Seed selection for viral marketing in online social networks : from influence maximization to profit maximization

Tang, Jing

2017

Tang, J. (2017). Seed selection for viral marketing in online social networks : from influence maximization to profit maximization. Doctoral thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/72857>

<https://doi.org/10.32657/10356/72857>

**NANYANG
TECHNOLOGICAL
UNIVERSITY**

SINGAPORE

**SEED SELECTION FOR VIRAL
MARKETING IN ONLINE SOCIAL
NETWORKS: FROM INFLUENCE
MAXIMIZATION TO PROFIT
MAXIMIZATION**

TANG Jing

**Interdisciplinary Graduate School
Multi-plAtform Game Innovation Centre (MAGIC)**

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2017

Abstract

Online Social Networks (OSNs) attract billions of users. Information can be disseminated widely and rapidly through OSNs with “word-of-mouth” effects. Viral marketing is such a typical application in which new products or commercial activities are advertised by some seed users in OSNs to other users in a cascading manner. A large amount of recent work has been focusing on viral marketing in OSNs. In this thesis, three seed selection problems in viral marketing are investigated. We start from the classical influence maximization. Then, we look at profit maximization for advertisers and OSN providers respectively that naturally combines the benefit with the cost of viral marketing.

Recent research has adopted the sampling approach to achieve a $(1 - 1/e - \varepsilon)$ -approximation guarantee for influence maximization. One fundamental step of this approach is to examine the approximation assurance of the seed set constructed under a given number of samples generated. We focus on this essential step and propose a framework to Maximize the online Approximation Guarantee (MAG). Our framework exploits instance-specific information during execution to construct online bounds that can potentially break the conventional approximation limit of $(1 - 1/e)$. The applications of MAG are two-fold. First, MAG can provide online approximation guarantees for runtime-restricted influence maximization in which only a limited amount of execution time is allowed to generate samples for influence estimation. Second, by deriving a better online approximation guarantee, MAG can be used to save the running time needed to reach an approximation target for traditional influence maximization.

The selection of initial seed users yields a tradeoff between the expense and reward of viral marketing. We define a profit metric that combines the benefit of influence spread with the cost of seed selection in viral marketing. We carry out a comprehensive study on finding a set of seed nodes to maximize the profit of viral marketing. We show that

the profit metric is significantly different from the influence metric in that it is no longer monotone. This characteristic differentiates the profit maximization problem from the traditional influence maximization problem. We develop new seed selection algorithms for profit maximization with strong approximation guarantees. We also derive several upper bounds to benchmark the practical performance of an algorithm on any specific problem instance.

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. We attempt 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, such a 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.

Extensive experimental evaluations with real OSN datasets demonstrate the effectiveness of our algorithms and techniques.

Acknowledgments

I would take this opportunity to express my deepest gratitude to my advisors in Nanyang Technological University (NTU). They are my supervisor **Prof. Tang Xueyan**, co-supervisor **Prof. Yuan Junsong**, and mentor **Prof. Cai Wentong**. They gave me many useful suggestions that inspired me in many aspects of my research direction. Without their guidance, the achievements of my research work will never be possible.

I would also like to thank all my fellow and colleagues for the projects we are working together, my relatives and friends for their encouragements and the staffs in Multiplatform Game Innovation Centre (MAGIC) for their technical supports. In particular, I appreciate Prof. Richard T.B. Ma, Prof. Xiao Xiaokui, Prof. Feng Liang, Prof. Li Yusen, Prof. Seah Hock Soon, Dr. Zhai Yiteng, Mr. Hou Yaqing, Mr. Wang Tianyi, Mr. Chen Caishun, Mr. Da Bingshui, Mr. Ren Runtian, Mr. Shi Chenwei and Ms. Elsie Sim Lee Boon. Working and living with them inspires my research and makes the Ph.D. life lively and interesting.

Last but not least, special thanks to all my family. I show my immense gratitude and love to my wife Ms. Du Jiali for all the support she did for my study and life. I would also like to show my great appreciation to my parents for their constant encouragement, understanding, support and love. Without these, I would not have finished my Ph.D. works easily.

Contents

Abstract	i
Acknowledgments	iii
List of Figures	viii
List of Tables	x
1. Introduction	1
1.1 Research Background	1
1.2 Research Motivations and Objectives	2
1.3 Core Contributions	4
1.4 Organization of the Thesis	5
2. Literature Review	8
2.1 Influence Spread in Online Social Networks	8
2.1.1 Independent Cascade Model	9
2.1.2 Linear Threshold Model	10
2.1.3 Other Diffusion Models	11
2.2 Viral Marketing in Online Social Networks	11
2.2.1 Influence Maximization	12
2.2.2 Profit Maximization	14
2.2.3 Other Marketing Strategies	15
2.3 Submodular Optimization	15
2.3.1 Submodular Function	15

2.3.2	Problem 1: Monotone Submodular Maximization with Cardinality Constraint	16
2.3.3	Problem 2: Unconstrained Submodular Maximization	17
2.3.4	Problem 3: Maximizing the Difference Between Submodular Functions	18
3.	Optimizing Approximation Guarantee for Influence Maximization in Online Social Networks	19
3.1	Introduction	19
3.2	Preliminaries	20
3.2.1	Influence Maximization	20
3.2.2	Reverse Influence Sampling	21
3.3	Framework Overview	22
3.3.1	Algorithm and Approximation Guarantee	22
3.3.2	Sampling Error	23
3.3.3	Framework Overview	23
3.4	Upper Bound of Maximum Achievable Coverage	25
3.5	Optimizing Sampling Error	27
3.5.1	Preliminaries	27
3.5.2	Fixing ε to Compute δ	28
3.5.3	Fixing δ to Compute ε	29
3.6	Extending MAG Framework for $(1 - 1/e - \varepsilon)$ -Approximation	36
3.7	Evaluation	41
3.7.1	Experimental Setup	41
3.7.2	Online Approximation Guarantee under The IC Model	43
3.7.3	Online Approximation Guarantee under The LT Model	45
3.7.4	Efficiency for Achieving $(1 - 1/e - \varepsilon)$ -Approximation	46
3.8	Summary	48
4.	Profit Maximization for Viral Marketing in Online Social Networks	49
4.1	Introduction	49
4.2	Problem Formulation	50

4.2.1	Preliminaries	50
4.2.2	The Profit Maximization Problem	51
4.3	Seed Selection Algorithms	53
4.3.1	Simple Greedy Heuristic	53
4.3.2	Double Greedy Algorithms	55
4.3.3	Warm-Start by Iterative Pruning	57
4.3.4	Generality and Time Complexity	61
4.4	Analysis of Upper Bounds	62
4.4.1	Upper Bound for Double Greedy	62
4.4.2	Upper Bounds Based on Submodularity	63
4.4.3	Tighten the Bounds	64
4.5	Evaluation	70
4.5.1	Experimental Setup	70
4.5.2	Profits Produced by Different Algorithms	72
4.5.3	Running Time	76
4.5.4	Iterative Pruning Technique	77
4.5.5	Upper Bounds	78
4.5.6	Weighted Benefit Setting	80
4.5.7	LT Model	80
4.6	Summary	81
5.	Towards Profit Maximization for Online Social Network Providers . .	83
5.1	Introduction	83
5.2	Problem Formulation	84
5.2.1	Preliminaries	84
5.2.2	The Profit Maximization Problem	85
5.3	Two-Phase Framework	87
5.3.1	Prune Search Space	87
5.3.2	Heuristic Algorithms	94
5.3.3	Discussions	95
5.4	Performance Analysis	95
5.4.1	Upper Bound of Optimum	96

5.4.2	Sampling Error	98
5.4.3	Reduce Sampling Error via Normalization	103
5.5	Evaluation	105
5.5.1	Experimental Setup	105
5.5.2	Profits Produced by Different Algorithms	106
5.5.3	Running Times of Different Algorithms	107
5.5.4	Iterative Pruning Technique	109
5.5.5	Online Approximation Guarantee of Solution Quality	110
5.5.6	Normalization	112
5.6	Summary	113
6.	Conclusion and Future Work	114
6.1	Conclusion	114
6.2	Limitations and Future Work	115
	Publications	118
	References	120

List of Figures

1.1	The organization of the thesis.	6
2.1	An social graph with 4 nodes and 4 edges.	10
3.1	Online approximation guarantee under the IC model ($k = 100$).	44
3.2	Online approximation guarantee under the LT model ($k = 100$).	45
3.3	Influence spread for influence maximization ($k = 100$).	46
3.4	Running time for influence maximization ($k = 100$).	47
3.5	Number of RR sets for influence maximization ($k = 100$).	47
4.1	A simple hill-climbing algorithm fails to achieve any bounded approxima- tion factor.	54
4.2	Double greedy algorithms fail to achieve any bounded approximation fac- tor when $\phi(V) < 0$	57
4.3	Profits produced under uniform cost setting, uniform benefit setting and IC model.	73
4.4	Profits produced under degree-proportional cost setting, uniform benefit setting and IC model.	74
4.5	Cost and benefit distributions by a single seed under degree-proportional cost setting, uniform benefit setting and IC model.	75
4.6	Upper bounds under weighted benefit setting and IC model on LiveJournal dataset.	79
4.7	Profits produced under degree-proportional cost setting, weighted benefit setting and IC model.	80
4.8	Profits produced under degree-proportional cost setting, weighted benefit setting and LT model.	81

5.1	Non-monotonicity and non-submodularity of the profit function $\phi(\cdot)$. . .	87
5.2	Iterative pruning under the Independent Cascade diffusion model.	92
5.3	Profits produced by different algorithms.	107
5.4	Running times of different algorithms.	108
5.5	Impact of iterative pruning technique on profit produced (algorithms with prefix “O-” do not use pruning technique).	110
5.6	Impact of iterative pruning technique on running time (algorithms with prefix “O-” do not use pruning technique).	111
5.7	Impact of normalization (algorithms with prefix “O-” do not use pruning technique and with postfix “-w/o” are without normalization).	112
6.1	Viral marketing in OSNs as a two-sided market.	116

List of Tables

3.1	Statistics of OSN datasets.	41
3.2	Online approximation guarantee for different seed set sizes k on the Live-Journal dataset when $2^{10} \times 10,000$ RR sets are generated under the IC model.	44
3.3	Online approximation guarantee for different seed set sizes k on the Live-Journal dataset when $2^{10} \times 10,000$ RR sets are generated under the LT model.	46
4.1	Running times of different algorithms (seconds).	76
4.2	Impact of iterative pruning.	77
4.3	Upper bounds (normalized by the profit of DGIP).	78
5.1	Search space reduction.	109
5.2	Online approximation guarantees of seed sets obtained.	111

Chapter 1

Introduction

1.1 Research Background

Online social networks (OSNs) are popular websites through which huge numbers of people communicate and share information. According to the statistics from Statista [102], as of April 2017, the most popular OSNs include Facebook (1.97 billion monthly active users or MAUs), WhatsApp (1.20 billion MAUs), Youtube (1.00 billion MAUs), WeChat (889 million MAUs), QQ (868 million MAUs), Tumblr (550 million MAUs), Instagram (600 million MAUs), Twitter (319 million MAUs), Weibo (313 million MAUs), etc. According to Nielsen’s latest report [92], people spent 20% of their PC time and 30% of their mobile time on social media, much more than that on other websites. Nowadays, the market of OSN advertisement is growing explosively due to the huge number of users in OSNs. Fortune [96] estimates that around 600 million US dollars of advertisement spending is expected to go to social media for the 2016 US president election.

Suppose that we are trying to market a product or campaign among a population of individuals. We can start out with some targeted individuals by offering these individuals with free samples of the products. An important question one must ask is whom we shall target. Individuals form complex social networks based on various interactions and relationships, according to which each individual’s decision to buy a product is often highly affected by her social friends and acquaintances besides her personal preference. Leveraging the “word-of-mouth” effects (social network effects), viral marketing in OSNs has become one of the most effective promotion channels for various products or campaigns [31]. The intuition behind is that the initial adopters can recommend a product

to their friends, who in turn may recommend it to their friends, and so forth. In this way, a cascade of recommendations is formed so that decisions can spread through the network from a small number of seed users to a potentially much larger population.

To make the viral marketing successful, a set of influential users in the OSN must be selected as seeds for spreading the information as widely as possible, which is known as *influence maximization*. Since the seminal work by Kempe et al. [63] that formulates the influence maximization problem with two basic diffusion models, namely the *Independent Cascade* (IC) and *Linear Threshold* (LT) models, a large amount of recent work [9, 18–20, 22, 23, 25, 41, 42, 49, 51, 52, 62, 72, 76, 90, 91, 93, 100, 101, 104, 106, 115, 116] has been focusing on influence maximization in viral marketing. Although finding the optimal seed set is NP-hard [63], a simple greedy algorithm has a $(1 - 1/e)$ -approximation guarantee due to the monotonicity and submodularity of the influence spread under these models [89].

1.2 Research Motivations and Objectives

It is #P-hard to compute the exact influence spread of a seed set under commonly adopted diffusion models such as IC and LT [18, 20]. Thus, sampling-based approaches are often used to estimate the influence spread, such as Monte-Carlo simulation [63] and Reverse Influence Sampling [9]. One fundamental step of the sampling-based approaches is to examine the approximation assurance of the seed set constructed under a given number of samples generated. The approximation guarantees derived by existing influence maximization algorithms are all restricted by the limit of $(1 - 1/e)$ which is the worst-case result of the greedy algorithm for maximizing monotone submodular functions. This constant approximation factor is often very loose for specific problem instances, leaving much space for optimization. By deriving a better estimate of the approximation guarantee than the fixed and instance-independent factor of $(1 - 1/e)$, it can help save the running time needed to reach an approximation target for influence maximization.

Note that influence maximization has assumed a fixed and pre-determined budget for seed selection. In essence, the cost of seed selection is the price to pay for viral marketing (e.g., providing the selected users with free samples or other incentives). The influence spread, on the other hand, is the reward of viral marketing, which can potentially be

translated into growth in the adoptions of products. Thus, the budget for seed selection reflects a tradeoff between the expense and reward of viral marketing. If the budget is set too low, it may not produce the desired extent of influence spread to fully exploit the potential of viral marketing in boosting sales and revenues. In contrast, if the budget is set too high, the benefit of the influence spread generated may not pay off the expense. A cost-effective budget setting should strike a balance between the expense and reward of viral marketing. Economic-wise, a common goal for companies conducting viral marketing is to maximize the profit return, which can be defined as the reward less the expense.

In addition, advertisers often delegate the operation of viral marketing campaigns to OSN providers who have the complete information of their social network structures [14]. 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 is willing to pay the OSN provider a commission for each user that adopts its product or shares its ads. Thus, 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. For example, for a cloud-based OSN running viral video marketing [86], 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.

This dissertation explores the above problems of seed selection for viral marketing in OSNs. The specific objectives of this thesis are two-fold: (1) to improve the approximation guarantee and computational efficiency of the classical influence maximization problem, and (2) to extend the goal of seed selection from influence maximization to profit maximization.

Due to the submodularity of the influence spread under commonly used diffusion models [6, 63, 98], our problems are essentially submodular maximization problems, which

are NP-hard in general. Analyzing the approximation guarantees of the algorithms can help to demonstrate their effectiveness. Furthermore, the influence spread is usually estimated via some sampling-based methods and the estimates are prone to sampling errors. Thus, to study the approximation guarantees, the sampling errors must also be analyzed. Therefore, our research focuses on developing effective seed selection solutions and analyzing these solutions with mathematical reasoning.

1.3 Core Contributions

First, we develop a framework to optimize the approximation guarantee for the classical influence maximization problem. Second, we define a profit metric that naturally combines the revenue of influence spread with the cost of seed selection to address their tradeoff and investigate the profit maximization problem for advertisers. Finally, we extend the profit maximization problem to OSN providers. The main contributions of this thesis are summarized as follows.

- **Optimizing Approximation Guarantee for Influence Maximization in Online Social Networks:**

In this study, we design a Maximizing online Approximation Guarantee (MAG) framework for influence maximization. MAG consists of two components: an upper bounding approach that can always certify an approximation guarantee no worse than $(1 - 1/e)$ for any instance of monotone submodular maximization, and a new method to analyze and optimize the sampling errors. We also extend our MAG framework to save the running time needed to achieve the conventional approximation guarantee of $(1 - 1/e - \varepsilon)$. We carry out extensive experiments with real OSN datasets to evaluate the performance of MAG against the state-of-the-art methods. The results demonstrate the superior performance of our MAG framework in terms of approximation guarantee and computational efficiency.

- **Profit Maximization for Viral Marketing in Online Social Networks:**

We define a general problem of profit maximization for viral marketing in OSNs. We show that the profit metric is submodular but not always monotone. We construct

a greedy hill-climbing algorithm and show that such an intuitive greedy algorithm does not have any bounded approximation factor for profit maximization. We present double greedy algorithms to optimize the profit. To expand the applicability of their approximation guarantees, we develop an iterative pruning technique to provide good warm-starts and relax the condition for the guarantees to hold. To provide online approximation guarantees and deal with cases where the required condition is not fulfilled, we derive several upper bounds on the optimal solution to the profit maximization problem. These bounds can be used to characterize the quality of the solutions constructed on any specific problem instance. We conduct extensive experiments with real OSN datasets. The results demonstrate the effectiveness of our profit maximization algorithms.

- **Towards Profit Maximization for Online Social Network Providers:**

In this work, 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 real OSN datasets. The results demonstrate the effectiveness and efficiency of our framework.

1.4 Organization of the Thesis

The organization of this thesis is visualized in Figure 1.1. The roadmap for the rest of this thesis is outlined as follows.

- Chapter 2 presents an overview of the recent research work on viral marketing in OSNs. In this chapter, the background knowledge of influence spread and its applications such as influence maximization and profit maximization are introduced. Meanwhile, the relevant methodologies for submodular optimization are discussed. The information provided through this chapter would help the readers better comprehend the main research of this thesis.

Figure 1.1: The organization of the thesis.

- Chapter 3 presents a study on optimizing the approximation guarantee of classical influence maximization, which is a problem of monotone submodular maximization with a cardinality constraint. A new sampling error analysis and a novel upper bounding approach are proposed to maximize the approximation guarantee derived for the seed set constructed. These techniques are also extended to speed up the achievement of the conventional $(1 - 1/e - \varepsilon)$ -approximation guarantee for influence maximization. To evaluate the proposed approaches, extensive experimental studies are conducted.
- Chapter 4 presents a study on profit maximization for viral marketing in OSNs. This is a problem of unconstrained submodular maximization. Several heuristics are proposed to address the problem, including simple greedy and double greedy algorithms. An iterative pruning technique is developed to provide good warm-starts and relax the condition for the theoretical guarantees to hold. Several upper bounds on the optimal solution are derived for evaluating the quality assurance of the solutions constructed. Extensive experiments with real OSN datasets are carried out to verify the effectiveness of our profit maximization algorithms.
- Chapter 5 presents a study on profit maximization for OSN providers. This problem is to maximize the difference between submodular functions. A general two-phase

framework is developed, which 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. The advantages of our pruning technique in improving the effectiveness and efficiency of any algorithm used in the search phase are theoretically proved. Several bounds on the maximum achievable profit are derived. Extensive experiments with real OSN datasets are conducted to demonstrate the effectiveness and efficiency of our framework.

- Chapter 6 concludes this thesis and outlines some potential future work in several directions.

Chapter 2

Literature Review

This chapter reviews the relevant research work in the domain of viral marketing in Online Social Networks (OSNs). We begin with an introduction of the influence spread in OSNs in Section 2.1, which is a core metric for measuring the effectiveness of viral marketing. Several diffusion models for influence propagation are also introduced in this section, including the widely used Independent Cascade and Linear Threshold models. Then, Section 2.2 surveys the hot problems of viral marketing, which mainly focuses on influence maximization and profit maximization. Finally, Section 2.3 reviews the submodular optimization methodologies that would be applied to seed selection for influence/profit maximization.

2.1 Influence Spread in Online Social Networks

Information can be disseminated widely and rapidly through OSNs with “word-of-mouth” effects. Leveraging OSNs as the medium for information spread has been increasingly adopted in many areas, such as viral marketing in which new products or commercial activities are advertised by some influential users in the OSN to other users in a cascading manner [31]. Kempe et al. [63] formulated two basic diffusion models for the processes by which influence propagates in social networks, namely the *Independent Cascade* (IC) and *Linear Threshold* (LT) models.

2.1.1 Independent Cascade Model

An OSN is usually abstracted as a directed graph $G = (V, E)$ with a set V of nodes (representing users) and a set E of edges (representing connections among users). For each directed edge $\langle u, v \rangle \in E$, we say that v is a *neighbor* of u , and u is an *inverse neighbor* of v . 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 the edge $\langle u, v \rangle$. Figure 2.1 gives an example of a social graph with 4 nodes and 4 edges. Each edge is associated with a propagation probability, e.g., $p_{v_1, v_2} = 0.5$.

Let N_u denote the set of node u 's neighbors, i.e., $N_u = \{v : v \in V, \langle u, v \rangle \in E\}$. Given a set of seed nodes S , the diffusion process of the IC model is described in Algorithm 1. Initially, at step 0, the seed nodes S are activated, while all the other nodes are not activated (line 1). When a node u first becomes activated at a step t (line 4), it has a single chance to activate its neighbors who are not yet activated. For each such neighbor $v \in N_u$, v would become activated with probability $p_{u,v}$ at step $t + 1$ (lines 5–7). Node u does not further attempt to activate any node beyond step $t + 1$. This process repeats until no more node can be activated (line 2). The *influence spread* of a seed set S , denoted by $\sigma(S)$, is defined as the expected number of nodes activated by the above process starting with S , i.e.,

$$\sigma(S) = \mathbb{E} \left[\left| \bigcup_t S_t \right| \right]. \quad (2.1)$$

Algorithm 1: IC(S)

```

1 initialize  $t \leftarrow 0$  and  $S_0 \leftarrow S$ ;
2 while  $S_t \neq \emptyset$  do
3    $S_{t+1} \leftarrow \emptyset$ ;
4   for each node  $u \in S_t$  do
5     for each node  $v \in N_u$  do
6       if  $v \notin \bigcup_{\tau=0}^t S_\tau$  and  $U(0, 1) \leq p_{u,v}$  then
7         update  $S_{t+1} \leftarrow S_{t+1} \cup \{v\}$ ;
8   update  $t \leftarrow t + 1$ ;
//  $U(0, 1)$  represents a value randomly generated from a uniform
   distribution between 0 and 1.
```

Figure 2.1: An social graph with 4 nodes and 4 edges.

For example, Figure 2.1 shows an social graph with 4 nodes and 4 edges. The value labeled on each edge describes its propagation probability, e.g., $p_{v_1, v_2} = 0.5$. If node v_1 is selected as a seed, then v_2 will be activated with probability 0.5 and v_3 will be activated with probability 0.4. The activation probability of v_4 can be derived as follows. Note that v_4 can be activated through two paths, i.e., $v_1 \rightarrow v_2 \rightarrow v_4$ and $v_1 \rightarrow v_3 \rightarrow v_4$. The probability of v_4 being activated through $v_1 \rightarrow v_2 \rightarrow v_4$ is $0.5 \times 0.2 = 0.1$ and that through $v_1 \rightarrow v_3 \rightarrow v_4$ is $0.4 \times 0.3 = 0.12$. Thus, the probability that none of these two paths activates v_4 is $(1 - 0.1) \times (1 - 0.12) = 0.792$, which indicates that v_4 will be activated with probability $1 - 0.792 = 0.208$. Therefore, the influence spread of the seed set $\{v_1\}$ under the IC model is $1 + 0.5 + 0.4 + 0.208 = 2.108$.

2.1.2 Linear Threshold Model

Different from the IC model, the LT model requires the propagation probabilities to satisfy $\sum_{w \in I_v} p_{w, v} \leq 1$, where I_v denotes the set of node v 's inverse neighbors, i.e., $I_v = \{w : w \in V, (w, v) \in E\}$. Another difference between the IC and LT models lies in the activation procedure. In the LT model, each node v uniformly and randomly selects a threshold λ_v from the interval $[0, 1]$. An inactive node v becomes activated only if $\sum_{w \in I_v, w \text{ is activated}} p_{w, v} \geq \lambda_v$. Algorithm 2 describes how the nodes are activated under the LT model. For example, in Figure 2.1, suppose again that node v_1 is selected as a seed. Then, the activation probabilities of v_2 and v_3 are still 0.5 and 0.4 respectively. For v_4 , the activation probability becomes $0.5 \times 0.2 + 0.4 \times 0.3 = 0.22$. Therefore, the influence spread of the seed set $\{v_1\}$ under the LT model is $1 + 0.5 + 0.4 + 0.22 = 2.12$.

Algorithm 2: LT(S)

```

1 randomly assign every node  $v \in V$  with a threshold  $\lambda_v$  from the interval  $[0, 1]$ ;
2 initialize  $t \leftarrow 0$  and  $S_0 \leftarrow S$ ;
3 while  $S_t \neq \emptyset$  do
4    $S_{t+1} \leftarrow \emptyset$ ;
5   for each node  $u \in S_t$  do
6     for each node  $v \in N_u$  do
7       if  $v \notin \bigcup_{\tau=0}^t S_\tau$  and  $\sum_{w \in I_v, w \text{ is activated}} p_{w,v} \geq \lambda_v$  then
8          $\text{update } S_{t+1} \leftarrow S_{t+1} \cup \{v\}$ ;
9    $\text{update } t \leftarrow t + 1$ ;

```

2.1.3 Other Diffusion Models

The *triggering model* [63] generalizes the IC and LT models. It randomly assigns each node v with a subset T_v of its inverse neighbors according to a *triggering distribution*. The diffusion process works as follows. All the nodes in the seed set S are initially activated. Any other node v would become activated if any inverse neighbor in its triggering set T_v is activated. This process repeats until no more node can be activated.

Continuous-time models [17, 30, 32, 73, 81, 82, 98] assume that the influence propagates through the OSN within a time window. In these models, each edge $e \in E$ is associated with a transmission time (or length) distribution $t(e)$. The nodes can only be activated within the time window since the propagation starts from the seed nodes.

Topic-aware models [6, 15, 113] consider each edge $\langle u, v \rangle$ to be associated with a propagation probability $p_{u,v}^z$ on each topic z . A node u can influence each of its neighbors v via a mixed probability dependent on the topics involved in the queries.

2.2 Viral Marketing in Online Social Networks

Viral marketing in OSNs has emerged as a new way to promote the sales of products. Domingos and Richardson [31, 95] were the first to exploit social influence for marketing optimization by modeling social networks as Markov random fields. Kempe et al. [63] formulated influence maximization as a discrete optimization problem, which targets at finding a fixed-size set of seed nodes to produce the largest influence spread. They showed

that the influence spreads under the IC and LT models are monotone and submodular and thus, they proposed a simple hill-climbing greedy algorithm to address the problem.¹ For monotone submodular maximization, the greedy algorithm could provide a $(1 - 1/e)$ -approximation guarantee although finding the optimal solution is NP-hard in general [89]. The greedy algorithm is described in Algorithm 3, where k is the given number of seed nodes to pick. In each step, it selects the node with the largest marginal influence gain of $\sigma(S \cup \{v\}) - \sigma(S)$ (line 3) and adds it to the seed set (line 4).

Algorithm 3: Greedy

```

1 initialize  $S \leftarrow \emptyset$ ;
2 for  $i \leftarrow 1$  to  $k$  do
3   find  $u \leftarrow \arg \max_{v \in V \setminus S} \{\sigma(S \cup \{v\}) - \sigma(S)\}$ ;
4    $S \leftarrow S \cup \{u\}$ ;
5 return  $S$ ;
```

2.2.1 Influence Maximization

Motivated by Kempe et al.’s seminal work [63], there has been considerable research concentrating on improving the implementation efficiency of the greedy algorithm for large-scale OSNs [9, 18–20, 22, 23, 25, 41, 42, 49, 51, 52, 56, 62, 72, 76, 90, 91, 93, 100, 101, 104–106, 109, 115, 116]. According to the seed selection processes of these approaches, we categorize them into two types, i.e., sampling-based and heuristic.

- **Sampling-Based Methods:** Computing the exact influence spread $\sigma(S)$ of a seed set S has been shown to be #P-hard for both the IC and LT models [18, 20]. Kempe et al. [63] used Monte-Carlo simulation to estimate the influence spread, which is very inefficient. Leskovec et al. [76] proposed a Cost-Effective Lazy Forward (CELf) algorithm that can accelerate Kempe et al.’s greedy algorithm by up to 700 times. The lazy forward approach was later further improved to CELf++ [51]. Wang et al. [109] proposed a community-based greedy algorithm to reduce the running time by more than an order of magnitudes. A more efficient Upper Bound

¹Besides the IC and LT models, the influence spreads under the aforementioned triggering model, continuous-time models, and topic-aware models are also submodular.

based Lazy Forward algorithm (UBLF) was proposed by Zhou et al. [115, 116] to avoid unnecessary influence estimation via an upper bounding approach. In addition, SAA [100], StaticGreedy [23] and PMC [93] were proposed to generate a fixed number of samples in advance and use these samples to estimate the influence spread of any node set. To address the efficiency problem, Borgs et al. [9] introduced a novel Reverse Influence Sampling approach to dramatically facilitate the influence estimation. Some follow-up work, such as TIM⁺ [106], IMM [105], BCT [90] and SSA [56, 91], focused on reducing the number of samples generated for estimating the influence spread and thus improving the efficiency of influence maximization. Some other optimization approaches, such as a sketch-based influence maximization (SKIM) method [25] and a community-based greedy algorithm (CGA) via a divide-and-conquer approach [101], were also used to improve the efficiency of the sampling-based methods. One of the most important advantages of the sampling-based approaches is that they can provide a $(1 - 1/e - \varepsilon)$ -approximation guarantee for influence maximization.

- **Heuristics:** The main characteristic of heuristic algorithms is to trade accuracy for efficiency. Chen et al. [19] developed degree discount heuristics that are very efficient. Later, they further proposed to compute the influence spread via maximum influence paths to further improve the efficiency of influence maximization algorithms, namely PMIA [18] under the IC model and LDAG [20] under the LT model. Goyal et al. [52] proposed the SIMPATH algorithm to address several limitations of LDAG for influence maximization under the LT model. Due to the fact that the majority of influence spread is produced within the first few hops of propagation as observed by several measurement-driven studies on real OSNs [13, 44, 75], the OGIS algorithm developed by Lee et al. [72] roughly estimates the influence within 2 hops while Tang et al. [104] proposed OneHop and TwoHop algorithms that can compute the *exact* influence spread for one-hop and two-hop of propagations. Rather than estimating influence locally by the aforementioned heuristics, some algorithms perform global estimations. Goyal et al. [49] leverage available diffusion traces to estimate expected influence spread. Jung et al. [62] took advantage of influence ranking to maximize the influence spread efficiently. Cheng et al. [22]

proposed IMRank that iterates multiple times to reorder the nodes according to their ranking-based marginal influence spreads computed. Galhotra et al. [41, 42] estimated the influence spread via a score-assignment procedure.

Arora et al. [4] compared various algorithms with extensive diffusion models and datasets in their benchmark work. They showed that no single state-of-the-art technique in influence maximization can stand strong on all three pillars: quality of spread, running time efficiency, and main memory footprint. There is also some complementary work on learning the parameters of diffusion models [7, 21, 46–48, 70]. In addition, some recent work [27, 50, 83, 114] studied the seed minimization problem that focuses on minimizing the seed set size (or cost) for achieving a given amount of influence spread, which is the dual problem of influence maximization.

2.2.2 Profit Maximization

Maximizing the influence spread alone has been shown to be ineffective for optimizing the profit return of viral marketing [103]. 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 [85, 103, 117]. These studies considered the cost of seed selection which is modular and implies that their profit metric is still submodular. They investigated greedy 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 [14, 64]. Therefore, the OSN providers are able to run viral marketing campaigns more efficiently than the advertisers. From the OSN provider’s perspective, its profit is given by the benefit of influence spread less the cost of information diffusion over the social network. This profit metric is the difference between two submodular functions, which is neither submodular nor monotone. From the algorithmic perspective, such a profit maximization problem generalizes those in [85, 103, 117] since a modular cost function is a special case of submodular cost functions.

2.2.3 Other Marketing Strategies

Besides the seed selection for viral marketing, some researchers looked at other marketing strategies such as pricing. Li et al. [78] modeled the product advertisement in large-scale OSNs through local mean field analysis. The model is designed to compute the expected proportion of users who would eventually buy the product, which may indirectly guide the advertisement to improve the profit. However, no specific strategy was given to maximize the profit. Hartline et al. [53] aimed to find the optimal marketing strategies by controlling the price and the order of sales to different customers to improve the profit. Some recent work [1, 5, 36, 37] studied strategic allocation of ads to users by leveraging social influence and the propensity of ads for viral propagation. The objectives are either to maximize the expected number of clicks or conversions or to help OSN hosts match their ad services with advertiser budgets as close as possible. Two recent studies [85, 117] also focused on finding the pricing strategies to optimize the profit return of viral marketing and they adopted a simple hill-climbing heuristic to select initial seed nodes. Lu et al. [84] considered competitive viral marketing that two or more players compete with similar products on the same network. Lei et al. [74] proposed to learn influence probabilities and maximize influence spread simultaneously on the input data. Yang et al. [111] looked at the friending recommendation via a cascade manner. Some work [16, 55] considered influence maximization with uncertainty in the edge influence probability estimates.

2.3 Submodular Optimization

Since the influence spread under many diffusion models, including the commonly used IC and LT models, is submodular [6, 63, 98], the influence and profit maximization problems for viral marketing in OSNs are closely related to submodular optimization.

2.3.1 Submodular Function

A set function $f : 2^V \rightarrow \mathbb{R}$ is *submodular* [89] iff for all $S, T \subseteq V$, it holds that $f(S) + f(T) \geq f(S \cup T) + f(S \cap T)$. Alternatively, defining $f(v \mid S) \triangleq f(S \cup \{v\}) - f(S)$ as the marginal gain of adding an element $v \in V$ to a set $S \subseteq V$, an equivalent definition

of a submodular set function f is that $f(v \mid S) \geq f(v \mid T)$ for all $S \subseteq T$ and $v \in V \setminus T$. The latter form of definition describes the concept of *diminishing return* in economics. The function f is *monotone* (by which we mean nondecreasing) iff $f(S) \leq f(T)$ for all $S \subseteq T$, or equivalently $f(v \mid S) \geq 0$.

Many well known combinatorial optimization problems are essentially submodular maximization, such as maximum facility location [2, 28], Max-Cut and Max-DiCut [34, 45, 54]. Additionally, a growing number of problems in real-world applications are shown to be submodular maximization. These problems include data subset selection [67, 110], sensor placement [68, 69], document summarization [79, 80] and image segmentation [10, 29, 61], to name a few.

In what follows, we review the related work on the following three maximization problems where both functions f (which is monotone in the first problem) and g are submodular.

Problem 1: $\max_{|S| \leq k} f(S)$,

Problem 2: $\max f(S)$,

Problem 3: $\max f(S) - g(S)$.

Problem 1 represents the scenario of monotone submodular maximization with a cardinality constraint, which captures the classical influence maximization problem. Problem 2 represents the scenario of unconstrained submodular maximization, which captures the profit maximization problem from the advertiser's perspective. Problem 3 represents the scenario of maximizing the difference between submodular functions, which captures the profit maximization problem from the OSN provider's perspective.

2.3.2 Problem 1: Monotone Submodular Maximization with Cardinality Constraint

Monotone submodular maximization subject to a cardinality constraint is a well known combinatorial optimization problem. A classic result in [89] shows that a simple hill-climbing greedy algorithm (Algorithm 3) can provide a $(1 - 1/e)$ -approximation guarantee for maximizing a monotone submodular function with at most k elements selected. This

is the best achievable guarantee for the general family of monotone submodular functions [88]. The result is also known to be tight for the maximum coverage problem in particular [33], which is used by the sampling-based methods for influence maximization. Complementary to the fixed and instance-independent approximation factor established in these studies, Krause et al. [69] derived an instance-specific upper bound that is useful for evaluating the practical performance of the greedy algorithm for any monotone submodular function.

In addition, Khuller et al. [65] investigated monotone submodular maximization when the constraint is generalized to a knapsack, i.e., each node v is associated with a weighted cost $c(v)$ and the constraint is $\sum_{v \in S} c(v) \leq b$. They proposed a modified greedy algorithm that can provide an approximation factor of $(1 - 1/\sqrt{e})$ which is an offline guarantee that is independent of the problem instance. This algorithm has two candidate sets. The first one is a single element set $\{v^*\}$ for which $f(\{v^*\})$ is maximized. The second one is constructed by a greedy procedure that iteratively selects an element u with the maximum ratio $\frac{f(u|S)}{c(u)}$ and adds it to S if the budget allows. Finally, it returns the one with a larger function value from the two candidate sets. This offline factor can be improved to $(1 - 1/e)$ by a partial enumeration heuristic which enumerates all subsets of up to 3 elements [65]. However, this approach would increase the time complexity by a multiplicative factor of $O(n^3)$.

2.3.3 Problem 2: Unconstrained Submodular Maximization

Feige et al. [35] developed local search algorithms for approximately maximizing non-monotone submodular functions, including a deterministic $(1/3)$ -approximation algorithm and a randomized $(2/5)$ -approximation algorithm. They also proved that no algorithm can provide an approximation guarantee better than $1/2$. Gharan et al. [43] improved the approximation factor of the local search to 0.41 via a simulated annealing approach [66]. Feldman et al. [38] provided a further improved approximation factor of roughly 0.42 by studying the structural similarity. Recently, Buchbinder et al. [11] proposed double greedy algorithms that have much lower computational complexities than those in [35] and improve the approximation guarantees to match the known hardness

result of the problem, i.e., $(1/2)$ -approximation. Note that all these approximation algorithms require the objective function to be non-negative. Instead of providing instance-independent approximation guarantees, Iyer et al. [58,60] gave two modular upper bounds on submodular functions and made use of them to benchmark the performance of submodular optimization.

There is also some work on maximizing non-monotone submodular functions under knapsack constraints [71]. The proposed algorithms require submodular functions to be non-negative and have very high time complexities (at least $O(n^4 \log n)$ for computation and $O(n^{13})$ for sampling where n is the size of the ground set). Feldman et al. [39] proposed a unified continuous greedy algorithm that can provide $(1/e - o(1))$ -approximation. Buchbinder et al. [12] improved the approximation factor to a value in the range $[1/e + 0.004, 1/2]$. Vondrák [107] considered the scenario that exactly k elements have to be chosen, and proposed a $(1/4 - o(1))$ -approximation algorithm via a fractional local search. The approximation factor is further improved to be in the range $[0.356, 1/2]$ by Buchbinder et al. [12].

2.3.4 Problem 3: Maximizing the Difference Between Submodular Functions

Inspired by the convex-concave procedure [112], Narasimhan et al. [87] proposed a submodular-supermodular algorithm to minimize set functions which can be represented as a difference between submodular functions. The idea of this algorithm is to replace the second submodular function with its modular lower bound [40] and thus the objective is transformed to a submodular function which can be minimized [94,99]. Interestingly, they showed that any set function can be expressed as a difference between submodular functions. Hence, every combinatorial optimization problem of a set function can be reduced to minimizing a difference between submodular functions. Iyer et al. [57] showed that this problem is multiplicatively inapproximable unless $P=NP$ and proposed several heuristic methods to address the problem that extend the work in [87], including a supermodular-submodular procedure and a modular-modular procedure. In addition, the greedy algorithm [89] and simulated annealing (SA) algorithm [66] are also widely used heuristics for submodular optimization.

Chapter 3

Optimizing Approximation Guarantee for Influence Maximization in Online Social Networks

3.1 Introduction

As discussed in Section 2.2, many algorithms have been developed for influence maximization in OSNs. The heuristic algorithms trade the solution quality for efficiency and normally cannot provide any theoretical guarantees for the seed sets constructed. The sampling-based algorithms can provide $(1 - 1/e - \epsilon)$ -approximation guarantees by the standard result of the greedy hill-climbing method. Although the $(1 - 1/e)$ -approximation guarantee of the greedy algorithm is tight in the worst case [89], it is often rather loose for specific problem instances. In this chapter, we investigate the optimization of the approximation guarantee for influence maximization.

We define an *online approximation guarantee* metric to measure the quality of the seed set returned and propose a framework to Maximize the online Approximation Guarantee (MAG). Our framework exploits instance-specific information during execution to construct online bounds that can potentially break the conventional approximation limit of $(1 - 1/e)$. MAG can be used to derive online approximation guarantees for *runtime-restricted influence maximization* [9] that stipulates a pre-defined execution time to generate samples for influence estimation. Moreover, by providing a better estimate of the

approximation guarantee than the fixed and instance-independent factor of $(1 - 1/e)$, MAG can help save the running time needed to reach an approximation target for traditional influence maximization. To make use of our MAG framework, the sampling errors that naturally exist in the sampling-based methods must be analyzed. The previous analysis that fixes the error parameter ε to compute the confidence level [9, 90, 91, 105, 106] is not compatible with our goal of maximizing the online approximation guarantee. Once the error parameter ε is fixed, the approximation guarantee is also fixed. To address this issue, we develop a new analysis technique that fixes the confidence level to compute the error ε . Then, we can optimize the sampling errors for maximizing the online approximation guarantee.

The rest of this chapter is organized as follows. Section 3.2 formally defines the influence maximization problem and introduces the Reverse Influence Sampling approach proposed by Borgs et al. [9] for influence estimation. Section 3.3 overviews our MAG framework. Section 3.4 elaborates a new online bound for the maximum coverage problem. Section 3.5 conducts analysis to optimize the sampling errors. Section 3.6 extends our MAG framework for traditional influence maximization with the $(1 - 1/e - \varepsilon)$ -approximation. Section 3.7 presents the experimental study. Finally, Section 3.8 summarizes the chapter.

3.2 Preliminaries

3.2.1 Influence Maximization

We abstract an OSN as a directed graph $G = (V, E)$ with a set V of nodes (representing users) and a set E of edges (representing connections among users). For each directed edge $\langle u, v \rangle \in E$, we say that v is a *neighbor* of u , and u is an *inverse neighbor* of v .

To facilitate discussion, we focus on two basic and widely adopted diffusion models, i.e., the Independent Cascade (IC) and Linear Threshold (LT) models, which are discussed in Section 2.1. In the IC and LT models, each edge $\langle u, v \rangle$ is associated with a propagation probability $p_{u,v}$, representing the probability for u to influence v . Given a set of seed nodes S , the diffusion process proceeds as follows. Initially, the seed nodes S are *activated* and the other nodes are *inactive*. When a node u first becomes activated, it

tries to further activate its inactive neighbors. This process repeats until no more node can be activated.

- *IC model.* When a node u first becomes activated, it has a *single chance* to activate each inactive neighbor v with a probability $p_{u,v}$. After that, u stops activating any other nodes.
- *LT model.* It requires the propagation probabilities to satisfy $\sum_{u \in I_v} p_{u,v} \leq 1$, where I_v denotes the set of node v 's inverse neighbors. Each node v uniformly and randomly selects a threshold λ_v from the interval $[0, 1]$. An inactive node v becomes activated only if $\sum_{u \in A_v} p_{u,v} \geq \lambda_v$, where $A_v \subseteq I_v$ denotes the set of v 's inverse neighbors that are activated.

Let $\sigma(S)$ denote the expected number of nodes activated by the seed set S via the above process. We refer to $\sigma(S)$ as the *influence spread* of the seed set S . Given a graph $G = (V, E)$ with a propagation probability $p_{u,v}$ for each edge $\langle u, v \rangle \in E$, the *influence maximization* problem [63] is to seek for a set S of k nodes that maximizes the influence spread $\sigma(S)$. Formally,

$$\max_{|S|=k} \sigma(S). \quad (3.1)$$

3.2.2 Reverse Influence Sampling

Computing the exact influence spread $\sigma(S)$ of a seed set S has been shown to be #P-hard for both the IC and LT models [18, 20]. Kempe et al. [63] used Monte-Carlo simulation to estimate the influence spread, which is very inefficient. To address the efficiency problem, Borgs et al. [9] introduced a novel Reverse Influence Sampling approach to dramatically facilitate the influence estimation.

Definition 3.1 (RR Set). *A random Reverse Reachable (RR) set R for a graph G is generated by 1) first selecting a random node $v \in V$, 2) then sampling a random graph G' from G according to the diffusion model, 3) finally taking the set of nodes in G' that can reach v as R .*

Intuitively, for a random RR set generated from a randomly selected node v , a diffusion process starting from a seed set that intersects with the RR set should have a certain probability to activate v . The following lemma formalizes the connection between the influence spread and RR sets.

Lemma 3.1 ([9]). *Given a fixed seed set $S \subseteq V$, for a random RR set R ,*

$$\sigma(S) = n \cdot \Pr[S \cap R \neq \emptyset], \quad (3.2)$$

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

Lemma 3.1 indicates that the influence spread of a seed set S is proportional to the probability that S intersects with a random RR set. We say that a seed set S *covers* a random RR set R if S contains at least one node in R . Thus, given a set of random RR sets generated for influence estimation, the influence maximization problem is transformed into a *maximum coverage* problem [33], which aims to select a set S of k nodes to cover the maximum number of RR sets.

3.3 Framework Overview

The challenges for influence maximization via the Reverse Influence Sampling approach are two-fold. (i) For the maximum coverage problem, how to construct a seed set solution to maximize the coverage and how close is the coverage of the constructed solution to the optimum? (ii) Back to the original influence maximization problem, how good is the seed set constructed? The former is characterized by the approximation factor of the algorithm designed for the maximum coverage problem, while the latter is characterized by the sampling errors of the Reverse Influence Sampling approach.

3.3.1 Algorithm and Approximation Guarantee

A simple hill-climbing greedy heuristic can achieve a $(1 - 1/e)$ -approximation [89] for maximizing the coverage which is monotone and submodular. Feige [33] showed that for any $\varepsilon > 0$, the maximum coverage problem cannot be approximated in polynomial time with a ratio of $(1 - 1/e + \varepsilon)$, unless $P = NP$. Thus, we make use of the greedy heuristic

as it is a good algorithm for seed selection. Though we cannot further optimize the approximation factor of $(1 - 1/e)$ for the maximum coverage problem in general, as shall be shown later, it is possible to certify an *online approximation guarantee* much better than $(1 - 1/e)$ for the solution constructed by the greedy algorithm. In contrast to the *offline* approximation guarantee that is the worst-case ratio and instance-independent, the *online* approximation guarantee that we shall derive measures the solution quality in an instance-specific manner.

3.3.2 Sampling Error

The influence spread estimated by the RR sets generated may not be accurate and is prone to sampling errors. Given a fixed number of RR sets generated, there is a tradeoff between the sampling error bound and its confidence level. To maximize the online approximation guarantee, we conduct analysis to minimize the sampling error bound under a required confidence level. Then, together with the approximation factor of the maximum coverage problem, we can maximize the online approximation guarantee.

3.3.3 Framework Overview

We first introduce some key notations used in our framework. Let $\Lambda(S)$ denote the coverage of S on a group $\mathcal{R}$ of RR sets, which is defined by $\Lambda(S) = \sum_{i=1}^{|\mathcal{R}|} \mathbf{1}_{\{S \cap R_i \neq \emptyset\}}$, where $\mathbf{1}_{\{S \cap R_i \neq \emptyset\}} = 1$ if and only if $S \cap R_i \neq \emptyset$. Let $\theta = |\mathcal{R}|$ be the total number of RR sets. Then, the influence spread $\sigma(S)$ of a seed set S in (3.2) can be rewritten as

$$\sigma(S) = n \cdot \frac{\mathbb{E}[\Lambda(S)]}{\theta}. \quad (3.3)$$

Algorithm 4 shows an overview of our framework for Maximizing the Approximation Guarantee (MAG) of influence spread under a given number of RR sets generated. The framework is built on the two-fold analysis in the above sections. In a nutshell, Algorithm 4 (i) constructs a seed set and derives an upper bound on the maximum achievable coverage via a greedy approach (lines 1–7), and (ii) computes the online approximation guarantee α for influence maximization via a sampling error analysis (lines 8 and 9). Specifically, we first use the greedy algorithm to construct a seed set S_k of k nodes to maximize the coverage on the RR sets. In the i -th iteration, it selects the node with

Algorithm 4: $\text{MAG}(k, \delta, \mathcal{R}, \mathcal{R}')$

Input: Target seed set size k , the required confidence $1 - \delta$, and two groups of random RR sets $\mathcal{R}$ and $\mathcal{R}'$
Output: A seed set S_k and its online approximation guarantee α (taking into account sampling errors)

```

1 initialize  $S_0 \leftarrow \emptyset, \hat{\Lambda} \leftarrow \infty$ ;
2 for  $i \leftarrow 1$  to  $k$  do                                     // on RR sets  $\mathcal{R}$ 
3   find  $u \leftarrow \arg \max_{v \in V \setminus S_{i-1}} \{\Lambda(S_{i-1} \cup \{v\}) - \Lambda(S_{i-1})\}$ ;
4    $S_i \leftarrow S_{i-1} \cup \{u\}$ ;
5   compute an upper bound based on  $S_{i-1}$ ;                       // Section 3.4
6   if  $\hat{\Lambda}$  is larger than the  $S_{i-1}$ -based upper bound then
7     | update  $\hat{\Lambda} \leftarrow S_{i-1}$ -based upper bound;
8 compute the validated coverage  $\Lambda'(S_k)$  of  $S_k$  on the RR sets  $\mathcal{R}'$ ;
9 compute the online approximation guarantee  $\alpha$  via  $\hat{\Lambda}$  and  $\Lambda'(S_k)$ ; // Section 3.5
10 return solution  $S_k$  and the online approximation guarantee  $\alpha$ ;
```

the largest marginal coverage gain of $\Lambda(S_{i-1} \cup \{v\}) - \Lambda(S_{i-1})$ (line 3) and adds it to the seed set (line 4). Meanwhile, in the i -th iteration of the greedy algorithm, we also compute the S_{i-1} -based upper bound on the maximum achievable coverage (line 5). The tightest upper bound $\hat{\Lambda}$ over all iterations is recorded (lines 6 and 7). We shall show how to compute the S_{i-1} -based upper bound and prove that the online approximation guarantee certified by $\hat{\Lambda}$ is never worse than $(1 - 1/e)$ in Section 3.4. Then, we compute the online approximation guarantee α by optimizing the sampling errors with confidence $1 - \delta$ (line 9). This is done via the upper bound $\hat{\Lambda}$ computed and S_k 's coverage $\Lambda'(S_k)$ validated on another group $\mathcal{R}'$ of random RR sets (line 8), which will be elaborated in Section 3.5.

Our framework can derive the online approximation guarantee of influence maximization on a given group of random RR sets. Since the running time of an influence maximization algorithm based on the Reverse Influence Sampling approach is mainly spent for generating RR sets [105], our framework can be directly applied for runtime-restricted influence maximization in which only a limited amount of execution time is allowed to generate RR sets. On the other hand, traditional influence maximization aims to achieve the $(1 - 1/e - \varepsilon)$ -approximation guarantee with confidence at least $1 - \delta$. Our framework can also be easily extended to cope with this target by gradually increasing

the number of RR sets generated until the online approximation guarantee obtained is no less than $(1 - 1/e - \varepsilon)$. We shall discuss the extension in Section 3.6.

3.4 Upper Bound of Maximum Achievable Coverage

To maximize the coverage of k nodes, the greedy algorithm can provide a $(1 - 1/e)$ -approximation guarantee due to the submodularity of the coverage [89]. In this section, we derive new upper bounds that can potentially break the approximation limit of $(1 - 1/e)$ for specific problem instances.

Let $\Lambda(v \mid S)$ denote the marginal coverage gain of adding a node v to a seed set S , which is defined by

$$\Lambda(v \mid S) = \Lambda(S \cup \{v\}) - \Lambda(S). \quad (3.4)$$

Let $\Lambda(S, l)$ denote the total of the largest l marginal coverage gains of adding nodes to S , which can be written as

$$\Lambda(S, l) = \max_{|T|=l} \sum_{v \in T} \Lambda(v \mid S). \quad (3.5)$$

By definition, it is easy to obtain that $\Lambda(S, l_1) \geq \Lambda(S, l_2)$ for any $l_1 \geq l_2$ and any seed set S .

The following theorem derives an upper bound of the maximum achievable coverage from any given seed set S .

Theorem 3.1. *For any given seed set S , we have*

$$\max_{|T|=k} \Lambda(T) \leq \Lambda(S) + \Lambda(S, k), \quad (3.6)$$

Proof. Let $S_\Lambda^* = \arg \max_{|T|=k} \Lambda(T)$ be an optimal seed set that maximizes the coverage. Then, we have

$$\begin{aligned} \max_{|T|=k} \Lambda(T) &= \Lambda(S_\Lambda^*) \\ &\leq \Lambda(S \cup S_\Lambda^*) \\ &\leq \Lambda(S) + \sum_{v \in S_\Lambda^* \setminus S} \Lambda(v \mid S) \\ &\leq \Lambda(S) + \Lambda(S, |S_\Lambda^* \setminus S|) \\ &\leq \Lambda(S) + \Lambda(S, k). \end{aligned}$$

where the first two inequalities are due to the monotonicity and submodularity of $\Lambda(\cdot)$ respectively, the third inequality is by the definition of $\Lambda(S, |S_\Lambda^* \setminus S|)$ and the last inequality is because that there are at most k nodes in $S_\Lambda^* \setminus S$. $\square$

Theorem 3.1 shows that for any given seed set S , $\Lambda(S) + \Lambda(S, k)$ is an upper bound on the optimum achievable coverage, called S -based upper bound. Let S_i be the seed set constructed by the greedy algorithm after the i -th iteration for any $0 \leq i \leq k$. Then, S_k is the seed set solution returned by the greedy algorithm. We observe that the smallest $\Lambda(S_i) + \Lambda(S_i, k)$ among all $0 \leq i \leq k$ is still an upper bound of the maximum achievable coverage. Thus, it can be used to certify the online approximation guarantee of S_k . We prove that such a guarantee is never worse than $(1 - 1/e)$.

Theorem 3.2. *Let $\hat{\Lambda} = \min_{0 \leq i \leq k} \{\Lambda(S_i) + \Lambda(S_i, k)\}$ be the smallest upper bound over all $0 \leq i \leq k$. The greedy algorithm obtains an online approximation guarantee of $\Lambda(S_k)/\hat{\Lambda}$ that satisfies*

$$\Lambda(S_k)/\hat{\Lambda} \geq 1 - (1 - 1/k)^k \geq 1 - 1/e. \quad (3.7)$$

Proof. By the definition of the greedy algorithm, for any $0 \leq i \leq k - 1$, we have

$$\Lambda(S_{i+1}) - \Lambda(S_i) = \max_{v \in V} \Lambda(v \mid S_i). \quad (3.8)$$

Thus,

$$\Lambda(S_i) + k(\Lambda(S_{i+1}) - \Lambda(S_i)) \geq \Lambda(S_i) + \Lambda(S_i, k) \geq \hat{\Lambda}. \quad (3.9)$$

Rearranging (3.9) yields

$$\hat{\Lambda} - \Lambda(S_{i+1}) \leq (1 - 1/k) \cdot (\hat{\Lambda} - \Lambda(S_i)). \quad (3.10)$$

Recursively, we have

$$\hat{\Lambda} - \Lambda(S_k) \leq (1 - 1/k)^k \cdot (\hat{\Lambda} - \Lambda(S_0)) = (1 - 1/k)^k \cdot \hat{\Lambda}. \quad (3.11)$$

The equality is due to $\Lambda(S_0) = \Lambda(\emptyset) = 0$ by definition. As a result, it holds that

$$\Lambda(S_k)/\hat{\Lambda} \geq 1 - (1 - 1/k)^k \geq 1 - 1/e. \quad (3.12)$$

Hence, the theorem is proved. $\square$

Remark: Based on the above proof, $\Lambda(S_k)/\hat{\Lambda} = 1 - (1 - 1/k)^k$ holds only if $\Lambda(S_i) + k(\Lambda(S_{i+1}) - \Lambda(S_i)) = \Lambda(S_i) + \Lambda(S_i, k)$ and $\Lambda(S_i) + \Lambda(S_i, k) = \Lambda(S_j) + \Lambda(S_j, k)$ for any $0 \leq i, j \leq k-1$. Let v_i denote the node selected at the i -th iteration. Then, to satisfy the first condition, the largest k marginal coverage gains must be the same in each iteration, i.e., they are all equal to $\Lambda(v_i | S_{i-1})$ at the i -th iteration. The second condition implies that $\Lambda(S_{i-1}) + k \cdot \Lambda(v_i | S_{i-1}) = \Lambda(S_i) + k \cdot \Lambda(v_{i+1} | S_i) = \Lambda(S_{i-1}) + \Lambda(v_i | S_{i-1}) + k \cdot \Lambda(v_{i+1} | S_i)$, which suggests that $\Lambda(v_{i+1} | S_i) = (1 - 1/k) \cdot \Lambda(v_i | S_{i-1})$. Therefore, to satisfy the second condition, the largest marginal coverage gain must be in geometric reduction with a multiplicative factor of $(1 - 1/k)$ over all iterations. In practice, these conditions rarely hold together and our derived online approximation guarantee can be much tighter than the factor of $1 - (1 - 1/k)^k \geq 1 - 1/e$.

We also remark that an earlier study on influence maximization [76] has developed an online bound on maximum coverage which can be written as $\Lambda(S_k) + \Lambda(S_k, k)$ in our notations. Although such a bound was empirically shown to provide a better online approximation guarantee than $(1 - 1/e)$, there is no theoretical assurance that the derived guarantee is always better than $(1 - 1/e)$. Our new upper bound is not only tighter than $\Lambda(S_k) + \Lambda(S_k, k)$ but also warrants a better guarantee than $(1 - 1/e)$.

3.5 Optimizing Sampling Error

3.5.1 Preliminaries

Let S_k be the set of k seeds constructed by the greedy algorithm and S^* be the optimal seed set for the influence maximization problem. There exist sampling errors for using the Reverse Influence Sampling approach to estimate the influence spread. Fundamentally, there are two types of sampling errors:

1. The sampling error on the influence spread $\sigma(S_k)$ of the solution constructed.
2. The sampling error on the optimal influence spread $\sigma(S^*)$.

The online approximation guarantee for S_k is derived by establishing a lower bound of $\sigma(S_k)/\sigma(S^*)$. To this end, a high-confidence lower bound of $\sigma(S_k)$ and a high-confidence upper bound of $\sigma(S^*)$ are needed. Recall that $\sigma(S_k)$ and $\sigma(S^*)$ are estimated by $\Lambda(S_k) \cdot \frac{n}{\theta}$

and $\Lambda(S^*) \cdot \frac{n}{\theta}$, respectively. Thus, we can mathematically describe the above two types of sampling errors as

$$\Pr \left[\Lambda(S_k) \cdot \frac{n}{\theta} \leq (1 + \varepsilon_1) \cdot \sigma(S_k) \right] \geq 1 - \delta_1, \quad (3.13)$$

and

$$\Pr \left[\Lambda(S^*) \cdot \frac{n}{\theta} \geq (1 - \varepsilon_2) \cdot \sigma(S^*) \right] \geq 1 - \delta_2. \quad (3.14)$$

Chernoff bounds (or Chernoff-like bounds via martingale) [24] are normally used to analyze these sampling errors. We shall make use of the following variant of Chernoff-like bounds from [105] that exploits the martingale technique [24].

Lemma 3.2 ([105]). *Given a fixed number of θ random RR sets $\mathcal{R} = \{R_i : 1 \leq i \leq \theta\}$ and a seed set S , let Z_i denote a random variable that satisfies $Z_i = 1$ if $S \cap R_i \neq \emptyset$ and $Z_i = 0$ otherwise. Consider the sum $Z = \sum_{i=1}^{\theta} Z_i$ with expectation $\mathbb{E}[Z]$. Then, for any $\lambda > 0$, we have*

$$\Pr [Z - \mathbb{E}[Z] \geq \lambda] \leq \exp \left(-\frac{\lambda^2}{2\mathbb{E}[Z] + \frac{2}{3}\lambda} \right), \quad (3.15)$$

$$\Pr [Z - \mathbb{E}[Z] \leq -\lambda] \leq \exp \left(-\frac{\lambda^2}{2\mathbb{E}[Z]} \right). \quad (3.16)$$

Basically, (3.13) and (3.14) can be derived from (3.15) and (3.16) respectively by making the following settings. Let $Z = \Lambda(S_k)$, $\lambda = \varepsilon_1 \mathbb{E}[Z]$ and $\exp \left(-\frac{\lambda^2}{2\mathbb{E}[Z] + \frac{2}{3}\lambda} \right) = \delta_1$ in (3.15). Let $Z = \Lambda(S^*)$, $\lambda = \varepsilon_2 \mathbb{E}[Z]$ and $\exp \left(-\frac{\lambda^2}{2\mathbb{E}[Z]} \right) = \delta_2$ in (3.16).

3.5.2 Fixing ε to Compute δ

Recent research [9, 90, 91, 105, 106] has focused on generating a sufficient number of RR sets to establish an (ε, δ) -approximation, i.e., achieving a conventional approximation guarantee of $(1 - 1/e - \varepsilon)$ with confidence at least $1 - \delta$. Given a fixed number of θ RR sets, these studies essentially compute δ_1 and δ_2 under some pre-defined ε_1 and ε_2 based on Lemma 3.2, i.e.,

$$\delta_1 = \exp \left(-\frac{\varepsilon_1^2 \cdot \theta \cdot \sigma(S_k)}{(2 + \frac{2}{3}\varepsilon_1)n} \right), \quad (3.17)$$

$$\delta_2 = \exp \left(-\frac{\varepsilon_2^2 \cdot \theta \cdot \sigma(S^*)}{2n} \right). \quad (3.18)$$

To achieve the overall approximation guarantee of $(1 - 1/e - \varepsilon)$, the pre-determined ε_1 and ε_2 shall satisfy $(1 - 1/e) \frac{\varepsilon_1 + \varepsilon_2}{1 + \varepsilon_1} \leq \varepsilon$ so that $\frac{\sigma(S_k)}{\sigma(S^*)} \geq \frac{\Lambda(S_k)}{\Lambda(S^*)} \cdot \frac{1 - \varepsilon_2}{1 + \varepsilon_1} \geq (1 - 1/e) \cdot \frac{1 - \varepsilon_2}{1 + \varepsilon_1} \geq 1 - 1/e - \varepsilon$. Then, it is checked whether the δ_1 and δ_2 obtained satisfy the requirement $\delta_1 + \delta_2 \leq \delta$. The number of RR sets θ is gradually increased until the required confidence $1 - \delta$ is satisfied.

However, the above method is not compatible with our target of maximizing the online approximation guarantee. By fixing the sampling errors ε_1 and ε_2 , this method requires a pre-determined approximation factor between $\Lambda(S_k)$ and $\Lambda(S^*)$ in the maximum coverage problem to optimize the confidence probabilities $1 - \delta_1$ and $1 - \delta_2$. On the other hand, the upper bound of $\Lambda(S^*)$ proposed in Section 3.4 is calculated during the seed selection process and is not known a priori. Thus, the “fixing ε to compute δ ” method cannot make use of our proposed upper bound to optimize the online approximation guarantee.

In addition, the above method also has some other drawbacks. First, δ is much more sensitive to ε than ε to δ . For example, in Lemma 3.2, let $\delta = \exp\left(-\frac{\varepsilon^2 \mathbb{E}[Z]}{2}\right)$. When doubling ε to 2ε , δ is reduced to δ^4 . In contrast, when doubling δ to 2δ , ε is only reduced by a relative factor of $\sqrt{\log_\delta(2\delta)}$ (which is decreasing with δ , e.g., $\sqrt{\log_\delta(2\delta)} = 0.95$ when $\delta = 10^{-3}$). Therefore, a slack setting of ε_1 and ε_2 would be much less effective in optimizing δ_1 and δ_2 compared to a slack setting of δ_1 and δ_2 in optimizing ε_1 and ε_2 . As a result, the former can lead to the need for a much larger number of RR sets than the latter.

Second, since both $\sigma(S_k)$ and $\sigma(S^*)$ are unknown in (3.17) and (3.18), the exact δ_1 and δ_2 values are hard to obtain. Thus, some lower bounds $\sigma^l(S_k)$ and $\sigma^l(S^*)$ of $\sigma(S_k)$ and $\sigma(S^*)$ (with high confidence) must be used to assure δ_1 and δ_2 can satisfy the confidence requirement $\delta_1 + \delta_2 \leq \delta$. In particular, $\sigma^l(S_k)$ is also used as the lower bound $\sigma^l(S^*)$ of $\sigma(S^*)$ in the studies of [90, 91, 105, 106], which leads to the dependence between the two types of sampling errors. This makes the analysis rather complicated. In fact, only a high-confidence upper bound $\sigma^u(S^*)$ of $\sigma(S^*)$ is needed for deriving the online approximation guarantee.

3.5.3 Fixing δ to Compute ε

To address the above issues, we propose a new method to analyze the sampling errors via fixing δ to compute ε . We first analyze the relation between the coverage $\Lambda(S)$ and

the real influence spread $\sigma(S)$ based on Lemma 3.2.

Theorem 3.3. *Given any value $\delta \in (0, 1)$, we have*

$$\Pr \left[\sigma(S) \geq \left(\left(\sqrt{\Lambda(S) + \frac{2a}{9}} - \sqrt{\frac{a}{2}} \right)^2 - \frac{a}{18} \right) \cdot \frac{n}{\theta} \right] \geq 1 - \delta, \quad (3.19)$$

$$\Pr \left[\sigma(S) \leq \left(\sqrt{\Lambda(S) + \frac{a}{2}} + \sqrt{\frac{a}{2}} \right)^2 \cdot \frac{n}{\theta} \right] \geq 1 - \delta, \quad (3.20)$$

where $a = \ln(1/\delta)$ is a function of δ .

Proof. We prove (3.19) first. According to (3.3), we have $\frac{\sigma(S) \cdot \theta}{n} = \mathbb{E}[\Lambda(S)]$. Thus,

$$\begin{aligned} & \Pr \left[\sigma(S) < \left(\left(\sqrt{\Lambda(S) + \frac{2a}{9}} - \sqrt{\frac{a}{2}} \right)^2 - \frac{a}{18} \right) \cdot \frac{n}{\theta} \right] \\ &= \Pr \left[\sqrt{\frac{\sigma(S) \cdot \theta}{n}} + \frac{a}{18} < \sqrt{\Lambda(S) + \frac{2a}{9}} - \sqrt{\frac{a}{2}} \right] \\ & \quad + \Pr \left[\sqrt{\frac{\sigma(S) \cdot \theta}{n}} + \frac{a}{18} < \sqrt{\frac{a}{2}} - \sqrt{\Lambda(S) + \frac{2a}{9}} \right] \\ &= \Pr \left[\sqrt{\frac{\sigma(S) \cdot \theta}{n}} + \frac{a}{18} < \sqrt{\Lambda(S) + \frac{2a}{9}} - \sqrt{\frac{a}{2}} \right] \\ &= \Pr \left[\sqrt{\mathbb{E}[\Lambda(S)] + \frac{a}{18}} < \sqrt{\Lambda(S) + \frac{2a}{9}} - \sqrt{\frac{a}{2}} \right] \\ &= \Pr \left[\left(\sqrt{\mathbb{E}[\Lambda(S)] + \frac{a}{18}} + \sqrt{\frac{a}{2}} \right)^2 < \Lambda(S) + \frac{2a}{9} \right] \\ &= \Pr \left[\sqrt{2a\mathbb{E}[\Lambda(S)] + \frac{a^2}{9}} + \frac{a}{3} < \Lambda(S) - \mathbb{E}[\Lambda(S)] \right] \\ &\leq \exp \left(- \frac{(\sqrt{2a\mathbb{E}[\Lambda(S)] + \frac{a^2}{9}} + \frac{a}{3})^2}{2\mathbb{E}[\Lambda(S)] + \frac{2}{3} \cdot (\sqrt{2a\mathbb{E}[\Lambda(S)] + \frac{a^2}{9}} + \frac{a}{3})} \right) \\ &= \exp \left(- \frac{(\sqrt{2a\mathbb{E}[\Lambda(S)] + \frac{a^2}{9}} + \frac{a}{3})^2}{(\sqrt{2\mathbb{E}[\Lambda(S)] + \frac{a}{9}} + \frac{\sqrt{a}}{3})^2} \right) \\ &= \exp(-a) \\ &= \delta, \end{aligned}$$

where the inequality is due to Lemma 3.2. Note that in the second equality, when $\sqrt{\frac{a}{2}} - \sqrt{\Lambda(S) + \frac{2a}{9}} \geq 0$ or equivalently $\Lambda(S) \leq \frac{5a}{18}$, since $\sqrt{\frac{\sigma(S) \cdot \theta}{n} + \frac{a}{18}} \geq \sqrt{\frac{a}{18}}$ and $\sqrt{\frac{a}{2}} - \sqrt{\Lambda(S) + \frac{2a}{9}} \leq \sqrt{\frac{a}{2}} - \sqrt{\frac{2a}{9}} = \sqrt{\frac{a}{18}}$, it follows that the probability $\Pr \left[\sqrt{\frac{\sigma(S) \cdot \theta}{n} + \frac{a}{18}} < \sqrt{\frac{a}{2}} - \sqrt{\Lambda(S) + \frac{2a}{9}} \right] = 0$.

The proof of (3.20) is analogous.

$$\begin{aligned}
 & \Pr \left[\sigma(S) > \left(\sqrt{\Lambda(S) + \frac{a}{2}} + \sqrt{\frac{a}{2}} \right)^2 \cdot \frac{n}{\theta} \right] \\
 &= \Pr \left[\sqrt{\mathbb{E}[\Lambda(S)]} > \sqrt{\Lambda(S) + \frac{a}{2}} + \sqrt{\frac{a}{2}} \right] \\
 &\leq \Pr \left[\left(\sqrt{\mathbb{E}[\Lambda(S)]} - \sqrt{\frac{a}{2}} \right)^2 > \Lambda(S) + \frac{a}{2} \right] \\
 &= \Pr \left[-\sqrt{2a\mathbb{E}[\Lambda(S)]} > \Lambda(S) - \mathbb{E}[\Lambda(S)] \right] \\
 &\leq \exp \left(-\frac{2a\mathbb{E}[\Lambda(S)]}{2\mathbb{E}[\Lambda(S)]} \right) \\
 &= \delta,
 \end{aligned}$$

Hence, the theorem is proved. $\square$

Thus, given the observation $\Lambda(S)$, we can define a lower bound and an upper bound of $\sigma(S)$ as

$$\begin{cases} \sigma^l(S) \triangleq \left(\left(\sqrt{\Lambda(S) + \frac{2a}{9}} - \sqrt{\frac{a}{2}} \right)^2 - \frac{a}{18} \right) \cdot \frac{n}{\theta}, \\ \sigma^u(S) \triangleq \left(\sqrt{\Lambda(S) + \frac{a}{2}} + \sqrt{\frac{a}{2}} \right)^2 \cdot \frac{n}{\theta}. \end{cases} \quad (3.21)$$

Theorem 3.3 indicates that $\sigma(S)$ is lower (or upper) bounded by $\sigma^l(S)$ (or $\sigma^u(S)$) with a high probability at least $1 - \delta$. Then, ε can be computed by $\frac{\Lambda(S) \cdot n / \theta - \sigma^l(S)}{\sigma^l(S)}$ or $\frac{\sigma^u(S) - \Lambda(S) \cdot n / \theta}{\sigma^u(S)}$.

Recall that we need a high-confidence lower bound of $\sigma(S_k)$ and a high-confidence upper bound of $\sigma(S^*)$, which are given by $\sigma^l(S_k)$ and $\sigma^u(S^*)$, respectively. On the other hand, since S^* is unknown, $\Lambda(S^*)$ and thus $\sigma^u(S^*)$ are difficult to obtain directly. Fortunately, the greedy algorithm can give an upper bound $\hat{\Lambda} = \frac{\Lambda(S_k)}{1-1/e}$ of $\Lambda(S^*)$ for the maximum coverage problem, which can be used to establish an upper bound of $\sigma(S^*)$.

Theorem 3.4. *Given any value $\delta \in (0, 1)$, for the solution S_k returned by the greedy algorithm, let $\hat{\Lambda} = \min_{0 \leq i \leq k} \{\Lambda(S_i) + \Lambda(S_i, k)\}$. We have*

$$\Pr \left[\sigma(S^*) \leq \left(\sqrt{\hat{\Lambda} + \frac{a}{2}} + \sqrt{\frac{a}{2}} \right)^2 \cdot \frac{n}{\theta} \right] \geq 1 - \delta. \quad (3.22)$$

Proof. By Theorem 3.1, it holds that $\Lambda(S^*) \leq \hat{\Lambda}$. As a result,

$$\sigma^u(S^*) = \left(\sqrt{\Lambda(S^*) + \frac{a}{2}} + \sqrt{\frac{a}{2}} \right)^2 \cdot \frac{n}{\theta} \leq \left(\sqrt{\hat{\Lambda} + \frac{a}{2}} + \sqrt{\frac{a}{2}} \right)^2 \cdot \frac{n}{\theta}.$$

Together with $\Pr [\sigma(S^*) \leq \sigma^u(S^*)] \geq 1 - \delta$ from Theorem 3.3, it completes the proof. $\square$

Theorem 3.4 gives an upper bound of $\sigma(S^*)$. For convenience, we denote the upper bound as

$$\hat{\sigma}(S^*) \triangleq \left(\sqrt{\hat{\Lambda} + \frac{a}{2}} + \sqrt{\frac{a}{2}} \right)^2 \cdot \frac{n}{\theta}. \quad (3.23)$$

Note that for the seed set solution S_k obtained on a group $\mathcal{R}$ of RR sets, $\mathcal{R}$ is not random with respect to S_k in general. Thus, we need another group $\mathcal{R}'$ of RR sets (whose number is denoted by θ') to validate the lower bound $\sigma^l(S_k)$ of $\sigma(S_k)$ as did in [91]. Suppose that the lower bound of the constructed solution holds with confidence $1 - \delta_1$ and the upper bound of the optimum holds with confidence $1 - \delta_2$. Then, the probability for either $\sigma(S_k)$ or $\sigma(S^*)$ to violate its bound is at most $\delta_1 + \delta_2$. Therefore, a high-confidence online approximation guarantee α can be obtained via $\sigma^l(S_k)$ and $\hat{\sigma}(S^*)$, i.e.,

$$\alpha = \frac{\sigma^l(S_k)}{\hat{\sigma}(S^*)}. \quad (3.24)$$

It is easy to see that

$$\Pr \left[\sigma(S_k) \geq \alpha \cdot \sigma(S^*) \right] \geq 1 - \delta_1 - \delta_2. \quad (3.25)$$

Thus, to satisfy the confidence requirement $1 - \delta$, we can simply set $\delta_1 + \delta_2 = \delta$. Empirically, we set $\delta_1 = \delta_2 = \delta/2$ and show that the online approximation guarantee obtained under this setting is close to optimal.

Theorem 3.5. *Let $\hat{\Lambda}$ be the upper bound of the maximum achievable coverage on $\mathcal{R}$ and $\Lambda'(S_k)$ be the coverage of S_k evaluated on $\mathcal{R}'$. Then, for any confidence requirement*

of $1 - \delta > 0$, the online approximation guarantee $\alpha = \sigma^l(S_k)/\hat{\sigma}(S^*)$ (if $\sigma^l(S_k) \geq 0$)¹ derived from the setting of $\delta_1 = \delta_2 = \delta/2$ is at least a factor $\frac{f(\ln \frac{2}{\delta}) \cdot g(\ln \frac{1}{\delta})}{f(\ln \frac{1}{\delta}) \cdot g(\ln \frac{2}{\delta})}$ of the maximum attainable online approximation guarantee, where $f(x) = \left(\sqrt{\Lambda'(S_k) + \frac{2x}{9}} - \sqrt{\frac{x}{2}} \right)^2 - \frac{x}{18}$ and $g(x) = \left(\sqrt{\hat{\Lambda} + \frac{x}{2}} + \sqrt{\frac{x}{2}} \right)^2$. Furthermore, when $\Lambda'(S_k) \geq 8 \ln \frac{1}{\delta}$ and $\hat{\Lambda} \geq 8 \ln \frac{1}{\delta}$, the above factor satisfies

$$\frac{f(\ln \frac{2}{\delta}) \cdot g(\ln \frac{1}{\delta})}{f(\ln \frac{1}{\delta}) \cdot g(\ln \frac{2}{\delta})} > \frac{\ln \frac{1}{\delta} - \frac{\ln 2}{2}}{\ln \frac{1}{\delta} + \frac{\ln 2}{3}}. \quad (3.26)$$

Proof. $f(x)$ can be rewritten as

$$\begin{aligned} f(x) &= \left(\sqrt{\Lambda'(S_k) + \frac{2x}{9}} - \sqrt{\frac{x}{2}} \right)^2 - \frac{x}{18} \\ &= \Lambda'(S_k) + \frac{2x}{3} - \sqrt{2x\Lambda'(S_k) + \frac{4x^2}{9}} \\ &= \Lambda'(S_k) + \frac{\frac{4x^2}{9} - 2x\Lambda'(S_k) - \frac{4x^2}{9}}{\frac{2x}{3} + \sqrt{2x\Lambda'(S_k) + \frac{4x^2}{9}}} \\ &= \Lambda'(S_k) - \frac{2\Lambda'(S_k)}{\frac{2}{3} + \sqrt{\frac{2\Lambda'(S_k)}{x} + \frac{4}{9}}}. \end{aligned}$$

Thus, $f(x)$ is decreasing with x . Meanwhile, $g(x)$ is increasing with x . Therefore, for any $0 \leq \delta_1, \delta_2 \leq \delta < 1$ that satisfy $\delta_1 + \delta_2 = \delta$,

$$\frac{\sigma^l(S_k)}{\hat{\sigma}(S^*)} = \frac{f(\ln \frac{1}{\delta_1})}{g(\ln \frac{1}{\delta_2})} \cdot \frac{\theta}{\theta'} \leq \frac{f(\ln \frac{1}{\delta})}{g(\ln \frac{1}{\delta})} \cdot \frac{\theta}{\theta'} \triangleq \alpha^*. \quad (3.27)$$

On the other hand, by setting $\delta_1 = \delta_2 = \delta/2$,

$$\frac{\sigma^l(S_k)}{\hat{\sigma}(S^*)} = \frac{f(\ln \frac{2}{\delta})}{g(\ln \frac{2}{\delta})} \cdot \frac{\theta}{\theta'} \triangleq \alpha. \quad (3.28)$$

Thus, by (3.27) and (3.28), we have

$$\frac{\alpha}{\alpha^*} = \frac{f(\ln \frac{2}{\delta}) \cdot g(\ln \frac{1}{\delta})}{f(\ln \frac{1}{\delta}) \cdot g(\ln \frac{2}{\delta})}. \quad (3.29)$$

¹When $\sigma^l(S_k) < 0$, there is no online approximation guarantee (i.e., 0-approximation) for the seed set S_k .

Furthermore, for any $0 \leq x_1 \leq x_2$, we have

$$\begin{aligned} f(x_1) - f(x_2) &= \frac{2(x_1 - x_2)}{3} + \left(\sqrt{2x_2\Lambda'(S_k) + \frac{4x_2^2}{9}} - \sqrt{2x_1\Lambda'(S_k) + \frac{4x_1^2}{9}} \right) \\ &\leq \frac{2(x_1 - x_2)}{3} + \sqrt{2\Lambda'(S_k)} \cdot (\sqrt{x_2} - \sqrt{x_1}) + \frac{2}{3} \cdot (x_2 - x_1) \\ &= \sqrt{2\Lambda'(S_k)} \cdot (\sqrt{x_2} - \sqrt{x_1}), \end{aligned}$$

where the inequality is due to $\sqrt{y_1 + z_1} - \sqrt{y_2 + z_2} \leq \sqrt{y_1} - \sqrt{y_2} + \sqrt{z_1} - \sqrt{z_2}$ for any $y_1 \geq y_2 \geq 0$ and $z_1 \geq z_2 \geq 0$. Similarly, by rewriting $g(x)$ as $g(x) = \hat{\Lambda} + x + \sqrt{2x\hat{\Lambda} + x^2}$, we have

$$\begin{aligned} g(x_2) - g(x_1) &= x_2 - x_1 + \left(\sqrt{2x_2\hat{\Lambda} + x_2^2} - \sqrt{2x_1\hat{\Lambda} + x_1^2} \right) \\ &\leq x_2 - x_1 + \sqrt{2\hat{\Lambda}} \cdot (\sqrt{x_2} - \sqrt{x_1}) + x_2 - x_1 \\ &= 2(x_2 - x_1) + \sqrt{2\hat{\Lambda}} \cdot (\sqrt{x_2} - \sqrt{x_1}), \end{aligned}$$

Thus,

$$\begin{aligned} \frac{f(\ln \frac{2}{\delta})}{f(\ln \frac{1}{\delta})} - 1 &= \frac{f(\ln \frac{2}{\delta}) - f(\ln \frac{1}{\delta})}{f(\ln \frac{1}{\delta})} \\ &\geq \frac{\sqrt{2\Lambda'(S_k)} \cdot (\sqrt{\ln \frac{1}{\delta}} - \sqrt{\ln \frac{2}{\delta}})}{f(\ln \frac{1}{\delta})} \\ &= \frac{\sqrt{2\Lambda'(S_k)} \cdot \frac{-\ln 2}{\sqrt{\ln \frac{1}{\delta}} + \sqrt{\ln \frac{2}{\delta}}}}{f(\ln \frac{1}{\delta})} \\ &> -\frac{\ln 2 \cdot \sqrt{\frac{\Lambda'(S_k)}{2 \ln \frac{1}{\delta}}}}{f(\ln \frac{1}{\delta})} \\ &= -\frac{\ln 2 \cdot z}{2 \ln \frac{1}{\delta} \cdot (z^2 + \frac{1}{3} - \sqrt{z^2 + \frac{1}{9}})} \\ &> -\frac{\ln 2 \cdot z}{2 \ln \frac{1}{\delta} \cdot (z^2 + \frac{1}{3} - z - \frac{1}{3})} \\ &= -\frac{\ln 2}{2 \ln \frac{1}{\delta} \cdot (z - 1)} \\ &\geq -\frac{\ln 2}{2 \ln \frac{1}{\delta}}, \end{aligned}$$

where $z \triangleq \sqrt{\frac{\Lambda'(S_k)}{2 \ln \frac{1}{\delta}}} \geq 2$ if $\Lambda'(S_k) \geq 8 \ln \frac{1}{\delta}$.

Similarly,

$$\begin{aligned}
\frac{g(\ln \frac{2}{\delta})}{g(\ln \frac{1}{\delta})} - 1 &= \frac{g(\ln \frac{2}{\delta}) - g(\ln \frac{1}{\delta})}{g(\ln \frac{1}{\delta})} \\
&\leq \frac{2 \ln 2 + \sqrt{2\hat{\Lambda}} \cdot (\sqrt{\ln \frac{2}{\delta}} - \sqrt{\ln \frac{1}{\delta}})}{g(\ln \frac{1}{\delta})} \\
&= \frac{\ln 2 \cdot \left(2 + \frac{\sqrt{2\hat{\Lambda}}}{\sqrt{\ln \frac{2}{\delta}} + \sqrt{\ln \frac{1}{\delta}}}\right)}{g(\ln \frac{1}{\delta})} \\
&< \frac{\ln 2 \cdot \left(2 + \sqrt{\frac{\hat{\Lambda}}{2 \ln \frac{1}{\delta}}}\right)}{g(\ln \frac{1}{\delta})} \\
&= \frac{\ln 2 \cdot (2 + z)}{2 \ln \frac{1}{\delta} \cdot (z^2 + \frac{1}{2} + \sqrt{z^2 + \frac{1}{4}})} \\
&< \frac{\ln 2 \cdot (2 + z)}{2 \ln \frac{1}{\delta} \cdot (z^2 + z)} \\
&= \frac{\ln 2 \cdot (\frac{2}{z} + 1)}{2 \ln \frac{1}{\delta} \cdot (z + 1)} \\
&\leq \frac{\ln 2}{3 \ln \frac{1}{\delta}},
\end{aligned}$$

where $z \triangleq \sqrt{\frac{\hat{\Lambda}}{2 \ln \frac{1}{\delta}}} \geq 2$ if $\hat{\Lambda} \geq 8 \ln \frac{1}{\delta}$.

As a result,

$$\frac{f(\ln \frac{2}{\delta}) \cdot g(\ln \frac{1}{\delta})}{f(\ln \frac{1}{\delta}) \cdot g(\ln \frac{2}{\delta})} \geq \frac{1 - \frac{\ln 2}{2 \ln \frac{1}{\delta}}}{1 + \frac{\ln 2}{3 \ln \frac{1}{\delta}}} = \frac{\ln \frac{1}{\delta} - \frac{\ln 2}{2}}{\ln \frac{1}{\delta} + \frac{\ln 2}{3}}.$$

Hence, the theorem is proved. $\square$

In Theorem 3.5, when $\delta = 10^{-6}$, it can be computed that $8 \ln \frac{1}{\delta} = 110$ and $\frac{\ln \frac{1}{\delta} - \frac{\ln 2}{2}}{\ln \frac{1}{\delta} + \frac{\ln 2}{3}} = 0.96$. This shows that if the coverage of S_k is more than 110 (which is very likely to hold for common settings of the Reverse Influence Sampling approach), the setting of $\delta_1 = \delta_2 = \delta/2$ can give at least 96% of the maximum attainable online approximation guarantee.

3.6 Extending MAG Framework for $(1 - 1/e - \varepsilon)$ -Approximation

Algorithm 5: MAGIM(k, ε, δ)

Input: Target seed set size k and approximation parameters ε and δ

Output: A seed set S_k of k nodes that can provide an approximation guarantee of $(1 - 1/e - \varepsilon)$ with confidence at least $1 - \delta$

```

1 initialize  $\theta_{\max}$  and  $\theta_0$  by (3.30) and (3.31), respectively;
2 compute  $i_{\max} \leftarrow \lceil \log_2 \frac{\theta_{\max}}{\theta_0} \rceil$ ;
3  $\mathcal{R} \leftarrow$  generate  $\theta_0$  random RR sets;
4  $\mathcal{R}' \leftarrow$  generate  $\theta_0$  random RR sets;
5 for  $i \leftarrow 1$  to  $i_{\max}$  do
6   find  $S_k, \alpha \leftarrow \text{MAG}(k, \frac{2\delta}{3i_{\max}}, \mathcal{R}, \mathcal{R}')$ ;
7   if  $\alpha \geq 1 - 1/e - \varepsilon$  or  $i = i_{\max}$  then return  $S_k$ ;
   // return the solution  $S_k$  if the stop condition is satisfied
8   double the sizes of  $\mathcal{R}$  and  $\mathcal{R}'$  with new RR sets;
```

As mentioned in Section 3.3.3, we can extend our MAG framework for traditional influence maximization. The extended algorithm called MAGIM is shown in Algorithm 5. MAGIM iteratively doubles the number of RR sets generated until the approximation guarantee of α is no less than $(1 - 1/e - \varepsilon)$ (lines 5-8). Tang et al. [105] derived an upper bound on the number of RR sets needed to assure the $(1 - 1/e - \varepsilon)$ -approximation guarantee with confidence at least $1 - \delta$. We substitute $\delta/3$ for δ in the upper bound to ensure that the approximation guarantee is achieved with confidence at least $1 - \delta/3$. The upper bound $\theta_{\max}$ after substitution is given below

$$\theta_{\max} = \frac{2n \left((1 - 1/e) \sqrt{\ln \frac{6}{\delta}} + \sqrt{(1 - 1/e) (\ln \binom{n}{k} + \ln \frac{6}{\delta})} \right)^2}{\varepsilon^2 k}. \quad (3.30)$$

We bound the largest number of RR sets at $\theta_{\max}$ and initialize the number of RR sets to be θ_0 as given below (line 1).

$$\theta_0 = 2 \left((1 - 1/e) \sqrt{\ln \frac{6}{\delta}} + \sqrt{(1 - 1/e) (\ln \binom{n}{k} + \ln \frac{6}{\delta})} \right)^2. \quad (3.31)$$

As a result, the number of iterations is at most $i_{\max} = \lceil \log_2 \frac{\theta_{\max}}{\theta_0} \rceil = \lceil \log_2 \frac{n}{\varepsilon^2 k} \rceil$ (line 2). In each iteration, we compute the approximation guarantee using the MAG algorithm

(Algorithm 4) with the required confidence at least $1 - \frac{2\delta}{3i_{\max}}$ (line 6). The algorithm stops once the approximation guarantee reaches $(1 - 1/e - \varepsilon)$ (line 7). By the union bound, if the solution S_k is returned before the $i_{\max}$ -th iteration, it can provide an $(1 - 1/e - \varepsilon)$ -approximation guarantee with confidence at least $1 - 2\delta/3$. On the other hand, if the solution S_k is returned at the $i_{\max}$ -th iteration, it can also provide an $(1 - 1/e - \varepsilon)$ -approximation guarantee with confidence at least $1 - \delta/3$. Therefore, the overall confidence is no worse than $1 - \delta$.

The following theorem studies the expected time complexity of MAGIM.

Theorem 3.6. *When $\delta \leq 1/2$, MAGIM runs in $O((k \ln n + \ln(1/\delta))(n+m)\varepsilon^{-2})$ expected time under the triggering model.*

Proof. The computation of $\hat{\Lambda}$ requires deriving $\Lambda(S_i) + \Lambda(S_i, k)$ for every $0 \leq i \leq k$. We observe that (i) both $\Lambda(S_i)$ and $\Lambda(v | S_i)$ for any v are available from the i -th iteration of the greedy algorithm (invoked on $\mathcal{R}$), and (ii) for any i , we can identify the nodes with the largest k marginal coverage gains in $O(n)$ time by using a linear-time selection algorithm to identify the k -th largest marginal coverage and then scanning through nodes to find those with marginal coverage no less than the k -th largest [8]. Therefore, each invocation of MAG takes $O(kn + \sum_{R \in \mathcal{R}} |R|)$ time to compute the upper bound $\hat{\Lambda}$ and $O(\sum_{R \in \mathcal{R}'} |R|)$ time to calculate $\Lambda'(S_k)$ on $\mathcal{R}'$. According to Wald's equation [108], MAGIM requires $O(kn + \mathbb{E}[|\mathcal{R} \cup \mathcal{R}'|] \cdot \mathbb{E}[|R|])$ expected time to execute the greedy algorithm and compute the online approximation guarantee α in the last iteration after which MAGIM stops. Recall that MAGIM doubles the sizes of $\mathcal{R}$ and $\mathcal{R}'$ in each iteration. Thus, the total number of RR sets examined in all the iterations is within twice of that in the last iteration. On the other hand, there are at most $i_{\max}$ iterations such that

$$i_{\max} \leq \log_2\left(\frac{n}{\varepsilon^2 k}\right) + 1 \leq \log_2\left(\frac{n}{\varepsilon^2}\right) + 1 \leq \log_2(n) + \frac{2}{\varepsilon} + 1,$$

for any $0 < \varepsilon < 1$. Thus, the total expected time used for executing the greedy algorithm and computing the online approximation guarantee α in all the iterations is

$$O(kn \cdot i_{\max} + 2\mathbb{E}[|\mathcal{R} \cup \mathcal{R}'|] \cdot \mathbb{E}[|R|]) = O(kn(\ln n + 1/\varepsilon) + \mathbb{E}[|\mathcal{R} \cup \mathcal{R}'|] \cdot \mathbb{E}[|R|]).$$

In addition, the time complexity of iteratively generating RR sets is $O(\sum_{R \in \mathcal{R} \cup \mathcal{R}'} w(R))$ [105, 106], where $w(R)$ denotes the number of edges in G that point to the nodes in an

RR set R . It is easy to see that $w(R) + 1 \geq |R|$. Thus, the expected time complexity of MAGIM is $O(kn(\ln n + 1/\varepsilon) + \mathbb{E}[|\mathcal{R} \cup \mathcal{R}'|] \cdot \mathbb{E}[w(R)])$. Previous work [105, 106] shows that $n \cdot \mathbb{E}[w(R)] \leq m \cdot \sigma(S^*)$. Thus, the expected time complexity can be derived via the expected number of RR sets generated, i.e., $O(kn(\ln n + 1/\varepsilon) + \frac{m}{n}\sigma(S^*)\mathbb{E}[|\mathcal{R} \cup \mathcal{R}'|])$. In what follows, we show that the expected number of RR sets generated by MAGIM is $O((k \ln n + \ln(1/\delta)) \cdot n \cdot \varepsilon^{-2}/\sigma(S^*))$, which implies that MAGIM runs in $O((k \ln n + \ln(1/\delta))(n + m)\varepsilon^{-2})$ expected time.

Specifically, let

$$\varepsilon_2 = \varepsilon, \quad \varepsilon_1 = \varepsilon - (1 - 1/e)\varepsilon_2 = \varepsilon/e,$$

$$\tilde{\varepsilon}_2 = \sqrt{\frac{2a_c n}{\sigma(S^*)\theta}}, \quad \tilde{\varepsilon}_1 = \sqrt{\frac{2a_c n}{\sigma(S_k)\theta'}}, \quad \text{and} \quad \hat{\varepsilon}_1 = \left(\sqrt{\frac{2a_c \sigma(S_k)\theta'}{n}} + \frac{a_c^2}{9} + \frac{a_c}{3} \right) \cdot \frac{n}{\sigma(S_k)\theta'},$$

where $a_c = c \ln(\frac{3i_{\max}}{\delta})$ for any $c \geq 1$. Furthermore, let

$$\theta^* = \max \left\{ \frac{2n \ln \frac{6}{\delta}}{\varepsilon_2^2 \sigma(S^*)}, \frac{(2 + 2\varepsilon_1/3)n \ln \frac{6\binom{n}{k}}{\delta}}{\varepsilon_1^2 \sigma(S^*)}, \frac{27n \ln \frac{3i_{\max}}{\delta}}{(1 - 1/e - \varepsilon)\varepsilon_2^2 \sigma(S^*)} \right\}. \quad (3.32)$$

It is easy to verify that $\theta^* = O((k \ln n + \ln(1/\delta)) \cdot n \cdot \varepsilon^{-2}/\sigma(S^*))$.

Then, when $\theta = \theta' = c\theta^*$, according to Lemma 3.2, we have the following results.

$$\Pr \left[\Lambda(S^*) \cdot \frac{n}{\theta} < (1 - \varepsilon_2) \cdot \sigma(S^*) \right] \leq \left(\frac{\delta}{6} \right)^c, \quad (3.33)$$

$$\Pr \left[\Lambda(S_k) \cdot \frac{n}{\theta} > \sigma(S_k) + \varepsilon_1 \cdot \sigma(S^*) \right] \leq \left(\frac{\delta}{6\binom{n}{k}} \right)^c, \quad (3.34)$$

$$\Pr \left[\Lambda(S^*) \cdot \frac{n}{\theta} < (1 - \tilde{\varepsilon}_2) \cdot \sigma(S^*) \right] \leq \left(\frac{\delta}{3i_{\max}} \right)^c, \quad (3.35)$$

$$\Pr \left[\Lambda'(S_k) \cdot \frac{n}{\theta'} < (1 - \tilde{\varepsilon}_1) \cdot \sigma(S_k) \right] \leq \left(\frac{\delta}{3i_{\max}} \right)^c, \quad (3.36)$$

$$\Pr \left[\Lambda'(S_k) \cdot \frac{n}{\theta'} > (1 + \hat{\varepsilon}_1) \cdot \sigma(S_k) \right] \leq \left(\frac{\delta}{3i_{\max}} \right)^c. \quad (3.37)$$

In particular, (3.33) is obtained by l.h.s. $\leq \exp \left(-\frac{\varepsilon_2^2 \sigma(S^*) \frac{\theta}{n}}{2} \right) \leq \left(\frac{\delta}{6} \right)^c$ since $\theta = c\theta^* \geq \frac{c2n \ln \frac{6}{\delta}}{\varepsilon_2^2 \sigma(S^*)}$, and (3.34) is obtained by l.h.s. $\leq \exp \left(-\frac{\varepsilon_1^2 \sigma(S^*) \frac{\theta}{n}}{2\frac{\sigma(S_k)}{\sigma(S^*)} + 2\varepsilon_1/3} \right) \leq \exp \left(-\frac{\sigma(S^*) \frac{\theta}{n} \varepsilon_1^2}{2 + 2\varepsilon_1/3} \right) \leq \left(\frac{\delta}{6\binom{n}{k}} \right)^c$ since $\sigma(S_k) \leq \sigma(S^*)$ and $\theta^* \geq \frac{2n(1+\varepsilon_1/3) \ln \frac{6\binom{n}{k}}{\delta}}{\sigma(S^*)\varepsilon_1^2}$, and (3.35)–(3.37) are obtained similarly. Note that S_k is obtained from the set $\mathcal{R}$ of RR sets, so it is not independent

of $\mathcal{R}$. To fulfill the requirement of Lemma 3.2, we multiply the failure probability with a factor of $\binom{n}{k}$ to ensure all size- k node sets to be included. By the union bound, the probability that none of the events in (3.33)–(3.37) happens is at least $1 - \left(\left(\frac{\delta}{6} \right)^c + \left(\frac{\delta}{6\binom{n}{k}} \right)^c \right)^{\binom{n}{k} + 3 \cdot \left(\frac{\delta}{3i_{\max}} \right)^c} \geq 1 - \delta^c$.

On the other hand, we can easily get that $\tilde{\varepsilon}_2 \leq \sqrt{\frac{2(1-1/e-\varepsilon)\varepsilon_2^2}{27}} < \frac{\varepsilon_2}{3}$. In addition, when the events in (3.33) and (3.34) do not happen, we have

$$\begin{aligned} \sigma(S_k) &\geq \Lambda(S_k) \cdot \frac{n}{\theta} - \varepsilon_1 \cdot \sigma(S^*) \\ &\geq (1 - 1/e) \Lambda(S^*) \cdot \frac{n}{\theta} - \varepsilon_1 \cdot \sigma(S^*) \\ &\geq (1 - 1/e)(1 - \varepsilon_2) \cdot \sigma(S^*) - \varepsilon_1 \cdot \sigma(S^*) \\ &= (1 - 1/e - \varepsilon) \cdot \sigma(S^*). \end{aligned} \quad (3.38)$$

Thus, we have $\tilde{\varepsilon}_1 = \sqrt{\frac{2(1-1/e-\varepsilon)\sigma(S^*)\varepsilon_2^2}{27\sigma(S_k)}} < \frac{\varepsilon_2}{3}$ and $\hat{\varepsilon}_1 = \sqrt{\frac{a_c(2+2/\hat{\varepsilon}_1)n}{\sigma(S_k)\theta'}} \leq \sqrt{\frac{(2+2/\hat{\varepsilon}_1)\varepsilon_2^2}{27}} < \frac{\varepsilon_2}{3}$.

Moreover, $\Lambda'(S_k) \cdot \frac{n}{\theta'} \leq (1 + \hat{\varepsilon}_1) \cdot \sigma(S_k)$ is equivalent to $\left(\sqrt{\Lambda'(S_k) + \frac{2a_c}{9}} - \sqrt{\frac{a_c}{2}} \right)^2 - \frac{a_c}{18} \leq \sigma(S_k) \cdot \frac{\theta'}{n}$. Thus, it holds that $\Lambda'(S_k) - \hat{\varepsilon}_1 \sigma(S_k) \cdot \frac{\theta'}{n} = \Lambda'(S_k) - \left(\sqrt{2a_c \sigma(S_k) \cdot \frac{\theta'}{n} + \frac{a_c^2}{9} + \frac{a_c}{3}} \right) \leq \Lambda'(S_k) - \left(\sqrt{2a \left(\sqrt{\Lambda'(S^*) + \frac{2a_c}{9}} - \sqrt{\frac{a_c}{2}} \right)^2 - \frac{a_c^2}{9} + \frac{a_c^2}{9} + \frac{a_c}{3}} \right) = \Lambda'(S_k) - \left(\sqrt{2a \Lambda'(S_k) + \frac{4a_c^2}{9}} - a_c + \frac{a_c}{3} \right) = \left(\sqrt{\Lambda'(S_k) + \frac{2a_c}{9}} - \sqrt{\frac{a_c}{2}} \right)^2 - \frac{a_c}{18}$. When setting $\delta_1 = \delta/(3i_{\max})$, we have $\ln(1/\delta_1) \leq a_c$. It follows from (3.21) that

$$\sigma^l(S_k) \geq \left(\left(\sqrt{\Lambda'(S_k) + \frac{2a_c}{9}} - \sqrt{\frac{a_c}{2}} \right)^2 - \frac{a_c}{18} \right) \cdot \frac{n}{\theta'} \geq \Lambda'(S_k) \cdot \frac{n}{\theta'} - \hat{\varepsilon}_1 \sigma(S_k). \quad (3.39)$$

Similarly, $\Lambda(S^*) \cdot \frac{n}{\theta} \geq (1 - \tilde{\varepsilon}_2) \cdot \sigma(S^*)$ is equivalent to $\left(\sqrt{\Lambda(S^*) + \frac{a_c}{2}} + \sqrt{\frac{a_c}{2}} \right)^2 \cdot \frac{n}{\theta} \geq \sigma(S^*)$. Thus, $1 - \tilde{\varepsilon}_2 = 1 - \sqrt{\frac{2a_c n}{\sigma(S^*)\theta}} \leq 1 - \frac{\sqrt{2a_c}}{\sqrt{\Lambda(S^*) + \frac{a_c}{2}} + \sqrt{\frac{a_c}{2}}} = \frac{\sqrt{\Lambda(S^*) + \frac{a_c}{2}} - \sqrt{\frac{a_c}{2}}}{\sqrt{\Lambda(S^*) + \frac{a_c}{2}} + \sqrt{\frac{a_c}{2}}} = \frac{\Lambda(S^*)}{\left(\sqrt{\Lambda(S^*) + \frac{a_c}{2}} + \sqrt{\frac{a_c}{2}} \right)^2} \leq \frac{\hat{\Lambda}}{\left(\sqrt{\hat{\Lambda} + \frac{a_c}{2}} + \sqrt{\frac{a_c}{2}} \right)^2}$. Hence, when setting $\delta_2 = \delta/(3i_{\max})$, we have $\ln(1/\delta_2) \leq a_c$. It follows from (3.23) that

$$\hat{\sigma}(S^*) \leq \left(\sqrt{\hat{\Lambda} + \frac{a_c}{2}} + \sqrt{\frac{a_c}{2}} \right)^2 \cdot \frac{n}{\theta} \leq \frac{\hat{\Lambda}}{1 - \tilde{\varepsilon}_2} \cdot \frac{n}{\theta}. \quad (3.40)$$

Putting (3.39) and (3.40) together, when none of the events in (3.33)–(3.37) happens,

we have

$$\begin{aligned}
\frac{\sigma^l(S_k)}{\hat{\sigma}(S^*)} &\geq \frac{\Lambda'(S_k) - \hat{\varepsilon}_1 \sigma(S_k) \cdot \frac{\theta'}{n}}{\hat{\Lambda}/(1 - \tilde{\varepsilon}_2)} \\
&\geq \frac{(1 - \tilde{\varepsilon}_2)(1 - \tilde{\varepsilon}_1 - \hat{\varepsilon}_1) \sigma(S_k) \cdot \frac{\theta'}{n}}{\hat{\Lambda}} \\
&> (1 - \tilde{\varepsilon}_2 - \tilde{\varepsilon}_1 - \hat{\varepsilon}_1) \frac{\sigma(S_k) \cdot \frac{\theta'}{n}}{\Lambda(S_k)} \frac{\Lambda(S_k)}{\hat{\Lambda}} \\
&> (1 - \varepsilon_2) \frac{\Lambda(S_k) - \varepsilon_1 \cdot \sigma(S^*) \cdot \frac{\theta'}{n}}{\Lambda(S_k)} (1 - 1/e) \\
&\geq (1 - \varepsilon_2) \frac{\Lambda(S_k) - \varepsilon_1 \Lambda(S^*)/(1 - \varepsilon_2)}{\Lambda(S_k)} (1 - 1/e) \\
&\geq (1 - \varepsilon_2) \left(1 - \frac{\varepsilon_1/(1 - \varepsilon_2)}{1 - 1/e}\right) (1 - 1/e) \\
&= 1 - 1/e - \varepsilon.
\end{aligned} \tag{3.41}$$

Therefore, when $\theta = \theta' = c\theta^*$ RR sets are generated, MAGIM does not stop only if at least one of the events in (3.33)–(3.37) happens. The probability is at most δ^c .

Let j be the first iteration that the number of RR sets generated by MAGIM reaches θ^* . From this iteration onward, the expected number of RR sets further generated is at most

$$\begin{aligned}
2 \cdot \sum_{z \geq j} \theta_0 \cdot 2^z \cdot \delta^{2^{z-j}} &= 2 \cdot 2^j \cdot \theta_0 \sum_{z=0}^{\infty} 2^z \cdot \delta^{2^z} \\
&\leq 4\theta^* \sum_{z=0}^{\infty} 2^{-2^z + z} \\
&\leq 4\theta^* \sum_{z=0}^{\infty} 2^{-z} \\
&\leq 8\theta^*.
\end{aligned}$$

The first inequality is due to $2^{j-1} \cdot \theta_0 < \theta^*$ and $\delta \leq 1/2$, and the second inequality is due to $-2^z + z \leq -z$. If the algorithm stops before this iteration, there are at most $2\theta^*$ RR sets generated. Therefore, the expected number of RR sets generated is less than $10\theta^*$, which is $O((k \ln n + \ln(1/\delta)) \cdot n \cdot \varepsilon^{-2}/\sigma(S^*))$.

Hence, the theorem is proved. $\square$

Remark: Among the existing influence maximization algorithms that can provide $(1 - 1/e - \varepsilon)$ -approximation, the best time complexity reported so far is that of the

IMM algorithm [105]. Comparing MAGIM with IMM, although they have the same approximation guarantees and expected time complexity, MAGIM would run much faster than IMM in practice since it requires a much smaller number of RR sets to be generated. Specifically, IMM uses the same set $\mathcal{R}$ of RR sets for constructing the seed set S_k and deriving the lower bound of its influence spread. To bound S_k 's influence spread from below using $\mathcal{R}$, IMM uses a union bound of all size- k node sets to ensure that the concentration bound in Lemma 3.2 can be applied to the seed set S_k obtained from $\mathcal{R}$ itself. This will increase the failure probability by a factor of $\binom{n}{k}$. In contrast, MAGIM divides the RR sets into two disjoint groups $\mathcal{R}$ and $\mathcal{R}'$. The solution S_k is obtained from $\mathcal{R}$ and the lower bound $\sigma^l(S_k)$ of $\sigma(S_k)$ is computed from $\mathcal{R}'$. Since S_k is independent of the RR sets in $\mathcal{R}'$, the concentration bound in Lemma 3.2 can be applied directly without using a union bound. Moreover, using $\hat{\Lambda}$ to derive the upper bound $\hat{\sigma}(S^*)$ of the optimal seed set S^* 's influence spread in (3.23) can further speed up the MAGIM algorithm. As a consequence, MAGIM's practical efficiency is much higher to that of IMM, which shall be shown in Section 3.7.

3.7 Evaluation

3.7.1 Experimental Setup

Datasets. We have experimented with many real OSN datasets available at [77], including Facebook, Wiki-Vote, Google+ and LiveJournal. Table 3.1 shows the statistics of these datasets.

Table 3.1: Statistics of OSN datasets.

Dataset	#Nodes ($ V $)	#Edges ($ E $)	Type	Avg. degree
Facebook	4.0K	88.2K	undirected	43.7
Wiki-Vote	7.1K	103.7K	directed	29.1
Google+	107.6K	13.7M	directed	254.1
LiveJournal	4.8M	68.9M	directed	28.5

Algorithms. The main performance metric used in our evaluation is the online approximation guarantee of the seed set solution constructed from a fixed number of RR

sets generated. Borgs et al. [9] designed a $\min\{1/4, \beta\}$ -approximation algorithm when $\beta \cdot 144 \cdot 48 \cdot 6^3 \cdot (n+m) \ln(n)$ RR sets are generated, where n and m are the numbers of nodes and edges in the social network. But this algorithm is not practical. Even for our smallest dataset (Facebook) with 4.0K nodes and 88.2K edges, $\beta \cdot 144 \cdot 48 \cdot 6^3 \cdot (n+m) \ln(n) \approx 1.1 \times 10^{12} \cdot \beta$. If 10^7 RR sets are generated (the largest number of RR sets used in our experiments), the approximation factor β is around 8.7×10^{-6} only, which is hardly meaningful. Due to this reason, we do not include this algorithm in our evaluations.

Recent work has focused on reducing the number of RR sets generated for achieving the conventional approximation guarantee of $(1 - 1/e - \varepsilon)$ [90, 91, 105, 106]. We manually compute the smallest *fake* ε that satisfies the stop conditions of the state-of-the-art algorithms IMM [105] and SSA [91]. Note that such an ε cannot be used to provide the online approximation guarantee under a given number of RR sets generated since it is invalid when ε is dependent on the observations and is not pre-defined. This issue has been discussed in [56].

- IMM [105]: When the number of RR sets θ and the required confidence $1 - \delta$ are fixed, the smallest ε satisfying the stop condition of IMM is given by

$$\varepsilon = \frac{1}{\sqrt{2} \cdot \left(\sqrt{\frac{\Lambda(S_k)}{2 \left(\ln \binom{n}{k} + \ln \frac{1}{\delta} + \ln \log_2 n \right)} + \frac{1}{9} - \frac{2}{3}} \right)}, \quad (3.42)$$

where k is the seed set size and $\Lambda(S)$ is the coverage of S_k constructed by the greedy algorithm on the group of RR sets.

- SSA [91]: Huang et al. [56] showed that there are some problems for the original conference versions of SSA and D-SSA. We evaluate the corrected version of SSA in [56]. Similar to IMM, given θ and δ , we can find the smallest ε satisfying the stop conditions of SSA.

We compare the online approximation guarantee of our MAG framework with those provided by the IMM and SSA algorithms. In our MAG framework (Algorithm 4), we use the same numbers of RR sets for $\mathcal{R}$ and $\mathcal{R}'$. All the algorithms are all implemented in C++ and the experiments are carried out on a machine with an Intel Xeon E5-2695 2.4GHz CPU and 64GB memory.

Parameter Settings. For both the Independent Cascade (IC) and Linear Threshold (LT) models, 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 in-neighbors), as widely adopted by other studies [18, 20, 63, 90, 91, 105, 106]. We set $\delta = 1/n$ and $k = 100$ by default. We vary the number of RR sets by $2^i \times 10,000$ for $i = 0, 1, 2, \dots, 10$ to study the online approximation guarantees provided by different numbers of RR sets. For each parameter setting, we perform 100 simulation runs with randomly generated RR sets and the average online approximation guarantee of these 100 runs is plotted together with the 90th and 10th percentile results.

3.7.2 Online Approximation Guarantee under The IC Model

We first compare the algorithms under the IC model. Figure 3.1 shows the online approximation guarantee provided by different algorithms on various graphs as a function of the number of RR sets used. In the figure, a red horizontal dash line is plotted to show the value of $(1 - 1/e)$ and a black dash line labelled “MAG (coverage)” shows our derived online approximation guarantee for the maximum coverage problem (see Theorem 3.2 in Section 3.4). We observe that for all the algorithms tested, including IMM, SSA, and MAG, the online approximation guarantee provided is increasing with the number of RR sets used. This is because the sampling errors reduce with increasing number of RR sets generated. We also observe that our MAG algorithm can provide much tighter online approximation guarantees than the state-of-the-art influence maximization algorithms IMM and SSA. As can be seen, even when there are only 10,000 RR sets generated, our MAG algorithm can still provide good online approximation guarantees (0.664 on Facebook, 0.650 on Wiki-Vote, 0.525 on Google+ and 0.256 on LiveJournal) while IMM hardly achieves any online approximation guarantees for the Wiki-Vote, Google+ and LiveJournal datasets and so does SSA on the Facebook, Google+ and LiveJournal datasets. Moreover, Figure 3.1 shows that our MAG algorithm can easily break the approximation limit of $(1 - 1/e)$, which demonstrates the effectiveness of our proposed techniques. We remark that compared to IMM and SSA, the additional overhead of MAG for computing the online approximation guarantee using Theorem 3.2 is negligible (within 1 second) in all the experiment runs.

Figure 3.1: Online approximation guarantee under the IC model ($k = 100$).

Table 3.2: Online approximation guarantee for different seed set sizes k on the LiveJournal dataset when $2^{10} \times 10,000$ RR sets are generated under the IC model.

k	IMM	SSA	MAG	MAG (coverage)
1	0.592	0.505	0.930	1.000
10	0.594	0.580	0.843	0.870
100	0.573	0.603	0.791	0.804
1000	0.515	0.613	0.822	0.835
10000	0.385	0.514	0.803	0.826

Table 3.2 presents the results for different seed set sizes k varying from 1 to 10,000 on the LiveJournal dataset (the largest dataset tested) when $2^{10} \times 10,000$ RR sets are used. It can be seen that our MAG method substantially outperforms the IMM and SSA

(a) Facebook

(b) Wiki-Vote

(c) Google+

(d) LiveJournal

Figure 3.2: Online approximation guarantee under the LT model ($k = 100$).

methods in terms of the online approximation guarantee provided. Recall that the online approximation guarantees of the IMM and SSA methods are all limited to $1 - 1/e \approx 0.632$. In contrast, our MAG method can achieve online approximation guarantees far beyond $(1 - 1/e)$ even with the sampling errors taken into account, which demonstrates the tightness of our MAG framework. Furthermore, this also indirectly shows that the greedy algorithm for seed selection performs quite close to the optimal.

3.7.3 Online Approximation Guarantee under The LT Model

We next compare the algorithms under the LT model. Similar to the case of the IC model, Figure 3.2 shows that MAG provides considerably tighter online approximation guarantees than the IMM and SSA methods for all the datasets and the numbers of RR

Figure 3.3: Influence spread for influence maximization ($k = 100$).

Table 3.3: Online approximation guarantee for different seed set sizes k on the LiveJournal dataset when $2^{10} \times 10,000$ RR sets are generated under the LT model.

k	IMM	SSA	MAG	MAG (coverage)
1	0.595	0.514	0.943	1.000
10	0.599	0.586	0.924	0.952
100	0.586	0.609	0.805	0.818
1000	0.548	0.618	0.813	0.821
10000	0.462	0.571	0.804	0.817

sets tested. Table 3.3 shows the results for different seed set sizes k on the LiveJournal dataset when $2^{10} \times 10,000$ RR sets are used. Again, our MAG method outperforms the state-of-the-art methods significantly. These results demonstrate the generality of our MAG method – it can be applied to various diffusion models to provide good online approximation guarantees.

3.7.4 Efficiency for Achieving $(1 - 1/e - \epsilon)$ -Approximation

As discussed in Section 3.6, our framework can be extended for traditional influence maximization with the conventional approximation guarantee of $(1 - 1/e - \epsilon)$. Now, we evaluate MAGIM (Algorithm 5) and focus on the large LiveJournal dataset with millions of nodes. We test three different values of ϵ at 0.1, 0.05 and 0.01.

Figure 3.3 shows the influence spreads produced by different algorithms under the

(a) IC model

(b) LT model

Figure 3.4: Running time for influence maximization ($k = 100$).

(a) IC model

(b) LT model

Figure 3.5: Number of RR sets for influence maximization ($k = 100$).

IC and LT models. In general, the three algorithms, IMM, SSA and MAGIM, produce similar influence spreads. In most cases, IMM and SSA generate slightly higher influence spreads than MAGIM, which indicates that the online approximation guarantee provided by IMM and SSA is under-estimated.

Figure 3.4 shows the running times of different algorithms. We observe that our MAGIM method runs significantly faster than the IMM and SSA methods (by up to 3 orders of magnitude) to achieve the $(1 - 1/e - \epsilon)$ -approximation guarantee. This is because IMM and SSA need to generate many more RR sets than MAGIM (also by up to 3 orders of magnitude, which can be seen from Figure 3.5) to achieve the same

approximation guarantee. When ε reduces from 0.1 to 0.01, the running times of IMM and SSA increase dramatically (note that the y-axis is in logscale) whereas that of our MAGIM method remains almost the same (within 5 seconds for both the IC and LT models). These results demonstrate the efficiency of our MAGIM method for achieving the conventional approximation guarantee of influence maximization.

3.8 Summary

In this chapter, we have proposed a MAG framework to maximize the online approximation guarantee for influence maximization. The framework includes a new online upper bound for the maximum coverage problem and a new method to analyze the sampling errors of the Reverse Influence Sampling approach. These techniques together can substantially improve the current best factor of $(1 - 1/e - \varepsilon)$ for influence maximization. Moreover, we have extended our MAG framework to speed up the achievement of the conventional approximation guarantee of $(1 - 1/e - \varepsilon)$. Experiments are conducted with real OSN datasets to compare the effectiveness and efficiency of our framework with the state-of-the-art methods. In terms of effectiveness (under a fixed number of RR sets), our MAG method can provide online approximation guarantees up to 0.93 for runtime-restricted influence maximization, which consistently and significantly outperforms the IMM and SSA methods. In terms of efficiency (for achieving the conventional $(1 - 1/e - \varepsilon)$ -approximation), our MAG method can find the top-100 influential nodes on the million-scale LiveJournal dataset within 5 seconds, which is up to 3 orders of magnitude faster than the IMM and SSA methods.

We remark that besides the IC and LT models, our proposed solutions in this chapter can also be applied to many other diffusion models discussed in Section 2.1, such as the *triggering model* [63] which generalizes the IC and LT models, the *continuous-time models* [17, 32, 81, 98] and the *topic-aware models* [6, 15].

Chapter 4

Profit Maximization for Viral Marketing in Online Social Networks

4.1 Introduction

A large amount of recent work [4,9,18–20,22,23,25,41,42,49,51,52,62,72,76,90,91,93,100,101,104,106,115,116] has been focusing on *influence maximization* in viral marketing, which targets at selecting a set of initial seed nodes in the OSN to spread the influence as widely as possible. These studies have assumed a fixed and pre-determined budget for seed selection. In essence, the cost of seed selection is the price to pay for viral marketing (e.g., providing the selected users with free samples or other incentives). The influence spread, on the other hand, is the reward of viral marketing, which can potentially be translated into growth in the adoptions of products. Thus, the budget for seed selection reflects a tradeoff between the expense and reward of viral marketing. If the budget is set too low, it may not produce the desired extent of influence spread to fully exploit the potential of viral marketing in boosting sales and revenues. In contrast, if the budget is set too high, the benefit of the influence spread generated may not pay off the expense. A cost-effective budget setting should strike a balance between the expense and reward of viral marketing.

Economic-wise, a common goal for companies conducting viral marketing is to maximize the profit return, which can be defined as the reward less the expense. Thus, in this chapter, we define a profit metric that naturally combines the benefit of influence spread with the cost of seed selection to eliminate the need for presetting the budget for seed

selection. We carry out a comprehensive study on finding a set of seed nodes to optimize the profit of viral marketing. We show that the profit metric is significantly different from the influence metric in that it is no longer monotone. Applying simple hill-climbing algorithms to the profit maximization problem would not provide any strong theoretical guarantee on the seed set selected. Observing that seed selection for profit maximization is an unconstrained submodular maximization problem, we develop new seed selection algorithms based on the ideas of the double greedy algorithms by Buchbinder et al. [11]. The original double greedy algorithms have a serious limitation for our profit maximization problem: they rely on a rather strict condition for offering non-trivial approximation guarantees which is not realistic in viral marketing. We propose several new techniques to address this limitation along different directions.

The rest of this chapter is organized as follows. Section 4.2 defines the profit maximization problem. Section 4.3 elaborates our algorithm design. Section 4.4 derives the upper bounds. Section 4.5 presents the experimental study. Finally, Section 4.6 summarizes the chapter.

4.2 Problem Formulation

4.2.1 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 .

As discussed in Section 2.1, there are many diffusion models for the processes by which influence propagates in social networks. While our problem formulation and solutions are general and not restricted to a particular diffusion model, to facilitate exposition, we shall discuss our examples using the Independent Cascade (IC) model — a representative and most widely-studied model for influence propagation [18, 19, 62, 63, 76, 91, 93, 101, 105, 106, 116]. Recall that 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. Let N_u denote the set of node u 's neighbors, i.e., $N_u = \{v: v \in V, \langle u, v \rangle \in E\}$.

Given a set of seed nodes S , the IC diffusion process proceeds as follows. Initially, the seed nodes S are activated, while all the other nodes are not activated. When a node u first becomes activated, it has a single chance to activate its neighbors who are not yet activated. For each such neighbor $v \in N_u$, v would become activated with probability $p_{u,v}$. This process repeats until no more node can be activated. Note that the IC diffusion process is a random process. Let $g \sim G$ be a sample outcome of influence propagation in the sample space and let $V_g(S)$ be the set of nodes activated starting from the initial 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.,

$$\sigma(S) = \mathbb{E}[|V_g(S)|].$$

4.2.2 The Profit Maximization Problem

As discussed, the influence spread is the benefit gained by viral marketing and the cost of seed selection is the price to pay for viral marketing. In general, the users in the social network are likely to bring different amounts of benefit if activated and have different costs for seed selection. For example, users may have different preferences for various models of a product due to their genders, ages, or occupations (e.g., most students may purchase iPhone 7 32G while most businessmen may purchase iPhone 7 Plus 256G, thus they offer different benefits), and sending free samples to users in distinct regions may incur different delivery expenses. Suppose that each node $v \in V$ is associated with a benefit $b(v)$ if v is activated and a cost $c(v)$ for seed selection. Then, we define a *profit* metric as the benefit of influence spread less the cost of seed selection, i.e., the profit of a seed set S , denoted by $\phi(S)$, is given by

$$\phi(S) = \beta(S) - c(S),$$

where $\beta(S) = \mathbb{E}[\sum_{v \in V_g(S)} b(v)]$ is the total benefit brought by all the nodes activated and $c(S) = \sum_{v \in S} c(v)$ is the total cost of all the seed nodes selected.

Our goal is to find a seed set S to maximize the profit $\phi(S)$. First, we study the submodularity of the profit function. To simplify the notations, we define $\beta(v \mid S) \triangleq \beta(S \cup \{v\}) - \beta(S)$ as the *marginal benefit gain* of adding a new seed node $v \in V$ into a

seed set $S \subseteq V$ and define $\phi(v \mid S) \triangleq \phi(S \cup \{v\}) - \phi(S)$ as the *marginal profit gain* of adding v into a seed set S .

Lemma 4.1. *The profit function $\phi(\cdot)$ is submodular if the benefit function $\beta(\cdot)$ is submodular.*

Proof. If $\beta(\cdot)$ is submodular, for any two seed sets S and T where $S \subseteq T$ and any node $v \notin T$, it holds that $\beta(v \mid S) \geq \beta(v \mid T)$. Therefore, we have $\phi(v \mid S) = \beta(v \mid S) - c(v) \geq \beta(v \mid T) - c(v) = \phi(v \mid T)$, which implies that $\phi(\cdot)$ is also submodular. $\square$

Kempe et al. [63] has proved that the influence function $\sigma(\cdot)$ is submodular under the IC model. Using a similar approach, it can be shown that the benefit function $\beta(\cdot)$ is also submodular under the IC model.

Lemma 4.2. *The benefit function $\beta(\cdot)$ is submodular under the IC model.*

Proof. A sample influence propagation outcome g can be generated by independently flipping a coin of bias $p_{u,v}$ for each edge $\langle u, v \rangle \in E$ to decide whether the edge is *live* or *blocked*. It is easy to see that the set of nodes $V_g(S)$ activated in the sample outcome g are those that can be reached from the seed set S in g . The total benefit generated from the seed set S in the outcome g is given by $\beta_g(S) = \sum_{v \in V_g(S)} b(v)$. Let $p(g)$ denote the probability of a specific outcome g in the sample space. Then, $\beta(S) = \sum_g (p(g) \cdot \beta_g(S))$.

In addition, the marginal benefit gain $\beta_g(u \mid S) = \sum_{v \in V_g(u \mid S)} b(v)$, where $V_g(u \mid S)$ is the set of nodes that are reachable from a node u but are not reachable from any node in a seed set S in the sample outcome g . For any two node sets S and T where $S \subseteq T$, we have $V_g(u \mid S) \supseteq V_g(u \mid T)$, which implies that $\beta_g(u \mid S) \geq \beta_g(u \mid T)$. Since $p(g) \geq 0$ for any g , taking the linear combination, we have $\beta(u \mid S) \geq \beta(u \mid T)$. Thus, $\beta(\cdot)$ is submodular. $\square$

By Lemmas 4.1 and 4.2, the profit function $\phi(\cdot)$ is also submodular under the IC model. Though both the profit and influence functions are submodular, it should be noted that the profit $\phi(\cdot)$ is significantly different from the influence spread $\sigma(\cdot)$ in that $\phi(\cdot)$ may no longer be monotone. The marginal profit gain by adding a new seed, i.e., $\phi(v \mid S) = \beta(v \mid S) - c(v)$, can be negative. As shall be shown soon, this makes the seed selection for profit maximization more challenging than that for influence maximization. Selecting seed nodes to maximize the profit becomes an unconstrained submodular maximization problem [11, 35].

4.3 Seed Selection Algorithms

In this section, we first study an intuitive greedy algorithm and show that it does not provide any bounded approximation guarantee. To provide strong approximation guarantees, we then borrow ideas from the double greedy algorithms [11], and develop new methods for profit maximization.

4.3.1 Simple Greedy Heuristic

If activating each node offers the same benefit, one may intuitively argue that the profit maximization problem can be easily solved by a straightforward approach that runs an influence maximization algorithm for every possible seed set size (from 1 to $|V|$) and then chooses the best solution. But unfortunately, this would not work when the nodes have different costs for seed selection because most existing influence maximization algorithms [18, 19, 62, 63, 90, 91, 100, 105, 106] do not differentiate the nodes by their costs in the seed selection. If a top influential node has a high cost (e.g., a popular user may require more incentives to be recruited as a seed), the total profit of the seed set selected can be low or even negative.

Algorithm 6: SimpleGreedy(G, ϕ)

```

1 initialize  $S \leftarrow \emptyset$ ;
2 while True do
3   find  $u \leftarrow \arg \max_{v \in V \setminus S} \{\phi(v \mid S)\}$ ;
4   if  $\phi(u \mid S) \leq 0$  then
5      $\quad$  break;
6    $S \leftarrow S \cup \{u\}$ ;
7 return  $S$ ;
```

We construct a simple greedy hill-climbing algorithm to optimize the profit, similar to that proposed by Kempe et al. [63] for influence maximization. Algorithm 6 describes the greedy heuristic. It starts with an empty seed set $S = \emptyset$. In each iteration, if the largest marginal profit gain $\phi(v \mid S)$ by choosing a new seed from the non-seed nodes $V \setminus S$ is positive, the greedy heuristic adds the corresponding node to S . Otherwise, it implies that the profit cannot be further increased by adding any new seed, so the

Figure 4.1: A simple hill-climbing algorithm fails to achieve any bounded approximation factor.

algorithm stops and returns the seed set S . This simple greedy algorithm shares the same spirit with the hill-climbing heuristic adopted by [85] and [117].

Unfortunately, the above simple greedy algorithm does not have any bounded approximation factor for profit maximization because the profit function is submodular but not monotone. This is true even if all nodes have the same unit costs $c(v) = 1$ for seed selection and the same unit benefits $b(v) = 1$ when activated. Figure 4.1 shows an example social network with $n + 1$ nodes ($n \geq 2$) and $2n$ edges. The propagation probabilities are given by $p_{u,v_i} = 2p$ and $p_{v_i,u} = np - \epsilon$ for each $1 \leq i \leq n$, where $\epsilon > 0$. When node u is chosen as the only seed, the probability for each node v_i to be activated is $2p$, so the profit $\phi(\{u\}) = 1 + 2np - 1 = 2np$. For each node v_i , when v_i is chosen as the only seed, the probability for node u to be activated is $np - \epsilon$ and only when u is activated, each remaining node v_j ($j \neq i$) can be activated with probability $2p$. Hence, the profit $\phi(\{v_i\}) = 1 + (np - \epsilon) \cdot (1 + 2(n - 1)p) - 1 = (np - \epsilon) \cdot (1 + 2(n - 1)p)$. When $p < \frac{1}{2(n-1)}$, we have $1 + 2(n - 1)p < 2$ and thus, $\phi(\{v_i\}) < \phi(\{u\})$. If the simple greedy algorithm is applied, u would be the first seed selected. Furthermore, for any node v_i , $\phi(\{u, v_i\}) = 2 + 2(n - 1)p - 2 = 2(n - 1)p$, which means $\phi(v_i | \{u\}) = 2(n - 1)p - 2np = -2p < 0$. Therefore, the greedy algorithm would stop after selecting u and return $S = \{u\}$ with a profit of $\phi(\{u\}) = 2np$. On the other hand, when nodes $v_1, v_2, \dots, v_n$ are all chosen as seeds, the probability for node u to be activated is $1 - (1 - np + \epsilon)^n$. As a result, $\phi(V \setminus \{u\}) = n + 1 - (1 - np + \epsilon)^n - n = 1 - (1 - np + \epsilon)^n$. Let $p = \frac{1}{n^2} < \frac{1}{2(n-1)}$ and $\epsilon = \frac{1}{4n^2}$. Then, we have $\phi(\{u\}) = \frac{2}{n}$ and $\phi(V \setminus \{u\}) = 1 - (1 - \frac{1}{n} + \frac{1}{4n^2})^n = 1 - (1 - \frac{1}{2n})^{2n}$. When $n \rightarrow \infty$, we have $\frac{2}{n} \rightarrow 0$ and $1 - (1 - \frac{1}{2n})^{2n} \rightarrow 1 - \frac{1}{e}$ which is a positive constant. Thus, the simple greedy algorithm can perform arbitrarily worse than the optimal solution and does not have any bounded approximation factor.

We further remark that selecting seeds based on seed minimization algorithms [50, 83, 114] for achieving a given amount of influence spread would not be able to provide strong approximation guarantees for profit maximization either. As analyzed in [50, 114], for any $\epsilon > 0$, the seed minimization problem cannot be approximated within a factor of $(1 - \epsilon) \ln |V|$ unless NP has $n^{O(\log \log n)}$ -time deterministic algorithms. Therefore, the seed minimization algorithms do not have any constant approximation factor by themselves. Thus, we do not adapt seed minimization algorithms to our profit maximization problem.

4.3.2 Double Greedy Algorithms

Buchbinder et al. [11] proposed double greedy algorithms to address the unconstrained submodular maximization problem with strong approximation guarantees for non-negative submodular functions. A deterministic double greedy algorithm yields $(1/3)$ -approximation, while a randomized double greedy algorithm yields $(1/2)$ -approximation. Algorithm 7 describes the ideas of these algorithms in our context of profit maximization. The algorithms start with an empty set S and a set T initialized with the entire node set of the social network. They iterate through all the nodes in the network in an arbitrary order to decide whether or not to include them in S and T . When the algorithms complete, it must hold that $S = T$ and this is the seed set selected. The decision for each node u is made based on the marginal profit gain of adding u into S (i.e., $\phi(S \cup \{u\}) - \phi(S) = \phi(u | S)$) and the marginal profit gain of removing u from T (i.e., $\phi(T \setminus \{u\}) - \phi(T) = -\phi(u | T \setminus \{u\})$). In the deterministic approach, each node u joins S if it generates higher marginal profit gain than that of quitting from T and vice versa (lines 5–8). In the randomized approach, each node u is added to S with probability $\phi(u | S) / (\phi(u | S) - \phi(u | T \setminus \{u\}))$, and is removed from T with probability $-\phi(u | T \setminus \{u\}) / (\phi(u | S) - \phi(u | T \setminus \{u\}))$.

In general, in the double greedy algorithms, if we initialize S with S_0 and T with T_0 where $\emptyset \subseteq S_0 \subseteq T_0 \subseteq V$ (line 1 in Algorithm 7) and only check the nodes in $T_0 \setminus S_0$ to decide whether or not to include them in S and T (line 2), we have the following lemma according to [11].¹

¹Lemma 4.3 is extracted through a detailed check of the proof of Theorem I.1 in [11] though it was not presented as a separate lemma therein.

Algorithm 7: DeterministicDoubleGreedy(G, ϕ) [11]

```

1 start with  $S \leftarrow \emptyset, T \leftarrow V$ ;
2 for each node  $u \in V$  do
3    $r^+ \leftarrow \phi(u \mid S)$ ;
4    $r^- \leftarrow -\phi(u \mid T \setminus \{u\})$ ;
5   if  $r^+ \geq r^-$  then
6      $S \leftarrow S \cup \{u\}$ ;
7   else
8      $T \leftarrow T \setminus \{u\}$ ;
9 return  $S (= T)$ ;
    // For randomized double greedy, change the condition of line 5 to
     $U(0,1) \leq r^+/(r^+ + r^-)$ , where  $U(0,1)$  is a uniformly distributed
    number between 0 and 1, and  $r^+/(r^+ + r^-) = 1$  if  $r^+ + r^- = 0$ .
    
```

Lemma 4.3. Let S^* denote the optimal solution such that $\phi(S^*) = \max_{S \subseteq V} \phi(S)$. If S and T are initialized with S_0 and T_0 respectively, Algorithm 7 returns a solution S_D satisfying

$$\phi((S^* \cup S_0) \cap T_0) + \phi(S_0) + \phi(T_0) \leq 3 \cdot \phi(S_D),$$

and the randomized version of Algorithm 7 returns a solution S_R satisfying

$$\phi((S^* \cup S_0) \cap T_0) + (\phi(S_0) + \phi(T_0))/2 \leq 2 \cdot \mathbb{E}[\phi(S_R)].$$

Lemma 4.3 gives rise to the approximation guarantees proved in [11].

Theorem 4.1. For the profit maximization problem, if the profit of selecting all nodes as seeds is non-negative, i.e., $\phi(V) \geq 0$, the profit of the seed set S_D returned by Algorithm 7 satisfies

$$\phi(S_D) \geq (1/3) \cdot \max_{S \subseteq V} \phi(S),$$

and the expected profit of the seed set S_R returned by the randomized version of Algorithm 7 satisfies

$$\mathbb{E}[\phi(S_R)] \geq (1/2) \cdot \max_{S \subseteq V} \phi(S).$$

Remark: Buchbinder et al. [11] established the above approximation guarantees for any non-negative submodular function $\phi(\cdot)$ when S and T are initialized with $\emptyset$ and V in the double greedy algorithms. In this case, since $(S^* \cup S_0) \cap T_0 = (S^* \cup \emptyset) \cap$

Figure 4.2: Double greedy algorithms fail to achieve any bounded approximation factor when $\phi(V) < 0$.

$V = S^* \cap V = S^*$, by Lemma 4.3, it follows that $\phi(S^*) + \phi(\emptyset) + \phi(V) \leq 3 \cdot \phi(S_D)$ and $\phi(S^*) + (\phi(\emptyset) + \phi(V))/2 \leq 2 \cdot \mathbb{E}[\phi(S_R)]$. Then, the approximation guarantees of Theorem 4.1 are derived from the non-negativity of function $\phi(\cdot)$. It is easy to see that, in fact, only a much looser condition $\phi(\emptyset) + \phi(V) \geq 0$ is required to provide the approximation guarantees. In our profit maximization problem, since $\phi(\emptyset) = 0$, we just need the condition $\phi(V) \geq 0$.

4.3.3 Warm-Start by Iterative Pruning

Theorem 4.1 provides strong theoretical guarantees for the approximability of Algorithm 7. However, the condition of $\phi(V) \geq 0$ may not be realistic in our profit maximization problem. $\phi(V) \geq 0$ means that selecting all the nodes (users) as seeds is still profitable, which is unlikely to be true for viral marketing, particularly in large-scale social networks. Apparently, providing every individual with incentives to advertise a new product defeats the purpose of viral marketing. On the other hand, when $\phi(V) < 0$, we cannot have any bounded approximation guarantees by directly applying Algorithm 7. For example, consider the social network in Figure 4.2 with $n + 1$ nodes ($n \geq 3$) and n edges. Let propagation probabilities $p_{u,v_i} = 1$, and let benefits $b(u) = 1$ and $b(v_i) = 1$ for each v_i . Let seed selection costs $c(u) = \frac{n}{2} + 2$ and $c(v_i) = 2$ for each v_i . Then, $\phi(V) = (n + 1) - (\frac{n}{2} + 2 + 2n) = -\frac{3n}{2} - 1 < 0$. Assume that Algorithm 7 initializes $S = \emptyset$ and $T = V$, and it iterates through the nodes in the order of $u, v_1, v_2, \dots, v_n$. In the first iteration, we have $\phi(u | S) = \phi(\{u\}) = \beta(\{u\}) - c(u) = n + 1 - (\frac{n}{2} + 2) = \frac{n}{2} - 1$. Meanwhile, $-\phi(u | T \setminus \{u\}) = -(\beta(V) - \beta(V \setminus \{u\}) - c(u)) = -((n + 1) - n - (\frac{n}{2} + 2)) = \frac{n}{2} + 1$. Thus, u quits from T so that $S = \emptyset$ and $T = V \setminus \{u\}$. In the second iteration, $\phi(v_1 | S) = \beta(\{v_1\}) - \beta(\emptyset) - c(v_1) = 1 - 2 = -1$ and $-\phi(v_1 | T \setminus \{v_1\}) =$

$-(\beta(V \setminus \{u\}) - \beta(V \setminus \{u, v_1\}) - c(v_1)) = -(n - (n - 1) - 2) = 1$. By Algorithm 7, v_1 quits from T as well. Similarly, in each subsequent iteration, v_i quits from T . Thus, Algorithm 7 finally returns $S = \emptyset$ with profit $\phi(S) = 0$. However, we already know that $\phi(\{u\}) = \frac{n}{2} - 1 > 0$. When $n \rightarrow \infty$, we have $\phi(\{u\}) \rightarrow \infty$. Therefore, the deterministic double greedy algorithm can perform arbitrarily worse than the optimal solution and does not have any bounded approximation factor when $\phi(V) < 0$.

To address this problem, we extend the result of Theorem 4.1 to maintain the same approximation guarantees with a much weaker condition. We start by proposing an approach to reduce the search space for maximizing the profit function. Our strategy is to find the nodes that must be selected as seeds and eliminate the nodes that are impossible to be chosen as seeds in an optimal solution. Given the profit function $\phi(\cdot)$, we define two node sets $A_1 = \{v: \phi(v \mid V \setminus \{v\}) > 0\}$ and $B_1 = \{v: \phi(v \mid \emptyset) \geq 0\}$. Due to the submodularity of $\phi(\cdot)$, we have $A_1 \subseteq B_1$ and this allows us to define a space $\Omega_1 = [A_1, B_1]$ that contains all the sets S satisfying $A_1 \subseteq S \subseteq B_1$.

Lemma 4.4. $\Omega_1 = [A_1, B_1]$ retains all global maximizers S^* for the profit function $\phi(\cdot)$, i.e., $A_1 \subseteq S^* \subseteq B_1$ for all S^* where $\phi(S^*) = \max_{S \subseteq V} \phi(S)$.

Proof. If $\phi(v \mid V \setminus \{v\}) > 0$, for any seed set $S \subseteq V \setminus \{v\}$, it follows from the submodularity of $\phi(\cdot)$ that $\phi(v \mid S) \geq \phi(v \mid V \setminus \{v\}) > 0$. Thus, $S \cup \{v\}$ always generates higher profit than S , so v must be selected as a seed in every optimal solution, which indicates $A_1 \subseteq S^*$. By similar arguments, if $\phi(v \mid \emptyset) < 0$, then v cannot be selected as a seed in any optimal solution, which implies $S^* \subseteq B_1$. $\square$

We can prune $\Omega_1 = [A_1, B_1]$ even further using an iterative strategy. Specifically, since the nodes in A_1 must be included in any global maximizer, we can shrink B_1 to $B_2 = \{v: \phi(v \mid A_1) \geq 0\}$. Similarly, since the nodes in $V \setminus B_1$ cannot be included in any global maximizer, we can expand A_1 to $A_2 = \{v: \phi(v \mid B_1 \setminus \{v\}) > 0\}$. This yields a smaller space $\Omega_2 = [A_2, B_2]$ than Ω_1 . These operations can be repeated alternately until A and B cannot be further broadened and narrowed respectively. Algorithm 8 presents the pseudo code of the iterative pruning process. Let A^* and B^* denote the node sets finally returned by Algorithm 8. Theorem 4.2 proves that $\Omega^* = [A^*, B^*]$ retains all global maximizers. For notational convenience, we define $A_0 = \emptyset$ and $B_0 = V$. To establish Theorem 4.2, we also make use of the following lemma about submodular functions.

Algorithm 8: IterativePrune(G, ϕ)

```

1 start with  $t = 0$ ,  $A_0 \leftarrow \emptyset$ ,  $B_0 \leftarrow V$ ;
2 repeat
3    $A_{t+1} \leftarrow \{v: \phi(v \mid B_t \setminus \{v\}) > 0\}$ ;
4    $B_{t+1} \leftarrow \{v: \phi(v \mid A_t) \geq 0\}$ ;
5    $t \leftarrow t + 1$ ;
6 until converged, i.e.,  $A_t = A_{t-1}$  and  $B_t = B_{t-1}$ ;
7 return  $A_t$  and  $B_t$  as  $A^*$  and  $B^*$ ;
```

Lemma 4.5 ([89]). *For any submodular function $\phi(\cdot)$ on the power set of V and any two subsets $X, Y \subseteq V$, it holds that*

$$\phi(Y) \leq \phi(X) - \sum_{v \in X \setminus Y} \phi(v \mid X \cup Y \setminus \{v\}) + \sum_{v \in Y \setminus X} \phi(v \mid X), \quad (4.1)$$

$$\phi(Y) \leq \phi(X) - \sum_{v \in X \setminus Y} \phi(v \mid X \setminus \{v\}) + \sum_{v \in Y \setminus X} \phi(v \mid X \cap Y). \quad (4.2)$$

Theorem 4.2. *For any global maximizer S^* , it holds that $A_t \subseteq A_{t+1} \subseteq A^* \subseteq S^* \subseteq B^* \subseteq B_{t+1} \subseteq B_t$ for any $t \geq 0$. Moreover, both $\phi(A_t)$ and $\phi(B_t)$ are non-decreasing with t .*

Proof. We first show that after each iteration, the newly generated space is reduced from that in the previous iteration, i.e., $A_t \subseteq A_{t+1} \subseteq B_{t+1} \subseteq B_t$. We prove it by induction. Obviously, $A_0 = \emptyset \subseteq A_1 \subseteq B_1 \subseteq V = B_0$ according to Lemma 4.4. Suppose that $A_{t-1} \subseteq A_t \subseteq B_t \subseteq B_{t-1}$ holds for some $t \geq 1$. For every node $v \in A_t$, we know $\phi(v \mid B_{t-1} \setminus \{v\}) > 0$. Due to the submodularity, we have $\phi(v \mid B_t \setminus \{v\}) \geq \phi(v \mid B_{t-1} \setminus \{v\}) > 0$. As a result, $A_t \subseteq A_{t+1}$. Similarly, for every node $v \in B_{t+1}$, we have $\phi(v \mid A_t) \geq 0$. Due to the submodularity, $\phi(v \mid A_{t-1}) \geq \phi(v \mid A_t) \geq 0$, which indicates that $B_{t+1} \subseteq B_t$. Furthermore, for all nodes $v \in A_{t+1} \cap A_t$, we have $\phi(v \mid A_t) = 0$, which implies that $(A_{t+1} \cap A_t) \subseteq B_{t+1}$ via line 4 in Algorithm 8. For all nodes $v \in A_{t+1} \setminus A_t$, we have $\phi(v \mid B_t \setminus \{v\}) > 0$. Since $v \notin A_t$ and $A_t \subseteq B_t$, we also have $A_t \subseteq B_t \setminus \{v\}$. Thus, $\phi(v \mid A_t) \geq \phi(v \mid B_t \setminus \{v\}) > 0$, which implies that $(A_{t+1} \setminus A_t) \subseteq B_{t+1}$. Consequently, we have $A_{t+1} = (A_{t+1} \cap A_t) \cup (A_{t+1} \setminus A_t) \subseteq B_{t+1}$. Therefore, $A_t \subseteq A_{t+1} \subseteq B_{t+1} \subseteq B_t$ holds for any $t \geq 0$.

Next, we explore the relationship of S^* to A^* and B^* . Obviously, $A_0 = \emptyset \subseteq S^* \subseteq V = B_0$ holds. As proved in Lemma 4.4, $A_1 \subseteq S^* \subseteq B_1$ also holds. Suppose that

$A_t \subseteq S^* \subseteq B_t$ holds for some $t \geq 0$. Then, any node v satisfying $\phi(v \mid B_t \setminus \{v\}) > 0$ must be in S^* . Otherwise, if $v \notin S^*$, we have $\phi(v \mid S^*) = \phi(v \mid S^* \setminus \{v\}) \geq \phi(v \mid B_t \setminus \{v\}) > 0$ by the submodularity, which indicates $\phi(S^* \cup \{v\}) > \phi(S^*)$, contradicting the optimality of S^* . The set of such nodes v satisfying $\phi(v \mid B_t \setminus \{v\}) > 0$ is exactly A_{t+1} , and hence $A_{t+1} \subseteq S^*$. Similarly, any node $v \in S^*$ must be in B_{t+1} , which means $S^* \subseteq B_{t+1}$. Otherwise, if $v \notin B_{t+1}$, it holds that $\phi(v \mid A_t) < 0$ by the definition, and hence $v \notin A_t$. As a result, $\phi(v \mid S^* \setminus \{v\}) \leq \phi(v \mid A_t \setminus \{v\}) = \phi(v \mid A_t) < 0$ by the submodularity, which implies $\phi(S^*) < \phi(S^* \setminus \{v\})$, contradicting the optimality of S^* . By induction, we have $A_t \subseteq S^* \subseteq B_t$ for any $t \geq 0$ and thus, $A^* \subseteq S^* \subseteq B^*$.

Finally, we show that $\phi(A_t) \leq \phi(A_{t+1})$ and $\phi(B_t) \leq \phi(B_{t+1})$ for any $t \geq 0$. In fact, for any node $v \in A_{t+1} \setminus A_t$, it holds that $\phi(v \mid A_{t+1} \setminus \{v\}) \geq \phi(v \mid B_{t+1} \setminus \{v\}) \geq \phi(v \mid B_t \setminus \{v\}) > 0$, where the first two inequalities are due to the submodularity (since $A_{t+1} \subseteq B_{t+1} \subseteq B_t$) and the third inequality is by the definition of A_{t+1} . Therefore, $\phi(A_{t+1}) \geq \phi(A_t) + \sum_{v \in A_{t+1} \setminus A_t} \phi(v \mid A_{t+1} \setminus \{v\}) \geq \phi(A_t)$, where the first inequality is due to (4.1) of Lemma 4.5 and the fact $A_t \subseteq A_{t+1}$. Similarly, for any node $v \in B_t \setminus B_{t+1}$, we have $\phi(v \mid B_{t+1}) \leq \phi(v \mid A_{t+1}) \leq \phi(v \mid A_t) < 0$, where the first two inequalities are due to the submodularity (since $A_t \subseteq A_{t+1} \subseteq B_{t+1}$) and the third inequality is by the definition of B_{t+1} as $v \notin B_{t+1}$. Hence, $\phi(B_t) \leq \phi(B_{t+1}) + \sum_{v \in B_t \setminus B_{t+1}} \phi(v \mid B_{t+1}) \leq \phi(B_{t+1})$, where the first inequality is due to (4.2) of Lemma 4.5 and the fact $B_{t+1} \subseteq B_t$. $\square$

Now, instead of starting with $S = \emptyset$ and $T = V$ (the entire node set) in the double greedy algorithms, we can initialize S with A^* and T with B^* and only check the nodes in $B^* \setminus A^*$ to decide whether or not to include them in the seed set. The following corollary establishes the approximation guarantees for the modified algorithms.

Corollary 4.1. *Suppose that S and T are initialized with A^* and B^* such that $\phi(A^*) + \phi(B^*) \geq 0$, the profit of the seed set $\hat{S}_D$ returned by Algorithm 7 satisfies*

$$\phi(\hat{S}_D) \geq (1/3) \cdot \max_{S \subseteq V} \phi(S),$$

and the expected profit of the seed set $\hat{S}_R$ returned by the randomized version of Algorithm 7 satisfies

$$\mathbb{E}[\phi(\hat{S}_R)] \geq (1/2) \cdot \max_{S \subseteq V} \phi(S).$$

Proof. By Theorem 4.2, $A^* \subseteq S^* \subseteq B^*$. Thus, $(S^* \cup A^*) \cap B^* = S^* \cap B^* = S^*$. As a result, by Lemma 4.3, it holds that $\phi(S^*) + \phi(A^*) + \phi(B^*) \leq 3 \cdot \phi(\hat{S}_D)$ when S and T are initialized with A^* and B^* . Hence, if $\phi(A^*) + \phi(B^*) \geq 0$, we obtain $\phi(\hat{S}_D) \geq (1/3) \cdot \phi(S^*) = (1/3) \cdot \max_{S \subseteq V} \phi(S)$. The proof of $\mathbb{E}[\phi(\hat{S}_R)] \geq (1/2) \cdot \max_{S \subseteq V} \phi(S)$ is similar. $\square$

Corollary 4.1 shows that conducting the iterative pruning prior to applying the double greedy algorithms allows us to maintain the same approximation guarantees with the condition $\phi(A^*) + \phi(B^*) \geq 0$. This condition is much weaker than the original condition $\phi(V) \geq 0$ of Theorem 4.1 since by Theorem 4.2, $\phi(V) = \phi(\emptyset) + \phi(V) = \phi(A_0) + \phi(B_0) \leq \phi(A_1) + \phi(B_1) \leq \dots \leq \phi(A^*) + \phi(B^*)$. Thus, Corollary 4.1 significantly expands the applicability of the theoretical guarantees.

For the example shown in Figure 4.2, $\phi(u \mid V \setminus \{u\}) = 1 - (\frac{n}{2} + 2) < 0$ and $\phi(v_i \mid V \setminus \{v_i\}) = 0 - 2 < 0$ for each $1 \leq i \leq n$. Thus, by Algorithm 8, $A_1 = \emptyset$. Furthermore, $\phi(u \mid \emptyset) = n + 1 - (\frac{n}{2} + 1) = \frac{n}{2} - 1 > 0$ and $\phi(v_i \mid \emptyset) = 1 - 2 = -1 < 0$ for each $1 \leq i \leq n$, which implies $B_1 = \{u\}$. In the second iteration, since $\phi(u \mid B_1 \setminus \{u\}) = \phi(u \mid \emptyset) = \frac{n}{2} - 1 > 0$, we have $A_2 = \{u\}$. B_2 remains the same as $B_1 = \{u\}$. Thus, Algorithm 8 returns $A^* = B^* = \{u\}$. As a result, $\phi(A^*) + \phi(B^*) = 2 \cdot \phi(\{u\}) = n - 2 \geq 0$ satisfying the condition given in Corollary 4.1. In fact, since $A^* = B^*$, it must be an optimal solution and the double greedy algorithms return exactly the optimal solution.

Finally, we remark that the sets A^* and B^* produced by the iterative pruning can also be used as warm-starts for the simple greedy algorithm in Section 4.3.1 so that only the nodes in $B^* \setminus A^*$ need to be further examined for seed selection.

4.3.4 Generality and Time Complexity

Our analysis and algorithms are general frameworks that can be adapted to any diffusion models which are submodular. These include the IC model, the LT model, the generalized triggering model [63], continuous-time models [98], and topic-aware models [6], to name a few.

Evaluating the profit metric involves estimating the influence spread given a seed set. Any existing influence estimation methods, such as Monte-Carlo simulation [63,

76, 93, 100] and Reverse Influence Sampling [90, 91, 105, 106], can be used. Suppose the time complexity for computing the marginal profit gain of adding a node u into a seed set S or removing a node u from a seed set T is $O(M)$. The simple greedy algorithm (Algorithm 6) takes at most $O((|V| - |S|)M)$ time to select one seed. Thus, the total time complexity of the simple greedy algorithm is $O((|V| + |V| - 1 + \dots + 1)M) = O(|V|^2 M)$. The double greedy algorithm (Algorithm 7) takes $O(2M)$ time for checking each node to decide whether to select it as a seed. Thus, the total time complexity of the double greedy algorithm is $O(|V| \cdot 2M) = O(|V|M)$. For the iterative pruning process (Algorithm 8), the size of the node set $B_t \setminus A_t$ to check reduces by at least 1 in each iteration. Therefore, it has a time complexity of $O(|V|^2 M)$.

4.4 Analysis of Upper Bounds

The previous section has established a uniform theoretical guarantee across all instances of the profit maximization problem under a certain condition. To provide online approximation guarantees for specific problem instances and to deal with the instances where the required condition is not satisfied, we next derive several upper bounds on the optimal solution to the profit maximization problem. These bounds are easy to compute and can be used to characterize the quality of the solutions constructed on any problem instances.

4.4.1 Upper Bound for Double Greedy

For the deterministic double greedy algorithm (Algorithm 7), besides the approximation guarantee provided by Corollary 4.1, we can also derive an upper bound on the optimal solution without any condition requirement. In fact, following the proof of Corollary 4.1, we have an upper bound

$$\mu_1 \triangleq 3 \cdot \phi(\hat{S}_D) - (\phi(A^*) + \phi(B^*))$$

on the profit of the optimal solution $\phi(S^*)$. Regardless of whether $\phi(A^*) + \phi(B^*)$ is non-negative or not, the above upper bound always holds. Thus, any seed set $\hat{S}_D$ returned by the double greedy algorithm has an online approximation guarantee of $\phi(\hat{S}_D)/\mu_1$. When $\phi(A^*) + \phi(B^*) > 0$, this guarantee is tighter than the constant factor $(1/3)$ of Corollary 4.1.

4.4.2 Upper Bounds Based on Submodularity

Now, we develop several upper bounds based solely on the submodularity of the profit function. In a nutshell, based on a given node set X , we first derive two modular upper bounds $m_X(Y)$ and $\bar{m}_X(Y)$ on $\phi(Y)$ for any node set Y . Then, we find the maxima of $m_X(Y)$ and $\bar{m}_X(Y)$ over all sets Y to derive two upper bounds $\mu(X)$ and $\bar{\mu}(X)$ on the maximum achievable profit $\max_{S \subseteq V} \phi(S)$. Specifically, leveraging Lemma 4.5 and restricting the sets X and Y within the boundaries A^* and B^* discovered by the iterative pruning (i.e., $A^* \subseteq X, Y \subseteq B^*$), we can obtain two upper bounds on $\phi(Y)$ as follows:

$$m_X(Y) \triangleq \phi(X) - \sum_{v \in X \setminus Y} \phi(v \mid B^* \setminus \{v\}) + \sum_{v \in Y \setminus X} \phi(v \mid X), \quad (4.3)$$

$$\bar{m}_X(Y) \triangleq \phi(X) - \sum_{v \in X \setminus Y} \phi(v \mid X \setminus \{v\}) + \sum_{v \in Y \setminus X} \phi(v \mid A^*). \quad (4.4)$$

Lemma 4.6. *For any two sets X and Y where $A^* \subseteq X, Y \subseteq B^*$, the above defined $m_X(Y)$ and $\bar{m}_X(Y)$ satisfy*

$$m_X(Y) \geq \phi(Y) \quad \text{and} \quad \bar{m}_X(Y) \geq \phi(Y).$$

Proof. For each node $v \in X \setminus Y$, since $X \cup Y \setminus \{v\} \subseteq B^* \setminus \{v\}$, it holds that $\phi(v \mid B^* \setminus \{v\}) \leq \phi(v \mid X \cup Y \setminus \{v\})$ due to the submodularity. By (4.1) and (4.3), we obtain $m_X(Y) \geq \phi(Y)$. Similarly, for each node $v \in Y \setminus X$, since $A^* \subseteq X \cap Y$, it holds that $\phi(v \mid A^*) \geq \phi(v \mid X \cap Y)$. By (4.2) and (4.4), we obtain $\bar{m}_X(Y) \geq \phi(Y)$. $\square$

Based on Lemma 4.6, we can find two series of upper bounds on the maximum value of $\phi(\cdot)$ as follows. For any set X where $A^* \subseteq X \subseteq B^*$, we define

$$\mu(X) \triangleq \max_{A^* \subseteq Y \subseteq B^*} m_X(Y) \quad \text{and} \quad \bar{\mu}(X) \triangleq \max_{A^* \subseteq Y \subseteq B^*} \bar{m}_X(Y).$$

Theorem 4.3. *For any set X where $A^* \subseteq X \subseteq B^*$,*

$$\mu(X) \geq \max_{S \subseteq V} \phi(S) \quad \text{and} \quad \bar{\mu}(X) \geq \max_{S \subseteq V} \phi(S).$$

Proof. In fact, $\mu(X) = \max_{A^* \subseteq Y \subseteq B^*} m_X(Y) \geq \max_{A^* \subseteq Y \subseteq B^*} \phi(Y) = \max_{S \subseteq V} \phi(S)$, where the first equality is by the definition of $\mu(X)$, the second inequality is due to Lemma 4.6, and the last equality is due to Theorem 4.2. Similarly, we have $\bar{\mu}(X) \geq \max_{S \subseteq V} \phi(S)$. $\square$

A nice feature of the above upper bounds is that $m_X(Y)$ and $\bar{m}_X(Y)$ are modular functions with respect to Y . It is easy to verify that

$$m_X(Y) = m_X(\emptyset) + \sum_{v \in Y} (m_X(\{v\}) - m_X(\emptyset)).$$

Thus, the upper bound $\mu(X)$ is given by

$$\mu(X) = m_X(\emptyset) + \sum_{v \in B^*} \max \{0, m_X(\{v\}) - m_X(\emptyset)\}. \quad (4.5)$$

Meanwhile, it is easy to show that

$$m_X(\{v\}) - m_X(\emptyset) = \begin{cases} \phi(v \mid B^* \setminus \{v\}), & \text{if } v \in X, \\ \phi(v \mid X), & \text{otherwise.} \end{cases} \quad (4.6)$$

Therefore, given X , $\mu(X)$ can be computed in $O(|B^*|)$ time. Similarly, $\bar{\mu}(X)$ can also be computed in $O(|B^*|)$ time.

Remark: Note that when the iterative pruning is not applied, B^* should be replaced by V in (4.3) and A^* should be replaced by $\emptyset$ in (4.4). We refer to these corresponding upper bounds on $\phi(Y)$ as $m'_X(Y)$ and $\bar{m}'_X(Y)$. Due to the submodularity of the profit function $\phi(\cdot)$, we have $\phi(v \mid B^* \setminus \{v\}) \geq \phi(v \mid V \setminus \{v\})$ and $\phi(v \mid A^*) \leq \phi(v \mid \emptyset)$ for any node v . Thus, it holds that $m_X(Y) \leq m'_X(Y)$ and $\bar{m}_X(Y) \leq \bar{m}'_X(Y)$. Therefore, an additional advantage of the iterative pruning is to make the upper bounds tighter.

4.4.3 Tighten the Bounds

Theorem 4.3 indicates that we can obtain upper bounds on the optimal profit by choosing any set X in $\mu(X)$ and $\bar{\mu}(X)$. However, it is not known yet what kind of X leads to tighter bounds. In the following, we show that the seed set returned by a greedy algorithm (either simple greedy or double greedy algorithms) can provide the *tightest* bound among all the seed sets examined by the greedy algorithm. To prove this claim, we start by showing that the second order marginal profit gain is diminishing if the set of nodes activated by a seed set S is a union of the nodes activated by each individual seed $u \in S$. Recall that $V_g(S)$ denotes the set of nodes activated by a seed set S in a sample outcome X of influence propagation.

Lemma 4.7. *If $V_g(S) = \bigcup_{u \in S} V_g(\{u\})$ holds for any set S under a diffusion model, then we have*

$$\phi(v \mid S) - \phi(v \mid T) \geq \phi(v \mid S \cup R) - \phi(v \mid T \cup R),$$

for any sets R, S, T where $S \subseteq T$, and any node $v \notin T \cup R$.

Proof. We know that $\phi(v \mid S) = \beta(v \mid S) - c(v)$ for any node $v \notin S$. So, for any node $v \notin T \supseteq S$,

$$\phi(v \mid S) - \phi(v \mid T) = \beta(v \mid S) - \beta(v \mid T). \quad (4.7)$$

On the other hand, in any sample outcome X of influence propagation, $V_g(S) \subseteq V_g(T)$ when $S \subseteq T$, which indicates $V_g(v \mid S) \supseteq V_g(v \mid T)$ where $V_g(v \mid S) \triangleq V_g(S \cup \{v\}) \setminus V_g(S)$ is the set of nodes activable by u but not S . Thus, by definition,

$$\begin{aligned} \beta(v \mid S) - \beta(v \mid T) &= \mathbb{E} \left[\sum_{u \in V_g(v \mid S)} b(u) \right] - \mathbb{E} \left[\sum_{u \in V_g(v \mid T)} b(u) \right] \\ &= \mathbb{E} \left[\sum_{u \in V_g(v \mid S)} b(u) - \sum_{u \in V_g(v \mid T)} b(u) \right] \\ &= \mathbb{E} \left[\sum_{u \in V_g(v \mid S) \setminus V_g(v \mid T)} b(u) \right]. \end{aligned}$$

As a result, (4.7) can be rewritten as

$$\phi(v \mid S) - \phi(v \mid T) = \mathbb{E} \left[\sum_{u \in V_g(v \mid S) \setminus V_g(v \mid T)} b(u) \right].$$

Similarly, we have

$$\phi(v \mid S \cup R) - \phi(v \mid T \cup R) = \mathbb{E} \left[\sum_{u \in V_g(v \mid S \cup R) \setminus V_g(v \mid T \cup R)} b(u) \right].$$

Thus, the lemma holds if $V_g(v \mid S) \setminus V_g(v \mid T) \supseteq V_g(v \mid S \cup R) \setminus V_g(v \mid T \cup R)$. We prove this relation as follows. First, since $V_g(S) = \bigcup_{u \in S} V_g(\{u\})$, we have $V_g(v \mid S) =$

$(\bigcup_{u \in S \cup \{v\}} V_g(\{u\})) \setminus (\bigcup_{u \in S} V_g(\{u\})) = V_g(\{v\}) \setminus V_g(S)$. Therefore,

$$\begin{aligned}
 & V_g(v \mid S) \setminus V_g(v \mid T) \\
 &= (V_g(\{v\}) \setminus V_g(S)) \setminus (V_g(\{v\}) \setminus V_g(T)) \\
 &= (V_g(\{v\}) \setminus V_g(S) \setminus V_g(\{v\})) \cup (V_g(\{v\}) \setminus V_g(S) \cap V_g(T)) \\
 &= \emptyset \cup (V_g(\{v\}) \setminus V_g(S) \cap V_g(T)) \\
 &= V_g(\{v\}) \setminus V_g(S) \cap V_g(T) \\
 &= V_g(\{v\}) \cap V_g(T) \setminus V_g(S).
 \end{aligned}$$

Similarly,

$$\begin{aligned}
 & V_g(v \mid S \cup R) \setminus V_g(v \mid T \cup R) \\
 &= V_g(\{v\}) \cap V_g(T \cup R) \setminus V_g(S \cup R) \\
 &= V_g(\{v\}) \cap (V_g(T \cup R) \setminus V_g(S \cup R)) \\
 &= V_g(\{v\}) \cap (V_g(T) \setminus V_g(S \cup R)) \\
 &= V_g(\{v\}) \cap V_g(T) \setminus V_g(S \cup R).
 \end{aligned}$$

Since $V_g(S) \subseteq V_g(S \cup R)$, the target relation is proved. $\square$

Many diffusion models, including the IC and LT models, fulfill the condition $V_g(S) = \bigcup_{u \in S} V_g(\{u\})$ in Lemma 4.7. By the submodularity of $\phi(\cdot)$, we already know that $\phi(v \mid S)$ decreases with the base set S . Lemma 4.7 shows that the decreasing rate is slowing down as the base set expands. This implies that the marginal profit gain drops faster in the earlier stage of seed selection than in the later stage. Based on Lemma 4.7, we can further establish the monotonicity of the upper bounds $\mu(X)$ and $\bar{\mu}(X)$ as X expands.

Lemma 4.8. *For any set X where $A^* \subseteq X \subseteq B^*$, and any node $u \in B^* \setminus A^*$, we have*

$$\begin{cases} \mu(X) \geq \mu(X \setminus \{u\}) & \text{if } \phi(X) \leq \phi(X \setminus \{u\}), \\ \bar{\mu}(X) \geq \bar{\mu}(X \cup \{u\}) & \text{if } \phi(X) \leq \phi(X \cup \{u\}). \end{cases}$$

Proof. Based on (4.5) and (4.6) in the earlier discussion, we can rewrite the upper bound $\mu(X)$ as

$$\mu(X) = m_X(\emptyset) + \sum_{v \in X} \max\{0, \phi(v \mid B^* \setminus \{v\})\} + \sum_{v \in B^* \setminus X} \max\{0, \phi(v \mid X)\}.$$

Similarly, for any node $u \in X \setminus A^*$,

$$\begin{aligned} \mu(X \setminus \{u\}) &= m_{X \setminus \{u\}}(\emptyset) + \sum_{v \in (X \setminus \{u\})} \max\{0, \phi(v \mid B^* \setminus \{v\})\} + \sum_{v \in B^* \setminus (X \setminus \{u\})} \max\{0, \phi(v \mid X \setminus \{u\})\}. \end{aligned}$$

It is easy to show that

$$m_X(\emptyset) - m_{X \setminus \{u\}}(\emptyset) = \phi(u \mid X \setminus \{u\}) - \phi(u \mid B^* \setminus \{u\}).$$

As a result,

$$\begin{aligned} \mu(X) - \mu(X \setminus \{u\}) &= \phi(u \mid X \setminus \{u\}) + \max\{0, -\phi(u \mid B^* \setminus \{u\})\} - \max\{0, \phi(u \mid X \setminus \{u\})\} \\ &\quad + \sum_{v \in B^* \setminus X} \left(\max\{0, \phi(v \mid X)\} - \max\{0, \phi(v \mid X \setminus \{u\})\} \right). \end{aligned}$$

If $\phi(X) \leq \phi(X \setminus \{u\})$, we have $\phi(u \mid B^* \setminus \{u\}) \leq \phi(u \mid X \setminus \{u\}) \leq 0$. Thus, $\max\{0, -\phi(u \mid B^* \setminus \{u\})\} = -\phi(u \mid B^* \setminus \{u\})$ and $\max\{0, \phi(u \mid X \setminus \{u\})\} = 0$. On the other hand, for any node $v \in B^* \setminus X$, we have $\max\{0, \phi(v \mid X)\} - \phi(v \mid X) = \max\{-\phi(v \mid X), 0\} \geq \max\{-\phi(v \mid X \setminus \{u\}), 0\} = \max\{0, \phi(v \mid X \setminus \{u\})\} - \phi(v \mid X \setminus \{u\})$. That is, $\max\{0, \phi(v \mid X)\} - \max\{0, \phi(v \mid X \setminus \{u\})\} \geq \phi(v \mid X) - \phi(v \mid X \setminus \{u\}) = \phi(X \cup \{v\}) - \phi(X) - \phi(X \cup \{v\} \setminus \{u\}) + \phi(X \setminus \{u\}) = \phi(u \mid X \cup \{v\} \setminus \{u\}) - \phi(u \mid X \setminus \{u\})$. Therefore,

$$\begin{aligned} \mu(X) - \mu(X \setminus \{u\}) &\geq \phi(u \mid X \setminus \{u\}) - \phi(u \mid B^* \setminus \{u\}) + \sum_{v \in B^* \setminus X} \left(\phi(u \mid X \cup \{v\} \setminus \{u\}) - \phi(u \mid X \setminus \{u\}) \right). \end{aligned}$$

Let $v_1, v_2, \dots, v_i$ be the set of elements in $B^* \setminus X$. Then,

$$\begin{aligned} &\sum_{v \in B^* \setminus X} \left(\phi(u \mid X \cup \{v\} \setminus \{u\}) - \phi(u \mid X \setminus \{u\}) \right) \\ &= \sum_{j=1}^i \left(\phi(u \mid X \cup \{v_j\} \setminus \{u\}) - \phi(u \mid X \setminus \{u\}) \right) \\ &\geq \sum_{j=1}^i \left(\phi(u \mid X \cup \{v_1, \dots, v_j\} \setminus \{u\}) - \phi(u \mid X \cup \{v_1, \dots, v_{j-1}\} \setminus \{u\}) \right) \\ &= \phi(u \mid B^* \setminus \{u\}) - \phi(u \mid X \setminus \{u\}). \end{aligned}$$

where the inequality is due to Lemma 4.7. Hence, we can conclude that $\mu(X) - \mu(X \setminus \{u\}) \geq 0$ if $\phi(X) \leq \phi(X \setminus \{u\})$.

The proof of $\bar{\mu}(X) \geq \bar{\mu}(X \cup \{u\})$ if $\phi(X) \leq \phi(X \cup \{u\})$ is analogous. $\square$

Lemma 4.8 indicates that the higher the profit of a seed set, the tighter the corresponding upper bound derived based on the seed set. Intuitively, the seed sets constructed by the greedy algorithms have increasing profits over iterations and thus, the final seed sets returned by the greedy algorithms can provide the tightest bounds. In the following, we prove it formally. Recall that the greedy algorithms start with the boundary node sets A^* and B^* discovered by the iterative pruning.

Theorem 4.4. *For the seed set S^g returned by the simple greedy algorithm or the double greedy algorithms, we have*

$$\begin{aligned} \mu(S^g) &\leq \mu(X_B) \leq \mu(B^*) = \bar{\mu}(B^*), \\ \text{and } \bar{\mu}(S^g) &\leq \bar{\mu}(X_A) \leq \bar{\mu}(A^*) = \mu(A^*), \end{aligned}$$

where X_B is any intermediate seed set from B^* to S^g and X_A is any intermediate seed set from A^* to S^g in the execution of the greedy algorithm.

Proof. For any set Y where $A^* \subseteq Y \subseteq B^*$, we have $A^* \setminus Y = \emptyset$. Thus, it follows from (4.3) and (4.4) that

$$\begin{aligned} m_{A^*}(Y) &= \phi(A^*) - \sum_{v \in A^* \setminus Y} \phi(v \mid B^* \setminus \{v\}) + \sum_{v \in Y \setminus A^*} \phi(v \mid A^*) \\ &= \phi(A^*) + \sum_{v \in Y \setminus A^*} \phi(v \mid A^*) \\ &= \phi(A^*) - \sum_{v \in A^* \setminus Y} \phi(v \mid A^* \setminus \{v\}) + \sum_{v \in Y \setminus A^*} \phi(v \mid A^*) \\ &= \bar{m}_{A^*}(Y). \end{aligned}$$

Therefore, by definition, $\mu(A^*) = \bar{\mu}(A^*)$.

Similarly, for any set Y where $A^* \subseteq Y \subseteq B^*$, we have $Y \setminus B^* = \emptyset$. As a result,

$$\begin{aligned}
 m_{B^*}(Y) &= \phi(B^*) - \sum_{v \in B^* \setminus Y} \phi(v \mid B^* \setminus \{v\}) + \sum_{v \in Y \setminus B^*} \phi(v \mid B^*) \\
 &= \phi(B^*) - \sum_{v \in B^* \setminus Y} \phi(v \mid B^* \setminus \{v\}) \\
 &= \phi(B^*) - \sum_{v \in B^* \setminus Y} \phi(v \mid B^* \setminus \{v\}) + \sum_{v \in Y \setminus B^*} \phi(v \mid A^*) \\
 &= \bar{m}_{B^*}(Y).
 \end{aligned}$$

So, by definition, $\mu(B^*) = \bar{\mu}(B^*)$.

Suppose there are n nodes in $B^* \setminus A^*$. We next show that these nodes can be arranged into a sequence $v_1, v_2, \dots, v_n$ such that

$$\begin{cases} \phi(X_j) \leq \phi(X_{j+1}) & \text{if } j < |S^g|, \\ \phi(X_j) \geq \phi(X_{j+1}) & \text{if } j \geq |S^g|. \end{cases}$$

where $|S^g|$ is the size of the seed set returned by the greedy algorithm, $X_0 = A^*$, $X_{|S^g|} = S^g$, $X_n = B^*$ and $X_{j+1} = X_j \cup \{v_{j+1}\}$ for any $0 \leq j < n$.

Specifically, for the simple greedy algorithm, let $\{v_1, v_2, \dots, v_{|S^g|}\}$ be the sequence of the seeds selected and $\{v_{|S^g|+1}, v_{|S^g|+2}, \dots, v_n\}$ be a random sequence of the unselected nodes in $B^* \setminus A^*$. Then, for any $0 \leq j < |S^g|$, we have $\phi(X_j) \leq \phi(X_{j+1})$ since the simple greedy algorithm selects a new seed with non-negative marginal profit gain in each iteration. On the other hand, for any $|S^g| \leq j < n$, we know that $\phi(X_{|S^g|}) \geq \phi(X_{|S^g|} \cup \{v_{j+1}\})$. Due to the submodularity of $\phi(\cdot)$, we have $\phi(X_{j+1}) - \phi(X_j) \leq \phi(X_{|S^g|} \cup \{v_{j+1}\}) - \phi(X_{|S^g|}) \leq 0$ since $X_{|S^g|} \subseteq X_j$.

For the double greedy algorithms, let $\{v_1, v_2, \dots, v_{|S^g|}\}$ be the sequence of the nodes added to S and $\{v_{|S^g|+1}, v_{|S^g|+2}, \dots, v_n\}$ be the reverse sequence of the nodes removed from T . From Lemma II.1 in [11], we know that $r^+ + r^- \geq 0$ in each iteration of Algorithm 7. Thus, the nodes with negative r^+ must have positive r^- . Then, according to the decision condition of Algorithm 7 (line 5), these nodes must be removed from T by both the deterministic and randomized double greedy algorithms. This implies that all the nodes added to S must have non-negative r^+ . As a result, for any $0 \leq j < |S^g|$, we have $\phi(v_{j+1} \mid X_j) = \phi(X_{j+1}) - \phi(X_j) \geq 0$ at the time when node v_{j+1} is added

to S . Similarly, the nodes with negative r^- must have positive r^+ . Then, these nodes must be added to S by the double greedy algorithms. This implies that all the nodes removed from T must have non-negative r^- . Consequently, for any $|S^g| \leq j < n$, we have $-\phi(v_{j+1} \mid X_j) = \phi(X_j) - \phi(X_{j+1}) \geq 0$ at the time when node v_{j+1} is removed from T .

Then, according to Lemma 4.8, we have $\mu(S^g) = \mu(X_{|S^g|}) \leq \mu(X_{|S^g|+1}) \leq \dots \leq \mu(B^*) = \bar{\mu}(B^*)$ and $\bar{\mu}(S^g) = \bar{\mu}(X_{|S^g|}) \leq \bar{\mu}(X_{|S^g|-1}) \leq \dots \leq \bar{\mu}(A^*) = \mu(A^*)$. Note that $X_0(= A^*)$, $X_1, \dots, X_{|S^g|-1}$, and $X_{|S^g|+1}, X_{|S^g|+2}, \dots, X_n(= B^*)$ are actually all the intermediate seed sets explored by the greedy algorithm.

Hence, the theorem is proved. $\square$

To obtain tighter upper bounds on the optimal profit, Theorem 4.4 suggests to derive bounds based on the seed sets returned by the simple greedy and double greedy algorithms. We refer to these two upper bounds as $\mu_2 \triangleq \mu(S^g)$ and $\mu_3 \triangleq \bar{\mu}(S^g)$. We shall study the effectiveness of these bounds in the experimental evaluation.

4.5 Evaluation

4.5.1 Experimental Setup

Datasets. We use the same real OSN datasets from [77] as those used in Chapter 3 (see Table 3.1 for their statistics).

Algorithms. The performance comparison includes the following algorithms.

- *Random* (Rand): It randomly selects a given number of k nodes. We run the algorithm 10 times and take their average as the expected profit. To explore different seed set sizes, we iterate through $k = \frac{|V|}{2^i}$ for $i = 0, 1, \dots, 10$ (where $|V|$ is the number of nodes in the social graph) and choose the one with the largest expected profit.
- *High Degree* (HD): It selects k nodes with the highest degrees. Similar to the random algorithm, we also iterate through different k values and choose the one producing the largest profit among $k = \frac{|V|}{2^i}$ for $i = 0, 1, \dots, 10$.

- *IMM/BCT*: IMM [105] is a state-of-the-art sampling-based method that can provide theoretical guarantees for finding the top- k influential nodes for influence maximization, where activating every node offers the same benefit. BCT [90] is proposed to handle the scenario that the nodes have weighted benefits if activated. Thus, in our experiments, IMM is adopted in the uniform benefit setting, whereas BCT is adopted in the weighted benefit setting. We set the algorithm parameters $\epsilon = 0.5$ and $l = 1$ according to the default setting in [105] for both IMM and BCT. We again choose the k value producing the largest profit among $k = \frac{|V|}{2^i}$ for $i = 0, 1, \dots, 10$.
- *Simple Greedy* (SG): We adopt the *Reverse Influence Sampling* method used in IMM/BCT for influence estimation in our proposed algorithms. The number of Reverse Reachable (RR) sets is set to the maximal number of RR sets generated in the IMM/BCT method among the above 11 cases of different k values. We also adopt the CELF technique [76] in the implementation of the simple greedy algorithm to enhance its efficiency.
- *Simple Greedy with Iterative Pruning* (SGIP): It runs simple greedy after conducting the iterative pruning as described in Section 4.3.3.
- *Double Greedy* (DG): We generate the same number of RR sets as that for SG. Since the deterministic and randomized double greedy algorithms perform quite similarly in terms of the profit generated and the running time taken, we shall report the results of only the deterministic algorithm in order to make the figures easier to read.
- *Double Greedy with Iterative Pruning* (DGIP): It runs double greedy after conducting the iterative pruning.

Parameter Settings. We use both the Independent Cascade (IC) and Linear Threshold (LT) diffusion models. The propagation probability $p_{u,v}$ of each edge $\langle u, v \rangle$ is set to the reciprocal of v 's in-degree, i.e., $p_{u,v} = 1/|I_v|$, as widely adopted by other studies [18, 62, 90, 91, 105, 106]. A monetary unit (e.g., one unit may be 1 US dollar or 100 Singapore dollars) is used to represent the benefit and cost. We test different benefit and cost settings. In the uniform benefit setting, every node has a unit benefit if activated.

In the weighted benefit setting, we assign each node v with a benefit value randomly generated from a normal distribution with a mean 3.0 and a standard deviation 1.0, i.e., $b(v) \sim N(3.0, 1.0)$. To avoid negative benefit values, we set $b(v) = 0$ if the randomly generated value is negative.² In the uniform cost setting, all nodes have the same costs for seed selection. In the degree-proportional cost setting, the cost of each node is set proportional to its out-degree to emulate that popular users need more incentives to participate. The ratio between the total seed selection cost of all nodes and total benefit of all nodes is controlled by a scale factor λ , i.e., $\sum_{v \in V} c(v) = \lambda \cdot \sum_{v \in V} b(v)$. The larger the factor λ , the higher the cost of seed selection relative to the benefit of influence spread. The default value of λ is set to 10. We have tested other values of λ and observed similar performance trends.

4.5.2 Profits Produced by Different Algorithms

Figures 4.3 and 4.4 show the profits produced by different algorithms under the IC model, uniform benefit setting and different cost settings. Comparing the seed selection algorithms, our greedy algorithms are more effective in optimizing the profit than the three baseline algorithms (Rand, HD and IMM). The iterative pruning technique can further improve the greedy algorithms (by up to 13%).

Under the uniform cost setting, our simple greedy algorithm degenerates to the IMM algorithm that iterates through all possible seed set sizes. Thus, as seen from Figure 4.3, the greedy algorithms and IMM produce similar profits, and they both outperform the high degree and random algorithms. The random algorithm, in particular, generates near-zero profit and is difficult to benefit from viral marketing.

Under the degree-proportional cost setting, as seen from Figure 4.4, the high degree and IMM algorithms perform even worse than the random algorithm with very negative profits produced. This is because under such a setting, the high-degree nodes have large costs. To further explore, we plot in Figure 4.5 the cost distribution over all the nodes and the influence spread of each node when it is chosen as the only seed (which indicates the maximum possible influence contributed by each node in any seed set according to

²The number of negative values is very small since about 99.7% of values drawn from a normal distribution are within three times of the standard deviation away from the mean, i.e., $\Pr[0 \leq b(v) \leq 6] \approx 99.7\%$.

Figure 4.3: Profits produced under uniform cost setting, uniform benefit setting and IC model.

the submodularity). Here, the nodes are indexed in decreasing order of their out-degrees, i.e., node #1 has the highest out-degree, node #2 has the second highest out-degree and so on. It can be seen from Figure 4.5 that most high-degree nodes are not profitable, i.e., their costs are larger than their influence spreads. In contrast, many low-degree nodes are profitable due to their lower costs.³ Thus, the best seed set to maximize profit should include mostly low-degree nodes. Therefore, influence maximization algorithms have poor performance.

We can also see that the double greedy algorithms perform considerably better than the simple greedy algorithm on the Facebook dataset in Figure 4.4(a). This can again

³Note that under the uniform benefit setting, the benefit is equivalent to the influence spread.

Figure 4.4: Profits produced under degree-proportional cost setting, uniform benefit setting and IC model.

be explained with the cost and influence distributions shown in Figure 4.5. As seen from Figure 4.5(a), there are a few high-degree nodes with influence larger than cost. In the simple greedy algorithm, these profitable high-degree nodes are selected first since their profits (influence spread less cost) are much higher than those of low-degree nodes. This, in consequence, would prevent many low-degree nodes from being further selected because their influence spreads largely overlap with the profitable high-degree nodes. On the other hand, the double greedy algorithm does not select the profitable high-degree nodes. Since the influence spreads of the profitable high-degree nodes overlap with the low-degree nodes, when these high-degree nodes are removed from a nearly-full seed set (which includes almost all the nodes), there is not much loss in the total influence spread.

Figure 4.5: Cost and benefit distributions by a single seed under degree-proportional cost setting, uniform benefit setting and IC model.

Thus, the marginal profit gains r^- for these high-degree nodes to quit from the nearly-full seed set are basically their seed selection costs (line 4 of Algorithm 7). Figure 4.5(a) shows that such costs are higher than the marginal profit gains r^+ generated by adding these high-degree nodes into an almost-empty seed set (line 3 of Algorithm 7). Thus, according to the double greedy algorithm (lines 5–8 of Algorithm 7), these high-degree nodes would not be selected. The nodes selected by the double greedy algorithm are mostly low-degree nodes which are able to generate similar total influence spread as the profitable high-degree nodes and have lower total cost than the latter. Therefore, the double greedy algorithm produces remarkably higher profit than the simple greedy algorithm. This phenomenon does not occur in the Wiki-Vote and Google+ datasets as all

the high-degree nodes are not profitable (Figures 4.5(b) and 4.5(c)). For the LiveJournal dataset, there are also a few profitable high-degree nodes. Meanwhile, there are quite many low-degree nodes offering even higher profits than these high-degree nodes. As a result, the profitable high-degree nodes are not selected by the simple greedy algorithm and the above phenomenon does not occur. These observations show that the double greedy algorithm is more robust than the simple greedy algorithm.

4.5.3 Running Time

Table 4.1: Running times of different algorithms (seconds).

Dataset	IMM	SG	SGIP	DG	DGIP
Facebook	1.28	0.21	0.45	0.22	0.45
Wiki-Vote	0.91	0.14	0.10	0.19	0.10
Google+	31.02	6.02	7.09	5.72	5.98
LiveJournal	3273.63	550.91	588.66	545.40	587.12

(a) Uniform Cost Setting

Dataset	IMM	SG	SGIP	DG	DGIP
Facebook	2.12	0.25	0.25	0.22	0.45
Wiki-Vote	1.10	0.08	0.09	0.08	0.09
Google+	29.42	6.87	6.53	6.94	6.55
LiveJournal	2756.32	563.16	581.76	555.90	582.53

(b) Degree-Proportional Cost Setting

Table 4.1 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 E5-2695 2.4GHz CPU and 64GB memory. As the running times for the random and high degree algorithms are very short (less than 0.01 second), we omit them in Table 4.1. It can be seen that the IMM algorithm runs significantly slower than our greedy algorithms. This is because IMM needs to test different seed set sizes separately to find the solution. On the other hand, the running times of different greedy algorithms are similar. This is because the major time for running these algorithms is taken by generating the RR sets and the numbers of RR sets used by different greedy algorithms are the same. The running times of our greedy algorithms are less than 600 seconds even

for the large LiveJournal dataset with millions of nodes. This demonstrates the efficiency of our algorithms.

4.5.4 Iterative Pruning Technique

Table 4.2: Impact of iterative pruning.

Dataset	$ A^* $	$ B^* $	$ B^* \setminus A^* $	$\phi(A^*) + \phi(B^*)$
Facebook	12	158	146	622
Wiki-Vote	54	241	187	2,104
Google+	715	856	141	33,624
LiveJournal	2,719	548,855	546,136	-2,460,900

(a) Uniform Cost Setting

Dataset	$ A^* $	$ B^* $	$ B^* \setminus A^* $	$\phi(A^*) + \phi(B^*)$
Facebook	53	2,589	2,536	-8,678
Wiki-Vote	4,808	4,808	0	9,537
Google+	36,070	43,062	6,992	54,947
LiveJournal	1,136,106	1,738,332	602,226	1,372,225

(b) Degree-Proportional Cost Setting

Table 4.2 summarizes the impact of the iterative pruning technique proposed in Section 4.3.3. As can be seen, pruning substantially reduces the number of nodes that need to be considered for seed selection (by at least 1 order of magnitude in most cases). In addition, with a scale factor $\lambda = 10$, the profit of selecting all the nodes as seeds is $\phi(V) = |V| - 10 \cdot |V| < 0$. Thus, running the double greedy algorithm with the entire node set would not offer any approximation guarantee. In contrast, as shown in Table 4.2, $\phi(A^*) + \phi(B^*) > 0$ holds for most of the cases tested. In these cases, by Corollary 4.1, the pruning technique enables strong theoretical guarantees on the seed sets constructed by double greedy algorithms. In particular, for the degree-proportional cost setting on Wiki-Vote, we have $A^* = B^*$ so that no node needs to be further checked for seed selection after pruning, which implies that the pruning process directly produces the optimal seed set for profit maximization.

Table 4.3: Upper bounds (normalized by the profit of DGIP).

Dataset	μ_1^{DG}	μ_2^{DG}	μ_3^{DG}	μ_1^{DGIP}	μ_2^{DGIP}	μ_3^{DGIP}
Facebook	49.79	1.21	8.80	2.26	1.07	1.25
Wiki-Vote	49.00	1.63	3.94	1.62	1.12	1.28
Google+	64.45	1.37	3.50	1.05	1.02	1.03
LiveJournal	56.44	1.51	26.21	5.87	1.28	5.64

(a) Uniform Cost Setting

Dataset	μ_1^{DG}	μ_2^{DG}	μ_3^{DG}	μ_1^{DGIP}	μ_2^{DGIP}	μ_3^{DGIP}
Facebook	101.99	2.31	12.76	27.05	2.15	11.22
Wiki-Vote	16.45	1.00	1.02	1.00	1.00	1.00
Google+	39.17	1.34	1.59	1.30	1.14	1.20
LiveJournal	63.30	2.30	7.89	1.70	1.31	1.31

(b) Degree-Proportional Cost Setting

4.5.5 Upper Bounds

Table 4.3 shows the upper bounds derived based on the seed sets returned by the double greedy algorithms without/with iterative pruning. To quantify their relative order, these bounds are normalized by the actual profit produced by the DGIP algorithm. As can be seen, for all the datasets tested, μ_1 is always the loosest upper bound among all those obtained while $\mu_2 = \mu(S^g)$ is always the tightest one no matter whether the pruning technique is used. Comparing the upper bounds derived from the DG and DGIP seed sets, the latter are considerably lower than the former. This implies that the iterative pruning technique can improve the bounds significantly.

Next, we examine the upper bounds derived from the DGIP seed sets in detail (right half of Table 4.3). For the cases where $\phi(A^*) + \phi(B^*) < 0$ (see Table 4.2), μ_1^{DGIP} is far above 3 times the profit returned by the algorithm. In these cases, μ_2^{DGIP} certifies online approximation guarantees from 46% to 78% for the seed set returned by the DGIP algorithm. For other cases where $\phi(A^*) + \phi(B^*) \geq 0$, all the bounds are rather close to the profit obtained by the DGIP algorithm, and μ_2^{DGIP} certifies at least 76% online approximation guarantees for the seed set returned by the algorithm. These observations imply that the DGIP algorithm usually performs quite close to the optimal solution.

Figure 4.6 shows the evolution of upper bounds (normalized by the profit of DGIP) derived from the intermediate seed sets explored by the DGIP algorithm. Note that

 (a) Normalized Upper Bound μ_2 under Uniform Cost Setting

 (b) Normalized Upper Bound μ_3 under Uniform Cost Setting

 (c) Normalized Upper Bound μ_2 under Degree-Proportional Cost Setting

 (d) Normalized Upper Bound μ_3 under Degree-Proportional Cost Setting

Figure 4.6: Upper bounds under weighted benefit setting and IC model on LiveJournal dataset.

the upper bounds are never smaller than the solution values and thus the normalized upper bounds are always larger than 1. We show both the bounds derived with pruning (using A^* and B^*) and without pruning (using $\emptyset$ and V as explained in the remark of Section 4.4.2). It can be seen that the upper bounds are progressively tightened as the algorithm executes. The final seed set S^g returned by DGIP provides the tightest bound among all the seed sets examined. This confirms the theoretical results of Theorem 4.4. It can also be seen that the bounds derived with pruning are much tighter than those without pruning, which verifies the remark in Section 4.4.2.

Figure 4.7: Profits produced under degree-proportional cost setting, weighted benefit setting and IC model.

4.5.6 Weighted Benefit Setting

Figure 4.7 shows the profits produced by different algorithms under the IC model, degree-proportional cost setting, and weighted benefit setting. The results are similar to those under the uniform benefit setting (Figure 4.4). Our greedy algorithms considerably outperform the three baseline algorithms (Rand, HD and BCT). The iterative pruning technique can further improve the greedy algorithms.

4.5.7 LT Model

As discussed, our solutions and analysis can be applied to a variety of diffusion models. In this set of experiments, we further evaluate the algorithms with the LT model.

Figure 4.8: Profits produced under degree-proportional cost setting, weighted benefit setting and LT model.

Figure 4.8 shows the profits produced by different algorithms under the LT model, degree-proportional cost setting, and weighted benefit setting. In general, the results are quite similar to those under the IC model. Our greedy algorithms perform much better than the baseline algorithms, and the double greedy algorithm is more robust than the simple greedy algorithm.

4.6 Summary

In this chapter, we have studied a profit maximization problem for advertisers conducting viral marketing in OSNs. The objective is to select initial seed nodes to maximize the total profit that accounts for the benefit of influence spread as well as the cost of

seed selection. The non-monotone characteristic of the profit metric differentiates the profit maximization problem from the traditional influence maximization problem. We have presented simple greedy and double greedy algorithms for seed selection, and proposed several new techniques to enhance/benchmark their performance and expand the applicability of their approximation guarantees. Experimental results with real OSN datasets show that: (1) our greedy algorithms substantially outperform several baseline algorithms; (2) our upper bounds on the maximum achievable profit offer much tighter guarantees on the quality of the solutions constructed by various algorithms; (3) our iterative pruning technique can substantially tighten the upper bounds and further improve the greedy algorithms.

Chapter 5

Towards Profit Maximization for Online Social Network Providers

5.1 Introduction

Advertisers often delegate the operation of viral marketing campaigns to OSN providers who have the complete information of their social network structures [14]. 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 is willing to pay the OSN provider a commission for each user that adopts its product or shares its ads. Thus, 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. For example, for a cloud-based OSN running viral video marketing [86], 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 chapter, 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.

This new profit metric is challenging to optimize as it is significantly different from the influence metric that has been studied widely. It is well known that the influence spread generated by a seed set is submodular and monotone under many diffusion models [6, 63, 98], which makes it easy to design influence maximization algorithms with strong approximation guarantees [89]. In Chapter 4, we have addressed profit maximization from the advertiser’s perspective by putting together the benefit of influence spread and the cost of seed selection, where 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 [85, 103, 117], the resultant profit metric is still submodular. In contrast, in the problem of this chapter, both the revenue and cost of the OSN provider are attached to the diffusion process. As a result, the 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 chapter, 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 chapter is organized as follows. Section 5.2 defines the profit maximization problem for the OSN provider. Section 5.3 elaborates the design of our two-phase framework. Section 5.4 develops the bounds for quality measurement. Section 5.5 presents the experimental evaluation. Finally, Section 5.6 summarizes the chapter.

5.2 Problem Formulation

5.2.1 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 .

As presented in Section 2.1, 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., $\sigma(S) = \mathbb{E}[|V_g(S)|]$.

5.2.2 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 advertiser). The benefit $\beta(S)$ of influence spread with a seed set S is given by the total benefit brought by all the nodes activated:

$$\beta(S) = \mathbb{E} \left[\sum_{v \in V_g(S)} b(v) \right]. \quad (5.1)$$

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 is activated. The cost $\gamma(S)$ of influence propagation with a seed set S is given by the total cost incurred by all the nodes activated:

$$\gamma(S) = \mathbb{E} \left[\sum_{v \in V_g(S)} c(v) \right]. \quad (5.2)$$

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

$$\phi(S) = \beta(S) - \gamma(S), \quad (5.3)$$

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 (5.3) as

$$\phi(S) = \mathbb{E} \left[\sum_{v \in V_g(S)} (b(v) - c(v)) \right] \triangleq \mathbb{E} \left[\sum_{v \in V_g(S)} w(v) \right]. \quad (5.4)$$

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 $\bar{b}(v) = \max\{0, w(v)\}$ and $\bar{c}(v) = \max\{0, -w(v)\}$. Then, the benefit and cost metrics become

$$\bar{\beta}(S) = \mathbb{E} \left[\sum_{v \in V_g(S)} \bar{b}(v) \right], \quad (5.5)$$

$$\text{and } \bar{\gamma}(S) = \mathbb{E} \left[\sum_{v \in V_g(S)} \bar{c}(v) \right], \quad (5.6)$$

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 [63]. 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 $\gamma(S \cup \{v\}) - \gamma(S) \geq \gamma(T \cup \{v\}) - \gamma(T)$. Although both the benefit and cost metrics are monotone and submodular, the profit metric is neither monotone nor submodular (an example is shown in Figure 5.1). Thus, the methods used for maximizing monotone submodular functions would perform poorly for our profit maximization problem as shall be demonstrated in our experiments. In the following, we develop heuristic algorithms to address our profit maximization problem.

Figure 5.1: Non-monotonicity and non-submodularity of the profit function $\phi(\cdot)$.

Example 5.1. Figure 5.1 gives an example to illustrate that the profit function $\phi(\cdot)$ is neither monotone nor submodular. In this example, each node is associated with a benefit weight b and a cost weight c . The propagation probability of each edge is 1. Once a node is activated, it will activate all of its neighbors, i.e., the propagation probability of each edge is 1. 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.

5.3 Two-Phase Framework

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

- **Pruning phase (Algorithm 9):** We develop an iterative pruning technique to cut the search space.
- **Search phase (Algorithms 10 and 11):** We use some heuristics, including greedy [63] and modular-modular [57], to find the solution in the reduced search space.

5.3.1 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

$$\phi(v \mid S) = \phi(S \cup \{v\}) - \phi(S), \quad (5.7)$$

$$\beta(v \mid S) = \beta(S \cup \{v\}) - \beta(S), \quad (5.8)$$

$$\gamma(v \mid S) = \gamma(S \cup \{v\}) - \gamma(S). \quad (5.9)$$

Algorithm 9: IterativePrune

```

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

We start by proposing an iterative pruning approach that can dramatically reduce the search space from the power set of V to a smaller space for maximizing the profit function. Algorithm 9 shows the pseudo code of the pruning algorithm, which generalizes the idea of the iterative pruning technique presented in Section 4.3.3. 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 9 extends this idea in an iterative manner to reduce the search space. It is easy to verify that after each iteration, the newly generated space is a subset of that in the previous iteration, i.e., $A_t \subseteq A_{t+1} \subseteq B_{t+1} \subseteq B_t$.

Lemma 5.1. $A_t \subseteq A_{t+1} \subseteq B_{t+1} \subseteq B_t$ for any $t \geq 0$.

Proof. By the definitions in lines 3 and 4 of Algorithm 9, A_t gradually expands over iterations and B_t gradually shrinks, i.e., $A_t \subseteq A_{t+1}$ and $B_{t+1} \subseteq B_t$. What is left is to show that A_t is always a subset of B_t . We prove it by induction. Obviously, $A_0 = \emptyset \subseteq V = B_0$. Suppose that $A_t \subseteq B_t$ holds for some $t \geq 0$. Then, we can easily get

that $A_t \subseteq B_{t+1}$ by the definition in line 4 of Algorithm 9 and that $A_{t+1} \subseteq B_t$ by the definition in line 3 of Algorithm 9. On the other hand, for any node $v \in B_t \setminus A_t$, it is obvious that $A_t \subseteq B_t \setminus \{v\}$. Thus, for any node $v \in A_{t+1} \setminus A_t \subseteq B_t \setminus A_t$, we have $\beta(v \mid A_t) - \gamma(v \mid B_t \setminus \{v\}) \geq \beta(v \mid B_t \setminus \{v\}) - \gamma(v \mid A_t) > 0$, where the first inequality is due to the submodularity of $\beta(\cdot)$ and $\gamma(\cdot)$ and the second inequality is by the definition of A_{t+1} in line 3 of Algorithm 9. It indicates that $v \in B_{t+1}$ as well. Therefore, $A_{t+1} \setminus A_t \subseteq B_{t+1}$, which implies $A_{t+1} \subseteq B_{t+1}$ since we already know that $A_t \subseteq B_{t+1}$. Thus, we can conclude that $A_{t+1} \subseteq B_{t+1}$. $\square$

Furthermore, it can be proved that any seed set outside the resultant space $[A^*, B^*]$ delimited by the node sets A^* and B^* returned by Algorithm 9 can be transformed to a seed set in $[A^*, B^*]$ with higher profit.

Theorem 5.1. *For any node set S and $t \geq 0$, let $S_t = S \cap B_t \cup A_t$, it holds that*

$$\phi(S_t) \leq \phi(S_{t+1}), \quad (5.10)$$

where the “=” holds if and only if $S_t = S_{t+1}$.

Proof. By the definition of S_t , we easily have $A_t \subseteq S_t \subseteq B_t$. We already know that $A_t \subseteq A_{t+1} \subseteq B_{t+1} \subseteq B_t$ by Lemma 5.1. Thus,

$$\begin{aligned} S_{t+1} &= S \cap B_{t+1} \cup A_{t+1} \\ &= S \cap (B_t \setminus (B_t \setminus B_{t+1})) \cup A_{t+1} \\ &= S \cap B_t \setminus (S \cap (B_t \setminus B_{t+1})) \cup A_{t+1} \\ &= S \cap B_t \cup A_{t+1} \setminus (S \cap (B_t \setminus B_{t+1}) \setminus A_{t+1}) \\ &= S \cap B_t \cup A_{t+1} \setminus (S \cap (B_t \setminus B_{t+1})) \\ &= S \cap B_t \cup A_t \cup (A_{t+1} \setminus A_t) \setminus (S \cap (B_t \setminus B_{t+1})) \\ &= S_t \cup (A_{t+1} \setminus A_t) \setminus (S \cap (B_t \setminus B_{t+1})). \end{aligned} \quad (5.11)$$

By line 3 of Algorithm 9, for any node $v \in A_{t+1} \setminus A_t$, it holds that $\beta(v \mid B_t \setminus \{v\}) - \gamma(v \mid A_t) > 0$. Then, for any $S_t \subseteq T \subseteq S_t \cup (A_{t+1} \setminus A_t)$, since $A_t \subseteq T$ and $T \subseteq B_t \setminus \{v\}$,

we have $\phi(v \mid T) = \beta(v \mid T) - \gamma(v \mid T) \geq \beta(v \mid B_t \setminus \{v\}) - \gamma(v \mid A_t) > 0$ due to the submodularity. Let $v_1, v_2, \dots, v_k$ be the set of nodes in $A_{t+1} \setminus A_t \setminus S_t$. It follows that

$$\begin{aligned}
 \phi(S_t) &< \phi(S_t \cup \{v_1\}) \\
 &< \phi(S_t \cup \{v_1, v_2\}) \\
 &< \dots \\
 &< \phi(S_t \cup \{v_1, v_2, \dots, v_k\}) \\
 &= \phi(S_t \cup (A_{t+1} \setminus A_t \setminus S_t)) \\
 &= \phi(S_t \cup (A_{t+1} \setminus A_t)).
 \end{aligned}$$

Similarly, by line 4 of Algorithm 9, for any node $v \in B_t \setminus B_{t+1}$, it holds that $\beta(v \mid A_t) - \gamma(v \mid B_t \setminus \{v\}) < 0$. Then, for any $S_t \cup (A_{t+1} \setminus A_t) \setminus (S \cap (B_t \setminus B_{t+1})) \subseteq T \subseteq S_t \cup (A_{t+1} \setminus A_t)$, since $A_t \subseteq T$ and $T \subseteq B_t \setminus \{v\}$, we have $\phi(v \mid T) = \beta(v \mid T) - \gamma(v \mid T) \leq \beta(v \mid A_t) - \gamma(v \mid B_t \setminus \{v\}) < 0$ due to the submodularity. Let $u_1, u_2, \dots, u_l$ be the set of nodes in $S \cap (B_t \setminus B_{t+1})$. It follows that

$$\begin{aligned}
 \phi(S_t \cup (A_{t+1} \setminus A_t)) &< \phi(S_t \cup (A_{t+1} \setminus A_t) \setminus \{u_1\}) \\
 &< \phi(S_t \cup (A_{t+1} \setminus A_t) \setminus \{u_1, u_2\}) \\
 &< \dots \\
 &< \phi(S_t \cup (A_{t+1} \setminus A_t) \setminus \{u_1, u_2, \dots, u_l\}) \\
 &= \phi\left(S_t \cup (A_{t+1} \setminus A_t) \setminus (S \cap (B_t \setminus B_{t+1}))\right) \\
 &= \phi(S_{t+1}),
 \end{aligned}$$

where the last equality is due to (5.11). Therefore, $\phi(S_t) < \phi(S_{t+1})$ if $A_{t+1} \setminus A_t \setminus S_t \neq \emptyset$ or $S \cap (B_t \setminus B_{t+1}) \neq \emptyset$. On the other hand, if $A_{t+1} \setminus A_t \setminus S_t = S \cap (B_t \setminus B_{t+1}) = \emptyset$, based on (5.11), we have $S_t = S_{t+1}$. $\square$

Theorem 5.1 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.

Corollary 5.1. *For any node set S , if $S \notin [A^*, B^*]$, then*

$$\phi(S) < \phi(S \cap B^* \cup A^*). \quad (5.12)$$

Proof. According to Theorem 5.1, $\phi(S) = \phi(S_0) \leq \phi(S_1) \leq \dots \leq \phi(S \cap B^* \cup A^*)$ since $S = S \cap B_0 \cup A_0 = S_0$. On the other hand, if every “=” holds, we have $S = S_0 = S_1 = \dots = S \cap B^* \cup A^*$, which implies $A^* \subseteq S \subseteq B^*$. This is contradictory to $S \notin [A^*, B^*]$. Thus, it holds that $\phi(S) < \phi(S \cap B^* \cup A^*)$. $\square$

By Corollary 5.1, 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 space $[A^*, B^*]$ retains all the optimal seed sets. As a result, we can reduce the search space from $[\emptyset, V]$ to $[A^*, B^*]$. Our pruning approach can be used prior to any seed selection algorithms to improve the solution quality.

Corollary 5.2. *For any seed set S^* producing the maximum achievable profit, it holds that $A^* \subseteq S^* \subseteq B^*$.*

Proof. Suppose $S^* \notin [A^*, B^*]$. By Corollary 5.1, we have $\phi(S^*) < \phi(S^* \cap B^* \cup A^*)$. This is contradictory to the fact that S^* produces the maximum achievable profit. Thus, it holds that $A^* \subseteq S^* \subseteq B^*$. $\square$

Example 5.2. *Figure 5.2 gives an example to illustrate how the pruning algorithm works as well as the above theorem and corollaries. In this example, each node is associated with a benefit weight b and a cost weight c . Each edge has a propagation probability p . This example assumes the Independent Cascade (IC) diffusion model, in which 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. Initially, the seed nodes S are activated, while all the other nodes are not activated. 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 Figure 5.2, 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

Figure 5.2: Iterative pruning under the Independent Cascade diffusion model.

$\phi_t^+(v) = \beta(v \mid A_t) - \gamma(v \mid B_t \setminus \{v\})$ describe the calculations in Algorithm 9. At iteration 1, $\phi_1^-(v_1) = \beta(v_1 \mid \{v_2, v_3, v_4\}) - \gamma(v_1 \mid \emptyset) = 1 \times b(v_1) - (1 \times c(v_1) + 0.3 \times c(v_2) + (1 - (1 - 0.4) \times (1 - 0.3 \times 0.2)) \times c(v_4)) = 1.5 - (1 + 0.3 + 2.18) = 1.5 - 3.48 = -1.98 < 0$. Similarly, $\phi_1^-(v_2) = -0.6 < 0$, $\phi_1^-(v_3) = 0.5 > 0$, $\phi_1^-(v_4) = -4.328 < 0$, $\phi_1^+(v_1) = 1.972 > 0$, $\phi_1^+(v_2) = 1.7 > 0$, $\phi_1^+(v_3) = 2.6 > 0$, and $\phi_1^+(v_4) = 0.32 > 0$. Thus, v_3 is added to A_1 , i.e., $A_1 = \{v_3\}$ 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 $\phi_2^+(v_4) = -0.28 < 0$. Thus, v_4 is removed from B_2 , i.e., $A_2 = \{v_3\}$ 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 5.1 and Corollary 5.1. Moreover, it is easy to verify that the optimal seed set $S^* = \{v_2, v_3\}$ belongs to $[A^*, B^*]$, which confirms Corollary 5.2.

Finally, it can be shown that the normalization of the benefit and cost weights as described in Section 5.2.2 can only increase the amount of the search space cut by our pruning technique.

Theorem 5.2. Let $\bar{A}^*$ and $\bar{B}^*$ be the node sets returned by Algorithm 9 under the normalized form. Then,

$$A^* \subseteq \bar{A}^* \subseteq \bar{B}^* \subseteq B^*. \quad (5.13)$$

Proof. Let $\alpha(S)$ denote the overlapping value between the benefit of influence spread and the cost of influence propagation for a seed set S , which is defined as

$$\alpha(S) = \mathbb{E} \left[\sum_{v \in V_g(S)} \min\{b(v), c(v)\} \right].$$

It is easy to verify that $\alpha(\cdot)$ is also submodular and monotone. By the definition of normalization in Section 5.2.2, we have $\beta(S) = \bar{\beta}(S) + \alpha(S)$ and $\gamma(S) = \bar{\gamma}(S) + \alpha(S)$. We prove that $A_t \subseteq \bar{A}_t \subseteq \bar{B}_t \subseteq B_t$ for any $t \geq 0$ by induction. Obviously, it holds that $\emptyset = A_0 = \bar{A}_0 \subseteq \bar{B}_0 = B_0 = V$. Suppose that $A_t \subseteq \bar{A}_t \subseteq \bar{B}_t \subseteq B_t$ holds for some $t \geq 0$.

Then, for any node $v \in B_t \setminus \bar{B}_t$, we have $\beta(v \mid B_t \setminus \{v\}) - \gamma(v \mid A_t) = \bar{\beta}(v \mid B_t \setminus \{v\}) - \bar{\gamma}(v \mid A_t) + \alpha(v \mid B_t \setminus \{v\}) - \alpha(v \mid A_t) \leq \bar{\beta}(v \mid B_t \setminus \{v\}) - \bar{\gamma}(v \mid A_t) \leq \bar{\beta}(v \mid \bar{B}_t \setminus \{v\}) - \bar{\gamma}(v \mid \bar{A}_t)$, where the inequalities are due to the submodularity of $\alpha(\cdot)$, $\bar{\beta}(\cdot)$ and $\bar{\gamma}(\cdot)$. By Lemma 5.1, $\bar{A}_{t-1} \subseteq \bar{A}_t \subseteq \bar{B}_{t-1} = \bar{B}_{t-1} \setminus \{v\} \subseteq \bar{B}_t \setminus \{v\}$. It follows from the submodularity that $\bar{\beta}(v \mid \bar{B}_t \setminus \{v\}) - \bar{\gamma}(v \mid \bar{A}_t) \leq \bar{\beta}(v \mid \bar{A}_{t-1}) - \bar{\gamma}(v \mid \bar{B}_{t-1} \setminus \{v\}) < 0$, where the last inequality is due to $v \notin \bar{B}_t$. This implies that $(A_{t+1} \setminus A_t) \cap (B_t \setminus \bar{B}_t) = \emptyset$. Since $A_{t+1} \setminus A_t \subseteq B_t$, we have $A_{t+1} \setminus A_t \subseteq \bar{B}_t$. Since $A_t \subseteq \bar{B}_t$, it follows that $A_{t+1} \subseteq \bar{B}_t$. Thus, for any node $v \in A_{t+1} \setminus \bar{A}_t \subseteq \bar{B}_t \setminus \bar{A}_t$, we have $\bar{\beta}(v \mid \bar{B}_t \setminus \{v\}) - \bar{\gamma}(v \mid \bar{A}_t) \geq \bar{\beta}(v \mid B_t \setminus \{v\}) - \bar{\gamma}(v \mid A_t) = \beta(v \mid B_t \setminus \{v\}) - \gamma(v \mid A_t) - \alpha(v \mid B_t \setminus \{v\}) + \alpha(v \mid A_t) \geq \beta(v \mid B_t \setminus \{v\}) - \gamma(v \mid A_t) > 0$, where the first two inequalities are due to the submodularity and the last inequality is due to $v \in A_{t+1}$. This implies that $A_{t+1} \setminus \bar{A}_t \subseteq \bar{A}_{t+1}$. Since $\bar{A}_t \subseteq \bar{A}_{t+1}$, it follows that $A_{t+1} \subseteq \bar{A}_{t+1}$.

Similarly, for any node $v \in \bar{A}_t \setminus A_t$, we have $\beta(v \mid A_t) - \gamma(v \mid B_t \setminus \{v\}) = \bar{\beta}(v \mid A_t) - \bar{\gamma}(v \mid B_t \setminus \{v\}) + \alpha(v \mid A_t) - \alpha(v \mid B_t \setminus \{v\}) \geq \bar{\beta}(v \mid A_t) - \bar{\gamma}(v \mid B_t \setminus \{v\}) \geq \bar{\beta}(v \mid \bar{A}_t) - \bar{\gamma}(v \mid \bar{B}_t \setminus \{v\}) \geq \bar{\beta}(v \mid \bar{B}_{t-1} \setminus \{v\}) - \bar{\gamma}(v \mid \bar{A}_{t-1}) > 0$, where the first three inequalities are due to the submodularity of $\alpha(\cdot)$, $\bar{\beta}(\cdot)$ and $\bar{\gamma}(\cdot)$, and the last inequality is due to $v \in \bar{A}_t$. This implies that $\bar{A}_t \setminus A_t \subseteq B_{t+1}$. Since $A_t \subseteq A_{t+1} \subseteq B_{t+1}$, it follows that $\bar{A}_t \subseteq B_{t+1}$. Thus, for any node $v \in \bar{B}_t \setminus B_{t+1} \subseteq \bar{B}_t \setminus \bar{A}_t$, we have $\bar{\beta}(v \mid \bar{A}_t) - \bar{\gamma}(v \mid \bar{B}_t \setminus \{v\}) \leq \bar{\beta}(v \mid A_t) - \bar{\gamma}(v \mid B_t \setminus \{v\}) = \beta(v \mid A_t) - \gamma(v \mid B_t \setminus \{v\}) - \alpha(v \mid A_t) + \alpha(v \mid B_t \setminus \{v\}) \leq \beta(v \mid A_t) - \gamma(v \mid B_t \setminus \{v\}) < 0$, where the first two inequalities are due to the submodularity and the last inequality is due to $v \notin B_{t+1}$. This implies that $\bar{B}_{t+1} \cap (\bar{B}_t \setminus B_{t+1}) = \emptyset$. Since $\bar{B}_{t+1} \subseteq \bar{B}_t$, it follows that $\bar{B}_{t+1} \subseteq B_{t+1}$. By induction, we have $A_t \subseteq \bar{A}_t \subseteq \bar{B}_t \subseteq B_t$ for any $t \geq 0$ (it also holds after converged) and thus, $A^* \subseteq \bar{A}^* \subseteq \bar{B}^* \subseteq B^*$. $\square$

We shall experimentally evaluate the additional reduction in the search space due to normalization with real data sets in Section 5.5.

5.3.2 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 $P=NP$ [57].

Greedy Algorithm: We apply a simple hill-climbing idea to optimize the profit function (5.3). Algorithm 10 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 $B^* \setminus S$ have negative marginal profit gains.

Algorithm 10: Greedy

```

1 initialize  $S \leftarrow A^*$ ;
2 while True do
3   find  $u \leftarrow \arg \max_{v \in B^* \setminus S} \{\phi(v \mid S)\}$ ;
4   if  $\phi(u \mid S) \leq 0$  then return  $S$ ;
5    $S \leftarrow S \cup \{u\}$ ;
```

Modular-Modular Algorithm: Iyer et al. [57] introduced a modular-modular (ModMod) algorithm for optimizing the difference between submodular functions. Since we just need to search the space $[A^*, B^*]$ after pruning, we adapt the ModMod algorithm as shown in Algorithm 11. In line 4 of Algorithm 11, $h_{X^t}^\pi(Y; \beta)$ is a modular lower bound of $\beta(Y)$ that is tight at set X^t , i.e., $h_{X^t}^\pi(Y; \beta) \leq \beta(Y)$ for any $Y \subseteq V$ and $h_{X^t}^\pi(X^t; \beta) = \beta(X^t)$, 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 $h_{X^t}^\pi(Y; \beta) - m_{X^t}(Y; \gamma)$ 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 $\phi(X^{t+1}) \geq h_{X^t}^\pi(X^{t+1}; \beta) - m_{X^t}(X^{t+1}; \gamma) \geq h_{X^t}^\pi(X^t; \beta) - m_{X^t}(X^t; \gamma) = \phi(X^t)$. This indicates that Algorithm 11 always increases the profit value at every iteration. Examples of the modular upper and lower bounds will be given in Section 5.4.1.

Algorithm 11: Modular-Modular (ModMod)

```

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

5.3.3 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 [63, 76, 93] and Reverse Influence Sampling [9, 90, 91, 105, 106], can be used. Suppose the time complexity for computing the marginal profit gain of adding/removing a node u into/from a seed set S is $O(M)$. For the iterative pruning process (Algorithm 9), 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 $O((|V| + |V| - 1 + \dots + 1)M) = O(|V|^2 M)$ 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 i th iteration. Thus, it takes at most $O((k_1 + k_1 - 1 + \dots + 1)M) = O(k_1^2 M)$ time, which means the total time complexity of the Greedy algorithm is $O((|V|^2 + k_1^2)M)$. 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 $O((|V|^2 + k_1 k_2)M)$.

Diffusion Models: Our analysis and algorithms are general frameworks that can be adapted to any diffusion models which are submodular. There are many submodular diffusion models, such as the *triggering model* [63], the *continuous-time models* [17, 32, 81, 98], and the *topic-aware models* [6, 15].

5.4 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 commonly adopted diffusion models are analytically intractable. For example, computing the exact influence spread under the Independent Cascade diffusion model is $\#P$ -hard [18]. Thus, the benefit brought and the cost incurred by a seed set can only be estimated via some sampling approaches [9, 63]. As a result, the *sampling error* also affects the quality measurement of the seed set.

In this section, we propose techniques to analyze the aforementioned gap and sampling error. These techniques enable us to evaluate the online 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 online approximation guarantee of S^o using the estimated values $\tilde{\phi}(S^o)$ and $\tilde{\mu}$.

5.4.1 Upper Bound of Optimum

To derive our upper 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_X^1 and m_X^2 that are tight at a given set X [57]:

$$m_X^1(Y) \triangleq f(X) - \sum_{v \in X \setminus Y} f(v \mid V \setminus \{v\}) + \sum_{v \in Y \setminus X} f(v \mid X), \quad (5.14)$$

$$m_X^2(Y) \triangleq f(X) - \sum_{v \in X \setminus Y} f(v \mid X \setminus \{v\}) + \sum_{v \in Y \setminus X} f(v \mid \emptyset). \quad (5.15)$$

In the previous section, we have reduced the search space to $[A^*, B^*]$. Thus, 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:

$$m_X^3(Y) \triangleq f(X) - \sum_{v \in X \setminus Y} f(v \mid B^* \setminus \{v\}) + \sum_{v \in Y \setminus X} f(v \mid X), \quad (5.16)$$

$$m_X^4(Y) \triangleq f(X) - \sum_{v \in X \setminus Y} f(v \mid X \setminus \{v\}) + \sum_{v \in Y \setminus X} f(v \mid A^*). \quad (5.17)$$

It is easy to show that the bounds $m_X^3(Y)$ and $m_X^4(Y)$ remain tight at X , i.e., $m_X^3(X) = m_X^4(X) = f(X)$, and they are tighter than $m_X^1(Y)$ and $m_X^2(Y)$ at other sets, i.e., $m_X^1(Y) \geq m_X^3(Y) \geq f(Y)$ and $m_X^2(Y) \geq m_X^4(Y) \geq f(Y)$ 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 [40]. Let π be any permutation of V that places all the nodes in X before the nodes in $V \setminus X$. Let $S_i^\pi = \{\pi(1), \pi(2), \dots, \pi(i)\}$ be a chain formed by the permutation, where $S_0^\pi = \emptyset$ and $S_{|X|}^\pi = X$. We define $h_X^\pi(\pi(i))$ as

$$h_X^\pi(\pi(i)) = f(S_i^\pi) - f(S_{i-1}^\pi). \quad (5.18)$$

Then, $h_X^\pi(Y) = \sum_{v \in Y} h_X^\pi(v)$ is a lower bound of $f(Y)$, which is tight at X , i.e., $h_X^\pi(Y) \leq f(Y)$ for any $Y \subseteq V$ and $h_X^\pi(X) = f(X)$. 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 the Optimum: 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

$$\mu_i(X) \triangleq \max_{A^* \subseteq Y \subseteq B^*} m_X^i(Y; \beta) - h_X^\pi(Y; \gamma), \quad (5.19)$$

where $m_X^i(Y; \beta)$ ($i = 3, 4$) denotes the modular upper bound on the benefit function β and $h_X^\pi(Y; \gamma)$ 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 5.3. *For any set X where $A^* \subseteq X \subseteq B^*$,*

$$\mu(X) \geq \max_{S \subseteq V} \phi(S). \quad (5.20)$$

Proof. Let S^* be an optimal seed set producing the maximum achievable profit. We can directly obtain that $\mu_i(X) \geq m_X^i(S^*; \beta) - h_X^\pi(S^*; \gamma) \geq \beta(S^*) - \gamma(S^*) = \phi(S^*)$, where the first inequality is by the definition of $\mu_i(X)$ and the second inequality is due to the modular upper and lower bounds. $\square$

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 online approximation guarantee of S^o can be estimated by $\phi(S^o)/\mu(S^o)$.

5.4.2 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 [9, 90, 91, 105, 106], which was discussed in Chapter 3.

Definition 5.1 (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 $p(v) = b(v)/b$, where $b = b(V) = \sum_{v \in V} b(v)$ is the total benefit weight of all nodes. For a seed set S , the connection between its benefit and a random RR set R is $\beta(S) = b \cdot \Pr[S \cap R \neq \emptyset]$, where $\Pr[S \cap R \neq \emptyset]$ is the probability that R contains at least one node in S . 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., $p(v) = c(v)/c$, where $c = c(V) = \sum_{v \in V} c(v)$ 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 $\Lambda_\beta(S) \cdot b / \theta_\beta$. To analyze the sampling error for the benefit estimation, we make use of the Chernoff-Hoeffding Theorem [24].

Lemma 5.2 (Chernoff-Hoeffding Theorem [24]). *Let $Z_1, Z_2, \dots, Z_\theta$ denote random variables that are independently and identically distributed in the interval $[0, 1]$. Consider the sum $Z = \sum_{i=1}^\theta Z_i$ with expectation $\mathbb{E}[Z]$. Then, for any fixed $\theta > 0$ and $\lambda > 0$, we have*

$$\Pr [Z - \mathbb{E}[Z] \geq \lambda] \leq \exp \left(-\frac{\lambda^2}{2\mathbb{E}[Z] + \frac{2}{3}\lambda} \right), \quad (5.21)$$

$$\Pr [Z - \mathbb{E}[Z] \leq -\lambda] \leq \exp \left(-\frac{\lambda^2}{2\mathbb{E}[Z]} \right). \quad (5.22)$$

Based on the Chernoff-Hoeffding Theorem, we can establish the following relation between $\Lambda_\beta(S)$ and the real benefit $\beta(S)$.

Theorem 5.4. *For θ_β random RR sets that are independent of S and any given $\delta \in (0, 1)$, we have*

$$\Pr \left[\beta(S) \geq \left(\left(\sqrt{\Lambda_\beta(S) + \frac{2a}{9}} - \sqrt{\frac{a}{2}} \right)^2 - \frac{a}{18} \right) \cdot \frac{b}{\theta_\beta} \right] \geq 1 - \delta, \quad (5.23)$$

$$\Pr \left[\beta(S) \leq \left(\sqrt{\Lambda_\beta(S) + \frac{a}{2}} + \sqrt{\frac{a}{2}} \right)^2 \cdot \frac{b}{\theta_\beta} \right] \geq 1 - \delta, \quad (5.24)$$

where $a = \ln(1/\delta)$ is a function of δ .

Proof. The proof is similar to that of Theorem 3.3 in Chapter 3. Let $\zeta_\beta(S)$ be the expected fraction of samples covered by the seed set S . We have $\zeta_\beta(S) = \beta(S)/b$. To simplify the notation, we omit the common symbol S in what follows, e.g., β represents $\beta(S)$ and Λ_β represents $\Lambda_\beta(S)$. Then, the inequality (5.23) is equivalent to

$$\Pr \left[\zeta_\beta < \left(\left(\sqrt{\Lambda_\beta + \frac{2a}{9}} - \sqrt{\frac{a}{2}} \right)^2 - \frac{a}{18} \right) \cdot \frac{1}{\theta_\beta} \right] \leq \delta$$

and (5.24) is equivalent to

$$\Pr \left[\zeta_\beta > \left(\sqrt{\Lambda_\beta + \frac{a}{2}} + \sqrt{\frac{a}{2}} \right)^2 \cdot \frac{1}{\theta_\beta} \right] \leq \delta.$$

We prove the former first. In fact,

$$\Pr \left[\zeta_\beta < \left(\left(\sqrt{\Lambda_\beta + \frac{2a}{9}} - \sqrt{\frac{a}{2}} \right)^2 - \frac{a}{18} \right) \cdot \frac{1}{\theta_\beta} \right]$$

$$\begin{aligned}
 &= \Pr \left[\sqrt{\zeta_\beta \theta_\beta + \frac{a}{18}} < \sqrt{\Lambda_\beta + \frac{2a}{9}} - \sqrt{\frac{a}{2}} \right] + \Pr \left[\sqrt{\zeta_\beta \theta_\beta + \frac{a}{18}} < \sqrt{\frac{a}{2}} - \sqrt{\Lambda_\beta + \frac{2a}{9}} \right] \\
 &= \Pr \left[\sqrt{\zeta_\beta \theta_\beta + \frac{a}{18}} < \sqrt{\Lambda_\beta + \frac{2a}{9}} - \sqrt{\frac{a}{2}} \right] \\
 &= \Pr \left[\left(\sqrt{\zeta_\beta \theta_\beta + \frac{a}{18}} + \sqrt{\frac{a}{2}} \right)^2 < \Lambda_\beta + \frac{2a}{9} \right] \\
 &= \Pr \left[\sqrt{2a\zeta_\beta \theta_\beta + \frac{a^2}{9}} + \frac{a}{3} < \Lambda_\beta - \zeta_\beta \theta_\beta \right] \\
 &\leq \exp \left(- \frac{(\sqrt{2a\zeta_\beta \theta_\beta + \frac{a^2}{9}} + \frac{a}{3})^2}{2\zeta_\beta \theta_\beta + \frac{2}{3} \cdot (\sqrt{2a\zeta_\beta \theta_\beta + \frac{a^2}{9}} + \frac{a}{3})} \right) \\
 &= \exp \left(- \frac{(\sqrt{2a\zeta_\beta \theta_\beta + \frac{a^2}{9}} + \frac{a}{3})^2}{(\sqrt{2\zeta_\beta \theta_\beta + \frac{a}{9}} + \frac{\sqrt{a}}{3})^2} \right) \\
 &= \exp(-a) \\
 &= \delta,
 \end{aligned}$$

where the inequality is due to Lemma 5.2. Note that in the second equality, when

$$\sqrt{\frac{a}{2}} - \sqrt{\Lambda_\beta + \frac{2a}{9}} \geq 0 \quad \text{or equivalently} \quad \Lambda_\beta \leq \frac{5a}{18},$$

since

$$\sqrt{\zeta_\beta \theta_\beta + \frac{a}{18}} \geq \sqrt{\frac{a}{18}}$$

and

$$\sqrt{\frac{a}{2}} - \sqrt{\Lambda_\beta + \frac{2a}{9}} \leq \sqrt{\frac{a}{2}} - \sqrt{\frac{2a}{9}} = \sqrt{\frac{a}{18}},$$

it follows that the probability

$$\Pr \left[\sqrt{\zeta_\beta \theta_\beta + \frac{a}{18}} < \sqrt{\frac{a}{2}} - \sqrt{\Lambda_\beta + \frac{2a}{9}} \right] = 0.$$

The proof of the latter is analogous.

$$\begin{aligned}
 &\Pr \left[\zeta_\beta > \left(\sqrt{\Lambda_\beta + \frac{a}{2}} + \sqrt{\frac{a}{2}} \right)^2 \cdot \frac{1}{\theta_\beta} \right] \\
 &= \Pr \left[\sqrt{\zeta_\beta \theta_\beta} > \sqrt{\Lambda_\beta + \frac{a}{2}} + \sqrt{\frac{a}{2}} \right]
 \end{aligned}$$

$$\begin{aligned}
 &= \Pr \left[\left(\sqrt{\zeta_\beta \theta_\beta} - \sqrt{\frac{a}{2}} \right)^2 > \Lambda_\beta + \frac{a}{2} \right] \\
 &= \Pr \left[-\sqrt{2a\zeta_\beta \theta_\beta} > \Lambda_\beta - \zeta_\beta \theta_\beta \right] \\
 &\leq \exp \left(-\frac{2a\zeta_\beta \theta_\beta}{2\zeta_\beta \theta_\beta} \right) \\
 &= \delta.
 \end{aligned}$$

Hence, the theorem is proved. $\square$

According to Theorem 5.4, given $\Lambda_\beta(S)$, we can define a lower bound and an upper bound of $\beta(S)$ with a probability at least $1 - \delta$ as

$$\begin{cases} \beta_l(S) \triangleq \left(\left(\sqrt{\Lambda_\beta(S) + \frac{2a}{9}} - \sqrt{\frac{a}{2}} \right)^2 - \frac{a}{18} \right) \cdot \frac{b}{\theta_\beta}, \\ \beta_u(S) \triangleq \left(\sqrt{\Lambda_\beta(S) + \frac{a}{2}} + \sqrt{\frac{a}{2}} \right)^2 \cdot \frac{b}{\theta_\beta}. \end{cases} \quad (5.25)$$

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 $\Lambda_\gamma(S) \cdot c / \theta_\gamma$. 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 $1 - \delta$. 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 $\Pr[\phi(S^o) \geq \beta_l(S^o) - \gamma_u(S^o)] \geq \Pr[\beta(S^o) \geq \beta_l(S^o) \text{ and } \gamma(S^o) \leq \gamma_u(S^o)] = 1 - \Pr[\beta(S^o) < \beta_l(S^o) \text{ or } \gamma(S^o) > \gamma_u(S^o)] \geq 1 - (\delta + \delta) > 1 - 2\delta$.

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 5.4, 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 $\tilde{\phi}(S) = \Lambda_\beta(S) \cdot b / \theta_\beta - \Lambda_\gamma(S) \cdot c / \theta_\gamma$. Then, $\beta_u(S) - \gamma_l(S)$ can be represented as a function η of $\Lambda_\beta(S)$ and $\tilde{\phi}(S)$, i.e., $\eta(\Lambda_\beta(S), \tilde{\phi}(S))$. We can prove that $\eta(\Lambda_\beta(S), \tilde{\phi}(S))$ 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 5.4.1. As a result, for any seed set S , the upper bound $\eta(\Lambda_\beta(S), \tilde{\phi}(S))$ of its profit is bounded above by $\eta(\theta_\beta, \tilde{\mu}(S^o))$. In this way, we can obtain the sampling error for the upper bound $\mu(S^o)$.

Theorem 5.5. *For an estimated upper bound $\tilde{\mu}(S^o)$ on the maximum achievable profit from the RR sets generated, we have*

$$\Pr \left[\max_{S \subseteq V} \phi(S) \leq \tilde{\mu}(S^o) + \varepsilon(\tilde{\mu}(S^o)) \right] \geq 1 - 2\delta, \quad (5.26)$$

where $\varepsilon(\tilde{\mu}(S^o))$ is the sampling error for $\tilde{\mu}(S^o)$ such that $\varepsilon(\tilde{\mu}(S^o)) = a(\rho_\beta - \frac{2\rho_\gamma}{3}) + \rho_\gamma \sqrt{2a(\rho_\beta \theta_\beta - \tilde{\mu}(S^o))/\rho_\gamma + 4a^2/9} + \rho_\beta \sqrt{2a\theta_\beta + a^2}$, and $\rho_\beta = b/\theta_\beta$ and $\rho_\gamma = c/\theta_\gamma$.

Proof. Imagine that we evaluate S^* with the benefit and cost samples. By Theorem 5.4, we know that

$$\begin{aligned} \Pr [\phi(S^*) \leq \beta_u(S^*) - \gamma_l(S^*)] &\geq \Pr [\beta(S^*) \leq \beta_u(S^*) \text{ and } \gamma(S^*) \geq \gamma_l(S^*)] \\ &\geq 1 - \left(\Pr [\beta(S^*) > \beta_u(S^*)] + \Pr [\gamma(S^*) < \gamma_l(S^*)] \right) \\ &\geq 1 - (\delta + \delta) \\ &= 1 - 2\delta. \end{aligned} \quad (5.27)$$

However, without knowing S^* , $\beta_u(S^*)$ and $\gamma_l(S^*)$ cannot be obtained. In what follows, we are going to bound $\beta_u(S^*) - \gamma_l(S^*)$ by an upper bound on the function $\eta(S) = \beta_u(S) - \gamma_l(S)$. To simplify the notations, and we omit the common symbol S in what follows, e.g., η represents $\eta(S)$. Then, the estimated profit $\tilde{\phi} = \rho_\beta \Lambda_\beta - \rho_\gamma \Lambda_\gamma$. Consider η as a function of $\tilde{\phi}$ and Λ_β . We study the monotonicity of η with respect to $\tilde{\phi}$ and Λ_β . By the definition of η , we have

$$\begin{aligned} \eta(\Lambda_\beta, \tilde{\phi}) &= \rho_\beta \left(\Lambda_\beta + \sqrt{2a\Lambda_\beta + a^2} + a \right) - \rho_\gamma \left(\Lambda_\gamma - \sqrt{2a\Lambda_\gamma + \frac{4a^2}{9}} + \frac{2a}{3} \right) \\ &= \tilde{\phi} + \rho_\beta \sqrt{2a\Lambda_\beta + a^2} + a(\rho_\beta - \frac{2\rho_\gamma}{3}) + \rho_\gamma \sqrt{\frac{2a(\rho_\beta \Lambda_\beta - \tilde{\phi})}{\rho_\gamma} + \frac{4a^2}{9}}, \end{aligned}$$

which is increasing with Λ_β under any given $\tilde{\phi}$. Meanwhile, we also have

$$\begin{aligned}\eta(\Lambda_\beta, \tilde{\phi}) &= \rho_\beta \left(\sqrt{\Lambda_\beta + \frac{a}{2}} + \frac{a}{2} \right)^2 - \rho_\gamma \left(\left(\sqrt{\Lambda_\gamma + \frac{2a}{9}} - \frac{a}{2} \right)^2 - \frac{a}{18} \right) \\ &= \rho_\beta \left(\sqrt{\Lambda_\beta + \frac{a}{2}} + \frac{a}{2} \right)^2 - \rho_\gamma \left(\left(\sqrt{\frac{\rho_\beta \Lambda_\beta - \tilde{\phi}}{\rho_\gamma} + \frac{2a}{9}} - \frac{a}{2} \right)^2 - \frac{a}{18} \right),\end{aligned}$$

which is increasing with $\tilde{\phi}$ under any given Λ_β .

For an estimated upper bound $\tilde{\mu}(S^o)$ on the maximum achievable profit, we have $\tilde{\phi}(S^*) \leq \tilde{\mu}(S^o)$. On the other hand, it naturally holds that $\Lambda_\beta(S^*) \leq \theta_\beta$. Thus,

$$\begin{aligned}\beta_u(S^*) - \gamma_l(S^*) &= \eta(\Lambda_\beta(S^*), \tilde{\phi}(S^*)) \\ &\leq \eta(\theta_\beta, \tilde{\mu}(S^o)) \\ &= \tilde{\mu}(S^o) + \rho_\beta \sqrt{2a\theta_\beta + a^2} + a(\rho_\beta - \frac{2\rho_\gamma}{3}) + \rho_\gamma \sqrt{\frac{2a(\rho_\beta \theta_\beta - \tilde{\mu}(S^o))}{\rho_\gamma} + \frac{4a^2}{9}}.\end{aligned}$$

Hence, the theorem is proved. $\square$

In summary, by Theorems 5.4 and 5.5, we have the online approximation guarantee that

$$\frac{\phi(S^o)}{\max_{S \subseteq V} \phi(S)} \geq \frac{\beta_l(S^o) - \gamma_u(S^o)}{\tilde{\mu}(S^o) + \varepsilon(\tilde{\mu}(S^o))} \quad (5.28)$$

with a probability at least $1 - 4\delta$.

5.4.3 Reduce Sampling Error via Normalization

As discussed in Section 5.2.2, 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.

Consider the sampling error limit ε under θ samples that can provide a probability guarantee of $1 - 2\delta$, i.e., $\Pr[-\varepsilon_1 \leq \Lambda - \zeta\theta \leq \varepsilon_2] \geq 1 - 2\delta$ (where ζ is the expected value of

the random variable and Λ is the sum of θ samples). According to the Chernoff-Hoeffding inequality (Lemma 5.2), ε_1 and ε_2 are given by

$$\varepsilon_1 = \sqrt{a\zeta\theta} \quad \text{and} \quad \varepsilon_2 = \sqrt{2a\zeta\theta + \frac{a^2}{9}} + \frac{a}{3}, \quad (5.29)$$

where $a = \ln(1/\delta)$. We denote ε_β and ε_γ as the sampling error limits for the benefit and cost metrics under θ_β and θ_γ samples that can satisfy the Chernoff-Hoeffding inequality, i.e., $\Pr[-\varepsilon_{\beta,1} \leq \tilde{\beta}(S) - \beta(S) \leq \varepsilon_{\beta,2}] \geq 1 - 2\delta$ and $\Pr[-\varepsilon_{\gamma,1} \leq \tilde{\gamma}(S) - \gamma(S) \leq \varepsilon_{\gamma,2}] \geq 1 - 2\delta$. Then, $\varepsilon_{\phi,1} = \varepsilon_{\beta,1} + \varepsilon_{\gamma,2}$ and $\varepsilon_{\phi,2} = \varepsilon_{\beta,2} + \varepsilon_{\gamma,1}$ give a total sampling error limits for the profit metric which guarantees $\Pr[-\varepsilon_{\phi,1} \leq \tilde{\phi}(S) - \phi(S) \leq \varepsilon_{\phi,2}] \geq 1 - 4\delta$. Likewise, let $\bar{\varepsilon}_\beta$, $\bar{\varepsilon}_\gamma$ and $\bar{\varepsilon}_\phi$ denote the sampling error limits under the normalized form.

Theorem 5.6. *The normalization can reduce the sampling error limits, i.e., $\bar{\varepsilon}_{\phi,1} \leq \varepsilon_{\phi,1}$ and $\bar{\varepsilon}_{\phi,2} \leq \varepsilon_{\phi,2}$.*

Proof. Similar to the definitions of b and c , we define $\bar{b} = \bar{b}(V) = \sum_{v \in V} \bar{b}(v)$ and $\bar{c} = \bar{c}(V) = \sum_{v \in V} \bar{c}(v)$. Then, $\bar{b} - \bar{c} = b - c$. For a given seed set S , let $\bar{\zeta}_\beta(S)$ and $\bar{\zeta}_\gamma(S)$ denote the expected fractions of benefit and cost samples covered by S under the normalized form. To simplify the notation, we omit the common symbol S in what follows, e.g., β represents $\beta(S)$ and $\bar{\zeta}_\beta$ represents $\bar{\zeta}_\beta(S)$. Since $\beta = \frac{b}{\theta_\beta} \cdot \zeta_\beta \theta_\beta$, based on the definitions of ε_1 and ε_2 in (5.29), we have

$$\varepsilon_{\beta,1} = \frac{b\varepsilon_1}{\theta_\beta} = \frac{b\sqrt{a\zeta_\beta\theta_\beta}}{\theta_\beta} = \sqrt{\frac{ab^2\zeta_\beta}{\theta_\beta}} = \sqrt{\frac{ab\beta}{\theta_\beta}}, \quad (5.30)$$

and

$$\varepsilon_{\beta,2} = \frac{b\varepsilon_2}{\theta_\beta} = \frac{b\left(\sqrt{2a\zeta_\beta\theta_\beta + \frac{a^2}{9}} + \frac{a}{3}\right)}{\theta_\beta} = \sqrt{\frac{2ab\beta}{\theta_\beta} + \frac{a^2b^2}{9\theta_\beta^2}} + \frac{ab}{3\theta_\beta}. \quad (5.31)$$

Similarly, we have

$$\bar{\varepsilon}_{\beta,1} = \sqrt{\frac{a\bar{b}\bar{\beta}}{\theta_\beta}}, \quad \bar{\varepsilon}_{\beta,2} = \sqrt{\frac{2a\bar{b}\bar{\beta}}{\theta_\beta} + \frac{a^2\bar{b}^2}{9\theta_\beta^2}} + \frac{a\bar{b}}{3\theta_\beta}, \quad (5.32)$$

$$\varepsilon_{\beta,1} = \sqrt{\frac{ac\gamma}{\theta_\gamma}}, \quad \varepsilon_{\gamma,2} = \sqrt{\frac{2ac\gamma}{\theta_\gamma} + \frac{a^2c^2}{9\theta_\gamma^2}} + \frac{ac}{3\theta_\gamma}, \quad (5.33)$$

$$\bar{\varepsilon}_{\gamma,1} = \sqrt{\frac{a\bar{c}\bar{\gamma}}{\theta_\gamma}}, \quad \bar{\varepsilon}_{\gamma,2} = \sqrt{\frac{2a\bar{c}\bar{\gamma}}{\theta_\gamma} + \frac{a^2\bar{c}^2}{9\theta_\gamma^2}} + \frac{a\bar{c}}{3\theta_\gamma}. \quad (5.34)$$

Note that $b \geq \bar{b}$, $c \geq \bar{c}$, $\beta \geq \bar{\beta}$ and $\gamma \geq \bar{\gamma}$. Together with (5.30)–(5.34), we have $\varepsilon_{\beta,1} \geq \bar{\varepsilon}_{\beta,1}$, $\varepsilon_{\beta,2} \geq \bar{\varepsilon}_{\beta,2}$, $\varepsilon_{\gamma,1} \geq \bar{\varepsilon}_{\gamma,1}$ and $\varepsilon_{\gamma,2} \geq \bar{\varepsilon}_{\gamma,2}$. Therefore, $\varepsilon_{\phi,1} = \varepsilon_{\beta,1} + \varepsilon_{\gamma,2} \geq \bar{\varepsilon}_{\beta,1} + \bar{\varepsilon}_{\gamma,2} = \bar{\varepsilon}_{\phi,1}$ and $\varepsilon_{\phi,2} = \varepsilon_{\beta,2} + \varepsilon_{\gamma,1} \geq \bar{\varepsilon}_{\beta,2} + \bar{\varepsilon}_{\gamma,1} = \bar{\varepsilon}_{\phi,2}$. $\square$

Theorem 5.6 states that the normalization can reduce the sampling error limit to provide a given probability guarantee and thus, it can improve the solution quality which shall be demonstrated in the experiments.

5.5 Evaluation

5.5.1 Experimental Setup

Datasets. We use the same real OSN datasets from [77] as those used in Chapters 3 and 4 (see Table 3.1 for their statistics).

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 5.4.1, we have described one such lower bound and two such upper bounds. We use ModMod-1 to refer to that using the upper bound $m_X^3(Y)$ defined in Eq. (5.16) and use ModMod-2 to refer to that using the upper bound $m_X^4(Y)$ defined in Eq. (5.17). 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 [9, 90, 91, 105, 106].

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 $i = 0, 1, \dots, 10$ (where $|V|$ is the network size) and choose the k value producing the largest profit.

Parameter Settings. In all the experiments, we use the Independent Cascade (IC) diffusion model (as described in Section 2.1.1), a uniform benefit distribution (where every node has a unit benefit if activated), and a degree-proportional cost distribution (where the cost of each node is set proportional to its out-degree) as the default settings. 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 [18, 62, 90, 91, 105, 106].

We use a normalized monetary unit (e.g., one unit may be 1 US dollar or 100 Singapore dollars) to represent the benefit and cost of each node. By default, we normalize the benefit and cost weights as described in Section 5.2.2. 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 5.4.

5.5.2 Profits Produced by Different Algorithms

Figure 5.3 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. In fact, BenefitMax even produces negative profit on the Google+ and LiveJournal datasets when a small number of RR sets are generated. The reason is that the sampling error is relatively large with few samples. Then, the seed set constructed by BenefitMax produces much lower benefit than estimated. As a result, the final profit may be negative since BenefitMax attempts to maximize the benefit

Figure 5.3: Profits produced by different algorithms.

only without considering the cost. 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.

5.5.3 Running Times of Different Algorithms

Figure 5.4 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 E5-2695 2.4GHz CPU and 64GB 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 Figure 5.4. It can be seen that

(a) Facebook

(b) Wiki-Vote

(c) Google+

(d) LiveJournal

Figure 5.4: Running times of different algorithms.

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). For most cases, our heuristic methods complete execution within 100 seconds. Even for the LiveJournal dataset with millions of nodes and 10 million RR sets, the ModMod algorithms can obtain the solution within 600 seconds while the Greedy algorithm takes about 1,500 seconds. These observations demonstrate the efficiency of our two-phase algorithms.

Table 5.1: Search space reduction.

Dataset	$ V $	$ A^* $	$ B^* $	$ B^* \setminus A^* $	Reduction
Facebook	4.0K	2.0K	3.1K	1.1K	72.3%
Wiki-Vote	7.1K	5,041	5,054	13	99.8%
Google+	107.6K	80.6K	82.4K	1.8K	98.3%
LiveJournal	4.8M	2.5M	3.3M	0.8M	83.7%

(a) With Normalization

Dataset	$ V $	$ A^* $	$ B^* $	$ B^* \setminus A^* $	Reduction
Facebook	4.0K	37	4,039	4,002	0.9%
Wiki-Vote	7.1K	5.0K	6.8K	1.8K	74.6%
Google+	107.6K	62.2K	97.1K	34.9K	67.6%
LiveJournal	4.8M	1.3M	4.3M	3.0M	38.3%

(b) Without Normalization

5.5.4 Iterative Pruning Technique

Table 5.1 shows the amount of search space reduction by the iterative pruning technique proposed in Section 5.3.1. 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 72.3%, 99.8%, 98.3% and 83.7% of the original network size for the Facebook, Wiki-Vote, 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 (0.9%, 74.6%, 67.6% and 38.3% for the Facebook, Wiki-Vote, Google+ and LiveJournal datasets respectively). This observation confirms Theorem 5.2.

Figure 5.5 shows the impact of the search space reduction on the profit produced by our heuristics, where O-X refers to heuristic X without the search space reduction. As claimed in Corollary 5.1, 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 (Figure 5.5(d)), 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. These observations demonstrate the effectiveness of our pruning technique.

Figure 5.5: Impact of iterative pruning technique on profit produced (algorithms with prefix “O-” do not use pruning technique).

Figure 5.6 shows the impact of the search space reduction on the running time of our heuristics. As can be seen, the pruning technique can also help reduce the running time considerably, especially for larger datasets. By pruning the search space first, our heuristic algorithms can run up to 3 orders faster. These observations demonstrate the efficiency of our pruning technique.

5.5.5 Online Approximation Guarantee of Solution Quality

We evaluate the quality of the seed sets returned by different algorithms using the techniques presented in Section 5.4. In our evaluation, we always choose the tighter bound between $\mu_3(X)$ and $\mu_4(X)$ as defined in (5.19). We set $\delta = 2.5 \times 10^{-7}$ so that the online

(a) Facebook

(b) Wiki-Vote

(c) Google+

(d) LiveJournal

Figure 5.6: Impact of iterative pruning technique on running time (algorithms with prefix “O-” do not use pruning technique).

Table 5.2: Online approximation guarantees of seed sets obtained.

Dataset	Greedy	ModMod-1	ModMod-2
Facebook	92.0%	92.2%	86.7%
Wiki-Vote	99.6%	99.7%	99.6%
Google+	98.7%	98.8%	98.6%
LiveJournal	72.9%	75.8%	74.3%

approximation guarantees obtained by (5.28) have a high confidence of $1 - 10^{-6}$. Table 5.2 shows the online 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 online approximation guarantees above 85%, 99%, 98% and 70% for

Figure 5.7: Impact of normalization (algorithms with prefix “O-” do not use pruning technique and with postfix “-w/o” are without normalization).

the Facebook, Wiki-Vote, Google+, LiveJournal datasets respectively. This implies that our proposed upper bounds on the maximum achievable profit are quite tight and the heuristic algorithms perform rather close to the optimal.

5.5.6 Normalization

In Section 5.5.4, 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. Figure 5.7 shows the profit produced by the Greedy algorithm with and without the normalization.¹ As can be seen, no matter whether the

¹The results for the ModMod algorithms are similar.

pruning technique is used, the Greedy algorithm with normalization can always produce higher profit than that without normalization for all the datasets tested. This confirms Theorem 5.6 that the normalization can reduce the sampling error limit and thus, it can further improve the solution quality.

5.6 Summary

In this chapter, 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. Experiments are conducted with real OSN datasets to compare different algorithms and demonstrate the effectiveness and efficiency of our techniques.

Our two-phase framework is general and can be applied to any problem of optimizing the difference between submodular functions. For instance, when the cost function in Chapter 4 is submodular rather than modular, the optimization problem changes from unconstrained submodular maximization to maximizing the difference between submodular functions. Often, the cost function naturally has a diminishing returns property. For example, there is typically a discount when purchasing free samples in bulk.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we have studied three seed selection problems for viral marketing, including the classical influence maximization problem and two profit maximization problems for advertisers and OSN providers respectively. Due to the submodularity of the influence spread under many diffusion models, all the three problems can be attributed to submodular optimization problems, i.e., monotone submodular maximization with a cardinality constraint, unconstrained submodular maximization, and maximizing the difference between submodular functions. The main contributions of the dissertation are the new technological advancements in the algorithms design and analysis for these problems. Extensive experiments are carried out to demonstrate the effectiveness and efficiency of the algorithms developed. In what follows, the contributions for each problem are summarized.

In Chapter 3, we have proposed a MAG framework to maximize the approximation guarantee for the classical influence maximization problem. We have derived a new online upper bound for the maximum coverage problem and developed a new method to analyze the sampling errors of the Reverse Influence Sampling approach. These techniques together can substantially improve the current best factor of $(1 - 1/e - \varepsilon)$ for influence maximization. Moreover, we have extended our MAG algorithm to speed up the achievement of the conventional approximation guarantee of $(1 - 1/e - \varepsilon)$.

In Chapter 4, we have studied a profit maximization problem for advertisers. The objective is to select initial seed nodes to maximize the total profit that accounts for

the benefit of influence spread as well as the cost of seed selection. The non-monotone characteristic of the profit metric differentiates the profit maximization problem from the traditional influence maximization problem. We have presented simple greedy and double greedy algorithms for seed selection, and proposed pruning and upper bounding techniques to enhance/benchmark their performance and expand the applicability of their approximation guarantees.

In Chapter 5, we have studied a profit maximization problem for OSN providers. 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 developed several upper bounds on the maximum achievable profit and derived online approximation guarantees to measure the quality of the solution obtained by any algorithm.

Our proposed techniques such as iterative pruning and various upper bounds, are general and can be applied to any problem of the corresponding submodular optimization.

6.2 Limitations and Future Work

Many research issues can be further explored for viral marketing in OSNs. In what follows, we discuss the limitations in this dissertation and propose some future directions.

For all the applications discussed in this dissertation, we assume that the propagation probability $p_{u,v}$ is a parameter known a priori for each edge $\langle u, v \rangle$. In the experimental evaluations, we set $p_{u,v}$ to the reciprocal of v 's in-degree (the number of v 's in-neighbors). Although this is the commonly adopted setting in many studies, the propagation probability is usually unknown in reality. Intuitively, a bad setting of the propagation probability in which $p_{u,v}$ is far away from the true value would make the estimate of influence spread (or benefit and cost) rather inaccurate. As a result, the solution constructed via the inexact influence estimation may produce very low actual influence spread (or profit). Therefore, learning the propagation probability accurately is essentially important. Due to the similar reason, the benefit and cost of each node also need to be explored. One potential solution is to estimate these parameters via historical data. The challenges

Figure 6.1: Viral marketing in OSNs as a two-sided market.

here are two-fold: (i) how to get the accurate values of these parameters, and (ii) how to analyze the corresponding errors occurred when applying these estimated parameters in viral marketing of OSNs.

In Chapter 5, our study on the profit maximization problem for OSN providers has assumed that there is no limit on the number of seed nodes that can be chosen. Sometimes, the advertiser may have a limit on the number of products it is willing to give away for free, which in turn would restrict the number of users that the OSN provider can choose to “seed” the viral marketing. Hence, we can further study the profit maximization problem for OSN providers with a limited number of seed nodes. This problem is to maximize the difference between submodular functions subject to a cardinality constraint.

Another future direction is to study the viral marketing problems in a two-sided market [3, 97] involving three parties including product advertisers, OSN providers and end users (see Figure 6.1). Most of the existing studies have just considered the simplest scenario of this general framework, in which there are one advertiser and one OSN provider only. It will be much more challenging to study a complex system with multiple advertisers and multiple OSN providers. In such a system, a single user may participate in multiple OSNs, an advertiser may conduct viral marketing through several OSN providers, and an OSN provider may run viral marketing campaigns for similar products of multiple advertisers. Since each party has multiple participants and different participants may share the revenue, cost or profit, or have conflicting concerns, competition and cooperation emerge naturally where game theoretical analysis is likely to be useful.

In addition, we can implement the proposed solutions as a software toolkit and explore practical use of these techniques. The main functionality of the software toolkit is to compute the seed set for influence/profit maximization based on a given OSN graph, a diffusion model of influence propagation, a cost model of seed selection, and a benefit model of influence spread. The software toolkit shall implement the influence/profit maximization algorithms developed. Besides validating our techniques, the functionalities provided by the software toolkit can also be used by the research community to further study viral marketing services and test new quality optimization technologies.

Finally, more theoretical analyses can be conducted for the submodular maximization problems discussed in this dissertation. Specifically, in Chapter 4, we have applied a pruning approach to provide worst-case theoretical guarantees for unconstrained submodular maximization with a weak condition that $\phi(A^*) + \phi(B^*) \geq 0$ (see Corollary 4.1). However, we observe in our experiments that this condition may not be satisfied in a few cases. The theoretical guarantees may be further enhanced by either relaxing the condition requirement or deriving weaker guarantees. Moreover, it would be interesting if it can be demonstrated that for some general subclasses of submodular functions (e.g., curvature-based submodular [26, 59]), the proposed algorithms always give better guarantees than the best known.

Publications

1. **Jing Tang**, Xueyan Tang, Xiaokui Xiao, Junsong Yuan, “Online Processing Algorithms for Influence Maximization”, Accepted to appear in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD)*, 2018.
2. **Jing Tang**, Xueyan Tang, Junsong Yuan, “Traffic-Optimized Data Placement for Social Media”, Accepted to appear in *IEEE Transactions on Multimedia*.
3. **Jing Tang**, Xueyan Tang, Junsong Yuan, “Influence Maximization Meets Efficiency and Effectiveness: A Hop-Based Approach”, In *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 64–71, 2017. **Acceptance Rate: 17.2%** (33 out of 192).
4. **Jing Tang**, Xueyan Tang, Junsong Yuan, “Profit Maximization for Viral Marketing in Online Social Networks”, In *Proceedings of the 24th IEEE International Conference on Network Protocols (ICNP)*, pp. 1–10, 2016. **Acceptance Rate: 20.0%** (46 out of 229).
5. **Jing Tang**, Xueyan Tang, Junsong Yuan, “Optimizing Inter-Server Communication for Online Social Networks”, In *Proceedings of the 35th IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp. 215–224, 2015. **Acceptance Rate: 12.9%** (70 out of 543).
6. **Jing Tang**, Richard T. B. Ma, “Regulating Monopolistic ISPs without Neutrality”, In *Proceedings of the 22nd IEEE International Conference on Network Protocols (ICNP)*, pp. 374–384, 2014. **Acceptance Rate: 20.0%** (32 out of 160). **Best Paper Award.**

Submitted Papers

1. **Jing Tang**, Richard T. B. Ma, “Regulating Monopolistic ISPs without Neutrality”, Submitted to *INFORMS Information Systems Research*.
2. **Jing Tang**, Xueyan Tang, Junsong Yuan, “Profit Maximization for Viral Marketing in Online Social Networks: Algorithms and Analysis”, Submitted to *IEEE Transactions on Knowledge and Data Engineering (TKDE)*.
3. **Jing Tang**, Xueyan Tang, Junsong Yuan, “An Efficient and Effective Hop-Based Approach for Influence Maximization in Online Social Networks”, Fast tracked to *Social Network Analysis and Mining (SNAM) Journal*.
4. **Jing Tang**, Xueyan Tang, Junsong Yuan, “Towards Profit Maximization for Online Social Network Providers”, Submitted to *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2018.
5. **Jing Tang**, Xueyan Tang, Junsong Yuan, “Data-dependent Approximation Analysis for Large-scale Submodular Maximization”, *Technical Report*.

References

- [1] Z. Abbassi, A. Bhaskara, and V. Misra, “Optimizing display advertising in online social networks,” in *Proce. WWW*, 2015, pp. 1–11.
- [2] A. A. Ageev and M. I. Sviridenko, “An 0.828-approximation algorithm for the uncapacitated facility location problem,” *Discrete Applied Mathematics*, vol. 93, no. 4, pp. 149–156, 1999.
- [3] M. Armstrong, “Competition in two-sided markets,” *The RAND Journal of Economics*, vol. 37, no. 3, pp. 668–691, 2006.
- [4] A. Arora, S. Galhotra, and S. Ranu, “Debunking the myths of influence maximization: An in-depth benchmarking study,” in *Proc. ACM SIGMOD*, 2017, pp. 651–666.
- [5] C. Aslay, W. Lu, F. Bonchi, A. Goyal, and L. V. S. Lakshmanan, “Viral marketing meets social advertising: Ad allocation with minimum regret,” *Proc. VLDB Endowment*, vol. 8, no. 7, pp. 814–825, 2015.
- [6] N. Barbieri, F. Bonchi, and G. Manco, “Topic-aware social influence propagation models,” in *Proc. IEEE ICDM*, 2012, pp. 81–90.
- [7] X. Bei, N. Chen, L. Dou, X. Huang, and R. Qiang, “Trial and error in influential social networks,” in *Proc. ACM KDD*, 2013, pp. 1016–1024.
- [8] M. Blum, R. W. Floyd, V. Pratt, R. L. Rivest, and R. E. Tarjan, “Time bounds for selection,” *Journal of Computer and System Sciences*, vol. 7, no. 4, pp. 448–461, 1973.

REFERENCES

- [9] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, “Maximizing social influence in nearly optimal time,” in *Proc. SODA*, 2014, pp. 946–957.
- [10] Y. Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images,” in *Proc. IEEE ICCV*, vol. 1, 2001, pp. 105–112.
- [11] N. Buchbinder, M. Feldman, J. S. Naor, and R. Schwartz, “A tight linear time $(1/2)$ -approximation for unconstrained submodular maximization,” in *Proc. IEEE FOCS*, 2012, pp. 649–658.
- [12] N. Buchbinder, M. Feldman, J. S. Naor, and R. Schwartz, “Submodular maximization with cardinality constraints,” in *Proc. SODA*, 2014, pp. 1433–1452.
- [13] M. Cha, A. Mislove, and K. P. Gummadi, “A measurement-driven analysis of information propagation in the Flickr social network,” in *Proc. WWW*, 2009, pp. 721–730.
- [14] 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.
- [15] 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.
- [16] W. Chen, T. Lin, Z. Tan, M. Zhao, and X. Zhou, “Robust influence maximization,” in *Proc. ACM KDD*, 2016, pp. 795–804.
- [17] 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.
- [18] 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.
- [19] W. Chen, Y. Wang, and S. Yang, “Efficient influence maximization in social networks,” in *Proc. ACM KDD*, 2009, pp. 199–208.

REFERENCES

- [20] W. Chen, Y. Yuan, and L. Zhang, “Scalable influence maximization in social networks under the linear threshold model,” in *Proc. IEEE ICDM*, 2010, pp. 88–97.
- [21] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec, “Can cascades be predicted?” in *Proc. WWW*, 2014, pp. 925–936.
- [22] S. Cheng, H. Shen, J. Huang, W. Chen, and X. Cheng, “IMRank: Influence maximization via finding self-consistent ranking,” in *Proc. ACM SIGIR*, 2014, pp. 475–484.
- [23] S. Cheng, H. Shen, J. Huang, G. Zhang, and X. Cheng, “Staticgreedy: Solving the scalability-accuracy dilemma in influence maximization,” in *Proc. ACM CIKM*, 2013, pp. 509–518.
- [24] F. Chung and L. Lu, “Concentration inequalities and martingale inequalities: A survey,” *Internet Mathematics*, vol. 3, no. 1, pp. 79–127, 2006.
- [25] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck, “Sketch-based influence maximization and computation: Scaling up with guarantees,” in *Proc. ACM CIKM*, 2014, pp. 629–638.
- [26] M. Conforti and G. Cornuéjols, “Submodular set functions, matroids and the greedy algorithm: Tight worst-case bounds and some generalizations of the rado-edmonds theorem,” *Discrete Applied Mathematics*, vol. 7, no. 3, pp. 251–274, 1984.
- [27] G. Cordasco, L. Gargano, and A. A. Rescigno, “Influence propagation over large scale social networks,” in *Proc. IEEE/ACM ASONAM*, 2015, pp. 1531–1538.
- [28] G. Cornuejols, M. L. Fisher, and G. L. Nemhauser, “Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms,” *Management Science*, vol. 23, pp. 789–810, 1977.
- [29] A. Delong, O. Veksler, A. Osokin, and Y. Boykov, “Minimizing sparse high-order energies by submodular vertex-cover,” in *Proc. NIPS*, 2012, pp. 962–970.

REFERENCES

- [30] T. N. Dinh, H. Zhang, D. T. Nguyen, and M. T. Thai, “Cost-effective viral marketing for time-critical campaigns in large-scale social networks,” *IEEE/ACM Transactions on Networking*, vol. 22, no. 6, pp. 2001–2011, 2014.
- [31] P. Domingos and M. Richardson, “Mining the network value of customers,” in *Proc. ACM KDD*, 2001, pp. 57–66.
- [32] 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.
- [33] U. Feige, “A threshold of $\ln n$ for approximating set cover,” *Journal of the ACM*, vol. 45, no. 4, pp. 634–652, 1998.
- [34] U. Feige and M. Goemans, “Approximating the value of two power proof systems, with applications to MAX 2SAT and MAX DICUT,” in *Proc. IEEE ISTCS*, 1995, pp. 182–189.
- [35] U. Feige, V. S. Mirrokni, and J. Vondrák, “Maximizing non-monotone submodular functions,” in *Proc. IEEE FOCS*, 2007, pp. 461–471.
- [36] J. Feldman, M. Henzinger, N. Korula, V. S. Mirrokni, and C. Stein, “Online stochastic packing applied to display ad allocation,” in *Proc. ESA*, 2010, pp. 182–194.
- [37] J. Feldman, N. Korula, V. Mirrokni, S. Muthukrishnan, and M. Pál, “Online ad assignment with free disposal,” in *Proc. WINE*, 2009, pp. 374–385.
- [38] M. Feldman, J. S. Naor, and R. Schwartz, “Nonmonotone submodular maximization via a structural continuous greedy algorithm,” in *Proc. ICALP*, 2011, pp. 342–353.
- [39] M. Feldman, J. S. Naor, and R. Schwartz, “A unified continuous greedy algorithm for submodular maximization,” in *Proc. IEEE FOCS*, 2011, pp. 570–579.
- [40] S. Fujishige, *Submodular Functions and Optimization*. Elsevier Science, 2005, vol. 58.

REFERENCES

- [41] S. Galhotra, A. Arora, and S. Roy, “Holistic influence maximization: Combining scalability and efficiency with opinion-aware models,” in *Proc. SIGMOD*, 2016, pp. 743–758.
- [42] S. Galhotra, A. Arora, S. Virinchi, and S. Roy, “Asim: A scalable algorithm for influence maximization under the independent cascade model,” in *Proc. WWW Companion*, 2015, pp. 35–36.
- [43] S. O. Gharan and J. Vondrák, “Submodular maximization by simulated annealing,” in *Proc. SODA*, 2011, pp. 1098–1116.
- [44] S. Goel, D. J. Watts, and D. G. Goldstein, “The structure of online diffusion networks,” in *Proc. ACM EC*, 2012, pp. 623–638.
- [45] M. X. Goemans and D. P. Williamson, “Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming,” *Journal of the ACM*, vol. 42, no. 6, pp. 1115–1145, 1995.
- [46] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, “Inferring networks of diffusion and influence,” in *Proc. ACM KDD*, 2010, pp. 1019–1028.
- [47] M. Gomez-Rodriguez, J. Leskovec, and A. Krause, “Inferring networks of diffusion and influence,” *ACM Transactions on Knowledge Discovery from Data*, vol. 5, no. 4, pp. 21:1–21:37, 2012.
- [48] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan, “Learning influence probabilities in social networks,” in *Proc. ACM WSDM*, 2010, pp. 241–250.
- [49] 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.
- [50] A. Goyal, F. Bonchi, L. V. S. Lakshmanan, and S. Venkatasubramanian, “On minimizing budget and time in influence propagation over social networks,” *Social Network Analysis and Mining*, vol. 3, no. 2, pp. 179–192, 2013.

REFERENCES

- [51] A. Goyal, W. Lu, and L. V. S. Lakshmanan, “Celf++: Optimizing the greedy algorithm for influence maximization in social networks,” in *Proc. WWW Companion*, 2011, pp. 47–48.
- [52] A. Goyal, W. Lu, and L. V. S. Lakshmanan, “Simpath: An efficient algorithm for influence maximization under the linear threshold model,” in *Proc. IEEE ICDM*, 2011, pp. 211–220.
- [53] J. Hartline, V. Mirrokni, and M. Sundararajan, “Optimal marketing strategies over social networks,” in *Proc. WWW*, 2008, pp. 189–198.
- [54] J. Håstad, “Some optimal inapproximability results,” *Journal of the ACM*, vol. 48, no. 4, pp. 798–859, 2001.
- [55] X. He and D. Kempe, “Stability of influence maximization,” in *Proc. ACM KDD*, 2014, pp. 1256–1265.
- [56] K. Huang, S. Wang, G. Bevilacqua, X. Xiao, and L. V. S. Lakshmanan, “Revisiting the stop-and-stare algorithms for influence maximization,” *Proc. VLDB Endowment*, vol. 10, no. 9, pp. 913–924, 2017.
- [57] R. Iyer and J. Bilmes, “Algorithms for approximate minimization of the difference between submodular functions, with applications,” in *Proc. UAI*, 2012, pp. 407–417.
- [58] R. Iyer and J. Bilmes, “Submodular optimization with submodular cover and submodular knapsack constraints,” in *Proc. NIPS*, 2013, pp. 2436–2444.
- [59] R. Iyer, S. Jegelka, and J. Bilmes, “Curvature and optimal algorithms for learning and minimizing submodular functions,” in *Proc. NIPS*, 2013, pp. 2742–2750.
- [60] R. Iyer, S. Jegelka, and J. Bilmes, “Fast semidifferential-based submodular function optimization,” in *Proc. ICML*, 2013, pp. 855–863.
- [61] S. Jegelka and J. Bilmes, “Submodularity beyond submodular energies: Coupling edges in graph cuts,” in *Proc. IEEE CVPR*, 2011, pp. 1897–1904.

REFERENCES

- [62] K. Jung, W. Heo, and W. Chen, “IRIE: Scalable and robust influence maximization in social networks,” in *Proc. IEEE ICDM*, 2012, pp. 918–923.
- [63] D. Kempe, J. Kleinberg, and E. Tardos, “Maximizing the spread of influence through a social network,” in *Proc. ACM KDD*, 2003, pp. 137–146.
- [64] A. Khan, B. Zehnder, and D. Kossmann, “Revenue maximization by viral marketing: A social network host’s perspective,” in *Proc. IEEE ICDE*, 2016.
- [65] S. Khuller, A. Moss, and J. Naor, “The budgeted maximum coverage problem,” *Information Processing Letters*, vol. 70, no. 1, pp. 39–45, 1999.
- [66] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [67] A. Krause and C. Guestrin, “Near-optimal observation selection using submodular functions,” in *Proc. AAAI*, 2007, pp. 1650–1654.
- [68] A. Krause, J. Leskovec, C. Guestrin, J. Vanbriesen, and C. Faloutsos, “Efficient sensor placement optimization for securing large water distribution networks,” *Journal of Water Resources Planning and Management*, vol. 134, no. 6, pp. 516–526, 2008.
- [69] A. Krause, A. Singh, and C. Guestrin, “Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies,” *Journal of Machine Learning Research*, vol. 9, no. 3, pp. 235–284, 2008.
- [70] K. Kutzkov, A. Bifet, F. Bonchi, and A. Gionis, “STRIP: Stream learning of influence probabilities,” in *Proc. ACM KDD*, 2013, pp. 275–283.
- [71] J. Lee, V. S. Mirrokni, V. Nagarajan, and M. Sviridenko, “Maximizing nonmonotone submodular functions under matroid or knapsack constraints,” *SIAM Journal on Discrete Mathematics*, vol. 23, no. 4, pp. 2053–2078, 2010.
- [72] J. R. Lee and C. W. Chung, “A fast approximation for influence maximization in large social networks,” in *WWW Companion*, 2014, pp. 1157–1162.

REFERENCES

- [73] W. Lee, J. Kim, and H. Yu, “CT-IC: Continuously activated and time-restricted independent cascade model for viral marketing,” in *Proc. IEEE ICDM*, 2013, pp. 57–68.
- [74] S. Lei, S. Maniu, L. Mo, R. Cheng, and P. Senellart, “Online influence maximization,” in *Proc. ACM KDD*, 2015, pp. 645–654.
- [75] J. Leskovec, L. A. Adamic, and B. A. Huberman, “The dynamics of viral marketing,” *ACM Transactions on the Web*, vol. 1, no. 1, pp. 5:1–5:39, 2007.
- [76] 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.
- [77] J. Leskovec and A. Krevl, “SNAP Datasets: Stanford large network dataset collection,” <http://snap.stanford.edu/data>, 2014.
- [78] Y. Li, B. Q. Zhao, and J. C. S. Lui, “On modeling product advertisement in large-scale online social networks,” *IEEE/ACM Transactions on Networking*, vol. 20, no. 5, pp. 1412–1425, 2012.
- [79] H. Lin and J. Bilmes, “Multi-document summarization via budgeted maximization of submodular functions,” in *Proc. NAACL-HLT*, 2010, pp. 912–920.
- [80] H. Lin and J. Bilmes, “A class of submodular functions for document summarization,” in *Proc. HLT*, 2011, pp. 510–520.
- [81] B. Liu, G. Cong, D. Xu, and Y. Zeng, “Time constrained influence maximization in social networks,” in *Proc. IEEE ICDM*, 2012, pp. 439–448.
- [82] B. Liu, G. Cong, Y. Zeng, D. Xu, and Y. M. Chee, “Influence spreading path and its application to the time constrained social influence maximization problem and beyond,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1904–1917, 2014.
- [83] C. Long and R. C.-W. Wong, “Minimizing seed set for viral marketing,” in *Proc. IEEE ICDM*, 2011, pp. 427–436.

REFERENCES

- [84] W. Lu, F. Bonchi, A. Goyal, and L. V. S. Lakshmanan, “The bang for the buck: Fair competitive viral marketing from the host perspective,” in *Proc. ACM KDD*, 2013, pp. 928–936.
- [85] W. Lu and L. V. S. Lakshmanan, “Profit maximization over social networks,” in *Proc. IEEE ICDM*, 2012, pp. 479–488.
- [86] A. Meyer, “Viral video marketing: What’s, why’s & how’s of going viral,” <http://www.marketgizmo.com/viral-video-marketing-cats-babies-and-your-company/>, 2015.
- [87] M. Narasimhan and J. Bilmes, “A submodular-supermodular procedure with applications to discriminative structure learning,” in *Proc. UAI*, 2005, pp. 404–412.
- [88] G. L. Nemhauser and L. A. Wolsey, “Best algorithms for approximating the maximum of a submodular set function,” *Mathematics of Operations Research*, vol. 3, no. 3, pp. 177–188, 1978.
- [89] 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.
- [90] H. T. Nguyen, T. N. Dinh, and M. T. Thai, “Cost-aware targeted viral marketing in billion-scale networks,” in *Proc. IEEE INFOCOM*, 2016.
- [91] 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.
- [92] Nielsen, “State of the media: The social media report 2012,” Dec. 2012. [Online]. Available: <http://www.nielsen.com/us/en/insights/reports/2012/state-of-the-media-the-social-media-report-2012.html>
- [93] 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.

REFERENCES

- [94] J. B. Orlin, “A faster strongly polynomial time algorithm for submodular function minimization,” *Mathematical Programming*, vol. 118, no. 2, pp. 237–251, 2009.
- [95] M. Richardson and P. Domingos, “Mining knowledge-sharing sites for viral marketing,” in *Proc. ACM KDD*, 2002, pp. 61–70.
- [96] J. J. Roberts, “Facebook and Google are big winners as political ad money moves online,” *Fortune*, 2016.
- [97] J.-C. Rochet and J. Tirole, “Platform competition in two-sided markets,” *Journal of the European Economic Association*, vol. 1, no. 4, pp. 990–1029, 2003.
- [98] M. G. Rodriguez, D. Balduzzi, and B. Schölkopf, “Uncovering the temporal dynamics of diffusion networks,” in *Proc. ICML*, 2011, pp. 561–568.
- [99] A. Schrijver, “A combinatorial algorithm minimizing submodular functions in strongly polynomial time,” *Journal of Combinatorial Theory Series B*, vol. 80, no. 2, pp. 346–355, 2000.
- [100] D. Sheldon, B. Dilkina, A. Elmachoub, R. Finseth, A. Sabharwal, J. Conrad, C. Gomes, D. Shmoys, W. Allen, O. Amundsen, and W. Vaughan, “Maximizing the spread of cascades using network design,” in *Proc. UAI*, 2010, pp. 517–526.
- [101] 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.
- [102] Statista, “Most famous social network sites worldwide as of April 2017,” April 2017. [Online]. Available: <https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/>
- [103] J. Tang, X. Tang, and J. Yuan, “Profit maximization for viral marketing in online social networks,” in *Proc. IEEE ICNP*, 2016.
- [104] 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.

REFERENCES

- [105] Y. Tang, Y. Shi, and X. Xiao, “Influence maximization in near-linear time: A martingale approach,” in *Proc. ACM SIGMOD*, 2015, pp. 1539–1554.
- [106] Y. Tang, X. Xiao, and Y. Shi, “Influence maximization: Near-optimal time complexity meets practical efficiency,” in *Proc. ACM SIGMOD*, 2014, pp. 75–86.
- [107] J. Vondrák, “Symmetry and approximability of submodular maximization problems,” *SIAM Journal on Computing*, vol. 42, no. 1, pp. 265–304, 2013.
- [108] A. Wald, *Sequential Analysis*. Wiley, 1947.
- [109] Y. Wang, G. Cong, G. Song, and K. Xie, “Community-based greedy algorithm for mining top-k influential nodes in mobile social networks,” in *Proc. ACM KDD*, 2010, pp. 1039–1048.
- [110] K. Wei, R. Iyer, and J. Bilmes, “Submodularity in data subset selection and active learning,” in *Proc. ICML*, 2015, pp. 1954–1963.
- [111] D. N. Yang, H. J. Hung, W. C. Lee, and W. Chen, “Maximizing acceptance probability for active friending in online social networks,” in *Proc. ACM KDD*, 2013, pp. 713–721.
- [112] A. L. Yuille and A. Rangarajan, “The concave-convex procedure (cccp),” in *Proc. NIPS*, 2002, pp. 1033–1040.
- [113] C. Zhang, J. Sun, and K. Wang, “Information propagation in microblog networks,” in *Proc. IEEE/ACM ASONAM*, 2013, pp. 190–196.
- [114] P. Zhang, W. Chen, X. Sun, Y. Wang, and J. Zhang, “Minimizing seed set selection with probabilistic coverage guarantee in a social network,” in *Proc. KDD*, 2014, pp. 1306–1315.
- [115] C. Zhou, P. Zhang, J. Guo, and L. Guo, “An upper bound based greedy algorithm for mining top-k influential nodes in social networks,” in *Proc. WWW Companion*, 2014, pp. 421–422.

REFERENCES

- [116] 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.
- [117] 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.