $\tilde{A}_i$ be the estimation of A_i and $\tilde{B}_i$ be the estimation of B_i using RR sets. According to Lemma 1, when θ RR sets $\mathcal{R}$ are generated for estimation, an RR set R contributes to $\tilde{A}_i$ by an additive factor of

$$\begin{aligned}&\Delta(\tilde{A}_i, R)\\ &= \frac{n}{\theta}\cdot\frac{\sum\limits_{S\subseteq C}\Big(\mathbf{1}_{\{S\cap R\neq\emptyset\}}\cdot\big(\mathbf{1}_{\{s_i\in S\}} - \frac{|S|}{n_c}\big)\Big)}{2^{n_c-2}}\\ &= \frac{n}{\theta}\cdot\frac{\sum\limits_{S\subseteq C\backslash\{s_i\}}\mathbf{1}_{\{(S\cup\{s_i\})\cap R\neq\emptyset\}} - \sum\limits_{S\subseteq C}\big(\mathbf{1}_{\{S\cap R\neq\emptyset\}}\cdot\frac{|S|}{n_c}\big)}{2^{n_c-2}}.\end{aligned}$$

If $s_i \in R$, we have $\mathbf{1}_{\{(S\cup\{s_i\})\cap R\neq\emptyset\}} = 1$ for every $S \in C\backslash\{s_i\}$, where the number of such S is 2^{n_c-1}. If $s_i \notin R$, for any $S \in C\setminus\{s_i\}$, we have $\mathbf{1}_{\{(S\cup\{s_i\})\cap R\neq\emptyset\}} = 1$ if and only if $S\cap R \neq \emptyset$, where the number of such S is $2^{n_c-1} - 2^{n_c-1-n_r}$, and $n_r = |R\cap C|$. Thus,

$$\sum_{S\subseteq C\backslash\{s_i\}}\mathbf{1}_{\{(S\cup\{s_i\})\cap R\neq\emptyset\}} = \begin{cases}2^{n_c-1}, & \text{if } s_i \in R,\\ 2^{n_c-1} - 2^{n_c-1-n_r}, & \text{otherwise.}\end{cases}$$

Meanwhile,

$$\begin{aligned}&\sum_{S\subseteq C}\Big(\mathbf{1}_{\{S\cap R\neq\emptyset\}}\cdot\frac{|S|}{n_c}\Big) = \frac{1}{n_c}\cdot\Big(\sum_{S\subseteq C}|S| - \sum_{S\subseteq(C\backslash R)}|S|\Big)\\ &= \frac{1}{n_c}\cdot\Big(n_c\cdot 2^{n_c-1} - (n_c - n_r)\cdot 2^{n_c-n_r-1}\Big)\\ &= 2^{n_c-1} - (1 - \tfrac{n_r}{n_c})\cdot 2^{n_c-n_r-1}.\end{aligned}$$

Therefore,

$$\Delta(\tilde{A}_i, R) = \begin{cases}\frac{n}{\theta}\cdot(1 - \frac{n_r}{n_c})\cdot 2^{1-n_r}, & \text{if } s_i \in R,\\ -\frac{n}{\theta}\cdot\frac{n_r}{n_c}\cdot 2^{1-n_r}, & \text{otherwise.}\end{cases} \quad (15)$$

When $b \geq \sigma(C)$, according to Lemma 1, an RR set R contributes to $\tilde{B}_i$ by an additive factor of

$$\Delta(\tilde{B}_i, R) \geq \frac{n}{\theta}\cdot\frac{\mathbf{1}_{\{C\cap R\neq\emptyset\}}}{n_c} + \Delta(\tilde{A}_i, R).$$

Note that when $n_r \geq 1$,

$$\Delta(\tilde{A}_i, R) \geq -\frac{n}{\theta}\cdot\frac{n_r}{2^{n_r-1}}\cdot\frac{1}{n_c} \geq -\frac{n}{\theta}\cdot\frac{1}{n_c},$$

and when $n_r = 0$, $\Delta(\tilde{A}_i, R) = 0$. As $C\cap R \neq \emptyset$ when $n_r \geq 1$, we then have

$$\Delta(\tilde{B}_i, R) \geq \frac{n}{\theta}\cdot\frac{1}{n_c} - \frac{n}{\theta}\cdot\frac{1}{n_c} = 0.$$

Since $B_i = \sum_{R\in\mathcal{R}}\Delta(\tilde{B}_i, R)$ when $\theta \to \infty$, we can obtain that $B_i = \lambda - c_i \geq 0$ for each $1 \leq i \leq n_c$.

As a result, according to (13), we have the conclusion that $\langle p_1^*, p_2^*, \ldots, p_{n_c}^*\rangle$ in (14) is an optimal price profile. □

Discussion on Privacy Issues. Privacy protection is critical for both OSN providers and influencers. On one hand, our pricing mechanism intrinsically protects the privacy of OSN providers since they do not need to unveil the network structures. On the other hand, as most information posts are publicly available and designed to attract followers on platforms such as Instagram and TikTok, many influencers are willing to monetize their public influence powers. To minimize the ethical issues, OSN providers can first ensure influencers' willingness of engagement in marketing campaigns and then post their prices for marketing campaigns. Furthermore, to protect the privacy of the candidate seeds, the prices can be posted anonymously such that the personal information can be protected.

5. ESTIMATION OF NODE PRICES

In this section, we study the estimation of the node prices $p_1^*, p_2^*, \ldots, p_{n_c}^*$ in the optimal price profile. Based on the proof of Corollary 1, under the condition that $b \geq \sigma(C)$, p_i^* can be represented as

$$p_i^* = \lambda - c_i = A_i + \frac{b}{n_c},$$

where A_i can be estimated using the RIS method. When θ RR sets are generated, an RR set R contributes to the estimation $\tilde{A}_i$ by an additive factor of $\Delta(\tilde{A}_i, R)$ given in (15). We generalize the stopping rule algorithm [11] to get an (ε, δ)-approximation of A_i. Similar to the work of [28], we also use the martingale-based concentration bounds [9] to tighten the threshold setting in the stopping rule algorithm. The key differences of our algorithm from [11] and [28] are as follows:

- The random variables $\Delta(\tilde{A}_i, R)$ may be negative. We shift the random variables to fall in the range of $[0, 1]$ so that the stopping rule algorithm can be applied.
- We invent a tighter threshold setting than [11] and [28] to improve the efficiency of the stopping rule algorithm. We also construct an algorithm to estimate all the A_i ($i = 1, 2, \ldots, n_c$) simultaneously.

5.1 Shifting $\Delta(\tilde{A}_i, R)$

By definition, it is possible for $\Delta(\tilde{A}_i, R)$ to be less than 0. The stopping rule algorithm [11] is designed to estimate the mean of a non-negative random variable distributed in $[0, 1]$. To apply the stopping rule algorithm, we first increase $\Delta(\tilde{A}_i, R)$ to make it non-negative. According to the proof of Corollary 1, $\Delta(\tilde{A}_i, R)$ can be made non-negative by adding a factor of $\frac{n}{\theta} \cdot \frac{\mathbf{1}_{\{C \cap R \neq \emptyset\}}}{n_c}$. Let

$$\Delta(\tilde{A}'_i, R) = \Delta(\tilde{A}_i, R) + \frac{n}{\theta} \cdot \frac{\mathbf{1}_{\{C \cap R \neq \emptyset\}}}{n_c}.$$

If we aggregate $\Delta(\tilde{A}'_i, R)$ over θ random RR sets, the actual value estimated is $A'_i = A_i + \frac{\sigma(C)}{n_c} = p^*_i + \frac{\sigma(C)-b}{n_c}$. Thus, we shall first estimate A'_i using the stopping rule algorithm and then compute p^*_i.

Let $R_1, R_2, \ldots, R_\theta$ be a sequence of random RR sets. Let $X_{i,j}$ be a random variable defined as

$$X_{i,j} = \begin{cases} 0 & \text{if } R_j \cap C = \emptyset, \\ \frac{1}{n_c} + 2^{1-n_r} \cdot (1 - \frac{n_r}{n_c}) & \text{if } s_i \in R_j \cap C, \\ \frac{1}{n_c} - 2^{1-n_r} \cdot \frac{n_r}{n_c} & \text{otherwise}, \end{cases} \tag{16}$$

where $n_c = |C|$ and $n_r = |R_j \cap C|$. It is easy to verify that $0 \leq X_{i,j} \leq 1$.

By definition, $\Delta(\tilde{A}'_i, R_j) = \frac{n}{\theta} \cdot X_{i,j}$. Thus, we have $A'_i = \frac{n}{\theta} \cdot \mathbb{E}[\sum_{j=1}^{\theta} X_{i,j}]$. To use $\frac{n}{\theta} \cdot \sum_{j=1}^{\theta} X_{i,j}$ as an estimator of A'_i, we need θ to be large enough in order to ensure that $\sum_{j=1}^{\theta} X_{i,j}$ does not deviate significantly from its expectation.

5.2 Stopping Rule Algorithm

To obtain an (ε, δ)-approximation of the mean of a random variable, the stopping rule algorithm first computes a threshold Υ and then continuously generates samples according to the distribution until their sum exceeds Υ. Finally, the stopping rule algorithm returns the average of these samples as the estimate. The basic stopping rule algorithm can estimate the mean of only one random variable. In our pricing problem, we need to estimate all the values A'_i $(i = 1, 2, \ldots, n_c)$ in order to derive the optimal price profile. Estimating each A'_i by a separate invocation of the stopping rule algorithm can result in generating an unnecessarily large number of samples (RR sets). In the following, we construct a stopping rule algorithm to estimate all the values A'_i $(i = 1, 2, \ldots, n_c)$ simultaneously.

Algorithm 1 shows the details. The algorithm first calculates the threshold Υ based on the required approximation parameters ε and δ (line 1). After that, samples are generated and aggregated until the minimum sum among all the nodes s_i's exceeds Υ (lines 4–10). The number of samples θ_i is recorded for each node s_i when its sum S_i exceeds Υ (line 8). Finally, the average $\tilde{\mu}_i$ of the samples is returned as the estimate for each node s_i (line 11).

5.3 Theoretical Analysis

A tighter threshold setting. The original stopping rule algorithm in [11] sets the threshold Υ as

$$\Upsilon_D = 1 + 4(1+\varepsilon)(e-2)\ln\left(\frac{2}{\delta}\right)\frac{1}{\varepsilon^2} > 1 + 2.87(1+\varepsilon)\ln\left(\frac{2}{\delta}\right)\frac{1}{\varepsilon^2}.$$

Algorithm 1: Stopping Rule Algorithm

Input: RR sets $R_1, R_2, \cdots$ and $0 < \varepsilon, \delta < 1$;
Output: an (ε, δ)-approximation $\tilde{\mu}_i$ of each μ_i $(i \leq n_c)$;
1 $\Upsilon \leftarrow (1+\varepsilon)(1 + (2 + \frac{2}{3}\varepsilon)\ln(\frac{2}{\delta})\frac{1}{\varepsilon^2})$, $\theta \leftarrow 0$;
2 **foreach** *node* $s_i \in C$ **do**
3   Initialize $S_i \leftarrow 0$;
4 **while** $\min S_i < \Upsilon$ **do**
5   **foreach** *node* $s_i \in C$ **do**
6   **if** $S_i < \Upsilon$ **then**
7             $S_i \leftarrow S_i + X_{i,\theta}$, where $X_{i,\theta}$ is based on (16);
8   **if** $S_i \geq \Upsilon$ **then**
9             $\theta_i \leftarrow \theta$;
10     $\theta \leftarrow \theta + 1$;
11 **return** $\{\tilde{\mu}_i = \frac{\Upsilon}{\theta_i} : 1 \leq i \leq n_c\}$;

In Algorithm 1, we set the threshold Υ as

$$\Upsilon = (1+\varepsilon)(1+(2+\frac{2}{3}\varepsilon)\ln\left(\frac{2}{\delta}\right)\frac{1}{\varepsilon^2}) < (1+\varepsilon)(1+2.67\ln\left(\frac{2}{\delta}\right)\frac{1}{\varepsilon^2}),$$

since $0 < \varepsilon \leq 1$. Hence, the Υ setting in our algorithm is tighter than that in [11] when $0.2(1+\varepsilon)\ln(\frac{2}{\delta})\frac{1}{\varepsilon^2} > \varepsilon$, which holds when $\varepsilon \leq 0.5 < \sqrt[3]{0.2 \cdot \ln 2}$.

Similar to our algorithm, the stopping rule algorithm in [28] also uses the martingale-based concentration bounds [9] to set the threshold Υ. Since $0 \leq X_{i,j} \leq 1$ in our problem, applying the algorithm in [28], the threshold Υ is set as

$$\Upsilon_N = (1+\varepsilon)\big(2 + \frac{2}{3}\varepsilon'\big)\ln\left(\frac{2}{\delta}\right)\frac{1}{\varepsilon'^2},$$

where $\varepsilon' = \varepsilon(1 - \frac{\varepsilon}{(2+\frac{2}{3}\varepsilon)\ln(\frac{2}{\delta})}) < \varepsilon$. In Algorithm 1, we replace ε' with ε in the setting of Υ and add an additive factor of $1+\varepsilon$. Next, we show that the Υ value in our algorithm is smaller than that in [28]. To prove

$$(1+\varepsilon)\big(2+\frac{2}{3}\varepsilon'\big)\ln\left(\frac{2}{\delta}\right)\frac{1}{\varepsilon'^2} > (1+\varepsilon)\big(1+\big(2+\frac{2}{3}\varepsilon\big)\ln\left(\frac{2}{\delta}\right)\frac{1}{\varepsilon^2}\big),$$

it is equivalent to show that

$$\big(1+\frac{1}{3}\varepsilon'\big)\frac{1}{\varepsilon'^2} - \big(1+\frac{1}{3}\varepsilon\big)\frac{1}{\varepsilon^2} > \frac{1}{2\ln(\frac{2}{\delta})}. \tag{17}$$

Let $\alpha := (2+\frac{2}{3}\varepsilon)\ln(\frac{2}{\delta})$ and $\beta := \frac{\varepsilon'}{\varepsilon} = 1 - \frac{\varepsilon}{\alpha}$. We have

$$\begin{aligned} &\big(1+\frac{1}{3}\varepsilon'\big)\frac{1}{\varepsilon'^2} - \big(1+\frac{1}{3}\varepsilon\big)\frac{1}{\varepsilon^2} \\ &= \big(1+\frac{1}{3}\beta\varepsilon\big)\frac{1}{\beta^2\varepsilon^2} - \big(1+\frac{1}{3}\varepsilon\big)\frac{1}{\varepsilon^2} \\ &= \big(\frac{1+\beta}{\beta\varepsilon} + \frac{1}{3}\big) \cdot \frac{1-\beta}{\beta\varepsilon} \\ &> \frac{2-\frac{\varepsilon}{\alpha}}{\varepsilon(1-\frac{\varepsilon}{\alpha})} \cdot \frac{\frac{\varepsilon}{\alpha}}{\varepsilon(1-\frac{\varepsilon}{\alpha})} \\ &= \frac{2\alpha - \varepsilon}{\varepsilon(\alpha-\varepsilon)^2} \\ &> \frac{2}{\varepsilon(\alpha-\varepsilon)}. \end{aligned}$$

Since $4\ln(\frac{2}{\delta}) > \frac{8}{3}\ln(\frac{2}{\delta}) \geq \alpha \geq \alpha - \varepsilon \geq \varepsilon(\alpha-\varepsilon)$, we have

$$\frac{2}{\varepsilon(\alpha-\varepsilon)} > \frac{1}{2\ln(\frac{2}{\delta})}.$$

Thus, (17) holds and the Υ setting in our algorithm is tighter. So, our algorithm would generate less samples than that in [28].

In the following, we prove that our Υ setting can guarantee an (ε, δ)-approximation of estimation. The proof is similar in spirit to the original stopping rule algorithm [11], but we make use of the martingale-based concentration bounds [9] in the derivation.

DEFINITION 2 ([9]). *A sequence of random variables $Y_1, Y_2, \dots$ is a martingale if and only if $\mathbb{E}[|Y_j|] < \infty$ and $\mathbb{E}[Y_j \mid Y_1, Y_2, \dots, Y_{j-1}] = Y_{j-1}$ for any j.*

Since each RR set R_j is generated randomly and independently of all the prior RR sets, we have

$$\mathbb{E}[X_{i,j} \mid X_{i,1}, X_{i,2}, \dots, X_{i,j-1}] = \mathbb{E}[X_{i,j}] = \frac{A'_i}{n}. \tag{18}$$

Let $\mu_i = \frac{A'_i}{n}$ and $M_{i,j} = \Sigma_{k=1}^{j}(X_{i,k} - \mu_i)$. Then, we have $\mathbb{E}[|M_{i,j}|] \le j < \infty$, and

$$\mathbb{E}[M_{i,j} \mid M_{i,1}, M_{i,2}, \dots, M_{i,j-1}] = M_{i,j-1}.$$

Therefore, $M_{i,1}, M_{i,2}, \dots, M_{i,\theta}$ is a martingale.

LEMMA 3 ([9]). *Let $Y_1, Y_2, \dots, Y_j$ be a martingale, such that $Y_1 \le a$, $Y_k - Y_{k-1} \le a$ for any $2 \le k \le j$, and $\mathrm{Var}[Y_1] + \sum_{k=2}^{j} \mathrm{Var}[Y_k \mid Y_1, Y_2, \dots, Y_{k-1}] \le b$. Then, for any $\eta > 0$,*

$$\Pr[Y_j - \mathbb{E}[Y_j] \ge \eta] \le \exp\Big(-\frac{\eta^2}{\frac{2}{3}a\eta + 2b}\Big). \tag{19}$$

Since $0 \le X_{i,j} \le 1$ for any $1 \le j \le \theta$, we have $M_{i,1} = X_{i,1} - \mu_i \le 1$ and $M_{i,j} - M_{i,j-1} = X_{i,j} - \mu_i \le 1$ for any $2 \le j \le \theta$. Let $\mathrm{Var}[\cdot]$ denote the variance of a random variable. It follows that $\mathrm{Var}[X_{i,j}] = \mathbb{E}[X_{i,j}^2] - \mathbb{E}[X_{i,j}]^2 = \mathbb{E}[X_{i,j}^2] - \mu_i^2 \le \mathbb{E}[X_{i,j}] - \mu_i^2 \le \mu_i(1-\mu_i)$. Hence,

$$\begin{aligned}&\mathrm{Var}[M_{i,1}] + \sum_{j=2}^{\theta} \mathrm{Var}[M_{i,j} \mid M_{i,1}, M_{i,2}, \dots, M_{i,j-1}]\\ &= \sum_{j=1}^{\theta} \mathrm{Var}[X_{i,j} - \mu_i] \le \theta\mu_i \cdot (1-\mu_i) \le \theta\mu_i.\end{aligned}$$

By Lemma 3, we have $\Pr[M_{i,\theta} \ge \varepsilon \cdot \theta\mu_i] \le \exp\big(-\frac{\varepsilon^2\theta\mu_i}{2+\frac{2}{3}\varepsilon}\big)$.

Similarly, $-M_{i,1}, -M_{i,2}, \dots, -M_{i,\theta}$ is a martingale such that $-M_{i,1} \le \mu_i$, $-M_{i,j} + M_{i,j-1} \le \mu_i$, and $\mathrm{Var}[-M_{i,1}] + \sum_{j=2}^{\theta} \mathrm{Var}[-M_{i,j} \mid -M_{i,1}, -M_{i,2}, \dots, -M_{i,j-1}] \le \theta\mu_i \cdot (1-\mu_i)$. Then, we have

$$\begin{aligned}\Pr[-M_{i,\theta} \ge \varepsilon \cdot \theta\mu_i] &= \Pr\Big[\sum_{j=1}^{\theta} X_{i,j} - \theta\mu_i \le -\varepsilon \cdot \theta\mu_i\Big]\\ &\le \exp\Big(-\frac{\varepsilon^2\theta^2\mu_i^2}{\frac{2}{3}\varepsilon\theta\mu_i^2 + 2\theta\mu_i(1-\mu_i)}\Big)\\ &\le \exp\Big(-\frac{\varepsilon^2\theta\mu_i}{2}\Big).\end{aligned}$$

To summarize, we have the following corollary.[1]

[1]Tang et al. [39] directly gave the lower tail result without providing the detailed proof. Our analysis is based on Lemma 3 requiring $Y_1 \le a$ and $Y_k - Y_{k-1} \le a$, whereas Tang et al. [39] utilized a similar lemma requiring $|Y_1| \le a$ and $|Y_k - Y_{k-1}| \le a$, which might be insufficient for deriving the lower tail result.

COROLLARY 2. *For any $\varepsilon > 0$,*

$$\Pr\Big[\sum_{j=1}^{\theta} X_{i,j} - \theta\mu_i \ge \varepsilon \cdot \theta\mu_i\Big] \le \exp\Big(-\frac{\varepsilon^2\theta\mu_i}{2+\frac{2}{3}\varepsilon}\Big), \tag{20}$$

$$\Pr\Big[\sum_{j=1}^{\theta} X_{i,j} - \theta\mu_i \le -\varepsilon \cdot \theta\mu_i\Big] \le \exp\Big(-\frac{\varepsilon^2\theta\mu_i}{2}\Big). \tag{21}$$

Based on Corollary 2, we show that Algorithm 1 returns an (ε, δ)-approximate $\tilde{\mu}_i$ of each μ_i $(1 \le i \le n_c)$.

THEOREM 2. *Algorithm 1 returns an (ε, δ)-approximate $\tilde{\mu}_i$ of each μ_i $(1 \le i \le n_c)$, i.e.,*

$$\Pr[(1-\varepsilon)\mu_i \le \tilde{\mu}_i \le (1+\varepsilon)\mu_i] \ge 1-\delta. \tag{22}$$

PROOF. Given any i where $1 \le i \le n_c$, we will prove the following two probabilistic inequalities:

$$\Pr[\tilde{\mu}_i < (1-\varepsilon)\mu_i] \le \frac{\delta}{2}, \tag{23}$$

$$\Pr[\tilde{\mu}_i > (1+\varepsilon)\mu_i] \le \frac{\delta}{2}. \tag{24}$$

First, we prove (23). Since $0 \le X_{i,j} \le 1$, by the definition of Algorithm 1, when it terminates, we have

$$\Upsilon \le S_i = \sum_{j=1}^{\theta_i} X_{i,j} \le \Upsilon + 1.$$

Let $L_1 = \lceil \frac{\Upsilon}{(1-\varepsilon)\mu_i} \rceil$. Then,

$$L_1 = \Big\lceil \frac{\Upsilon}{(1-\varepsilon)\mu_i} \Big\rceil \ge \frac{\Upsilon}{(1-\varepsilon)\mu_i}, \tag{25}$$

and hence,

$$\frac{\Upsilon}{L_1} \le (1-\varepsilon)\mu_i.$$

Since θ_i is an integer, we have

$$\begin{aligned}\Pr[\tilde{\mu}_i < (1-\varepsilon)\mu_i] &= \Pr[\Upsilon < (1-\varepsilon)\mu_i\theta_i]\\ &= \Pr\Big[\frac{\Upsilon}{(1-\varepsilon)\cdot\mu_i} < \theta_i\Big]\\ &= \Pr[L_1 \le \theta_i]\\ &\le \Pr\Big[\sum_{j=1}^{L_1} X_{i,j} \le \sum_{j=1}^{\theta_i} X_{i,j}\Big]\\ &\le \Pr\Big[\sum_{j=1}^{L_1} X_{i,j} \le \Upsilon + 1\Big]\\ &= \Pr\Big[\frac{\sum_{j=1}^{L_1} X_{i,j}}{L_1} \le \frac{\Upsilon+1}{L_1}\Big]\\ &\le \Pr\Big[\frac{\sum_{j=1}^{L_1} X_{i,j}}{L_1} \le (1-\varepsilon)\mu_i + \frac{1}{L_1}\Big].\end{aligned}$$

Moreover, by the definition of L_1, we have

$$\frac{1}{L_1} \le \frac{(1-\varepsilon)\mu_i}{\Upsilon} = \frac{(1-\varepsilon)\mu_i}{(1+\varepsilon)(1+(2+\frac{2}{3}\varepsilon)\ln(\frac{2}{\delta})\frac{1}{\varepsilon^2})} < \frac{\varepsilon^2\mu_i}{1+\varepsilon}.$$

Therefore,

$$\begin{aligned}\Pr[\tilde{\mu}_i < (1-\varepsilon)\mu_i] &\le \Pr\Big[\frac{\sum_{j=1}^{L_1} X_{i,j}}{L_1} < (1-\varepsilon)\mu_i + \frac{\varepsilon^2\cdot\mu_i}{1+\varepsilon}\Big]\\ &\le \Pr\Big[\frac{\sum_{j=1}^{L_1} X_{i,j}}{L_1} < (1-\frac{\varepsilon}{1+\varepsilon})\cdot\mu_i\Big]\\ &= \Pr\Big[\sum_{j=1}^{L_1} X_{i,j} - L_1\mu_i < -\frac{\varepsilon}{1+\varepsilon}\cdot L_1\mu_i\Big].\end{aligned}$$

Meanwhile,

$$L_1 \geq \frac{\Upsilon}{(1-\varepsilon)\mu_i} > \frac{2(1+\varepsilon)\ln(\frac{2}{\delta})}{(1-\varepsilon)\varepsilon^2\mu_i} > \frac{2(1+\varepsilon)^2\ln(\frac{2}{\delta})}{\varepsilon^2\mu_i}.$$

Note that $\frac{\sum_{j=1}^{L_1} X_{i,j}}{L_1}$ is an estimate of μ_i using the first L_1 random samples. Applying (21), we obtain

$$\begin{aligned}\Pr[\tilde{\mu}_i \leq (1-\varepsilon)\mu_i] &\leq \exp\Big(-\frac{\varepsilon^2 L_1\mu_i}{2(1+\varepsilon)^2}\Big)\\ &< \exp\Big(-\frac{\varepsilon^2\frac{2(1+\varepsilon)^2\ln(\frac{2}{\delta})}{\varepsilon^2\mu_i}\mu_i}{2(1+\varepsilon)^2}\Big)\\ &= \frac{\delta}{2}.\end{aligned}$$

This completes the proof of (23).

Next, we prove (24), which is similar. Let $L_2 = \lfloor\frac{\Upsilon}{(1+\varepsilon)\mu_i}\rfloor$. Then, we have

$$\begin{aligned}\Pr[\tilde{\mu}_i > (1+\varepsilon)\mu_i] &= \Pr[\Upsilon > (1+\varepsilon)\mu_i\theta_i]\\ &= \Pr\Big[\frac{\Upsilon}{(1+\varepsilon)\mu_i} > \theta_i\Big]\\ &= \Pr[L_2 \geq \theta_i]\\ &\leq \Pr\Big[\sum_{j=1}^{L_2} X_{i,j} \geq \sum_{j=1}^{\theta_i} X_{i,j}\Big]\\ &\leq \Pr\Big[\sum_{j=1}^{L_2} X_{i,j} \geq \Upsilon\Big]\\ &= \Pr\Big[\frac{\sum_{j=1}^{L_2} X_{i,j}}{L_2} \geq \frac{\Upsilon}{L_2}\Big].\end{aligned}$$

By the definition of L_2, we have

$$L_2 \leq \frac{\Upsilon}{(1+\varepsilon)\mu_i},$$

which indicates that

$$\frac{\Upsilon}{L_2} \geq (1+\varepsilon)\mu_i.$$

In addition, $L_2 > \frac{\Upsilon}{(1+\varepsilon)\mu_i} - 1 = \frac{1}{\mu_i} + (2+\frac{2}{3}\varepsilon)\ln(\frac{2}{\delta})\frac{1}{\varepsilon^2} - 1 > (2+\frac{2}{3}\varepsilon)\ln\frac{2}{\delta}\cdot\frac{1}{\varepsilon^2}\cdot\frac{1}{\mu_i}$. By (20), we obtain

$$\begin{aligned}\Pr[\tilde{\mu}_i > (1+\varepsilon)\mu_i] &\leq \Pr\Big[\frac{\sum_{j=1}^{L_2} X_{i,j}}{L_2} \geq (1+\varepsilon)\mu_i\Big]\\ &= \Pr\Big[\sum_{j=1}^{L_2} X_{i,j} - L_2\mu_i \geq \varepsilon\cdot L_2\mu_i\Big]\\ &\leq \exp\Big(-\frac{\varepsilon^2 L_2\mu_i}{2+\frac{2}{3}\varepsilon}\Big)\\ &< \exp\Big(-\frac{\varepsilon^2(2+\frac{2}{3}\varepsilon)\ln\frac{2}{\delta}\frac{1}{\varepsilon^2}\frac{1}{\mu_i}\mu_i}{2+\frac{2}{3}\varepsilon}\Big)\\ &= \frac{\delta}{2}.\end{aligned}$$

By the union bound, (23) and (24) give rise to (22). □

With the values $\tilde{\mu}_i$ returned by Algorithm 1, $\tilde{A}'_i$ is computed as $\tilde{A}'_i = n\cdot\tilde{\mu}_i$. By Theorem 2, each value $\tilde{A}'_i$ ($i = 1, 2, \ldots, n_c$) can be estimated within a factor of $1+\varepsilon$ with probability at least $1-\delta$.

5.4 Choice of b

As discussed in Section 5.1, besides A'_i, we also need to estimate $\frac{b-\sigma(C)}{n_c}$ in order to compute p^*_i as $p^*_i = A'_i + \frac{b-\sigma(C)}{n_c}$. According to Corollary 1, in order to guarantee $p^*_i \geq 0$, the total price b of all the candidate nodes must be set no less than $\sigma(C)$, i.e., $b \geq \sigma(C)$. In general, we can also use the stopping rule algorithm and the RIS method to estimate $b - \sigma(C)$. In the following, we illustrate it by assuming b is set such that the expected price of a randomly chosen seed set is the same as its expected influence spread. Suppose that all the subsets of C are equally likely to be chosen as the seed set. Then, the expected price of a randomly chosen seed set is $\sum_{i=1}^{n_c} p^*_i\cdot\frac{1}{2}$. On the other hand, the expected influence spread of the set is $\frac{\sum_{S\subseteq C}\sigma(S)}{2^{n_c}}$. Therefore, b is set as

$$b = \sum_{i=1}^{n_c} p^*_i = \frac{\sum_{S\subseteq C}\sigma(S)}{2^{n_c-1}}. \tag{26}$$

It is easy to verify that $b \geq \sigma(C)$ since

$$\begin{aligned}b &= \frac{2\sum_{S\subseteq C}\sigma(S)}{2^{n_c}}\\ &= \frac{\sum_{S\subseteq C}(\sigma(S)+\sigma(C\backslash S))}{2^{n_c}}\\ &\geq \frac{\sum_{S\subseteq C}\sigma(C)}{2^{n_c}} = \sigma(C).\end{aligned}$$

Let $\tilde{\psi}$ be the estimation of $b-\sigma(C)$. According to Lemma 1, when a sequence of θ RR sets are generated, each RR set R contributes to $\tilde{\psi}$ by an additive factor of

$$\begin{aligned}\Delta(\tilde{\psi}, R) &= \frac{n}{\theta}\cdot\frac{\sum_{S\subseteq C}\mathbf{1}_{\{S\cap R\neq\emptyset\}}}{2^{n_c-1}} - \frac{n}{\theta}\cdot\mathbf{1}_{\{C\cap R\neq\emptyset\}}\\ &= \frac{n}{\theta}\cdot\frac{2^{n_c}(1-\frac{1}{2^{n_r}})}{2^{n_c-1}} - \frac{n}{\theta}\cdot\mathbf{1}_{\{C\cap R\neq\emptyset\}}\\ &= \frac{n}{\theta}\cdot\big(2(1-\frac{1}{2^{n_r}}) - \mathbf{1}_{\{C\cap R\neq\emptyset\}}\big),\end{aligned}$$

where $n_r = |R\cap C|$.

Since $0 \leq 2(1-\frac{1}{2^{n_r}}) - \mathbf{1}_{\{C\cap R\neq\emptyset\}} \leq 1$, an (ε,δ)-approximation of $b-\sigma(C)$ can be obtained using the stopping rule algorithm similar to the estimation of A'_i. By the union bound, we can set the failure probability $\delta' = \frac{\delta}{n_c+1}$ in the stopping rule algorithm so that all the values $\tilde{A}'_i$ and $\tilde{\psi}$ are estimated within a factor of $1+\varepsilon$ with probability at least $1-\delta$, which gives rise to an (ε,δ)-approximation of all the prices p^*_i.

6. EXPERIMENTS

This section experimentally evaluates the quality and scalability of our proposed algorithms. We implement our algorithms using C++. All experiments are run on a machine with Intel Xeon 2.4GHz CPU and 384GB memory.

6.1 Experimental Setup

Datasets. We evaluate our algorithms by several real datasets including Facebook, Google+, LiveJournal, Orkut and Twitter. The first four datasets are available at `http://snap.stanford.edu/data` and the Twitter dataset is obtained from `http://an.kaist.ac.kr/traces/WWW2010.html` [25]. Table 1 gives the details of these datasets.

Table 1: Datasets.

Dataset	#nodes	#edges	Type	Avg. degree
Facebook	4.0K	88.2K	Undirected	43.7
Google+	107.6K	13.7M	Directed	254.1
LiveJournal	4.8M	69.0M	Directed	28.5
Orkut	3.1M	117.2M	Undirected	76.3
Twitter	41.7M	1.5G	Directed	70.5

Parameter Settings. We adopt the Independent Cascade diffusion model and set the propagation probability $p_{u,v}$ of each edge (u, v) to the reciprocal of v's in-degree which is a commonly used setting by other studies [37–40]. We set the number of candidate nodes $n_c = 200$, 500 or 1000, the failure probability $\delta = \frac{1}{n}$ (n is the number of nodes in the OSN) and the error threshold $\varepsilon = 0.1$ by default. We assume that the candidate node set C includes the top-n_c nodes with the highest out-degrees. These nodes are offered by the OSN provider to advertisers for seed selection. We set the total price of all the candidate nodes at $b = \frac{\sum_{S\subseteq C}\sigma(S)}{2^{n_c-1}}$ as discussed in Section 5.4.

Algorithms. We compare the price profile calculated by our pricing algorithm (Algorithm 1), referred to as OptPrice, with the following baselines:

- Uniform: The prices of all the candidate nodes are set the same.
- Degree: The price of each candidate node is set proportional to its out-degree.
- SingletonInf: The price of each candidate node is set proportional to the influence spread it can produce when selected as the only seed. We estimate the influence spread using the RIS method and the stopping rule algorithm.
- IMRank [8]: A ranking of candidate nodes is generated in decreasing order of their marginal gains in influence spread. We generate the ranking by applying the greedy hill-climbing algorithm for influence maximization [23]. The price of each candidate node is set proportional to its marginal gain.

Time Complexity. In the stopping rule algorithm (Algorithm 1), by the analysis in Theorem 2, the number of samples θ_i generated for estimating μ_i satisfies

$$\Pr\Big[\frac{\Upsilon}{(1-\varepsilon)\cdot\mu_i} < \theta_i\Big] = \Pr[\tilde{\mu}_i < (1-\varepsilon)\mu_i] < \frac{\delta}{2}.$$

To estimate all the node prices in the candidate set, our stopping rule algorithm finishes with $O(\frac{\Upsilon}{\min_{s_i\in C}\mu_i})$ samples generated with probability at least $1-\frac{\delta}{2}$. In addition, the expected number of RR sets generated is $\frac{\Upsilon}{\min_{s_i\in C}\mu_i}$. Let EPT be the expected time complexity to generate an RR set. According to [40], let v^* be a random node chosen from V with probability proportional to its in-degree and we have $\mathrm{EPT} = \mathbb{E}[\sigma(\{v^*\})]\cdot\frac{m}{n}$ where the expectation is over the randomness of v^*, $n = |V|$ is the number of nodes and $m = |E|$ is the number of edges in the network. The expected time complexity of our pricing algorithm is then $O(\frac{\Upsilon}{\min_{s_i\in C}\mu_i}\cdot\mathrm{EPT})$. Let α_i be the estimation variable for the singleton influence spread for a seed $s_i \in C$, i.e., $\alpha_i = \frac{\sigma(s_i)}{n}$. Similarly, the expected time complexity of the SingletonInf pricing is $O(\frac{\Upsilon}{\min_{s_i\in C}\alpha_i}\cdot\mathrm{EPT})$. When using the RR sets generated by our stopping rule algorithm, the IMRank algorithm has the same time complexity as our pricing algorithm. Since the baselines of Degree and Uniform pricing do not incur computational cost for sampling, their time complexities are $O(|C|)$.

For fair comparison, the total price b of all the candidate nodes is set the same for all the pricing algorithms.

6.2 Experimental Results

6.2.1 Efficiency of Our Algorithm

Table 2 shows the running time of our pricing algorithm and the number of RR sets generated for various datasets. As can be seen, our OptPrice algorithm can compute the price profile within hours even for large-scale datasets. This demonstrates the scalability of our OptPrice algorithm.

6.2.2 Evaluation of Divergence Function

A straightforward evaluation is to compare the values of the divergence function (2) produced by the pricing profiles of different algorithms. According to Lemma 2, the divergence function can be divided into two parts: $\frac{1}{2^{n_c}}\sum_{S\subseteq C}\sigma(S)^2$ and $\sum_{i=1}^{n_c} g(p_i)$.

For each $g(p_i)$, the value of $\frac{1}{2^{n_c-1}}\sum_{S\subseteq C\setminus\{s_i\}}\sigma(S\cup\{s_i\})$ can be estimated using the RIS method and the stopping rule algorithm. Then, $\sum_{i=1}^{n_c} g(p_i)$ can be computed based on (3). The challenge lies in evaluating $\frac{1}{2^{n_c}}\sum_{S\subseteq C}\sigma(S)^2$. This part is non-linear with respect to the influence spread. There are an exponential number of seed sets to measure in order to obtain the sum. To make the evaluation tractable, we use the sample average to estimate the value of the sum. Note that this sum is an additive term in the divergence function that is independent from the price profile, which indicates that its estimation accuracy will not affect the relative performance of different algorithms. We show below the theoretical guarantees of the estimation accuracy when a given number of T seed sets are measured.

In the multiplicative-additive error form of martingale-based concentration bounds, the failure probability is independent of the mean of the tested random variable. The following lemma describes this phenomenon.

LEMMA 4. *Let $Z_1-\mathbb{E}[Z_1],\dots,Z_T-\mathbb{E}[Z_T]$ be a martingale difference sequence such that $Z_j\in[0,1]$ for each j. Let $\bar{Z} = \frac{1}{T}\sum_{j=1}^{T} Z_j$. If $\mathbb{E}[Z_j]$ is identical for every j, i.e., $\mathbb{E}[Z_j] = \mathbb{E}[\bar{Z}]$, then,*

$$\Pr[\bar{Z} \le (1-\varepsilon)\mathbb{E}[\bar{Z}] - \beta] \le \mathrm{e}^{-2\varepsilon\beta T}, \tag{27}$$

$$\Pr[\bar{Z} \ge (1+\varepsilon)\mathbb{E}[\bar{Z}] + \beta] \le \mathrm{e}^{-2\varepsilon\beta T/(1+\varepsilon/3)^2}. \tag{28}$$

PROOF. By (21) in Corollary 2, we have

$$\begin{aligned}\Pr\big[\bar{Z} \le (1-\varepsilon)\mathbb{E}[\bar{Z}] - \beta\big] &\le \exp\Big(-\frac{(\varepsilon\mathbb{E}[\bar{Z}]+\beta)^2 T}{2\mathbb{E}[\bar{Z}]}\Big)\\ &\le \exp\Big(-\frac{(2\sqrt{\varepsilon\mathbb{E}[\bar{Z}]\beta})^2 T}{2\mathbb{E}[\bar{Z}]}\Big)\\ &= \exp\big(-2\varepsilon\beta T\big).\end{aligned}$$

Table 2: **Running time and number of RR sets generated.**

#candidates	Metric	Facebook	Google+	LiveJournal	Orkut	Twitter
$n_c = 200$	Time (s)	19.21	64.49	701.54	2164.62	8655.96
	#RR sets	4.35E+06	1.34E+07	5.70E+07	1.76E+07	6.20E+06
$n_c = 500$	Time (s)	29.28	136.43	2422.94	5009.15	26965.50
	#RR sets	5.64E+06	2.61E+07	1.73E+08	3.96E+07	1.75E+07
$n_c = 1000$	Time (s)	44.16	276.62	5838.89	10376.70	53464.40
	#RR sets	6.47E+06	4.59E+07	2.87E+08	7.41E+07	3.39E+07

Table 3: Comparison between ω and $\beta \cdot \sigma(C)^2$.

#candidates	Metric	Facebook	Google+	LiveJournal	Orkut	Twitter
$n_c = 200$	ω	6.82E+05	7.34E+07	1.51E+10	2.83E+10	8.10E+13
	$\beta \cdot \sigma(C)^2$	3.29E+03	3.93E+05	7.94E+07	1.14E+08	3.24E+11
$n_c = 500$	ω	1.43E+06	1.50E+08	2.92E+10	4.24E+10	1.13E+14
	$\beta \cdot \sigma(C)^2$	5.86E+03	7.32E+05	1.42E+08	1.63E+08	4.27E+11
$n_c = 1000$	ω	2.99E+06	2.30E+08	4.62E+10	5.47E+10	1.29E+14
	$\beta \cdot \sigma(C)^2$	1.09E+04	1.06E+06	2.14E+08	2.05E+08	4.75E+11

Table 4: Value of divergence function.

#candidates	Algorithm	Facebook	Google+	LiveJournal	Orkut	Twitter
$n_c = 200$	Uniform	1.90E+04	2.50E+05	5.50E+07	4.46E+07	1.12E+11
	Degree	1.41E+04	1.32E+05	4.12E+07	4.30E+07	4.64E+10
	SingletonInf	8.19E+03	3.66E+04	3.20E+07	3.84E+07	1.06E+11
	IMRank	8.73E+03	1.78E+05	4.38E+07	2.57E+08	1.04E+12
	OptPrice	**1.24E+03**	**1.13E+04**	**1.14E+07**	**2.49E+07**	**3.20E+10**
$n_c = 500$	Uniform	1.21E+04	2.38E+05	5.34E+07	3.03E+07	1.34E+11
	Degree	9.44E+03	1.32E+05	3.82E+07	2.58E+07	1.09E+11
	SingletonInf	6.11E+03	6.73E+04	3.03E+07	2.17E+07	1.30E+11
	IMRank	1.91E+04	3.34E+05	6.17E+07	2.99E+08	1.42E+12
	OptPrice	**4.78E+02**	**2.15E+04**	**6.73E+06**	**7.41E+06**	**5.75E+10**
$n_c = 1000$	Uniform	6.74E+03	2.19E+05	7.06E+07	3.80E+07	8.87E+10
	Degree	5.39E+03	1.20E+05	5.58E+07	3.27E+07	8.52E+10
	SingletonInf	3.46E+03	7.67E+04	4.90E+07	2.92E+07	7.50E+10
	IMRank	3.31E+04	4.55E+05	1.02E+08	3.53E+08	1.43E+12
	OptPrice	**4.96E+02**	**1.78E+04**	**2.52E+07**	**1.49E+07**	**1.10E+10**

Similarly, by (20) in Corollary 2, we have

$$\Pr[\bar{Z} \geq (1+\varepsilon)\mathbb{E}[\bar{Z}] + \beta] \leq \exp\big(-h(\lambda)\big),$$

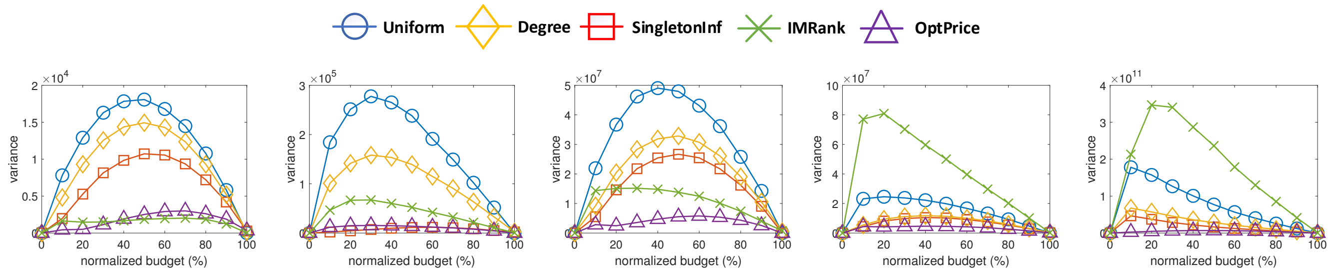
where $h(\lambda) = \frac{(\lambda^2 T)}{2(\lambda-\beta)/\varepsilon + 2\lambda/3}$ and $\lambda = \varepsilon\mathbb{E}[\bar{Z}] + \beta$. Let

$$\frac{\mathrm{d}h(\lambda)}{\mathrm{d}\lambda} = \frac{\big(2\lambda((\lambda-\beta)/\varepsilon + \lambda/3) - (1/\varepsilon + 1/3)\lambda^2\big)T}{2\big((\lambda-\beta)/\varepsilon + \lambda/3\big)^2} \triangleq 0.$$

Thus, $h(\lambda)$ achieves its minimum at $\lambda = \frac{2\beta}{\varepsilon(1/\varepsilon+1/3)}$ such that $h(\lambda) = \frac{2\varepsilon\beta T}{(1+\varepsilon/3)^2}$. This completes the proof. □

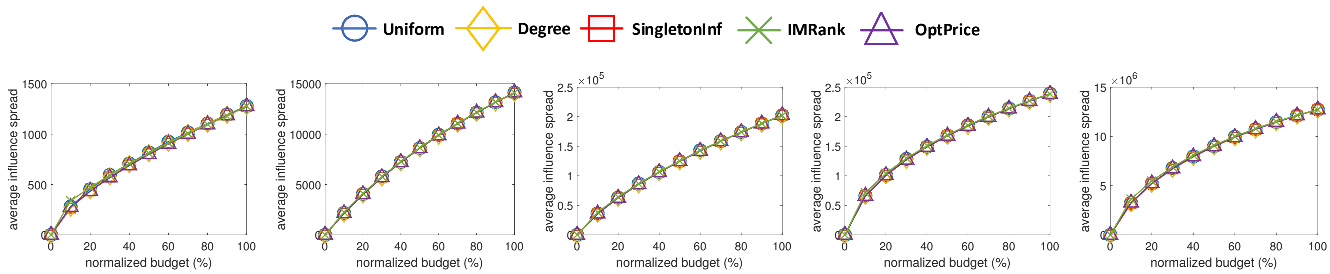
Let $S_1, S_2, \ldots, S_T$ be a sequence of randomly generated subsets of the candidate node set C. Let $\omega = \frac{1}{T}\sum_{j=1}^{T}\sigma(S_j)^2$. Then, for each $1 \leq j \leq T$, we have $\mathbb{E}[\sigma(S_j)^2] = \mathbb{E}[\omega] = \frac{1}{2^{n_c}}\sum_{S\subseteq C}\sigma(S)^2$. To make $\sigma(S_j)^2$ fall in the range of $[0, 1]$, we normalize the value of $\sigma(S_j)^2$ by $\sigma(C)^2$ so that $0 \leq \frac{\sigma(S_j)^2}{\sigma(C)^2} \leq 1$ since $\sigma(S_j) \leq \sigma(C)$.

Suppose that we set $T = \frac{(1+\varepsilon/3)^2 \ln(2/\delta)}{2\varepsilon\beta}$. Then, according to Lemma 4, the estimation $\frac{1}{T}\sum_{j=1}^{T}\sigma(S_j)^2$ is in the range of $[(1-\varepsilon)\mathbb{E}[\omega] - \beta\cdot\sigma(C)^2, (1+\varepsilon)\mathbb{E}[\omega] + \beta\cdot\sigma(C)^2]$ with



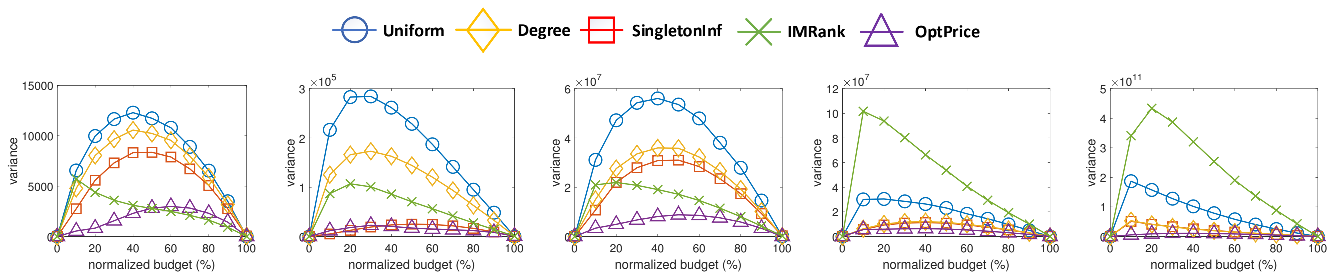
(a) Facebook (b) Google+ (c) LiveJournal (d) Orkut (e) Twitter

Figure 1: Variance of 10,000 seed sets for different budgets ($n_c = 200$).



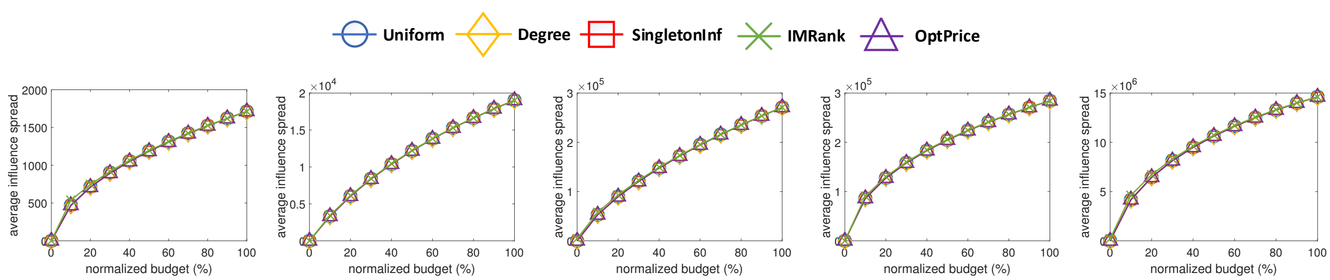
(a) Facebook (b) Google+ (c) LiveJournal (d) Orkut (e) Twitter

Figure 2: Average influence spread of 10,000 seed sets for different budgets ($n_c = 200$).



(a) Facebook (b) Google+ (c) LiveJournal (d) Orkut (e) Twitter

Figure 3: Variance of 10,000 seed sets for different budgets ($n_c = 500$).

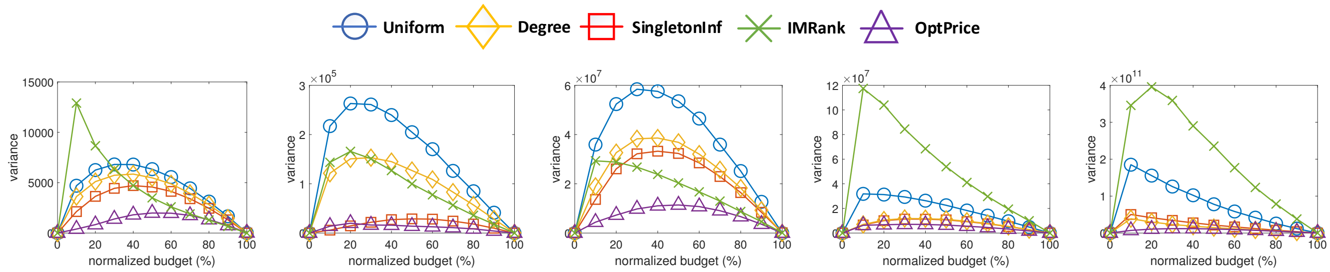


(a) Facebook (b) Google+ (c) LiveJournal (d) Orkut (e) Twitter

Figure 4: Average influence spread of 10,000 seed sets for different budgets ($n_c = 500$).

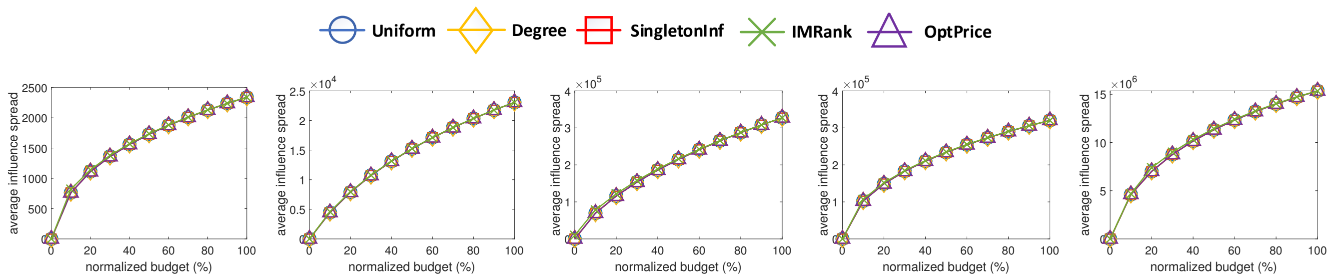
probability at least $1 - \delta$. In the experiments, we set $\varepsilon = 0.01$, $\beta = 0.002$, and $\delta = 0.01$. Then, $T = 133{,}342$. So, we randomly generate 133,342 subsets of C to estimate the value of $\frac{1}{2^{n_c}} \sum_{S \subseteq C} \sigma(S)^2$. As can be seen from Table 3, under this setting, the additive estimation error of $\beta \cdot \sigma(C)^2$ is negligible compared to the estimated value ω.

Table 4 shows the divergence function values produced by different pricing algorithms. It can be seen that our OptPrice



(a) Facebook (b) Google+ (c) LiveJournal (d) Orkut (e) Twitter

Figure 5: Variance of 10,000 seed sets for different budgets ($n_c = 1000$).



(a) Facebook (b) Google+ (c) LiveJournal (d) Orkut (e) Twitter

Figure 6: Average influence spread of 10,000 seed sets for different budgets ($n_c = 1000$).

algorithm results in significantly lower values than all the baselines. This implies that the price profile set produced by our OptPrice algorithm can better reflect the influence spread of any seed set.

6.2.3 Additional Experiments

Recall that the objective of our pricing profile is to minimize the divergence between the influence spreads and the seed prices for all possible seed sets. When such a divergence is minimized, the influence spread of any seed set would closely match its price. This implies that two seed sets with the same price would also be close in their influence spreads. This property is valuable to the advertisers because the advertisers, holding a budget to purchase seeds according to their prices, are expecting to achieve a more predictable influence spread from the selected seeds. To verify this property, we design the experiments to construct a collection of seed sets subject to a budget limit and compare their influence spreads. The budget limit is expressed as a percentage of b (the total price of all the candidate nodes). Given a budget, we randomly choose seeds among the candidate nodes to fill up the budget. We construct 10,000 random seed sets and estimate the influence spreads achieved by these seed sets using the RIS method with 100,000 RR sets. We test different budget limits from 10% to 90%.

Figures 1, 3 and 5 show the variance of the influence spreads of the 10,000 seed sets under different pricing algorithms when there are $n_c = 200$, 500 and 1000 candidate nodes respectively. It can be seen that different pricing algorithms result in quite different variances. In general, our OptPrice algorithm has significantly lower variance of influence spread than the baselines. This shows that our OptPrice algorithm can capture the influence potentials of the candidate nodes more accurately and give the advertisers a more stable and predictable return (influence spread) for their purchasing (seeding) activities. Figures 2, 4 and 6 show the average influence spread of the 10,000 seed sets. As can be seen, the average influence spread increases with the given budget of seed selection for all the datasets. The seed sets chosen under our proposed pricing algorithm achieve comparable average influence spreads to those under the baselines. This shows that our pricing profile is able to maintain the same expected influence spread as other baselines under a given budget.

7. CONCLUSION

In this work, we build a bridge between OSN providers and advertisers by proposing a pricing mechanism to facilitate the initiator selection of marketing campaigns without the knowledge of OSN structures. In particular, we study the problem of minimizing the pricing divergence from the influence spread and derive an optimal price profile. A scalable estimation algorithm is devised to yield an (ε, δ)-approximation of the optimal prices. Through extensive experiments, we demonstrate the performance advantages of our approach over other baselines.

8. ACKNOWLEDGMENTS

This work is supported by Singapore Ministry of Education Academic Research Fund Tier 1 under Grant 2018-T1-002-063, and by Singapore National Research Foundation under grant NRF-RSS2016-004.

9. REFERENCES

[1] C. Aslay, F. Bonchi, L. V. Lakshmanan, and W. Lu. Revenue maximization in incentivized social advertising. *PVLDB*, 10(11):1238–1249, 2017.

[2] C. Aslay, W. Lu, F. Bonchi, A. Goyal, and L. V. Lakshmanan. Viral marketing meets social advertising: Ad allocation with minimum regret. *PVLDB*, 8(7):814–825, 2015.

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

[4] P. Chalermsook, A. Das Sarma, A. Lall, and D. Nanongkai. Social network monetization via sponsored viral marketing. *ACM SIGMETRICS Performance Evaluation Review*, 43(1):259–270, 2015.

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

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

[7] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *Proc. IEEE ICDM*, pages 88–97, 2010.

[8] S. Cheng, H. Shen, J. Huang, W. Chen, and X. Cheng. Imrank: influence maximization via finding self-consistent ranking. In *Proc. ACM SIGIR*, pages 475–484, 2014.

[9] F. Chung and L. Lu. Concentration inequalities and martingale inequalities: a survey. *Internet Mathematics*, 3(1):79–127, 2006.

[10] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck. Sketch-based influence maximization and computation: Scaling up with guarantees. In *Proc. ACM CIKM*, pages 629–638, 2014.

[11] P. Dagum, R. Karp, M. Luby, and S. Ross. An optimal algorithm for monte carlo estimation. *SIAM Journal on Computing*, 29(5):1484–1496, 2000.

[12] P. Domingos and M. Richardson. Mining the network value of customers. In *Proc. ACM KDD*, pages 57–66, 2001.

[13] eMarketer. Social network ad spending worldwide, by region, 2013–2017. https://www.emarketer.com/Chart/Social-Network-Ad-Spending-Worldwide-by-Region-2013-2017/168356, 2015.

[14] eMarketer. Influencer marketing is about data, not celebrity deals. https://www.emarketer.com/Article/Influencer-Marketing-About-Data-Not-Celebrity-Deals/1016683, 2017.

[15] eMarketer. How much are brands paying influencers? https://www.emarketer.com/content/how-much-are-brands-paying-influencers, 2019.

[16] eMarketer. Is everyone on instagram an influencer? https://www.emarketer.com/content/is-everyone-on-instagram-an-influencer, 2019.

[17] https://newsroom.fb.com/company-info/.

[18] https://famebit.com/.

[19] K. Han, K. Huang, X. Xiao, J. Tang, A. Sun, and X. Tang. Efficient algorithms for adaptive influence maximization. *PVLDB*, 11(9):1029–1040, 2018.

[20] K. Huang, J. Tang, K. Han, X. Xiao, W. Chen, A. Sun, X. Tang, and A. Lim. Efficient approximation algorithms for adaptive influence maximization. *The VLDB Journal*, 2020.

[21] K. Huang, J. Tang, X. Xiao, A. Sun, and A. Lim. Efficient approximation algorithms for adaptive target profit maximization. In *Proc. IEEE ICDE*, pages 649–660, 2020.

[22] I. M. Hub. The remarkable rise of influencer marketing. https://influencermarketinghub.com/the-rise-of-influencer-marketing/, 2020.

[23] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proc. ACM KDD*, pages 137–146, 2003.

[24] A. Khan, B. Zehnder, and D. Kossmann. Revenue maximization by viral marketing: A social network host's perspective. In *Proc. IEEE ICDE*, pages 37–48, 2016.

[25] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *Proc. WWW*, pages 591–600, 2010.

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

[27] W. Lu and L. V. Lakshmanan. Profit maximization over social networks. In *Proc. IEEE ICDM*, pages 479–488, 2012.

[28] H. T. Nguyen, T. P. Nguyen, T. N. Vu, and T. N. Dinh. Outward influence and cascade size estimation in billion-scale networks. In *Proc. ACM SIGMETRICS*, pages 63–63, 2017.

[29] 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*, pages 695–710, 2016.

[30] Reuters. Social media ad spending is expected to pass newspapers by 2020. http://fortune.com/2016/12/05/social-media-ad-spending-newspapers-zenith-2020/, 2016.

[31] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *Proc. ACM KDD*, pages 61–70, 2002.

[32] J. Tang, K. Huang, X. Xiao, L. V. Lakshmanan, X. Tang, A. Sun, and A. Lim. Efficient approximation algorithms for adaptive seed minimization. In *Proc. ACM SIGMOD*, pages 1096–1113, 2019.

[33] J. Tang, X. Tang, X. Xiao, and J. Yuan. Online processing algorithms for influence maximization. In *Proc. ACM SIGMOD*, pages 991–1005, 2018.

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

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

[36] J. Tang, X. Tang, and J. Yuan. An efficient and effective hop-based approach for inluence maximization in social networks. *Social Network Analysis and Mining*, 8(10), 2018.

[37] 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*, 30(6):1095–1108, 2018.

[38] J. Tang, X. Tang, and J. Yuan. Towards profit maximization for online social network providers. In *Proc. IEEE INFOCOM*, pages 1178–1186, 2018.

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

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