Welfare maximization for complementary and competing items and their applications

Analysis and algorithm using a utility driven model for achieving better social influence

by

Prithu Banerjee

B.Tech., West Bengal University of Technology, 2010
M.Tech., Indian Institute of Technology, Guwahati, 2012

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate and Postdoctoral Studies

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

April 2022

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Welfare maximization for complementary and competing items and their applications: analysis and algorithm using a utility driven model for achieving better social influence

submitted by **Prithu Banerjee** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy** in **Computer Science**.

Examining Committee:

Laks V.S. Lakshmanan, Professor, Computer Science, UBC
Supervisor

Rachel Pottinger, Professor, Computer Science, UBC
Supervisory Committee Member

Mark Schmidt, Associate Professor, Computer Science, UBC
Supervisory Committee Member

Anne Condon, Professor, Computer Science, UBC
University Examiner

Jane Z. Wang, Professor, Electrical and Computer Engineering, UBC
University Examiner

Ambuj K. Singh, Professor, Computer Science, University of California, Santa Barbara
External Examiner

Abstract

Motivated by applications such as viral marketing, the problem of influence maximization (IM) has been extensively studied in the literature. The goal is to select a small number of users to adopt an item such that it results in a large cascade of adoptions by others. Existing works have three key limitations. (1) They do not account for the economic considerations of a user in buying/adopting items. (2) They cannot model the complex interactions between multiple items. (3) For the network owner, maximizing social welfare is important to ensure customer loyalty, which is not addressed in prior work in the IM literature. In this work, we address all three limitations and propose a novel model called Utility driven Independent Cascade (UIC) that combines utility-driven item adoption with influence propagation over networks. We focus on several types of items such as mutually complementary only, competing only, and a mix of the two in the context of the filter bubble problem. We formulate the problem of social welfare maximization under each of these settings. We show that while the objective function is neither submodular nor supermodular, a constant or instance-dependent approximation can still be achieved. With comprehensive experiments on real and synthetic datasets, we demonstrate that our algorithms significantly outperform all baselines on large real social networks.

Lay Summary

Currently, there is a significant interest in the problem of influence maximization. In economics, it is well accepted that the adoption of items is governed by the utility that a user derives from their adoption. In this work, we propose a model that combines utility-driven item adoption with the viral network effect helping to propagate adoption of and desire for items from users to their peers. We study the model for influence maximization and filter bubble problem.

Preface

This thesis is a product of a continuous research collaboration with Dr. Wei Chen from Microsoft Research, and my supervisor Prof. Laks V.S. Lakshmanan. The chapters are based on papers that are either published, or are under review.

- Chapter 2 is based on the research paper published in *The ACM Special Interest Group on Management of Data (SIGMOD), 2019* [12]. For this work I have developed the propagation model, analyzed its properties, proposed the greedy algorithm, conducted the experiments and wrote the paper. Wei Chen developed the block accounting using which the approximation ratio of the algorithm was established. Laks V.S. Lakshmanan proposed the key idea using which the prefix-preserving seed selection algorithm was developed. He also helped significantly in terms of writing the paper. Dr. Wei Lu and Ritika Jain, both were graduate students of Laks V.S. Lakshmanan, and provided valuable feedback during the early stages of this research.

- The second chapter, Chapter 3, is based on a research paper published in *International Conference on Very Large Data Bases (VLDB), 2020* [13]. In this paper I developed the propagation model and proved the key properties. Wei Chen and Laks V.S. Lakshmanan provided valuable opinion to establish the inapproximation result. My lab-mate Glenn S. Bevilacqua provided useful feedback during the analysis. I designed the algorithms and Wei Chen helped proving the guarantees. Laks V.S. Lakshmanan also helped with the writing of this paper. Experiments were conducted by me, however, anonymous reviewer brought the utility choice paper in our attention [14]. The real parameters were learnt using the method proposed in paper [14] for some of the experiments.

- The work of the last chapter, Chapter 4 is currently under submission. Laks V.S. Lakshmanan proposed the filter bubble problem, I then formulated the model and studied its properties. I developed the

algorithms and Wei Chen helped to analyze the performance of the algorithms. I also designed the experiments and wrote the paper with ample feedback from Laks V.S. Lakshmanan and Wei Chen.

Table of Contents

List of Tables

List of Figures

Acknowledgements

First of all, I would like to acknowledge how incredibly lucky I have been as a person, both before and during my graduate studies. I was surrounded by amazing colleagues and mentors who always provided the right help and direction that I needed to make this thesis possible. I am highly indebted to my supervisor Prof. Laks V.S. Lakshmanan for his relentless support and supervision throughout my Ph.D. journey. Needless to say that his unparalleled research ethics and passion are the cornerstones of this work.

I would also like to express my deepest gratitude to Dr. Wei Chen. It was a privilege to learn from his technical acumen which has helped my research on countless occasions. I thank him for being patient with my questions and for showing me how to truly enjoy research, and I hope that the collaboration would continue even after the completion of my graduate studies.

I sincerely thank my supervisory committee members - Prof. Rachel Pottinger and Prof. Mark Schmidt for their generous time and valuable feedback on the dissertation. I would also like to thank Prof. Hu Fu and Prof. Will Evans for serving on the committee during my research proficiency evaluation (RPE). I am also expressing my gratitude to Prof. Anne Condon, Prof. Jane Z. Wang and Prof. Ambuj K. Singh for examining my dissertation and providing very helpful comments. Prof. Jim Little chaired my proposal defense and Prof. David Michelson chaired the final defense; I am thankful for their help and time.

I am thankful to all the co-authors for the other research projects I worked on - Dr. Sayan Ranu, Dr. Lingyang Chu, Yu Tang, and Dr. Mostafa Khaghani Milani. All these projects were a great learning experience for me that indirectly helped in shaping up the research for this thesis. I am also grateful to everyone who let me intern and works with them: Amazon Research, for giving me the valuable experience of working in the team of, Dr. Vineet Chaoji and Pooja A.; Huawei Research, for the excellent opportunity in the team of, Dr. Yong Zhang and Dr. Lanjun Wang. A grateful thank you to all the staff and particularly Joyce Poon, and Lara Hall for all their help in dealing with the administrative issues.

The acknowledgments would not be complete without mentioning all of my friends and family members for making this journey joyful. Special thanks to my DMM lab buddies and friends in Vancouver for making my Vancouver life such amazing and for all the support whenever I needed something. Most importantly, I am greatly indebted to my parents, my sister, and my wife. Their unconditional love and care mean everything to me.

To those who were more deserving

Chapter 1

Introduction

The reach of online social networking websites such as Facebook, Twitter, etc. has rapidly increased over the past decade. Fueled by this growth, the study of the *social influence propagation* has received immense interest from several communities within computer science such as data mining [38, 83, 110, 126], machine learning [134, 135], theoretical computer science [24, 46, 102], algorithmic game theory [49, 69, 122], etc.

The study of social influence involves understanding how an individual's belief, opinion, or behavior impacts other individuals via social interactions [120]. Sociologists and psychologists have been studying the problem of characterizing social influence for several decades [11, 15, 26, 48, 54, 87, 88]. However, the advent of social networks has reshaped the dynamics of influence, and has led to the study of computational social influence. As individuals are getting more connected in these social platforms, their actions such as sharing a news article or tweeting about a product are frequently triggering a viral effect over the entire network [71]. There are several studies confirming the role of this effect in the context of marketing [73, 119], elections [21, 44], news propagation [7, 136], etc. Therefore, the study of computational social influence on a social network, has drawn a lot of interest. A fundamental algorithmic problem that has garnered significant attention in the context of computational social influence, is the *influence maximization* (IM) problem. This thesis identifies and addresses the shortcomings of the existing works on the IM problem, particularly in the context of multi-item propagation where complex relationships among the items play a key role. In what follows, an overview of the classical influence maximization problem involving single item propagation is given in Section 1.1; the basic constructs introduced in this section, will be used in the later chapters of the thesis. Then in Section 1.2 a review of a body of research is presented. These works have extended the study of IM for multiple items. Lastly, Section 1.3 highlights some fundamental shortcomings of the existing research works. Then the section summarizes the way this thesis addresses the existing shortcomings, and how that enables the study of some interesting applications under the IM paradigm.

1.1 Overview of the classical influence maximization problem

Influence maximization (IM) problem in social network was first introduced by Kempe et al [83], where the goal is to select influential users in the network such that if a campaign is started from those influential users, maximum number of other users can be influenced. Formally speaking, the inputs are the following:

1. A probabilistic graph $G = (V, E, p)$, where a node $v \in V$ represents an individual, edge $(u, v) \in E$ represents the existence of a connection between individuals, and the probability of an edge $p(u, v) \rightarrow [0, 1]$ denotes the edge weight.

2. A positive integer $k < |V|$ as budget.

3. A diffusion model M that describes the random process of how influence propagates from a set of nodes of initial adopters to other nodes in G.

Given the input, the goal of IM is to *find a set of nodes S such that $|S| \leq k$, and by targeting nodes in S as initial adopters, following the diffusion model M, the expected number of activated nodes in the entire graph G is maximized.*

The set S is called the *seed set* in IM literature. The *influence spread function* is denoted as $\sigma\colon 2^V \rightarrow \mathbb{R}_{\geq 0}$, where $\sigma(S)$ denotes the expected number of activated nodes at the end of the propagation, called the *influence spread*.

A diffusion model governs how the influence propagates from one node to another, in other words, how nodes get *activated*. Various diffusion models have been proposed in the literature, but the two classic and fundamental ones are the *independent cascade* (IC) model [62] and the *linear threshold* (LT) model [67]. The nodes can be either active or inactive under these models. Models are progressive, meaning once an inactive node becomes active, it stays so till the end of the propagation. This thesis primarily focuses on the IC model, hence the model is described with an example next. Readers are referred to [35] for more details of these models and their generalizations.

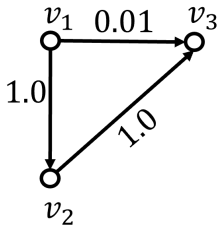


Figure 1.1: Graph of the example propagation shown in Figure 1.2.

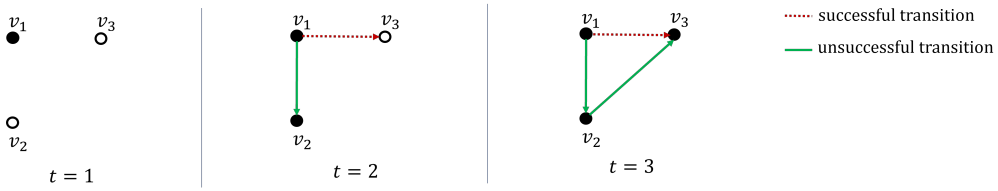


Figure 1.2: Illustrating propagation of influence under IC

1.1.1 Independent cascade (IC) model

IC model was first introduced in [83] and shares close resemblances with propagation models used in marketing research [62, 63] and epidemic modeling [6]. In more recent studies [77] on the so-called patient-0 problem, where the goal is to find the source of an infectious disease, propagation models similar to IC have been found to be useful to model the disease spread. The propagation under IC is described next.

In the beginning, all nodes are inactive. Propagation proceeds at discrete time-steps starting from time $t = 1$ when the only seed nodes become active. At any time $t > 1$, every node u that became active at $t - 1$ makes one attempt to activate its inactive out-neighbor v, $v \in N^{out}(u)$, in other words, node u tests if the edge (u, v) is "live" or "blocked". The attempt succeeds (the edge (u, v) is live) with probability $p_{uv} := p(u, v)$. The propagation process ends when there is no additional node that can be activated.

A sample diffusion under IC is illustrated using an example next.

Example 1. A graph G with edge probabilities is shown in Figure 1.1. Suppose that the budget $k = 1$ and node v_1 is selected as the only seed node. The diffusion from v_1 is shown in Figure 1.2. At a given timestep t, a black ring denotes a node that is not activated by time t; if a node is activated node, then it is shown in solid black. At time $t = 1$, only the seed node v_1 is activated. Then at $t = 2$, outgoing edges of v_1 are tested: edge

(v_1, v_3) fails (shown as red dotted line), but edge (v_1, v_2) succeeds (green solid line). Consequently v_2 is activated at time $t = 2$. Next at $t = 3$, v_2's outgoing edge (v_2, v_3) is tested. As it succeeds, v_3 is activated. Since there is no outgoing edge from v_3, no more node can be activated, therefore the propagation ends.

Also note that, v_1 is in fact the best seed that can be selected under the budget constraint $k = 1$. $\sigma(S) = 3$, when $S = \{v_1\}$; whereas for $S = \{v_2\}$, $\sigma(S) = 2$, and for $S = \{v_3\}$, $\sigma(S) = 1$. □

Next some key properties of the spread function $\sigma(.)$ are presented. These properties help design effective algorithms for the IM problem.

1.1.2 Key properties of the objective

Using a reduction from the set cover problem [82], the problem of IM is shown to be NP-hard under IC [83]. Further, it was also shown in a later work [38], using a reduction from the counting problem of s-t connectedness in a directed graph [133], that it is #P-hard to even compute $\sigma(S)$ for a given S. However, approximation algorithms are still designed leveraging some properties of the set function $\sigma(S)$.

Algorithm 1: Greedy seed selection(G, k)

```
S = ∅
for i = 1 to k do
    v = arg max_{u∈V\S} [σ(S ∪ {v}) − σ(S)]
    S = S ∪ {v}
Return S
```

A few important properties of a set function are often considered while designing IM solutions. A set function $f : 2^V \to \mathbb{R}$ is *monotone* if $f(S) \leq f(T)$ whenever $S \subseteq T \subseteq V$; *submodular* if for any $S \subseteq T \subseteq V$ and any $x \in V \setminus T$, $f(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T)$; f is *supermodular* if $-f$ is submodular; and f is *modular* if it is both submodular and supermodular.

$\sigma(S)$ is monotone and submodular with respect to S under IC. In fact $\sigma(\cdot)$ is monotone and submodular under a general propagation model called *general threshold model* that subsumes IC as a special case [102].

Nemhauser et al. in [107] showed that using a simple Greedy hill-climbing algorithm, maximizing a monotone and submodular function subject to a cardinality constraint can be approximated with a factor of $1 - \frac{1}{e}$. Therefore, the same $1 - \frac{1}{e}$ approximation can be achieved for the IM problem

by selecting the seed set S in the following way: Initialize $S = \emptyset$. Choose the k seeds one by one by repeatedly picking a seed that offers the maximal marginal gain w.r.t. the spread function. The pseudo-code is shown in Algorithm 1.

Although Greedy enables the constant approximation, it cannot circumvent the #P-hardness of computing $\sigma(S)$; this computation is required in Line 3. Use of Monte-Carlo (MC) simulation was proposed to address this, which causes the approximation to drop to $1-\frac{1}{e}-\epsilon$, for any $\epsilon > 0$, where the exact value of ϵ is a function of the number of MC iterations. These repeated MC simulations pose a great challenge in terms of efficiency. In particular, it takes weeks to select even a mere 50 seeds for graphs containing only a few thousand nodes [35]. Next, the contributions of some seminal works on IM for addressing the efficiency challenge are presented.

1.1.3 Scalable influence maximization

Leskovec et al. [89] proposed a cost-effective lazy forward method, called CELF, that improved the running time of Greedy by a factor of 700. Goyal et al [65] devised $CELF++$ which improved the efficiency by another 61%. However, all these techniques were still variants of MC simulations that could not scale for real networks having billion-sized node sets [22].

Borg et al. [22] introduced the concept of Reverse Reachable (RR) sets to estimate the influence spread. Tang et al. [128] proposed Two-phase Influence Maximization algorithm (TIM) which is a randomized algorithm using RR sets. Later the same authors improved the sampling efficiency of TIM using a martingale approach and developed an algorithm Martingale based Influence Maximization (IMM) [127]. IMM is shown to be orders of magnitude faster than its predecessors. Nguyen et al. parallelly developed a tighter sampling approach, called SSA, to sample RR sets [110], their algorithm outperforms IMM, especially for the initial few rounds of seed selection [74].

Several effective heuristics are also proposed to solve the IM problem efficiently. Chen et al. [39] proposed Maximum Influence Arborescence (MIA) for IC propagation model [37]. Local Directed Acyclic Graph (LDAG) [40] and SimPath [66] algorithms were introduced for LT propagation model. Galhotra et al. [55] introduced their OSIM heuristic for opinion aware influence propagation.

All these works focused on single-item propagation in the network, whereas this thesis focuses on multi-item propagation that is more prevalent in real social networks. In the next section multi-item IM works are discussed.

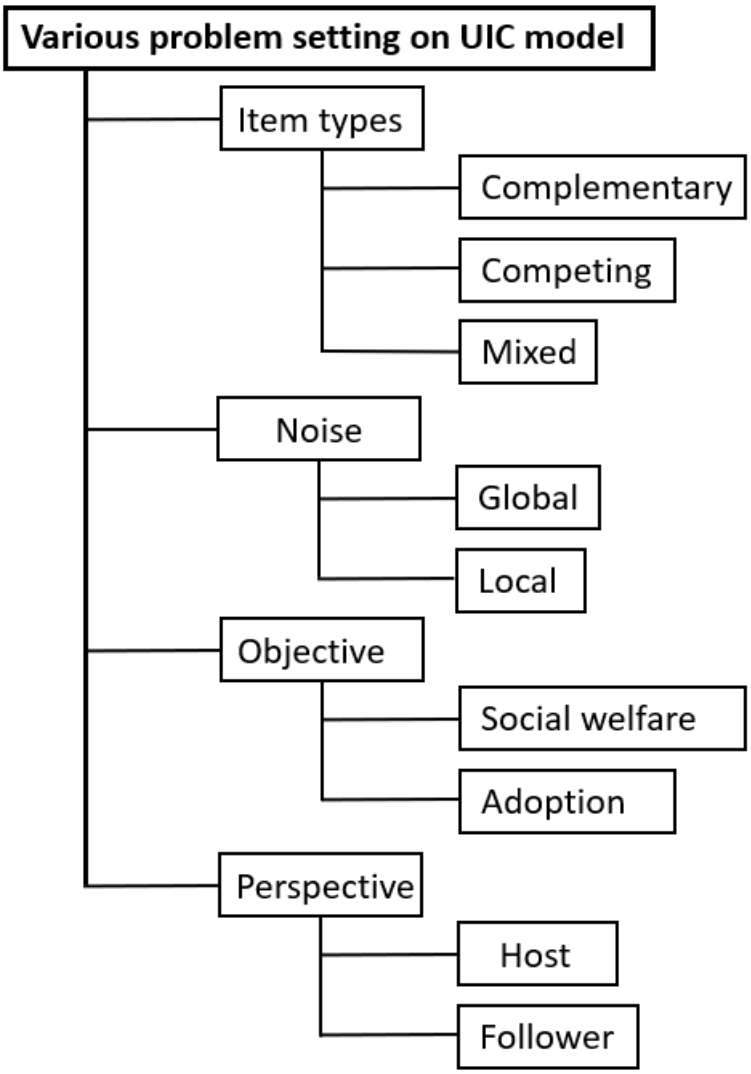


Figure 1.3: Taxonomy of different configurations that can be considered under the UIC model

1.2 Multi-item influence maximization

While the classical IM problem looked at the propagation of one item, several works studied the problem of influence maximization when multiple items are propagating. Two early works [47, 106] studied multi-item IM for non-competing items. Datta et al. [47] studied IM where under the assumption that items propagate independently, and devised a 1/3-approximation algorithm. In [106], Narayanam et al. proposed a model for multiple items where items are partitioned into two sets; an item can be adopted by a node only when it has already adopted a corresponding product in the other set. Thus between the two sets, items are complementary.

In competitive influence maximization, nodes are allowed to adopt at most one item from the set of multiple items that are being propagated [17, 23, 25, 28, 34, 70, 90, 96, 116]. Many of these works focused exclusively on the "follower's perspective" [17, 25, 28, 70], where given competitor's seed placement, the aim is to select seeds that maximize the spread of a specific item, corresponding to the follower's item, oftentimes at the cost of minimizing the competitor's spread. Later Lu et al. [96] proposed a method to maximize the total influence spread of all campaigners from a network host perspective while ensuring fair allocation of the seeds among them.

Multi-item propagation for a mix of competing and complementary items was first studied in [97]. Garimella et al. [57] introduced the problem of balanced exposure for two items, to break the filter-bubble effect using IM. Filter bubbles are observed when users of social networks are confined to only a specific type of items (or opinions). Matakos et al. [98] extended it for multiple items where both items and nodes (users) are known to have a leaning score on a polarity spectrum of items/opinions. In [132], propagation of two items is studied, where the goal is to ensure a balanced exposure of both the items to the maximum number of users in order to reduce the polarity in the network.

Despite these advancements, there are some fundamental limitations with all of these existing works. The next section highlights those limitations and provides an overview of how this thesis aims to address those.

1.3 Challenges and contributions

Even after a substantial amount of research, there are several key limitations, that are not addressed in these prior studies. Those limitations are presented here using a running example.

Example 2. Consider three items $iPhonePro$, $iPhoneMini$, and $AirPod$, that are in propagation. Notice that for a typical user owning $iPhonePro$ or $iPhoneMini$ individually is useful. However, $AirPod$ cannot be used without owning $iPhonePro$ or $iPhoneMini$ first. Further, although $iPhonePro$ and $iPhoneMini$ are competing products of the same type (which is, phone), for a user owning both can be useful to reap the advantages of the two different sizes that the two phones offer. Therefore the competition is not "pure" competition, rather partial competition. □

The limitations of the existing works in modeling propagation, involving item of different types, are listed below.

Firstly, in multi-item propagation, items exhibit a varying degree of competing or complementary relationship among them [14]. Earlier works have restricted this choice: for example, in the context of competing items, earlier studies focused primarily on *pure* competition, where each node can adopt at most one item. On the contrary, items may be involved in more complex relationships. As shown in Example 2, $iPhonePro$ and $AirPod$ exhibit a complementary relationship. Additionally, while $iPhonePro$ and $iPhoneMini$ do exhibit competition, it is not necessarily pure competition. Accommodating these arbitrary relationships in items is, therefore, necessary to correctly model how items are co-adopted and propagated in the network.

Secondly, in most of the propagation models it is assumed that whenever v is a seed node or node v is influenced by a neighboring node u on a certain item, node v will certainly adopt the item. However, this is far from reality. Suppose that a node v has not adopted $iPhonePro$ or $iPhoneMini$. If v is influenced by u with $AirPod$, then clearly v will not adopt $AirPod$ as $AirPod$ by itself has no use without owning $iPhonePro$ or $iPhoneMini$ first. Even in the case when v has adopted say $iPhonePro$ and is later influenced with $iPhoneMini$, then she may or may not adopt $iPhoneMini$ depending on the value proposition of owning the two phones. In fact, it has been noted in economics [104] that users take a stochastic decision regarding adoption after being informed about an item, instead of always deterministically adopting it. The only propagation model that takes this into account, is the work of Lu et al. [97]. In their model, a node-level automaton (NLA), governed by transition probabilities, is used to decide whether a non-seed user will adopt an item given the user's earlier adoptions. However, their main study was limited to the diffusion of two items, and their general parameter settings could lead to anomalies such as one item complementing a second item but the second one competing with the first

one or being indifferent to it. Therefore a more well-founded formalism is needed to model how users adopt items after being influenced.

As a consequence of focusing solely on maximizing item adoptions, the earlier work neglects the users' satisfaction. Recent studies, however, have shown users' overall satisfaction from using the service is of utmost importance for user retention. Therefore in this thesis a different maximization objective is studied, called welfare maximization. After introducing the concept of social welfare, it will be shown with an example (Example 3), that maximizing item adoption is not the same as welfare maximization that indeed focuses on users' satisfaction. A related observation on most studies [95, 96, 116, 138] on competition is that they study the objective of maximizing the influence of one item given all other items, or minimizing the influence of existing items, and do not consider maximizing the overall experience caused by all item adoptions.

1.3.1 Contributions of this thesis

The economics literature has extensively studied how users decide to adopt items [105, 113]. It is noted in those studies that item adoption by a user is driven by the *utility* that the user can derive from the item (or itemset). The utility of a user for an item(set) is computed as the difference between the *valuation* that the user has for the item(set) and the *price* she pays. In the field of combinatorial auctions, several studies [52, 81, 84] have focused on the problem of finding an optimal allocation of items to users that maximizes the sum of users' utilities, or social welfare, where users' valuations are provided as input. However, note that the economics literature does not consider the effect of influence propagation in networks. As stated earlier, in a social network, users desire items because of the influence from their in-neighbors who already adopted items, and then these users may, in turn, adopt the items if they could obtain positive utility from the adoption. Consequently, their out-neighbors get influenced by these items. Considering such network propagation is important for the application of viral marketing [83] which this work focuses on.

This thesis takes the first step to combine viral marketing (influence maximization) with a framework of item adoption grounded in the economic principle of item utility. A novel and powerful framework is proposed for capturing the interaction between these two paradigms, called *utility driven independent cascade (UIC)* model. In the UIC model, after being influenced following an IC model, nodes adopt items based on items' utilities. The utility of an item (or itemset) is defined to be the valuation minus the price.

Our formalism also enables the study of *social welfare* in this context, which is the sum of utilities of itemsets adopted by users at the end of a campaign, in expectation. Different from maximizing item adoption, maximizing social welfare results in higher user utilities and user satisfaction. Therefore this welfare maximization objective not only helps the users but also benefits the network host as more satisfied users are more likely to continue using the network, which in turn helps to grow the user base. Additionally, it is shown later in the thesis (Chapter 4) how welfare maximization can help build a better social space overall by mitigating problems such as filter bubbles.

Next, it is illustrated how welfare maximization using utilities differ from classical adoption maximization objective using a simple example.

Example 3. Consider again the three items $iPhonePro$, $iPhoneMini$, and $AirPod$, abbreviated resp. iPp, iPm, AP. As mentioned before, owning iPp or iPm individually is useful, but AP is not without owning iPp or iPm first. iPp, and iPm exhibit partial competition. Finally, owning all three provides a user with many advantages that the apple ecosystem offers. Inspired by this, assume the following utilities for itemsets: $\mathcal{U}(\{\}) = 0$, $\mathcal{U}(\{iPp\}) = 1$, $\mathcal{U}(\{iPm\}) = 1$, $\mathcal{U}(\{AP\}) = -1$, $\mathcal{U}(\{iPp, AP\}) = 2$, $\mathcal{U}(\{iPm, AP\}) = 2$, $\mathcal{U}(\{iPp, iPm\}) = 1.5$, $\mathcal{U}(\{iPp, iPm, AP\}) = 6$.

Let the campaign budgets for iPp, iPm, and AP be $1, 1$ and 2 respectively. Now consider a network consisting of two isolated nodes A and B and two allocations α and β. In α, all three items are assigned to user A (say) and one remaining copy of AP is assigned to user B. In β, iPp and AP are assigned to user A and iPm and AP are assigned to user B. Note that a user adopts an itemset that has the highest utility, whereas no user adopts an itemset that has a negative utility. Since users will not adopt itemsets with negative utility, α results in user A adopting all three items and user B adopting none. However, under allocation, β, users A and B are both assigned itemsets with positive utility and so they each adopt two items. Since the network is disconnected, there is no diffusion to consider. Allocation α leads to 3 adoptions and β to 4. However, the net social welfare, which is the sum of utilities, for the two allocations, are 6 and 4 respectively. Thus an allocation that maximizes the adoption count, does not necessarily maximize the welfare and vice versa. □

This thesis studies adoptions of items that are governed by the utility of items, by combining a basic stochastic diffusion model with the utility model for item adoption. IC model is used as the diffusion model, however, the diffusion does not readily cause activation of nodes. Instead, nodes become aware of the items via diffusion. Then, based on the utility model, the nodes

are activated. The utility model is grounded in the theory of economics where the utility of items is studied extensively [45, 105, 111, 113]. Recall that, the utility is defined to be the difference between the value and the price. The previous works suggest that in practice, prices of items may be known, but our knowledge of users' valuation for items is generally uncertain. Thus in our formulation, random noise is added to the utility function to model the uncertainty. The final model is UIC, which is formally described in detail in Section 2.3.

Different interesting objectives can be formulated in UIC. This thesis formulates the novel optimization problem of finding the optimal allocation of items to seed nodes under item budget constraints to maximize the expected social welfare. Maximizing welfare ensures that users of the social network earn a higher utility from their adoptions. Since the adoptions are resulted from the recommendations users received via campaigns, a higher utility means that the campaigns are more effective in keeping the users happy. This in turn results in better user retention and helps the network owner to grow the network. Moreover, it enables the host to foster a better social space for the users where the effect of problems such as filter bubbles can be minimized.

Across different chapters of the thesis, the social welfare maximization objective is studied under different settings, namely, when items are complementary, competing or a mix of the two types. As a preview, readers can refer to Figure 1.3, which demonstrates how different parameters of UIC can be configured to formulate different problems. Under any of those different configurations, the task is NP-hard, but more challenging is the result that the expected social welfare does not exhibit properties such as monotonicity, submodularity, or supermodularity that help design efficient approximation algorithms. The major configurations that this thesis will focus on, and an outline of different chapters of the thesis is presented in the following section.

1.3.2 Thesis outline

Chapter 2 studies the complementary items and considers noise to be global, which implies that every node shares the same noise parameter. A novel propagation model, UIC, is also formally introduced first in Chapter 2. The key contributions of this chapter are as follows:

- In Section 2.3 it is shown how to incorporate utility-based item adoption with influence diffusion into a novel multi-item diffusion model,

called *Utility-driven IC* (UIC) model. The model, UIC is shown to support any mix of competing and complementary items that different chapters of this thesis study individually.

- In Section 2.3.4 UIC model is studied for complementary items. The differences from existing works are highlighted by surveying several categories of the relevant literature in Section 2.2. It is shown in Section 2.4 that under the reasonable assumptions that price and noise are additive, the expected social welfare is monotone but neither submodular nor supermodular.

- Despite the lack of submodularity, in Section 2.5 an efficient algorithm is designed that achieves a $(1 - 1/e - \epsilon)$-approximation to the optimal expected social welfare, for any small $\epsilon > 0$. While the main algorithm is still based on the greedy approach for solving submodular function maximization, its analysis is far from trivial, because the objective function is neither submodular nor supermodular. As part of the proof strategy, a novel *block accounting* method is developed for reasoning about expected social welfare for properly defined blocks of items. Another important feature of the algorithm is that it does not require the valuations or prices of items as the input, and merely the fact that item valuation is supermodular while price and noise are additive is sufficient to guarantee the approximation ratio. This means that the algorithm does not need to obtain the valuations or marginal valuations of items, which may not be straightforward to get in practice.

- The proposed algorithm is evaluated with other baselines on five large real networks, with both real and synthetic utility configurations in Section 2.6.

Chapter 3 focuses on competing products. The expected social welfare is not even monotone in this setting. The chapter also shows that it is not possible to approximate the general problem within any constant c, unless $P = NP$. Therefore different algorithms are developed that provide approximation guarantees only under specific assumptions. The major contributions of this chapter are as follows:

- The propagation model, UIC is extended for competing items in Chapter 3. Then a novel problem, CWelMax, is formulated in the context of competitive social welfare maximization in Section 3.3. CWelMax

considers a more general seed assignment problem, where some of the seeds are already frozen, in other words, they are already chosen and remain fixed.

- Under competition, it is even more challenging to design an approximation algorithm, because unlike Chapter 2 the social welfare function is also not monotone, besides being neither submodular nor supermodular. Further, it is NP-hard to approximate the general version of CWelMax within any constant factor. In Section 3.4), the inapproximability result is shown by designing a non-trivial gadget that highlights the specific challenges in approximating the problem.

- Since the general problem is inapproximable, in Section 3.5, a first algorithm having utility-dependent approximation guarantee is proposed for the general problem. Then other algorithms are developed which provide approximation guarantees only under specific assumptions, which render the objective monotone and submodular. With stronger assumptions, the approximation quality is shown to progressively improve, leading to a constant $1 - \frac{1}{e}$ approximation.

- In Section 3.6 an extensive experimental evaluation is conducted over several real social networks comparing the proposed algorithms of this chapter with the existing baseline algorithms. Under various utility configurations, it is shown that the proposed algorithms constantly outperform the baselines both on social welfare and running time.

Chapter 4 uses UIC for a different application, which is to tackle the problem of mitigating filter bubbles. It is shown that the objective of mitigating filter bubble requires a utility function that combines the flavor of competition and complementarity. Therefore, maximizing the utility is akin to maximizing social welfare for a mix of items showing both competition and complementarity among them. As a result, the problem, is significantly more difficult to solve. The significant contributions made in this chapter are as follows:

- In Section 4.2, an extensive review of existing research works on mitigating the filter bubble problem is presented. It is shown that a competition parameter is essential to correctly model the spirit of filter bubble which existing works lack. Consequently, in Section 4.3, a utility driven propagation model is designed which combines both competition and complementarity in the adoption decisions of nodes.

- In Section 4.4, it is shown that the objective function emulates a combination of competing and complementary items. As a result, the resulting social welfare objective is significantly hard to maximize. Unlike Chapter 3, even under natural restrictive assumptions, the objective is shown to be not monotone and not submodular.

- In Section 4.5, instance dependent approximation algorithms are devised first. It is shown that for one of the algorithms, the approximation it achieves is tight. Further by leveraging graph structure, a non-trivial heuristic is developed, which is empirically shown to have superior performance. In Section 4.6, the proposed algorithms are compared against existing baselines. It is shown that the algorithms proposed in this chapter outperform the baselines in terms of both efficiency and quality.

Finally, Chapter 5 presents some of the several interesting problems that can still be formulated using the utility driven diffusion model that this thesis introduces. The challenges that remain to be solved are highlighted and in conclusion a brief hypothesis of how future research can address those challenges is presented.

Chapter 2

Maximizing Welfare in Social Networks under A Utility Driven Influence Diffusion model

2.1 Introduction

Motivated by applications such as viral marketing, the problem of influence maximization has been extensively studied in the literature [93]. The seminal paper of Kempe et al. [83] formulated *influence maximization* (IM) as a discrete optimization problem: given a directed graph $G = (V, E, p)$, with nodes V, edges E, a function $p : E \to [0, 1]$ associating influence weights with edges, a stochastic diffusion model M, and a seed budget k, select a set $S \subset V$ of up to k seed nodes such that by activating the nodes S, the expected number of nodes of G that get activated under M is maximized. One of the fundamental diffusion models is independent cascade (IC). In IC every activated node makes a one-shot attempt to activate its neighbor based on the edge probability. The neighbor is activated if the attempt succeeds. IC based models and their extensions on IM have mostly focused on a single item or phenomenon propagating through the network and have developed efficient and scalable heuristic and approximation algorithms for IM [22, 38, 40, 42, 66, 127].

Subsequent work studies multiple items propagating through a network [17, 25, 31, 70, 96, 97, 116], mostly focusing on the objective of maximizing the number of items that get adopted. Among these, the majority of studies have concentrated on items, products, or campaigns *competing* with each other (see Section 2.2 for more details). One exception is the Com-IC model by Lu et al. [97], which studied the effect of *complementary* products propagating through a network. The efficiency of most of these algorithms depends on two key properties of the objective function, namely: *mono-*

tonicity and *submodularity*. Lu et al. [97] for their general model which is non-submodular, achieved a data-dependent approximation by developing a novel *sandwich approximation* strategy.

A significant omission from the literature on IM and viral marketing is a study with item adoptions grounded in a sound economic footing, where users rationally adopt itemsets to maximize their own utility.

Adoption of items by users is a well-studied concept in economics [105, 113]: item adoption by a user is driven by the *utility* that the user can derive from the item (or itemset). Precisely, a user's utility for an item(set) is the difference between the *valuation* that the user has for the item(set) and the *price* she pays. A rich body of literature in combinatorial auctions (see [52, 81, 84]) studies the optimal allocation of goods to users, given the users' valuation for various sets of goods. These studies are not concerned with the influence propagation in networks, whereby users' desire for items arises due to the influence from their network neighbors who already adopted items, and then these users may in turn adopt the items if they could obtain positive utility from them and start influencing their neighbors about these items. Considering such network propagation is important for applications such as viral marketing [83].

This thesis takes the first step toward combining viral marketing (influence maximization) with a framework of item adoption grounded in the economic principle of item utility and proposes a simple, elegant, yet powerful framework for capturing the interaction between these two paradigms. As a result of this seamless integration, this thesis studies an optimization objective, called *social welfare*, which is the sum of utilities of itemsets adopted by users at the end of a campaign, in expectation. While social welfare is well studied in combinatorial auctions, to our knowledge, it has not been studied in the context of viral marketing, taking into account the recursive network effect of items propagating via user adoptions and desires. This chapter specifically focuses on a setting where the items are mutually complementary. This is achieved by modeling user valuation for itemsets as a *supermodular* function (definition in Section2.2). Supermodularity nicely captures the intuition that between complementary items, the marginal value-gain of an item w.r.t. a set of items increases as the set grows. Many companies offer complementary products, such as, Apple offers iPhone X, Apple Watch, and AirPower, and Microsoft offers XBox console, XBox controller, and XBox games, etc. This chapter studies adoptions of complementary items, by combining a basic stochastic diffusion model for propagation with the utility model for item adoption. While the framework, results, and techniques carry over to any triggering model (see [83])

this chapter uses the well-known IC model for ease of exposition.

In practice, prices of items may be known however, the knowledge of users' valuation for items may be uncertain. In this thesis, noise is explicitly introduced in the framework to model this uncertainty. *No distributional assumptions about the noise is made*, except for the standard assumption that it is zero mean, which implies that the distribution is centered at zero. This thesis studies a novel objective of optimizing expected social welfare using the proposed diffusion model. Specifically, the problem is formulated as finding the optimal allocation of items to seed nodes so as to maximize the expected social welfare. Expected social welfare naturally generalizes the expected spread and hence is readily shown to be NP-hard to optimize. The resulting objective function of expected social welfare is monotone, but unlike most known studies in IM, it is neither submodular nor supermodular. It is shown that under the simple and reasonable assumption that price and noise are additive and valuation is supermodular, it is possible to efficiently find an allocation of items to seed users that achieves an expected social welfare that is within a factor of $(1 - 1/e - \epsilon)$ of the optimum. *To our knowledge, this is the first such result in the context of IM where, for a non-submodular (and non-supermodular) objective function such a high-quality approximation is achieved.*

While the main algorithm for item allocation, proposed in this chapter, is a simple greedy algorithm, proving that it achieves the approximation guarantee above is far from simple and involves intricate reasoning about the propagation of items, some combinations of which may have negative utility but may combine with other (complementary!) items to become positive. A novel accounting method is developed for reasoning about expected social welfare, which may be of independent interest.

A remarkable property of the allocation algorithm is that it does not require valuations or prices as input, in other words, it is agnostic to this information. In contrast, the existing adoption maximization algorithms for complementary products [97] need the adoption probabilities of an item given an already adopted itemset, as input.

The algorithm and analysis demonstrate in a rigorous way the power of bundling — one should bundle complementary items together as much as possible to achieve high social welfare, irrespective of the actual valuations, prices, or noise distributions of items. The next example illustrates the framework as well as the challenges.

Example 4. Consider the network shown in Figure 2.1, where all edge probabilities are assumed to be 1. Clearly nodes v_5 and v_1 are the two nodes

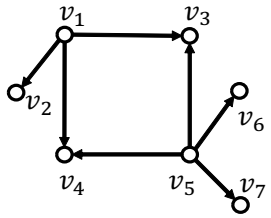


Figure 2.1: An illustrative network

Itemset	Utility
$\{i_1, i_2\}$	1
$\{i_1, i_3\}$	1
$\{i_1, i_2, i_3\}$	3

Table 2.1: Item subsets having nonnegative utility

with the highest expected spread, in order. There are three items $\{i_1, i_2, i_3\}$ propagating in the network. Assume no (that means, zero) noise. Table 2.1 shows item subsets with non-negative utilities. All other item subsets have strictly negative utility and hence are not adopted by any node. Let the budget for i_1, i_2, i_3 be $2, 1, 1$ respectively. Now a greedy allocation simply allocates all three items to node v_5 (the top node in terms of expected spread), and i_1 to v_1. After the end of the propagation, nodes v_3—v_7 adopt $\{i_1, i_2, i_3\}$, because the item-set $\{i_1, i_2, i_3\}$ is of the highest utility. Thus the expected social welfare, the sum of utilities of all nodes, obtained by greedy allocation is $5 \times 3 = 15$. Notice that the expected number of item adoptions is also 15.

Now consider another allocation that allocates $\{i_1, i_2\}$ to node v_1 and $\{i_1, i_3\}$ to v_5. In this case after the propagation ends, nodes v_1, v_2 adopt $\{i_1, i_2\}$, v_3, v_4 adopt $\{i_1, i_2, i_3\}$ and the remaining nodes adopt $\{i_1, i_3\}$. In terms of the number of adoptions, this allocation is superior to greedy: it results in a total of 16 adoptions among all items. However, the social welfare produced is 11, which is inferior to that of greedy. The example can be extended so that the (expected) number of adoptions for greedy is much worse than the alternative allocation, while greedy allocation has superior social welfare. This shows an important aspect of the problem — optimizing the number of item adoptions is not aligned with optimizing social welfare, which makes the latter problem challenging. □

A byproduct of the algorithmic solution is an adaptation of the state-of-

the-art single-item influence maximization algorithm IMM [127] to satisfy the *prefix-preserving property* for multiple items with diverse budgets. That is, assuming every item i has budget b_i, the algorithm outputs an ordered set S of $\bar{b} = \max_i b_i$ seeds, such that with high probability, *simultaneously* for all items i, the prefix of S with b_i nodes is a $(1 - 1/e - \epsilon)$-approximate solution to the optimal solution with b_i seeds, where $\epsilon > 0$ is a small constant on the approximation accuracy. The prefix-preserving property is achieved efficiently with the near-linear expected running time in the same order as the IMM algorithm on the largest budget $\bar{b}$, essentially independently of the number of items considered in the diffusion. This prefix-preserving algorithm could be of independent interest in its own right in multi-item IM.

Finally, in this chapter extensive empirical evaluations are conducted of the proposed algorithm together with a number of baselines, on both real and synthetic datasets. The empirical results demonstrate that the proposed algorithm significantly outperforms all baselines in terms of one or both of the quality of social welfare achieved and the running time.

The rest of the chapter is organized as follows:

- A survey of different categories of the existing relevant literature is presented in Section 2.2. In addition, the section highlights the differences that this chapter has from those works.
- In Section 2.3 it is shown how to incorporate utility-based item adoption with influence diffusion into a novel multi-item diffusion model, *Utility-driven IC* (UIC) model. The proposed model, UIC is shown to support any mix of competing and complementary items.
- UIC model is studied for complementary items in Section 2.3.4. A greedy allocation algorithm is shown to achieve a $(1-1/e-\epsilon)$-approximation ratio, even though the social welfare function is neither submodular nor supermodular (Section 2.4 and Section 2.5).
- Along the way, a prefix-preserving IM algorithm is designed for multi-item influence maximization that may be of independent interest, with running time in the same order as the IMM algorithm on the maximum budgeted item, independent of the number of items in the diffusion (Section 2.5).
- A detailed set of experiments is conducted comparing the performance of the proposed algorithm with baselines on 5 large real networks, with

both real and synthetic utility configurations. The results show that the proposed algorithm significantly dominates the baselines in terms of running time or expected social welfare or both, and also provide insights on the effect of budget skew between items on the expected social welfare (Section 2.6).

- In Section 2.7, the chapter is summarized with some future research directions that is conducted in the later chapters.

2.2 Background & Related Work

2.2.1 Single-item Influence Maximization

As mentioned in Chapter 1, in classical IM, social network is represented as a directed graph $G = (V, E, p)$, V being the set of nodes (users), E the set of edges (connections), with $|V| = n$ and $|E| = m$. The function $p : E \to [0, 1]$ specifies influence probabilities (or weights) between users. The *influence spread* of a seed set S, denoted $\sigma(S)$, is the expected number of active nodes after the diffusion that starts from the seed set S ends. *Influence maximization* (IM) is the problem of finding, for a given number k and a diffusion model, a set $S \subset V$ of k seed nodes that generates the maximum influence spread $\sigma(S)$.

Most existing studies on IM rely on the corresponding influence spread function $\sigma(S)$ being monotone and submodular (see Section 1.1.2 for the definitions). Borgs et al. [22] proposed the notion of random reverse reachable sets (rr-sets) for estimating the spread function $\sigma(S)$. Building on the notion of rr-sets, a family of scalable approximation algorithms such as TIM, IMM, and SSA, have been developed for IM [74, 110, 127, 128] which are orders of magnitude faster than the classic greedy algorithm making use of MC simulations for estimating the spread [83].

Motivated by designing an influence oracle, that responds to queries to find seeds for any given budget, Cohen et al. [42] proposed an IM algorithm called SKIM that leverages bottom-k sketches. A noteworthy property of SKIM is that it produces an ordering of the nodes such that any prefix of the ordering consisting of k nodes is guaranteed to have a spread that is at least $(1 - 1/e - \epsilon)$ times the optimal spread for a seed budget of k. Thus, SKIM is essentially a *prefix-preserving algorithm* in context of single item IM. However, as shown in [42], SKIM does not dominate TIM in performance. Given that IMM is orders of magnitude faster than TIM, there is a natural motivation to build a prefix-preserving IM algorithm by adapting IMM to a multi-item context, which is done in this chapter.

Influence maximization under non-submodular models has been studied in previous works [36, 91, 97, 121]. Most of them show hardness of approximation results [36, 91, 121]. In terms of approximation algorithms, Chen et al. rely on a low-rank assumption to provide an algorithm solving the non-submodular amphibious influence maximization problem with an approximation ratio of $(1-1/e-\epsilon)^3$ [36]. Lu et al. use the sandwich approximation to give a problem instance dependent approximation ratio [97]. Schoenebeck and Tao provide a dynamic programming algorithm for influence maximiza-

tion in the restricted one-way hierarchical blockmodel [121]. Li et al. provide an approximation algorithm with approximation ratio $(1-\epsilon)^{\ell}(1-1/e)$, in a network when at most ℓ nodes are ϵ-almost submodular and the rest of the nodes are submodular [91]. In contrast, the algorithm presented in this chapter achieves the $(1-1/e-\epsilon)$ approximation ratio (same as the ratio for submodular maximization) for a non-submodular objective function, under a general network without further assumptions.

2.2.2 Multi-item Influence Maximization

More recently, multiple items have been considered in the context of viral marketing of non-competing items [47, 106]. However their proposed solutions do not provide the typical $(1-1/e-\epsilon)$-approximation guarantee. Specifically Datta et al. [47] studied IM where propagations of items are assumed to be independent and provided a 1/3-approximate algorithm. In [106], Narayanam et al. propose an extension, where items are partitioned into two sets. A product can be adopted by a node only when it has already adopted a corresponding product in the other set. Such partition of itemsets, with strong dependencies on mutual adoptions of items in the two sets, represents a restricted special case of item adoptions in the real world.

Competitive influence maximization is studied in [17, 23, 25, 28, 34, 70, 96, 116] (see [35] for a survey), where a user adopts at most one item from the set of items being propagated. The works mainly focus on the "follower's perspective" [17, 25, 28, 70], where, given competitor's seed placement, select seeds so as to maximize one's own spread, or minimize the competitor's spread. Lu et al. [96] focused on maximizing the total influence spread of all campaigners from a network host perspective, while ensuring fair allocation.

Lu et al. [97] introduced a model called Com-IC capturing both competition and complementarity between a pair of items, leveraging the notion of a node level automaton (NLA). An NLA is a stochastic decision-making automaton governed by transition probabilities for a user adopting an item given what it has already adopted. Their model subsumes perfect complementarity and pure competition as special cases. However, their main study is confined to the diffusion of two items, and a straightforward extension to multiple items would need an exponential number of parameters in the number of items. Moreover, their general parameter settings could lead to anomalies such as one item complementing a second item but the second one competing with the first one, or being indifferent to it.

All of the above works on multiple item propagations focus on maximizing the expected number of item adoptions, which is not aligned with social

welfare (see Example 4).

Myers and Leskovec analyzed the effects of different cascades on users and predicted the likelihood that a user will adopt an item, seeing the cascades in which the user participated [104]. McAuley et al.[100] proposed a method to learn complementary relationships between products from user reviews. None of the works models the diffusion of complementary items, nor study the IM problem in this context.

2.2.3 Combinatorial Auctions

Combinatorial auctions are widely studied and a complete survey is beyond the scope of this section. Instead, a few key papers are discussed. In economics, adoption of items by users is modeled in terms of the utility that the user derives from the adoption [72, 105, 113, 118]. A classic problem is given m users and n items and the utility function of users for various subsets of items, find an allocation of items to users such that the social welfare, which is the sum of utilities of users resulting from the allocation, is maximized. This intractable problem has been studied in both offline and online settings [45, 52, 81, 84] and various approximation algorithms have been developed. All of them assume access to a value oracle or a demand oracle. A value oracle is a black box, which given a set of items as a query, returns the value of the itemset. A demand oracle is a black box, which given an assignment of prices to items, returns the itemset with maximum utility, where utility is value minus price. Also, the utility function in these settings is typically assumed to be sub-additive and as a result, this property extends to social welfare. Notably, these works do not consider the interaction of utility-maximizing item adoption with recursive propagation through a network. On the other hand, they consider more general settings where the utility functions are user-specific.

Inspired by the economics literature, the proposed model base item adoptions on item utility. Specifically, items have a price and a valuation and the difference is the utility. It is a well-accepted principle in economics and auction theory [45, 111] that users (agents), presented with a set of items, adopt a subset of items that maximizes their utility. It is this principle that is used in the framework proposed in this work, to govern which users adopt what items.

The use of utility naturally leads to the notion of *social welfare* and this chapter studies the problem of assigning seed nodes to various items in order to maximize expected social welfare, in a setting where items are complementary. To our knowledge, in the context of viral marketing, this work is

the first to study the problem of maximizing (expected) social welfare.

2.2.4 Welfare maximization on social networks

There are a few studies related to welfare maximization on social networks, but they all have significant differences with the proposed model and problem setting of this chapter. Sun et al. [124] study participation maximization in the context of online discussion forums. An item in that context is a discussion topic, and adopting an item means posting or replying on the topic. Item adoptions do propagate in the network, but (a) item propagations are independent (in other words, valuation of itemsets is additive rather than supermodular or submodular), and (b) they have a budget on the number of items each seed node can be allocated with, rather than on the number of seeds each item can be allocated to as studied in the model of this chapter. Bhattacharya et al. [19] consider item allocations to nodes for welfare maximization in a network with network externalities, but the major differences with problem of this chapter are: (a) they use network externalities to model social influence, hence, a user's valuation of an item is affected by the number of her one- or two-hop neighbors in the network adopting the same item, but network externalities do not model the *propagation* of influence and item adoptions, the main focus in this chapter is in modeling the viral marketing effect; (b) they consider unit demand or bounded demand on each node, which means items are competing against one another on every node, while this chapter focuses on the case of complementary items rather than competing items, and item bundling is a key component in the solution presented in this work; (c) they do not have budget on items so an item could be allocated to any number of nodes, while the problem studied in this chapter, has a budget on the number of nodes that can be allocated to an item as seeds and the proposed model relies on propagation for more nodes to adopt items. Despite these major differences, an empirical comparison of the proposed algorithm versus their algorithms is performed to demonstrate that with propagation the proposed algorithm can achieve the same social welfare with only a fraction of item budgets used in their solution. Abramowitz and Anshelevich [1] study network formation with various constraints to maximize social welfare, but it has no item allocation, no item complementarity, and no influence propagation, and thus is further away from the work of this chapter. In summary, to our knowledge, this chapter is the only one addressing social welfare maximization in a network with influence propagation, complementary items, and budget limits on items.

G, V, E, n and m	Graph, node set, edge set, number of nodes and number of edges
$p : E \to [0, 1]$	Influence weight function
$\mathbf{I}$	Universe of items
$\mathcal{P}$, $\mathcal{V}$, $\mathcal{N}$ and $\mathcal{U}$	Price, Value, Noise and Utility
$\vec{b}$	Budget vector
$\bar{b}$	Maximum budget
$\boldsymbol{\mathcal{S}}$	Seed allocation, as set of node-item pairs
S	Seed nodes
$S_i^{\boldsymbol{\mathcal{S}}}$	Seed nodes of item i in allocation $\boldsymbol{\mathcal{S}}$
$S^{\boldsymbol{\mathcal{S}}}$	All seed nodes of allocation $\boldsymbol{\mathcal{S}}$
$\mathbf{I}_v^{\boldsymbol{\mathcal{S}}}$	Items allocated to seed node v in allocation $\boldsymbol{\mathcal{S}}$
$\mathcal{R}^{\boldsymbol{\mathcal{S}}}(u,t)$ and $\mathcal{A}^{\boldsymbol{\mathcal{S}}}(u,t)$	Desire and adoption set of u at time t in allocation $\boldsymbol{\mathcal{S}}$
σ and ρ	Expected adoption and social welfare
W, W^E, W^N	Possible world, edge and noise possible world
Grd and OPT	Greedy and optimal allocation
B and $\mathcal{B}$	A block and a sequence of item disjoint blocks
e_i	Effective budget of block B_i
B_i^a and a_i	Anchor block and anchor item of block B_i

Table 2.2: Table of notations

2.3 UIC Model

In this section, a novel propagation model is proposed, called *utility driven independent cascade* model (UIC for short), that combines the diffusion dynamics of the classic IC model with an item adoption framework where decisions are governed by utility. Table 2.2 summarizes the notations used henceforth.

2.3.1 Utility based adoption

Utility is a widely studied concept in economics and is used to model item adoption decisions of users [52, 105, 113]. Next, in this section idea of utility is briefly reviewed, and the specific formulation is provided which is used in this chapter. For general definitions related to utility, the reader is referred to [52, 113]

Let $\mathbf{I}$ denote a finite universe of items. The utility of a set of items $I \subseteq \mathbf{I}$ for a user is the pay-off of I to the user and depends on the aggregate effect of three components: the price $\mathcal{P}$ that the user needs to pay, the valuation $\mathcal{V}$ that the user has for I and a random noise term $\mathcal{N}$, used to model the uncertainty in our knowledge of the user's valuation on items, where $\mathcal{P}$, $\mathcal{V}$ and $\mathcal{N}$ are all set functions over items. For an item $i \in \mathbf{I}$, $\mathcal{P}(i) > 0$ denotes its price. Price is assumed to be additive. Thus, for an itemset $I \subseteq \mathbf{I}$, $\mathcal{P}(I) = \sum_{i \in I} \mathcal{P}(i)$. Notice that UIC can handle any generic valuation function. Section 2.3.4 focuses on complementary products. Hence it is

assumed that $\mathcal{V}$ is supermodular (definition in Section 1.1.2), meaning that the marginal value of an item with respect to an itemset I increases as I grows. Later in the thesis, Chapter 3 focuses on competing products, where the value is assumed to be submodular, in other words, marginal value of an item with respect to an itemset I decreases as I grows. It is also assumed that $\mathcal{V}$ is monotone since it is a natural property for valuations. For $i \in \mathbf{I}$, $\mathcal{N}(i) \sim \mathcal{D}_i$ denotes the noise term associated with item i, where the noise may be drawn from any distribution $\mathcal{D}_i$ having a zero mean. Every item has an independent noise distribution. For a set of items $I \subseteq \mathbf{I}$, it is assumed that the noise is additive, therefore, the noise of I, $\mathcal{N}(I) := \sum_{i \in I} \mathcal{N}(i)$. Similar assumptions on additive noise are used in economics theory [72, 76].

Finally, the utility of an itemset I is $\mathcal{U}(I) = \mathcal{V}(I) - \mathcal{P}(I) + \mathcal{N}(I)$. Since noise is a random variable, utility is also random. Since noise is drawn from a zero mean distribution, $\mathbb{E}[\mathcal{U}(I)] = \mathcal{V}(I) - \mathcal{P}(I)$. Also, $\mathcal{V}(\emptyset) = 0$.

2.3.2 Diffusion Dynamics

2.3.2.1 Seed allocation

Let $\vec{b} = (b_1, ..., b_{|\mathbf{I}|})$ be a vector of natural numbers representing the budgets associated with the items. An item's budget specifies the number of seed nodes that may be assigned to that item. Sometimes, notation is slightly abused and $b_i \in \vec{b}$ is used to indicate that b_i is one of the item budgets. The maximum budget is denoted as $\overline{b} := max\{b_i \mid b_i \in \vec{b}\}$. An *allocation* is defined as a relation $\boldsymbol{\mathcal{S}} \subset V \times \mathbf{I}$ such that $\forall i \in \mathbf{I} : |\{(v, i) \mid v \in V\}| \leq b_i$. In words, each item is assigned a set of nodes whose size is under the item's budget. The nodes $S_i^{\boldsymbol{\mathcal{S}}} := \{v \mid (v, i) \in \boldsymbol{\mathcal{S}}\}$ are referred to as the *seed nodes* of $\boldsymbol{\mathcal{S}}$ for item i and nodes $S^{\boldsymbol{\mathcal{S}}} := \bigcup_{i \in \mathbf{I}} S_i^{\boldsymbol{\mathcal{S}}}$ are referred to as the *seed nodes* of $\boldsymbol{\mathcal{S}}$. The set of items allocated to a node is denoted as $v \in V$ as $\mathbf{I}_v^{\boldsymbol{\mathcal{S}}} := \{i \in \mathbf{I} \mid (v, i) \in \boldsymbol{\mathcal{S}}\}$.

2.3.2.2 Desire and adoption

Every node maintains two sets of items – desire set and adoption set. Desire set is the set of items that the node has been informed about (and thus potentially desires), via propagation or seeding. Adoption set is the subset of the desire set that the node adopts. At any time a node selects, from its desire set at that time, the subset of items that maximizes the utility, and adopts it. If there is a tie in the maximum utility between itemsets, then it is broken in favor of larger itemsets. Later in Lemma 1 of Section 2.4, it is shown that breaking ties in this way results in a well-defined adoption

behavior of the nodes. Following previous literature, a progressive model is considered: once a node desires an item, it remains in the node's desire set forever; similarly, once an item is adopted by a node, it cannot be unadopted later.

For a node u, $\mathcal{R}^{\boldsymbol{\mathcal{S}}}(u,t)$ denotes its desire set and $\mathcal{A}^{\boldsymbol{\mathcal{S}}}(u,t)$ denotes its adoption set at time t, pertinent to an allocation $\boldsymbol{\mathcal{S}}$. The time argument t is omitted to refer to the adoption (desire) set at the end of diffusion.

The diffusion process of UIC is presented next.

2.3.2.3 The diffusion model

In the beginning of any diffusion, the noise terms of all items are sampled, which are then used until the diffusion terminates. The diffusion then proceeds in discrete time steps, starting from $t = 1$. Given an allocation $\boldsymbol{\mathcal{S}}$ at $t = 1$, the seed nodes have their desire sets initialized : $\forall v \in S^{\boldsymbol{\mathcal{S}}}$, $\mathcal{R}^{\boldsymbol{\mathcal{S}}}(v,1) = \mathbf{I}_v^{\boldsymbol{\mathcal{S}}}$. Seed nodes then adopt the subset of items from the desire set that maximizes the utility, breaking ties if needed in favor of sets of larger cardinality. Thus, a seed node may adopt just a subset of items allocated to it.

Once a seed node u' adopts an item i, it influences its out-neighbor u with probability $p_{u',u}$, and if it succeeds, then i is added to the desire set of u at time $t = 2$. The rest of the diffusion process is described in Fig. 2.2.

The diffusion under UIC is illustrated using an example next.

Example 5. The graph G with edge probabilities and the utilities of the two items after sampling the noise terms, are shown in Figure 2.3. Note that the graph is the same as that of Example 1. The items i_1 and i_2 are, respectively, *iPhonePro* and *AirPod* of Example 3, hence they have the same corresponding utilities. The diffusion is shown in Figure 2.4. At time $t = 1$, node v_1 is seeded with item i_1 and v_3 with i_2, hence they desire those items respectively. Since i_1 (resp. i_2) has a positive (resp. negative) individual utility, v_1 adopts i_1 (resp. v_3 does not adopt i_2). However i_2 remains in the desire set of v_3. Then at $t = 2$, outgoing edges of v_1 are tested for transition: edge (v_1, v_3) fails (shown as red dotted line), but edge (v_1, v_2) succeeds (green solid line). Consequently v_2 desires and adopts i_1. Next at $t = 3$, v_2's outgoing edge (v_2, v_3) is tested. As it succeeds, v_3 desires i_1. Since it already had i_2 in its desire set, it adopts the set $\{i_1, i_2\}$. Since there is no outgoing edge from v_3, the propagation ends.

Note that the influence propagation is similar to the propagation of Example 1, but it is appended with item utilities that govern the final item adoptions of this example. □

1. Edge transition. At every time step $t > 1$, for a node u' that has adopted at least one new item at $t-1$, its outgoing edges are tested for transition. For an untested edge (u', u), flip a biased coin independently: (u', u) is *live* w.p. $p_{u',u}$ and *blocked* w.p. $1 - p_{u',u}$. Each edge is tested *at most once* in the entire diffusion process and its status is remembered for the duration of a diffusion process.

Then for each node u that has at least one in-neighbor u' (with a live edge (u', u)) which adopted at least one item at $t-1$, u is tested for possible item adoption (2-3 below).

2. Generating desire Set. The desire set of node u at time t, $\mathcal{R}^{\boldsymbol{\mathcal{S}}}(u,t) = \mathcal{R}^{\boldsymbol{\mathcal{S}}}(u,t-1) \cup_{u' \in N^-(u)} (\mathcal{A}^{\boldsymbol{\mathcal{S}}}(u', t-1))$, where $N^-(u) = \{u' \mid (u', u)$ is live$\}$ denotes the set of in-neighbors of u having a live edge connecting to u.

3. Node adoption. Node u determines the utilities for all subsets of items of the desire set $\mathcal{R}^{\boldsymbol{\mathcal{S}}}(u,t)$. u then adopts a set $T^* \subseteq \mathcal{R}^{\boldsymbol{\mathcal{S}}}(u,t)$ such that $T^* = \arg\max_{T \in 2^{\mathcal{R}^{\boldsymbol{\mathcal{S}}}(u,t)}} \{\mathcal{U}(T) \mid T \supseteq \mathcal{A}^{\boldsymbol{\mathcal{S}}}(u, t-1) \ \wedge \ \mathcal{U}(T) \geq 0\}$. $\mathcal{A}^{\boldsymbol{\mathcal{S}}}(u,t)$ is set to T^*.

Figure 2.2: Diffusion dynamics under UIC model

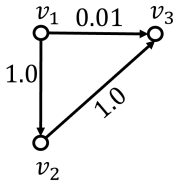
2.3.3 Definition of Social welfare Maximization

Let $G = (V, E, p)$ be a social network, $\mathbf{I}$ the universe of items under consideration. A novel utility-based objective is considered here. The objective is called *social welfare*, which is the sum of all users' utilities of itemsets adopted by them after propagation converges. Formally, $\mathbb{E}[\mathcal{U}(\mathcal{A}^{\boldsymbol{\mathcal{S}}}(u))]$ is the expected utility that a user u enjoys for a seed allocation $\boldsymbol{\mathcal{S}}$ after propagation ends. Then the *expected social welfare* (also known as "consumer surplus" in algorithmic game theory) for $\boldsymbol{\mathcal{S}}$, is $\rho(\boldsymbol{\mathcal{S}}) = \sum_{u \in V} \mathbb{E}[\mathcal{U}(\mathcal{A}^{\boldsymbol{\mathcal{S}}}(u))]$, where the expectation is over both the randomness of propagation and randomness of noise.

Key features of UIC.

The proposed utility driven model has several benefits over existing models.

Firstly, the seed users in the proposed model are treated as rational



	$\mathscr{U}(.)$
i_1	1
i_2	−1
i_1, i_2	2

Figure 2.3: Graph and utility configuration of the example propagation shown in Figure 2.4; for simplicity, assume noise is zero.

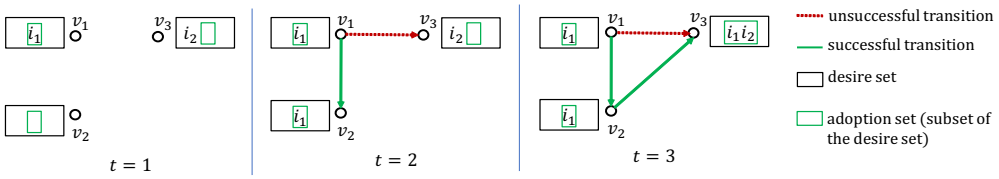


Figure 2.4: Illustrating propagation of items under UIC model

users. Thus they also go through the same utility based decision making like every other user of the network.

Secondly, Com-IC cannot handle the arrival of a set of items together. It had to use arbitrary tie-breaking in a case when a node becomes aware more than one items simultaneously, to put an order in the adoption. In UIC, this is treated by creating an explicit desire set for nodes first. The utility is a set function as opposed to the point probability of GAP. Therefore even if more than one items arrive at the same time instance, the utility can treat them as a set, without the need of enforcing an explicit order.

Third, the notion of utility opens up a whole new objective of social welfare maximization, where instead of maximizing just the adoption, the utility earned from the adoptions is aimed to be maximized. No other model reported to date, has been able to study social welfare maximization. UIC is the first framework which enables the study of utility in IM context.

Fourth, for complimentary products under UIC, a greedy allocation algorithm that preserves $1 - \frac{1}{e}$ approximation guaranty with respect to optimal social welfare, although social welfare is not submodular in seed size. This greedy algorithm is independent of the model parameters. Hence it can be easily extended to multiple items, whereas for Com-IC extending the algorithm beyond two items was difficult due to parameter explosion.

The problem of maximizing expected social welfare is defined ($WelMax$) as follows. This chapter refers to $\mathcal{V}, \mathcal{P}, \mathcal{N}$, as the model parameters and denotes them collectively as Param.

Problem 1. *[WelMax] Given $G = (V, E, p)$, the set of model parameters* Param*, and budget vector $\vec{b}$, find a seed allocation $\boldsymbol{\mathcal{S}}^*$, such that $\forall i \in \mathbf{I}$, $|S_i^{\boldsymbol{\mathcal{S}}^*}| \leq b_i$ and $\boldsymbol{\mathcal{S}}^*$ maximizes the expected social welfare, thus, $\boldsymbol{\mathcal{S}}^* = \arg\max_{\boldsymbol{\mathcal{S}}} \rho(\boldsymbol{\mathcal{S}})$.*

Unfortunately, WelMax is NP-hard.

Proposition 1. *WelMax in the UIC model is NP-hard.*

Proof. It is easy to verify that Influence maximization under the IC model, an NP hard problem, is a special case of WelMax.

The result follows from the fact that the IM problem under the IC model is a special case of WelMax: let $\mathbf{I} = \{i\}$, set $\mathcal{V}(i) = 1$, $\mathcal{P}(i) = 0$ and set the noise term for item i to 0. This makes $\mathcal{U}(i) = 1$ so any influenced node will adopt i. Thus, the expected social welfare is simply the expected spread. It is known that maximizing expected spread under the IC model is NP-hard [83]. □

2.3.3.1 Function Types

Notice that the functions $\mathcal{V}$ and $\mathcal{U}$ are functions over sets of items, whereas σ is a function over sets of network nodes, and ρ is a function over allocations, which are sets of (node, item) pairs. When a certain property (such as, submodularity) of a function of a given type is discussed, the property is meant w.r.t. the applicable type. For example, σ is monotone and submodular w.r.t. sets of nodes.

2.3.3.2 Design choices

In the UIC model, the desire set of a user is triggered either by seeding or by the influence on a user as her peers adopt items. Thus following standard practice in IM models, the model is kept to be progressive: a desire set never shrinks. On the other hand, the adoption decisions are driven by a standard assumption in economics [20], that users aim to maximize the utility when they adopt item(sets). UIC inherits this assumption to govern adoption decisions of the users. In UIC, it is assumed that price is additive. There are different ways of pricing a bundle of items: additivity is a simple and natural pricing model in the absence of discounts [32]. Further, supermodular value functions to model the effect of complementarity and competition respectively, among the products. This again follows the standard practice in the economics literature [101, 129].

The zero mean assumption for the noise distribution is an technical assumption that is made. The results of the chapter will hold if the mean is non-zero, as long as all the noise distributions of the items have the same mean value. Finally, the way noise is modeled as a global parameter, can be viewed as reflecting the uncertainty in the population's reaction to an item. One may further introduce personalized noise to model individual uncertainty, thereby users can have vastly different utilities for different items. Alternatively, one can consider where utilities can change based on different times of the diffusion. All these considerations would make algorithm design and analysis more difficult. The approximation bound of this chapter would not hold when noise is personalized and when valuation is not supermodular. Although specific design choices are made in this chapter for simplicity and tractability of the model, the UIC model can encompass any general form of value, price, and noise parameters and works for any triggering model [83].

2.3.4 UIC For Complementary items

This section, focuses on a setting where the items are mutually complementary, by modeling user valuation for itemsets as a *supermodular* function. Recall that a function $f : 2^U \to \mathbb{R}$ is supermodular if for any subsets $S \subset T \subset U$ and item $x \in U \setminus T$, $f(S \cup \{x\}) - f(S) \leq f(T \cup \{x\}) - f(T)$. Supermodularity captures the intuition that between complementary items, the marginal value-gain of an item w.r.t. a set of items increases as the set grows. Many companies offer complementary products, e.g., Apple offers iPhone, and AirPod. The marginal value-gain of AirPod is higher for a user who has bought an iPhone, compared to a user who hasn't. Complementary items have been well studied in the economics literature and supermodular function is a typical way for modeling their valuations (e.g., see [27, 129]). As a preview, the experiments conducted later indeed show that complementary items are natural and that their valuation is indeed supermodular. We study adoptions of complementary items, by combining a basic stochastic diffusion model with the utility model for item adoption. The highlights of the section are as follows:

1. We propose a greedy allocation algorithm, and show that the algorithm achieves a $(1 - 1/e - \epsilon)$-approximation ratio, even though the social welfare function is neither submodular nor supermodular (Section 2.4 and Section 2.5). Our main technical contribution is the block accounting method, which distributes social welfare to properly defined item blocks. The analysis is highly nontrivial and may be of independent interest to other studies.

2. We design a *prefix-preserving* seed selection algorithm for multi-item IM that may be of independent interest, with running time and memory usage in the same order as the scalable approximation algorithm IMM [127] on the maximum budgeted item, regardless of the number of items (Section 2.5).

3. We conduct detailed experiments comparing the performance of our algorithm with baselines on five large real networks, with both real and synthetic utility configurations. Our results show that our algorithm significantly dominates the baselines in terms of running time or expected social welfare or both (Section 2.6).

2.4 Properties Of UIC Under Supermodular Valuations

Since WelMax is NP-hard, this section first explores properties of the welfare function – monotonicity and submodularity, which can help to design efficient approximation strategies. An equivalent possible world model is presented to help the analysis.

2.4.1 Possible world model

Given an instance $\langle G, \mathsf{Param}\rangle$ of UIC, where $G = (V, E, p)$, a *possible world* associated with the instance is defined as a pair $W = (W^E, W^N)$, where W^E is an *edge possible world* (edge world), and W^N is a *noise possible world* (noise world); W^E is a sample graph drawn from the distribution associated with G by sampling edges, and W^N is a sample of noise terms for each item in $\mathbf{I}$, drawn from the corresponding item's noise distribution in Param.

As all the random terms are sampled, propagation and adoption in W is fully deterministic. For nodes $u, v \in V$, v is said to be reachable from u in W if there is a directed path from u to v in the deterministic graph W^E. $\mathcal{N}_W(i)$ denotes the sampled noise for item i and $\mathcal{U}_W(I)$ denotes the (deterministic) utility of itemset I, in world W. For a node u and an allocation $\boldsymbol{\mathcal{S}}$, its desire and adoption sets at time t in world W are denoted as $\mathcal{R}^{\boldsymbol{\mathcal{S}}}_W(u,t)$ and $\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(u,t)$ respectively. When only the noise terms are sampled, in other words, in a noise world W^N, the utilities are deterministic, but the propagation remains random.

Given a possible world $W = (W^E, W^N)$ and an allocation $\boldsymbol{\mathcal{S}}$, a node $v \in V$ adopts a set of items as follows: (i) if v is a seed node, then it desires $\mathbf{I}^{\boldsymbol{\mathcal{S}}}_v$ at time $t = 1$ and adopts an itemset $\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(v,1) := \arg\max\{\mathcal{U}_W(I) \mid I \subseteq \mathbf{I}^{\boldsymbol{\mathcal{S}}}_v\}$; (ii) if v is a non-seed node, and $t > 1$, then it desires the itemset $\mathcal{R}^{\boldsymbol{\mathcal{S}}}_W(v,t) := (\bigcup_{u \in N^{-1}_W(v)} \mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(u,t-1)) \cup \mathcal{R}^{\boldsymbol{\mathcal{S}}}_W(v,t-1)$, where $N^{-1}_W(v)$ denotes the in-neighbors of v in the deterministic graph W^E, thus, at time t, node v desires items that it desired at $(t-1)$ as well as items any of its in-neighbors in W^E adopted at $(t-1)$; node v then adopts the itemset $\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(v,t) := \arg\max\{\mathcal{U}_W(I) \mid I \subseteq \mathcal{R}^{\boldsymbol{\mathcal{S}}}_W(v,t)\ \&\ \mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(v,t-1) \subseteq I\}$. If there is more than one itemset in $\mathcal{R}^{\boldsymbol{\mathcal{S}}}_W(v,t)$ with the same maximum utility, it is assumed that v breaks ties in favor of the set with the larger cardinality.

$\mathcal{V}(\cdot)$ is supermodular while $\mathcal{P}(\cdot)$ and $\mathcal{N}_W(\cdot)$ are additive and hence modular, so it immediately follows that $\mathcal{U}_W(\cdot)$ is supermodular with respect to sets of items. Thus the expectation of utility w.r.t. edge worlds is supermodular. However, $\mathcal{U}_{W^N}$ is not monotone, because adding an item with a

very high price may decrease the utility.

A basic property is proven next, which helps in showing that the adoption behavior of the nodes is well defined in UIC. In any possible world, given a set of items that a node desires, there is a unique set of items that it adopts. Specifically, if there are multiple sets tied for utility, the node will adopt their union. For a set function $f : 2^U \to R$, let $f(T \mid S) = f(S \cup T) - f(S)$.

An itemset A is said to be a *local maximum* w.r.t. the utility function $\mathcal{U}_W$, if the utility of A is the maximum among all its subsets, formally speaking, $\mathcal{U}_W(A) = \max_{A' \subseteq A} \mathcal{U}_W(A')$. The following lemma is based on simple algebraic manipulations on the definitions of supermodularity and local maximum.

Lemma 1. *(Local maximum). Let W be a possible world and $A, B \subseteq \mathbf{I}$ be any itemsets such that A and B are local maximum with respect to $\mathcal{U}_W$. Then $(A \cup B)$ is also a local maximum with respect to $\mathcal{U}_W$, that is, $\mathcal{U}_W(A \cup B) = \max_{C \subseteq A \cup B} \mathcal{U}_W(C)$.*

Proof. For any subset $C \subseteq A \cup B$,

$$\begin{aligned}
\mathcal{U}_W(C) &= \mathcal{U}_W(C \setminus B \mid B \cap C) + \mathcal{U}_W(B \cap C) \\
&\leq \mathcal{U}_W(C \setminus B \mid B) + \mathcal{U}_W(B) && (2.1) \\
&= \mathcal{U}_W(C \cup B) \\
&= \mathcal{U}_W(B \mid C \setminus B) + \mathcal{U}_W(C \setminus B) \\
&\leq \mathcal{U}_W(B \mid A) + \mathcal{U}_W(A) && (2.2) \\
&= \mathcal{U}_W(A \cup B).
\end{aligned}$$

Inequality (2.1) follows from applying supermodularity of $\mathcal{U}_W$ on the first term, and applying local maximum of B on the second term. Inequality (2.2) follows applying supermodularity of $\mathcal{U}_W$ on the first term, and applying local maximum of A on the second term. □

An immediate consequence of Lemma 1 is that when two itemsets have the same largest utility, their union must also have the largest utility, and thus the tie-breaking rule is well-defined. Another consequence is the following lemma.

Lemma 2. *For any node u and any time t, the itemset adopted by u at time t, $\mathcal{A}^{\mathcal{S}}_W(u, t)$, must be a local maximum.*

Proof. It is proven by an induction on t. The base case of $t = 1$ is true because by the model, node u adopts the local maximum among all subsets of

items allocated to it. For the induction step, suppose for a contradiction that $\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(u,t)$ is not a local maximum but $\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(u,t-1)$ is a local maximum. Then there must exist a $C \subset \mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(u,t)$ that is a local maximum and $\mathcal{U}_W(C) > \mathcal{U}_W(\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(u,t))$. By Lemma 1, $C \cup \mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(u,t-1)$ is also a local maximum, and thus $C \cup \mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(u,t-1)$ cannot be $\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(u,t)$. But since $\mathcal{U}_W(C \cup \mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(u,t-1)) \geq \mathcal{U}_W(C) > \mathcal{U}_W(\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(u,t))$, u should adopt $C \cup \mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(u,t-1)$ instead of $\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(u,t)$, a contradiction. □

The next result shows that in any given possible world, adoption of items propagates through reachability. Reachability is a key property to be used later in Lemmas 5 and 7 while establishing the approximation guarantee of the algorithm.

Lemma 3. *(Reachability). For any item i and any possible world W, if a node u adopts i under allocation $\boldsymbol{\mathcal{S}}$, then all nodes that are reachable from u in the world W also adopt i.*

Proof. Consider a possible world W and a node u that adopts item i. Consider any node v reachable from u in W that does not adopt i. Let $(u, v_1, ..., v_k, v)$ be a path in W^E. Assume w.l.o.g. that v is the first node on the path that does not adopt i. $\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(v_k,t)$ and $\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(v,t+1)$ respectively are the itemsets adopted by v_k at time t and by v at time $t+1$. Let $J = \mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(v_k,t) \cup \mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(v,t+1)$. Clearly $i \in J$ and $J \subset \mathcal{R}^{\boldsymbol{\mathcal{S}}}_W(v,t+1)$, desire set of v at $t+1$. Notice that both $\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(v_k,t)$ and $\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(v,t+1)$ are local maximums by Lemma 2. Then by Lemma 1, J is also a local maximum, hence $util_W(J) \geq util_W(\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(v,t+1))$, as $\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(v,t+1) \subset J$. Also, $|J| > |\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(v,t+1)|$, as J contains at least one more item i. Thus as per the diffusion model v at time t should adopt the larger cardinality set J. Hence i is adopted by v. □

The *social welfare* of an allocation $\boldsymbol{\mathcal{S}}$ in a possible world $W = (W^E, W^N)$ is defined as the sum of utilities of itemsets adopted by nodes, formally, $\rho_W(\boldsymbol{\mathcal{S}}) := \sum_{v \in V} \mathcal{U}(\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(v))$. The *expected social welfare* of an allocation $\boldsymbol{\mathcal{S}}$ is $\rho(\boldsymbol{\mathcal{S}}) := \mathbb{E}_{W^E}[\mathbb{E}_{W^N}[\rho_W(\boldsymbol{\mathcal{S}})]] = \mathbb{E}_{W^N}[\mathbb{E}_{W^E}[\rho_W(\boldsymbol{\mathcal{S}})]]$. It is straightforward to show that the expected social welfare of allocation $\boldsymbol{\mathcal{S}}$ defined in Section 2.3.3 is equivalent to the above definition.

The properties of social welfare are presented next.

2.4.2 Properties of social welfare

The following theorem summarizes the property of social welfare function. The key intuition is that in each possible world, the social welfare is mono-

tone, a result proved by induction on the propagation time. However it is not submodular because the valuation is supermodular, and it is not supermodular because the propagation based on IC model would have submodular influence coverage.

Theorem 1. *Expected social welfare is monotone with respect to the sets of node-item allocation pairs. However it is neither submodular nor supermodular.*

Proof. Monotonicity is shown by induction on propagation time that the social welfare in any world W is monotone. The result follows upon taking expectation. Consider allocations $\boldsymbol{\mathcal{S}} \subseteq \boldsymbol{\mathcal{S}}'$ and any node v.
Base Case: At $t = 1$, desire happens by seeding. By assumption, $\mathbf{I}_v^{\boldsymbol{\mathcal{S}}} \subseteq \mathbf{I}_v^{\boldsymbol{\mathcal{S}}'}$. Thus, $\mathcal{R}_W^{\boldsymbol{\mathcal{S}}}(v,1) \subseteq \mathcal{R}_W^{\boldsymbol{\mathcal{S}}'}(v,1)$, where $\mathcal{R}_W^{\boldsymbol{\mathcal{S}}}(v,1)$ denotes the desire set of v in world W under allocation $\boldsymbol{\mathcal{S}}$. Suppose $J := \mathcal{A}_W^{\boldsymbol{\mathcal{S}}}(v,1) \setminus \mathcal{A}_W^{\boldsymbol{\mathcal{S}}'}(v,1)$ is non-empty. From the semantics of adoption of itemsets, $\mathcal{U}_W(J \mid \mathcal{A}_W^{\boldsymbol{\mathcal{S}}}(v,1) \setminus J) \geq 0$. Now, $\mathcal{A}_W^{\boldsymbol{\mathcal{S}}}(v,1) \setminus J \subseteq \mathcal{A}_W^{\boldsymbol{\mathcal{S}}'}(v,1)$. By supermodularity of utility, $\mathcal{U}_W(J \mid \mathcal{A}_W^{\boldsymbol{\mathcal{S}}'}(v,1)) \geq 0$. Since $J \subseteq \mathcal{A}_W^{\boldsymbol{\mathcal{S}}}(v,1) \subseteq \mathcal{R}_W^{\boldsymbol{\mathcal{S}}}(v,1) \subseteq \mathcal{R}_W^{\boldsymbol{\mathcal{S}}'}(v,1)$, by the semantics of itemset adoption, the set $J \cup \mathcal{A}_W^{\boldsymbol{\mathcal{S}}'}(v,1)$ will be adopted by v at time 1, a contradiction to the assumption that $\mathcal{A}_W^{\boldsymbol{\mathcal{S}}'}(v,1)$ is the adopted itemset by v at time 1.
Induction: By Lemma 3, it is known that once a node adopts an item, all nodes reachable from it in W^E also adopt that item. Furthermore, reachability is monotone in seed sets. From this, it follows that $\mathcal{A}_W^{\boldsymbol{\mathcal{S}}}(v, \tau + 1) \subseteq \mathcal{A}_W^{\boldsymbol{\mathcal{S}}'}(v, \tau + 1)$. Define $\mathcal{A}_W^{\boldsymbol{\mathcal{S}}}(v) := \bigcup_t \mathcal{A}_W^{\boldsymbol{\mathcal{S}}}(v,t)$. By definition, an adopted itemset has a non-negative utility, so, $\rho_W(\boldsymbol{\mathcal{S}}) = \sum_{v \in V} \mathcal{U}_W(\mathcal{A}_W^{\boldsymbol{\mathcal{S}}}(v)) \leq \sum_{v \in V} \mathcal{U}_W(\mathcal{A}_W^{\boldsymbol{\mathcal{S}}'}(v)) = \rho_W(\boldsymbol{\mathcal{S}}')$. This shows that the social welfare in any possible world is monotone, as was to be shown.

For submodularity and supermodularity, counterexamples are given. Consider a network with single node u and two items i_1 and i_2. Let $\mathcal{P}(i_1) > \mathcal{V}(i_1)$ and $\mathcal{P}(i_2) > \mathcal{V}(i_2)$. However $\mathcal{V}(\{i_1, i_2\}) > \mathcal{P}(i_1) + \mathcal{P}(i_2)$. Assume that noise terms are bounded random variables, hence, $|\mathcal{N}(i_j)| \leq |\mathcal{V}(i_j) - \mathcal{P}(i_j)|$, $j = 1, 2$. Thus expected individual utility of i_1 or i_2 is negative, but when they are offered together, the expected utility is positive. Now consider two seed allocations $\boldsymbol{\mathcal{S}} = \emptyset$ and $\boldsymbol{\mathcal{S}}' = \{(u, i_1)\}$. Let the additional allocation pair be (u, i_2). Now $\rho(\boldsymbol{\mathcal{S}} \cup \{(u, i_2)\}) - \rho(\boldsymbol{\mathcal{S}}) = 0 - 0 = 0$: for $\boldsymbol{\mathcal{S}}$, no items are adopted and for $\boldsymbol{\mathcal{S}} \cup \{(u, i_2)\}$ the noise $\mathcal{N}(i_2)$ cannot affect adoption decision in any possible world, so i_2 will not be adopted by u in any world.

However, $\rho(\boldsymbol{\mathcal{S}}' \cup \{(u, i_2)\}) - \rho(\boldsymbol{\mathcal{S}}') > 0$, as under allocation $\boldsymbol{\mathcal{S}}'$, i_1 is not adopted by u in any world, while under allocation $\boldsymbol{\mathcal{S}}' \cup \{(u, i_2)\}$, u will

adopt $\{i_1, i_2\}$ in every world, resulting in positive social welfare and breaking submodularity.

For supermodularity, consider a network consisting of two nodes v_1 and v_2 with a single directed edge from v_1 to v_2, with probability 1. Let there be one item i whose deterministic utility is positive, thus, $\mathcal{V}(i) > \mathcal{P}(i)$. Again, assume that the noise term $\mathcal{N}(i)$ is a bounded random variable, hence, $|\mathcal{N}(i)| \leq |\mathcal{V}(i) - \mathcal{P}(i)|$. Now consider two seed allocations $\boldsymbol{\mathcal{S}} = \emptyset$ and $\boldsymbol{\mathcal{S}}' = \{(v_1, i)\}$. Let the additional pair be (v_2, i). Under allocation $\boldsymbol{\mathcal{S}}'$, both nodes v_1 and v_2 will adopt i in every possible world. Hence adding the additional pair (v_2, i) does not change item adoption in any world and consequently the expected social welfare is unchanged. Thus,

$$\begin{aligned}\rho(\boldsymbol{\mathcal{S}} \cup \{(v_2, i)\}) - \rho(\boldsymbol{\mathcal{S}}) &= \mathbb{E}[\mathcal{U}(i_1)] > 0 \\ &= \rho(\boldsymbol{\mathcal{S}}' \cup \{(v_2, i)\}) - \rho(\boldsymbol{\mathcal{S}}')\end{aligned}$$

which breaks supermodularity. □

The node level adoption exhibits supermodularity because the utility function is supermodular, but the propagation behavior is governed by reachability (Lemma 3), and thus exhibits submodularity. Therefore, the combined propagation and adoption behavior in UIC exhibits a complicated behavior that is neither submodular nor supermodular. In the next section, it is shown that surprisingly, despite such complicated behavior, it is still possible to design a greedy algorithm that achieves a $(1 - 1/e - \epsilon)$-approximation to optimal expected social welfare.

2.5 Approximation Algorithm

2.5.1 Greedy algorithm overview

Given that the welfare function is neither submodular nor supermodular, designing an approximation algorithm for WelMax is challenging. Nevertheless, in this section we show that for any given $\epsilon > 0$ and number $\ell \geq 1$, a $(1 - \frac{1}{e} - \epsilon)$-approximation to the optimal social welfare can be achieved with probability at least $1 - \frac{1}{|V|^\ell}$, using a simple greedy algorithm. To the best of our knowledge, this is the first instance in the context of viral marketing where an *efficient approximation algorithm is proposed for a non-submodular objective, at the same level as submodular objectives*. We first present our algorithm and then analyze its correctness and efficiency.

Our algorithm, called bundleGRD (for bundle greedy) and shown in Algorithm 2, is based on a greedy allocation of seed nodes to items. Given a graph G, the universe of items $\mathbf{I}$, item budget vector $\vec{b}$, ϵ, and ℓ, bundleGRD first selects (line 2) the top-$\overline{b}$ seed nodes $S^{Grd} := S_{\overline{b}}$ for the IC model (disregarding item utilities), where $\overline{b} = max\{b_i \mid b_i \in \vec{b}\}$. Then, (line 4) for each item i with budget b_i, it assigns the top-b_i nodes from S^{Grd} to i. We will show that this allocation achieves a $(1 - \frac{1}{e} - \epsilon)$-approximation to the optimal expected social welfare. For this to work, the seed selection algorithm must ensure that the $\overline{b}$ seeds selected, $S_{\overline{b}}$, satisfy a *prefix-preserving* property (definition in Section 2.5.3). That is, intuitively, for every budget $b_i \in \vec{b}$, the top-b_i seeds among S^{Grd} must provide a $(1-\frac{1}{e}-\epsilon)$-approximation to the optimal expected spread under budget b_i. This property is ensured by invoking the PRIMA algorithm (Algorithm 3) in line 2 of Algorithm 2. The following is the main result for the bundleGRD algorithm.

Theorem 2. *Let $\boldsymbol{\mathcal{S}}^{Grd}$ be the greedy allocation generated by bundleGRD, and $\boldsymbol{\mathcal{S}}^{OPT}$ be the optimal allocation. Given $\epsilon > 0$ and $\ell > 0$, with probability at least $1 - \frac{1}{|V|^\ell}$, we have*

$$\rho(\boldsymbol{\mathcal{S}}^{Grd}) \geq (1 - \frac{1}{e} - \epsilon) \cdot \rho(\boldsymbol{\mathcal{S}}^{OPT}). \tag{2.3}$$

The running time is $O((\overline{b} + \ell + \log_n |\vec{b}|)(m + n) \log n/\epsilon^2)$.

We note that our bundleGRD algorithm has the interesting property that it does not need the valuation functions, prices, and the distributions of noises as input, and thus works for all possible utility settings. It reflects the power of bundling — as long as we know that all items are mutually complementary, then bundling them together as much as possible would always provide a good solution in terms of social welfare, no matter what the actual utilities. This is in stark contrast with the algorithmic solution in [97] for the complementary setting. Further, known algorithms for social welfare maximization in the combinatorial auction literature typically assume a value oracle (e.g., see [52, 81, 84]), which given a query as an itemset, returns the utility of the itemset. Works on IM for complementary items [97], require the knowledge of adoption probabilities of every item given already adopted item subsets. However, such an oracle can be quite expensive to realize in practice for non-additive utility functions, since there are exponentially many itemsets. In Section 2.5.2, we show the approximation guarantee of our algorithm through the novel block accounting method, then

Algorithm 2: bundleGRD($\mathbf{I}, \vec{b}, G, \epsilon, \ell$)

$\boldsymbol{\mathcal{S}}^{Grd} \leftarrow \emptyset$;
$S^{Grd} \leftarrow$ PRIMA($\vec{b}, G, \epsilon, \ell$)
for $i \in \mathbf{I}$ **do**
  Assign item i to the first b_i nodes of the ranked set S^{Grd}, i.e., $S_i^{Grd} \leftarrow$ top b_i nodes from S^{Grd}
  $\boldsymbol{\mathcal{S}}^{Grd} \leftarrow \boldsymbol{\mathcal{S}}^{Grd} \cup (S_i^{Grd} \times \{i\})$
return $\boldsymbol{\mathcal{S}}^{Grd}$ as the final allocation

in Section 2.5.3 we describe the prefix preserving influence maximization algorithm PRIMA. Algorithm 3 is described and its correctness and running time complexity are established in Section 2.5.3.

2.5.2 Block accounting to analyze bundleGRD

The analysis of the algorithm is highly non-trivial, because it needs to consider all possible seed allocations, propagation scenarios, with budgets possibly being non-uniform among items. Our main idea is a "block" based accounting method: we break the set of items into a sequence of "atomic" blocks, such that each block has non-negative marginal utility given previous blocks, and it can be counted as an atomic unit in the diffusion process. Then we account for each block's contribution to the social welfare during a propagation, and argue that for every block, the contribution of the block achieved by the greedy allocation is always at least $(1 - 1/e - \epsilon)$ times the contribution under any allocation. In Section 2.5.2.1 we first introduce the block generation process. Then using block based accounting, in Section 2.5.2.2 we establish the welfare produced by bundleGRD, and later in Section 2.5.2.3, show an upper bound on the welfare produced by any arbitrary allocation. The technical subtlety includes properly defining the blocks, showing why each block can be accounted for as an atomic unit separately, dealing with partial item propagation within blocks, etc.

In the rest of the analysis, we fix the noise world W^N, and prove that $\rho_{W^N}(\boldsymbol{\mathcal{S}}^{Grd}) \geq (1 - \frac{1}{e} - \epsilon) \cdot \rho_{W^N}(\boldsymbol{\mathcal{S}}^{OPT})$, where ρ_{W^N} denotes the expected social welfare under the fixed noise world W^N. We could then simply take another expectation over the distribution of W^N to obtain Inequality (2.3). Let $\mathcal{U}_{W^N}$ be the utility function under the noise possible world W^N.

Given W^N, let $\mathbf{I}^*_{W^N} \subseteq \mathbf{I}$ be the subset of items that gives the largest utility in W^N, with ties broken in favor of larger sets. By Lemma 1, $\mathbf{I}^*_{W^N}$

is unique. This implies that the marginal utility of any (non-empty) subset of $\mathbf{I} \setminus \mathbf{I}^*_{W^N}$ given $\mathbf{I}^*_{W^N}$ is strictly negative. Further recall that $\mathcal{U}_{W^N}$ is supermodular. Hence the marginal utility of any subset of $\mathbf{I} \setminus \mathbf{I}^*_{W^N}$ given any subset of $\mathbf{I}^*_{W^N}$ is strictly negative, which means no items in $\mathbf{I} \setminus \mathbf{I}^*_{W^N}$ can ever be adopted by any user under the noise world W^N. Thus, once we fix W^N, we can safely remove all items in $\mathbf{I} \setminus \mathbf{I}^*_{W^N}$ from consideration. In the rest of Section 2.5.2, for simplicity we use $\mathbf{I}^*$ as a shorthand for $\mathbf{I}^*_{W^N}$.

2.5.2.1 Block generation process

We divide items in $\mathbf{I}^*$ into a sequence of disjoint blocks such that each block has a non-negative marginal utility w.r.t. the union of all its preceding blocks. We also need to carefully arrange items according to their budgets for later accounting analysis. We next discuss how the blocks are generated.

Let $\mathbf{I}^* = \{i_1, ..., i_{|\mathbf{I}^*|}\}$. We order the items in non-increasing order of their budgets, which is, $b_1 \geq b_2 \geq \cdots \geq b_{|\mathbf{I}^*|}$.

Figure 2.5 shows the process of generating the blocks. Note that this *block generation process is solely used for our accounting analysis and is not part of our seed allocation algorithm. Hence it has no impact on the running time whatsoever* Given $\mathbf{I}^*$ and W^N, we first generate a global sequence $\mathcal{I}$ of all non-empty subsets of $\mathbf{I}^*$, following a precedence order $\prec$ (Step **2**), explained next.

For any two distinct subsets $S, S' \subseteq \mathbf{I}^*$, arrange items in each of S, S' in decreasing order of item indices. Compare items in S, S', starting from the highest indexed items of S and S'. If they match then compare the second highest indexed items and so on until one of the following rules applies:
1. One of S or S' exhausts. If say S exhausts first, then $S \prec S'$.
2. The current pair of items in S and S' do not match. Then $S \prec S'$, if the current item of S has a lower index than the current item of S'.

We illustrate this step using the following example.

Example 6 (Generation of $\mathcal{I}$)**.** Suppose we have three items $\mathbf{I}^* = \{i_1, i_2, i_3\}$ with $b_1 \geq b_2 \geq b_3$, then we order the subsets in the following way: $\mathcal{I} = (\{i_1\}$, $\{i_2\}$, $\{i_1, i_2\}$, $\{i_3\}$, $\{i_1, i_3\}$, $\{i_2, i_3\}$, $\{i_1, i_2, i_3\})$. Between subsets $\{i_3\}$ and $\{i_1, i_3\}$, $\{i_3\}$ is ordered first according to rule 1, whereas between $\{i_1, i_2\}$ and $\{i_3\}$, $\{i_1, i_2\}$ is ordered first according to rule 2. □

The sequence $\mathcal{I}$ has the following useful property:

Property 1. *For any subsets S and T in the sequence $\mathcal{I}$, if (a) T is a proper subset of S, or (b) the highest index among all items in T is strictly lower than the highest index among all items in S, then T appears before S in $\mathcal{I}$.*

1. Input for the process contains $\mathbf{I}^*$ and W^N.

2. Generate the $2^{|\mathbf{I}^*|} - 1$ non-empty subsets of $\mathbf{I}^*$
Sort the subsets following the precendence order $\prec$. Put the sorted subsets in sequence $\mathcal{I}$
$\mathcal{B} \leftarrow \emptyset$; $B \leftarrow$ the first entry in $\mathcal{I}$

3. Repeat the following steps until $\mathcal{I}$ is empty
(1) *If* $\mathcal{U}_{W^N}(B \mid \bigcup \mathcal{B}) \geq 0$ then,
$\mathcal{B} \leftarrow \mathcal{B} \oplus B$ i.e., append B at the end of sequence $\mathcal{B}$
remove all sets B' from $\mathcal{I}$ with $B' \cap B \neq \emptyset$
$B \leftarrow$ the first entry in $\mathcal{I}$

(2) *Else* $B \leftarrow$ the next entry in $\mathcal{I}$ after B

4. $\mathcal{B}$ is the final sequence of blocks

Figure 2.5: The block generation process

From $\mathcal{I}$, blocks are selected following an iterative process, as shown in Step **3** of Figure 2.5. We scan through this sequence, with the purpose of generating a sequence $\mathcal{B}$ of disjoint blocks. For each subset B being scanned, if its marginal utility given all previously selected blocks is non-negative, thus, $\mathcal{U}_{W^N}(B \mid \bigcup \mathcal{B}) \geq 0$, where $\mathcal{B}$ is the currently selected sequence of blocks, and $\bigcup \mathcal{B}$ is the union of all items in these selected blocks, then we append B to the end of selected sequence $\mathcal{B}$, thus, $\mathcal{B} = \mathcal{B} \oplus B$, where $\oplus$ denotes "append". After selecting B, we remove all subsets in $\mathcal{I}$ that overlap with B, and restart the scan from the beginning of the remaining sequence. If $\mathcal{U}_{W^N}(B \mid \bigcup \mathcal{B}) < 0$, then we skip this set and go to the next one.

Example 7 illustrates the process.

Example 7 (Block generation)**.** Continuing from Example 6, assume the

following utility assignments for noise world W^N:

$$\mathcal{U}_{W^N}(i_1) = \mathcal{U}_{W^N}(i_2) = \mathcal{U}_{W^N}(i_3) = \mathcal{U}_{W^N}(i_1, i_2) = -1$$
$$\mathcal{U}_{W^N}(i_1, i_3) = \mathcal{U}_{W^N}(i_2, i_3) = 1;\ \mathcal{U}_{W^N}(i_1, i_2, i_3) = 4$$

Then as per the block generation process, $\{i_1, i_3\}$ will be chosen as the first block B_1, since it is the first block in $\mathcal{I}$ with non-negative marginal utility w.r.t. $\emptyset$. Once B_1 is chosen all itemsets containing i_1 or i_3 are deleted from $\mathcal{I}$, thus only $\{i_2\}$ remains in $\mathcal{I}$. Since $\mathcal{U}_{W^N}(\{i_2\} \mid \{i_1, i_3\}) = \mathcal{U}_{W^N}(i_1, i_2, i_3) - \mathcal{U}_{W^N}(i_1, i_3) = 4 - 1 > 0$, $\{i_2\}$ is chosen as B_2 and the process terminates with $\mathcal{B} = (\{i_1, i_3\}, \{i_2\})$. □

By the fact that $\mathbf{I}^*$ is a local maximum, it is easy to see that the blocks generated form a partition of $\mathbf{I}^*$. Let $\mathcal{B} = \{B_1, B_2, \ldots, B_t\}$ be the sequence of blocks generated, where t is the number of blocks in the block partition. We define the marginal gain of each block B_i as

$$\Delta_i = \mathcal{U}_{W^N}(B_i \mid \cup_{j=1}^{i-1} B_j). \tag{2.4}$$

We have the following properties regarding the marginal gains.

Property 2. *$\forall i \in [t]$, $\Delta_i \geq 0$, and $\mathcal{U}_{W^N}(\mathbf{I}^*) = \sum_{i=1}^{t} \Delta_i$.*

Let $A \subseteq \mathbf{I}^*$ be an arbitrary subset of items. We partition A based on block partition $\mathcal{B}$: Define $A_i = A \cap B_i, \forall i \in [t]$. If $A_i = B_i$, we call A_i a full block, if $A_i = \emptyset$, then it is an empty block, otherwise, we call it a partial block. Define $\Delta_i^A = \mathcal{U}_{W^N}(A_i \mid A_1 \cup \ldots \cup A_{i-1})$. By Property 1 and the fact that B_i is the first block in $\mathcal{I}$ with non-negative marginal utility w.r.t. $\bigcup_{j=1}^{i-1} B_j$, it follows that

Property 3. *$\forall i \in [t]$, $\Delta_i^A \leq \Delta_i$, and $\mathcal{U}_{W^N}(A) = \sum_{i=1}^{t} \Delta_i^A$.*

Using this property, we devise our accounting where each A_i contributes Δ_i^A in its social welfare.

2.5.2.2 Social welfare under greedy allocation

We are now ready to analyze the social welfare of our greedy allocation (Algorithm 2) using block accounting. We first show that, before the propagation starts, each seed node would adopt exactly the prefix of full blocks allocated until the first non-full block, and then show that all these adopted full blocks will propagate together, so we can exactly account for the contribution of each block to the expected social welfare. The following lemma gives the exact statement of the first part.

Lemma 4. *Under the greedy allocation, suppose that at a seed node v, A_i is the first non-full block assigned to v, then before the propagation starts, v adopts exactly $B_1 \cup ... \cup B_{i-1}$.*

Proof. This proof relies on the supermodularity of $\mathcal{U}_{W^N}$, the block generation process, the greedy allocation procedure, and Property 3. Let M be the set of items adopted by v before the propagation starts, and let $M_1 = M \cap (B_1 \cup ... \cup B_{i-1})$ and $M_2 = M \setminus M_1$. Since A_i is a partial block, we know that $M_2 \neq B_i$. We first show that $M_2 = \emptyset$ and then $M_1 = B_1 \cup ... \cup B_{i-1}$.

Suppose, for a contradiction, that $M_2 \neq \emptyset$. We know that $\mathcal{U}_{W^N}(M_2 \mid M_1) \geq 0$, and by supermodularity $\mathcal{U}_{W^N}(M_2 \mid B_1 \cup ... \cup B_{i-1}) \geq 0$. If M_2 is ordered before B_i in sequence $\mathcal{I}$, then M_2 should be selected instead of B_i, a contradiction. If M_2 is ordered after B_i in $\mathcal{I}$, by the block generation process we can conclude that all items in B_i have budgets no less than the minimum budget for items in M_2, which by greedy allocation implies that all items in B_i should be allocated to v, contradicting the fact A_i is a partial block. Thus $M_2 = \emptyset$ and $M = M_1 \subseteq B_1 \cup ... \cup B_{i-1}$.

Next, by Property 3,

$$\mathcal{U}_{W^N}(M) = \sum_{j=1}^{i-1} \Delta_i^M \leq \sum_{j=1}^{i-1} \Delta_i = \mathcal{U}_{W^N}(B_1 \cup ... \cup B_{i-1})$$

Thus v should adopt $B_1 \cup ... \cup B_{i-1}$ instead of M. □

Effective budget of blocks. For a block B_i, we define its *effective budget* $e_i = \min_{j \in B_1 \cup \cdots \cup B_i} b_j$. In bundleGRD (Algorithm 2), the first e_i seed nodes of S^{Grd} are assigned all the full blocks $\{B_1 \cup ... \cup B_i\}$. By Lemma 4, only those nodes actually adopt the block B_i before the propagation starts. Such seed nodes are called *effective seed nodes* of block B_i and denoted as $S_{B_i}^{GrdE}$. Thus in summary, under the greedy allocation, before the propagation starts, all seed nodes in $S_{B_i}^{GrdE}$ adopt B_i together with $B_1, \ldots, B_{i-1}$, and none of the seed nodes outside $S_{B_i}^{GrdE}$ adopts any items in $B_i, B_{i+1}, \ldots, B_t$.

As established earlier that the nodes in $S_{B_i}^{GrdE}$ always adopt B_i together with $B_1, \ldots, B_{i-1}$ and without considering the effect of propagation, no other seed nodes outside the set $S_{B_i}^{GrdE}$ adopts B_i or any other blocks $B_{i+1}, \ldots, B_t$. B_i is not adopted because at least one of the previous $B_1, ..., B_{i-1}$ blocks is not allocated to those nodes. Also since B_i is not adopted, none of the subsequent blocks can be adopted. We illustrate this using an example next.

Example 8 (Block budgets)**.** Revisit the blocks shown in Example 7. Let us assume that $b_1 > b_2 > b_3$. Recall that $B_1 = \{i_1, i_3\}$ and $B_2 = \{i_2\}$. Let S_2, S_3 be the top b_2, b_3 nodes in the greedy allocation respectively, and $S_3 \subset S_2$. Then under the greedy allocation, B_2 as a full block will be allocated to nodes in S_2. The effective budget of B_2 is $e_2 = \min_{j \in B_1 \cup B_2} b_j = b_3$. The effective seed set of B_2 is $S_{B_2}^{GrdE} = S_3$, since nodes in S_3 are allocated both B_1 and B_2 and will adopt both B_1 and B_2 according to Lemma 4 (can also be verified by checking the utility settings given in Example 7 manually). For nodes in $S_2 \setminus S_3$, even though they are allocated the full block B_2, they are only allocated a partial block $A_1 = \{i_1\}$, and thus by Lemma 4 they will not adopt B_2 or A_1. □

We are now ready to show the social welfare of the allocation made by bundleGRD.

Lemma 5. *Let $\boldsymbol{\mathcal{S}}^{Grd}$ be the greedy allocation obtained using Algorithm 2. Then the expected social welfare of the allocation $\boldsymbol{\mathcal{S}}^{Grd}$ in W^N is $\rho_{W^N}(\boldsymbol{\mathcal{S}}^{Grd}) = \sum_{i \in [t]} \sigma(S_{B_i}^{GrdE}) \cdot \Delta_i$, where $S_{B_i}^{GrdE}$ are the effective seed nodes of block B_i under allocation $\boldsymbol{\mathcal{S}}^{Grd}$, $\sigma(\cdot)$ is the expected spread function under the IC model, and Δ_i is as defined in Eq.* (2.4).

Proof. To account for the effect of propagation, we use the Reachability Lemma (Lemma 3). By that lemma, nodes reachable from $S_{B_i}^{GrdE}$ adopt all the blocks $B_1, ..., B_i$. For a full block B_i only the effective seeds of B_i and nodes reachable from them adopt B_i. Thus the expected number of nodes that are reached by block B_i and consequently adopt B_i, is $\sigma(S_{B_i}^{GrdE})$. From Property 2, adoption of every such B_i contributes Δ_i to the overall social welfare. Moreover, the only item adoptions are disjoint union of full blocks. Hence $\rho_{W^N}(\boldsymbol{\mathcal{S}}^{Grd}) = \sum_{i \in [t]} \sigma(S_{B_i}^{GrdE}) \cdot \Delta_i$. □

2.5.2.3 Social welfare under an arbitrary allocation

Unlike greedy, in an arbitrary allocation, for the effective seed nodes, we cannot conclude that a block B_i is offered with all previous full blocks $B_1, \ldots, B_{i-1}$. Thus our accounting method needs to be adjusted. Our idea is to define the key concept of an *anchor item* a_i for every block B_i, which appears in $B_1 \cup \cdots \cup B_i$. We want to show that only when B_i is co-adopted with a_i by any node, B_i could contribute positive marginal social welfare (Lemma 6), and in this case its marginal contribution is upper bounded by Δ_i (Property 3). Hence we only need to track the diffusion of the anchor item a_i to account for the marginal contribution of B_i. Finally by showing

that the budget of a_i is exactly the effective budget $e_i = |S_{B_i}^{GrdE}|$ of B_i, we conclude that $\sigma(S_{a_i}) \leq (1 - 1/e - \epsilon)\ \sigma(S_{B_i}^{GrdE})$ by the prefix preserving property explained in Section 2.5.1.

We define the budget of a block to be the minimum budget of any item in the block. Then the *anchor block* B_i^a, of a block B_i is the block from $B_1, \ldots, B_i$ that has the minimum budget. In case of a tie, the block having highest index is chosen as the anchor block. Notice that anchor item a_i is the highest indexed and consequently minimum budgeted item in its corresponding anchor block B_i^a. Notice that, by definition, if block B_j is the anchor block of block B_i with $j < i$, then block B_j is also the anchor block for all blocks $B_j, B_{j+1}, \ldots, B_i$. Moreover, the effective budget e_i of a block B_i, is the budget of its anchor item a_i, which is, the minimum budget of all items in $B_1 \cup \cdots \cup B_i$. We illustrate the concept of anchor block and item using the example below.

Example 9 (Anchor block and item)**.** Anchor block of block B_2 in Example 8, is $B_2^a = B_1$. Its corresponding anchor item a_2 is the highest indexed item of block B_2^a, which is, i_3. Block B_1's anchor block is the block itself and consequently its anchor item a_1 is again i_3. □

Lemma 6. *Let a_i be the anchor item of B_i, and suppose a_i appears in B_j, $j \leq i$. During the diffusion process from an arbitrary seed allocation $\boldsymbol{\mathcal{S}}$, let A be the set of items in $B_j \cup \ldots \cup B_i$ that have been adopted by v by time t. If $a_i \notin A$ and $A \neq \emptyset$, then $\mathcal{U}_{W^N}(A \mid B_1 \ldots, B_{j-1}) < 0$.*

Proof. Suppose that $\mathcal{U}_{W^N}(A \mid B_1 \cup \cdots \cup B_{j-1}) \geq 0$. By the definition of the anchor item, we know that all items in $A \setminus B_j$ have strictly larger budget than the budget of a_i, otherwise one of items in $A \setminus B_j$ should be the anchor item for B_i. This means all items in $A \setminus B_j$ have index strictly lower than a_i. Notice $a_i \notin A$, and thus all items in $A \cap B_j$ also have index strictly lower than a_i. Then by Property 1, A should appear before B_j in sequence $\mathcal{I}$. Since $\mathcal{U}_{W^N}(A \mid B_1 \ldots, B_{j-1}) \geq 0$, the block generation process should select A as the j-th block instead of the current B_j, a contradiction. □

Using the above result, we establish the following lemma, which upper bounds the welfare produced by an arbitrary allocation.

Lemma 7. *For any arbitrary seed allocation $\boldsymbol{\mathcal{S}}$, the expected social welfare in W^N is $\rho_{W^N}(\boldsymbol{\mathcal{S}}) \leq \sum_{i \in [t]} \sigma(S_{a_i}) \cdot \Delta_i$, where S_{a_i} is the seed set assigned to the anchor item a_i of block B_i, and Δ_i is as defined in Eq.* (2.4).

Proof. For an edge possible world W^E, suppose that after the diffusion process under W^E, every node v adopts item set A_v. Let $A_{v,i} = A_v \cap B_i$ for all $i \in [t]$, and $\Delta_i^{A_v} = \mathcal{U}_{W^N}(A_{v,i} \mid A_{v,1} \cup \ldots \cup A_{v,i-1})$. Thus, we have

$$\begin{aligned}\rho_{W^N}(\boldsymbol{\mathcal{S}}) &= \mathbb{E}_{W^E}\left[\sum_{v\in V} \mathcal{U}_{W^N}(A_v)\right] = \mathbb{E}_{W^E}\left[\sum_{v\in V}\sum_{i\in[t]} \Delta_i^{A_v}\right] \\ &= \sum_{i\in[t]} \mathbb{E}_{W^E}\left[\sum_{v\in V} \Delta_i^{A_v}\right], \end{aligned} \tag{2.5}$$

where the expectation is taken over the randomness of the edge possible worlds, and thus we use subscript W^E under the expectation sign to make it explicit. By switching the summation signs and the expectation sign in the last equality above, we show that the expected social welfare can be accounted as the summation among all blocks B_i of the expected marginal gain of block B_i on all nodes. We next bound $\mathbb{E}_{W^E}\left[\sum_{v\in V} \Delta_i^{A_v}\right]$ for each block B_i.

Under the edge possible world W^E, for each $v \in V$, there are three possible cases for $A_{v,i}$. In the first case, $A_{v,i} = \emptyset$. In this case, $\Delta_i^{A_v} = 0$, so we do not need to count the marginal gain $\Delta_i^{A_v}$. In the second case, $A_{v,i}$ is not empty but it does not co-occur with block B_i's anchor a_i, that is $a_i \notin A_v$, and $A_{v,i} \neq \emptyset$. In this case, Let $A' = A \cap (B_j \cup \ldots \cup B_i)$, where B_j is the anchor block of B_i. Then A' is not empty and we know $\mathcal{U}_{W^N}(A' \mid B_1 \cup \ldots \cup B_{j-1}) < 0$. Since we have $\mathcal{U}_{W^N}(A' \mid B_1 \cup \ldots \cup B_{j-1}) = \sum_{j'=j}^{i} \Delta_{j'}^{A_v}$. Thus the cumulative marginal gain of $\Delta_{j'}^{A_v}$ with $j \leq j' \leq i$ is negative, so we can relax them to 0, effectively not counting the marginal gain of $\Delta_i^{A_v}$ either.

Finally, $A_{v,i}$ is non-empty and co-occur with its anchor a_i, hence $a_i \in A$ and $A_{v,i} \neq \emptyset$. Since A_v is a partial block, $\Delta_i^{A_v} \leq \Delta_i$, we relax $\Delta_i^{A_v}$ to Δ_i. This relaxation occurs only on nodes that adopt a_i. A node v could adopt a_i only when there is a path in W^E from a seed node that adopts a_i to node v. As defined in the lemma, S_{a_i} is the set of seed nodes of a_i. Let $\Gamma(S_{a_i}, W^E)$ be the set of nodes that are reachable from S_{a_i} in W^E. Then, there are at most $|\Gamma(S_{a_i}, W^E)|$ nodes at which we relax $\Delta_i^{A_v}$ to Δ_i for block B_i. Hence,

$$\sum_{v\in V} \Delta_i^{A_v} \leq |\Gamma(S_{a_i}, W^E)| \cdot \Delta_i. \tag{2.6}$$

Furthermore, notice that $\mathbb{E}_{W^E}[|\Gamma(S_{a_i}, W^E)|] = \sigma(S_{a_i})$, by the live-edge representation of the IC model. Therefore, together with Eq. (2.5) and (2.6), we have

$$\begin{aligned}\rho_{W^N}(\boldsymbol{\mathcal{S}}) &\leq \sum_{i\in[t]} \mathbb{E}_{W^E}\left[|\Gamma(S_{a_i}, W^E)| \cdot \Delta_i\right] \\ &= \sum_{i\in[t]} \sigma(S_{a_i}) \cdot \Delta_i.\end{aligned}$$

This concludes the proof of the lemma. □

Notice in Lemma 7, $|S_{a_i}| \leq e_i$, whereas in Lemma 5 $|S^{GrdE}_{B_i}| = e_i$. Hence the combination of Lemma 5 and Lemma 7, together with the fact that $S^{GrdE}_{B_i}$ is a $(1-1/e-\epsilon)$-approximation of the optimal solution with e_i seeds (by the prefix-preserving property), leads to the approximation guarantee of bundleGRD (Eq. (2.3) of Theorem 2), which we prove next.

Theorem 3. (Correctness of bundleGRD) *Let $\boldsymbol{\mathcal{S}}^{Grd}$ be the greedy allocation and $\boldsymbol{\mathcal{S}}$ be any arbitrary allocation. Given $\epsilon > 0$ and $\ell > 0$, the expected social welfare $\rho(\boldsymbol{\mathcal{S}}^{Grd}) \geq (1 - \frac{1}{e} - \epsilon) \cdot \rho(\boldsymbol{\mathcal{S}})$ with at least $1 - \frac{1}{|V|^\ell}$ probability.*

Proof. From Lemma 5 , we have for a possible world $W^N = (W^E, W^N)$, $\rho_{W^N}(\boldsymbol{\mathcal{S}}^{Grd}) = \sum_{i\in[t]} \sigma(S^{GrdE}_{B_i}) \cdot \Delta_i$, where the size of $S^{GrdE}_{B_i}$ is the effective budget of B_i.

For an arbitrary allocation $\boldsymbol{\mathcal{S}}$, since a_i is the anchor item of B_i, by its definition we know that $|S_{a_i}| = |S^{GrdE}_{B_i}|$. By the correctness of the prefix-preserve influence maximization algorithm we use in line 2 (Definition 1, to be instantiated in Section 2.5.3), we have that with probability at least $1 - \frac{1}{|V|^\ell}$, $\sigma(S^{GrdE}_{B_i}) \geq (1-\frac{1}{e}-\epsilon)\sigma(S_{a_i})$, for all blocks B_i's and their corresponding anchors a_i's.

Let the distribution of world W^N be $\mathcal{D}^N$. Then, together with Lemma 7,

we have that with probability at least $1 - \frac{1}{|V|^\ell}$,

$$\begin{aligned}
\rho(\boldsymbol{\mathcal{S}}^{Grd}) &= \mathbb{E}_{W^N \sim \mathcal{D}^N}[\rho_{W^N}(\boldsymbol{\mathcal{S}}^{Grd})] \\
&= \mathbb{E}_{W^N \sim \mathcal{D}^N}\left[\sum_{i \in [t]} \sigma(S_{B_i}^{GrdE}) \cdot \Delta_i\right] \\
&\geq \mathbb{E}_{W^N \sim \mathcal{D}^N}\left[\sum_{i \in [t]} (1 - \frac{1}{e} - \epsilon)\sigma(S_{a_i}) \cdot \Delta_i\right] \\
&\geq (1 - \frac{1}{e} - \epsilon)\mathbb{E}_{W^N \sim \mathcal{D}^N}[\rho_{W^N}(\boldsymbol{\mathcal{S}})] \\
&= (1 - \frac{1}{e} - \epsilon)\rho(\boldsymbol{\mathcal{S}}).
\end{aligned}$$

Therefore, the theorem holds. □

In the following section, we explain the component PRIMA that provides the prefix preserving property.

2.5.3 Item-wise prefix preserving IMM

We first formally define the prefix-preserving property.

Definition 1. (Prefix-Preserving Property). *Given $G = (V, E, p)$ and budget vector $\vec{b}$, an influence maximization algorithm $\mathbb{A}$ is said to be prefix-preserving w.r.t. $\vec{b}$, if for any $\epsilon > 0$ and $\ell > 0$, $\mathbb{A}$ returns an ordered set $S_{\overline{b}}^{Grd}$ of size $\overline{b}$, such that with probability at least $1 - \frac{1}{|V|^\ell}$, for every $b_i \in \vec{b}$, the top-b_i nodes of $S_{\overline{b}}^{Grd}$, denoted $S_{b_i}^{Grd}$, satisfies $\sigma(S_{b_i}^{Grd}) \geq (1 - \frac{1}{e} - \epsilon)$ OPT_{b_i}, where OPT_{b_i} is the optimal expected spread of b_i nodes.*

Unfortunately, state-of-the-art IM algorithms such as IMM [127], SSA [110], and OPIM [126] are not prefix-preserving out-of-the-box. In this section, we present a non-trivial extension of IMM, called PRIMA (PRefix preserving IM Algorithm) (Algorithm 3), to make it prefix-preserving. The classical models of influence propagation assume a single item and IMM is one of the state of the art algorithms for influence maximization. For a single item, as well as for multiple items with uniform budgets, the prefix property is trivial. In the presence of multiple items with non-uniform budgets, an algorithm that returns a seed set of high quality with only a probabilistic guarantee need *not* satisfy the prefix preserving property (Definition 1). We present PRIMA (*PRefix* IMM), shown in Algorithm 3, which

is a prefix-preserving extension of IMM for multiple items. Notice that *NodeSelection*$(\mathcal{R}, k)$ is the standard greedy algorithm for finding a seed set of size k by solving max k-cover on the set of RR sets $\mathcal{R}$. For more details, the reader is referred to [127]. The *NodeSelection* algorithm used in PRIMA is same as Alg 1 of IMM, which we donot repeat for brevity.

State-of-the-art IM algorithms including IMM use reverse influence sampling (RIS) approach [22] governed by reverse-reachable (RR) sets. An RR set is a random set of nodes sampled from the graph by (a) first selecting a node v uniformly at random from the graph, and (b) then simulating the reverse propagation of the model (e.g., IC model) and adding all visited nodes into the RR set. The main property of a random RR set R is that: influence spread $\sigma(S) = n \cdot \mathbb{E}[\mathbb{I}\{S \cap R \neq \emptyset\}]$ for any seed set S, where $\mathbb{I}$ is the indicator function. After finding large enough number of RR sets, the original influence maximization problem is turned into a k-max coverage problem – finding the set of k nodes that covers the most number of RR sets, where a set S covers an RR set R if $S \cap R \neq \emptyset$. All RIS algorithms use the same well-known coverage procedure, denoted as *NodeSelection*$(\mathcal{R}, k)$ in [22], and thus we omit its description here. These algorithms mainly differ in estimating the number of RR sets needed for the approximation guarantee. The number of RR sets generated by these algorithms is in general not monotone with the budget k, making them not prefix preserving. Our PRIMA algorithm carefully addresses this issue, even with nonuniform item budgets, while keeping the efficiency of the algorithm.

PRIMA ingests four inputs, namely the budget vector $\vec{b}$, graph G, ϵ and ℓ, with $\vec{b}$ sorted in non-increasing order as stated in Definition 1. Given ℓ, for a budget k, IMM generates a set of RR sets $\mathcal{R}$, such that $|\mathcal{R}| \geq \lambda_k^*/OPT_k$ with probability at least $1 - 1/n^\ell$. PRIMA derives a number $\ell' > \ell$ as a function of ℓ (Algorithm 3, line 2), the details of which we provide in Lemma 9. Before that, we briefly describe PRIMA. Extending the bounding technique of [127], for each budget k, we set

$$\lambda'_k = \frac{(2 + \frac{2}{3}\epsilon') \cdot (\log \binom{n}{k} + \ell' \cdot \log\ n + \log\log_2\ n) \cdot n}{\epsilon'^2}, \tag{2.7}$$

$$\lambda_k^* = 2n \cdot ((1 - 1/e) \cdot \alpha + \beta_k)^2 \cdot \epsilon^{-2}, \tag{2.8}$$

where, $\alpha = \sqrt{\ell' \log n + \log 2}$ is a constant independent of k, and $\beta_k = \sqrt{(1 - 1/e) \cdot (\log \binom{n}{k} + \ell' \log\ n + \log 2)}$. Note that we use log without a base to represent the natural logarithm.

Algorithm 3: PRIMA $(\vec{b}, G, \epsilon, \ell)$

Initialize $\mathcal{R} = \emptyset$, $s = 1$, $n = |V|$, $i = 1$, $\epsilon' = \sqrt{2} \cdot \epsilon$, *budgetSwitch* = **false**;
$\ell = \ell + \log 2 / \log n$, $\ell' = \log_n(n^\ell \cdot |\vec{b}|)$;
while $i \leq \log_2(n) - 1$ *and* $s \leq |\vec{b}|$ **do**
    $k = b_s$, $LB = 1$;
    $x = \frac{n}{2^i}$; $\theta_i = \lambda'_k / x$, where λ'_k is defined in Eq. (2.7);
    while $|\mathcal{R}| \leq \theta_i$ **do**
        Generate an RR set for a randomly selected node v of G and insert in $\mathcal{R}$;
    if *budgetSwitch* **then**
        S_k = the first k nodes in the ordered set $S_{b_{s-1}}$ returned from the previous call to *NodeSelection*
    else
        $S_k = \mathit{NodeSelection}(\mathcal{R}, k)$
    if $n \cdot F_{\mathcal{R}}(S_k) \geq (1 + \epsilon') \cdot x$ **then**
        $LB = n \cdot F_{\mathcal{R}(S_k)} / (1 + \epsilon')$;
        $\theta_k = \lambda^*_k / LB$, where λ^*_k is defined in Eq. (2.8);
        while $|\mathcal{R}| < \theta_k$ **do**
            Generate an RR set for a randomly selected node v of G and insert in $\mathcal{R}$;
        $s = s + 1$; *budgetSwitch* = **true**
    else
        $i = i + 1$; *budgetSwitch* = **false**
if $s \leq |\vec{b}|$ **then**
    $\theta_k = \lambda^*_{b_s} / LB$;
$\mathcal{R} = \emptyset$;
while $|\mathcal{R}| < \theta_k$ **do**
    Generate an RR set for a randomly selected node v of G and insert in $\mathcal{R}$;
$S_{\overline{b}} = \mathit{NodeSelection}(\mathcal{R}, \overline{b})$;
return $S_{\overline{b}}$ as the final seed set;

The basic idea of PRIMA is to generate enough RR sets such that for any budget $k \in \vec{b}$, $|\mathcal{R}| \geq \lambda^*_k / OPT_k$, with probability at least $1 - 1/n^{\ell'}$. Since OPT_k is unknown, we rely on a good lower bound of OPT_k, i.e., LB_k, as proposed in IMM [127]. Specifically PRIMA starts from the highest budget,

i.e., b_1. For a given budget $k \in \vec{b}$ and i it samples enough RR sets into $\mathcal{R}$ first (lines 6-7) and then checks the coverage condition on the sampled set of RR sets (line 12). Note if $\mathcal{R}$ already had enough number of RR sets (generated at a previous budget), then it skips RR set generation and moves directly to coverage check. If the coverage condition succeeds, then a good LB for the budget k is determined. It uses the LB to find the required number of RR sets (lines 14-16) for k and moves to the next budget. It then reuses the prefix of the ordered seed set found for budget k as the seed set found for the new budget, avoiding a redundant call to the *NodeSelection* procedure. This is fine because *NodeSelection* is a deterministic greedy procedure in finding seed nodes, and the last call to *NodeSelection* before the budget switch, is using the same RR set collection $\mathcal{R}$ with a larger budget, and thus it already found all the seed nodes for the new budget. If the coverage condition fails, it increments i to sample more RR sets for the current budget k (line 19).

If for any budget, all possible i values are tested, PRIMA breaks the for-loop and generates RR sets (for that budget) using $LB = 1$ (line 21), which is the lowest possible value of LB. Further, since budgets are sorted in non-increasing order and λ^*_k is monotone in k (Eq. (2.8)), there cannot be any remaining budget k', where $k' \leq k$, for which $\lambda^*_{k'}/LB$ (line 21) is higher. Hence the RR set generation process terminates.

Lastly, after determining $|\mathcal{R}|$, those many RR sets are generated from scratch (line 23) on which the final *NodeSelection* is invoked. This addresses a recently found issue of the original IMM algorithm [33]. PRIMA then returns the top-$\bar{b}$ seeds obtained from *NodeSelection* (line 25).

The correctness and the running time of the PRIMA algorithm mainly follow the proof of the IMM algorithm [33]. We first show the correctness and towards that we prove that the following lemma holds.

Lemma 8. *Let $\mathcal{R}$ be the final set of RR sets generated by PRIMA at the end and let $k \in \vec{b}$ be any budget. Then $|\mathcal{R}| \geq \lambda^*_k / OPT_k$ holds with probability at least $1 - 1/n^{\ell'}$.*

Proof. Given $x \in [1, n]$, ϵ' and $\delta_3 \in (0, 1)$ and a budget k. Let S_k be the seed set of size k obtained by invoking *NodeSelection*$(\mathcal{R}, k)$, where,

$$|\mathcal{R}| \geq \frac{(2 + \frac{2}{3}\epsilon') \cdot (\log \binom{n}{k} + \log(1/\delta_3))}{\epsilon'} \cdot \frac{n}{x}. \tag{2.9}$$

Then, from Lemma 6 of [127], if $OPT_k < x$, then $n \cdot F_{\mathcal{R}}(S_k) < (1+\epsilon') \cdot x$ with probability at least $(1 - \delta_3)$. Now let $j = \lceil \log_2 \frac{n}{OPT_k} \rceil$. By union bound, we can infer that PRIMA has probability at most $(j-1)/(n^{\ell'} \cdot \log_2 n)$ to satisfy

the coverage condition of line 12 for the budget k. Then by Lemma 7 of [127] and the union bound, PRIMA will satisfy $LB_k \leq OPT_k$ with probability at least $1 - n^{\ell'}$. We know that for any $k \in \vec{b}$, $|\mathcal{R}| \geq \lambda_k^*/LB_k$, hence the lemma follows. □

We are now ready to prove the correctness of PRIMA.

Lemma 9. *PRIMA returns a prefix preserving* $(1 - 1/e - \epsilon)$*-approximate solution* $S_{\overline{b}}$ *to the optimal expected spread, with probability at least* $1 - 1/n^{\ell}$.

Proof. We know from Lemma 8 that the RR set sampling for any budget can result in the coverage condition (Algorithm 3, line 12) failing with probability at most $1/n^{\ell'}$. By applying union bound over all the budgets, we have that the failure probability of the coverage condition in PRIMA is at most $\sum_{k \in \vec{b}} 1/n^{\ell'} = |\vec{b}| \cdot 1/n^{\ell'}$. By setting $\ell' = \log_n(n^{\ell} \cdot |\vec{b}|)$, we bound this failure probability to at most $1/n^{\ell}$. Thus ℓ' is used for computing α and β_k in Eq. (2.8). Further once θ_k is determined, we generate those many RR set from scratch. This follows the fix proposed in [33]. Without the fix, the top $S_{\overline{b}}$ nodes returned by the last call to *NodeSelection* (line 25), cannot be shown to have a $(1-1/e-\epsilon)$-approximate solution with probability at least $1 - 1/n^{\ell}$. For every budget $b_i \in \vec{b}$, we can then choose the prefix of top-b_i nodes of $S_{\overline{b}}$ and use that as a solution S_{b_i} for that budget, with the guarantee that with probability at least $1-1/n^{\ell}$ each S_{b_i} is a $(1-1/e-\epsilon)$-approximate solution to OPT_{b_i}.

By union bound, PRIMA returns a $(1 - 1/e - \epsilon)$-approximate prefix preserving solution with probability at least $1 - 2/n^{\ell}$.

Finally by increasing ℓ to $\ell + \log 2/\log n$ in line 2, we raise PRIMA's probability of success to $1 - 1/n^{\ell}$. □

Running time

The running time of PRIMA essentially involves two parts: the time needed to generate the set of RR sets $\mathcal{R}$ and the total time of all *NodeSelection* invocations. From Lemma 9 of [127], we have for any budget k, the set of RR sets generated for that budget $\mathcal{R}_k$ satisfies,

$$\mathbb{E}[|\mathcal{R}_k|] \leq \frac{3max\{\lambda_k^*, \lambda_k'\} \cdot (1+\epsilon')^2}{(1-1/e) \cdot OPT_k}.$$

Since λ_k' and λ_k^* are both monotone in k (Eq. ((2.7)) and ((2.8))), we know their maximums are achieved for $k = \overline{b}$.

Further let $OPT_{min} := OPT_{b_{|\mathbf{I}|}}$ be the minimum expected spread, i.e., minimum value of OPT, across all budgets, then for any $\mathcal{R}_k$,

$$\begin{aligned}\mathbb{E}[|\mathcal{R}_k|] &\leq \frac{3max\{\lambda^*_{\bar{b}}, \lambda'_{\bar{b}}\} \cdot (1+\epsilon')^2}{(1-1/e) \cdot OPT_{min}} \\ &= O((\bar{b}+\ell')n \log\ n \cdot \epsilon^{-2}/OPT_{min}).\end{aligned}$$

Further since PRIMA reuses the RR sets instead of generating them from scratch for every budget, for the RR set $\mathcal{R}$ generated by PRIMA,

$$\begin{aligned}\mathbb{E}[|\mathcal{R}|] &= max_{k\in\vec{b}}\{\mathbb{E}[|\mathcal{R}_k|]\} \\ &= O((\bar{b}+\ell')n \log\ n \cdot \epsilon^{-2}/OPT_{min}).\end{aligned} \tag{2.10}$$

For an RR set $R \in \mathcal{R}$, let $w(R)$ denote the number of edges in G pointing to nodes in R. If EPT is the expected value of $w(R)$, then we know, $n \cdot EPT \leq m \cdot OPT_{min}$. Hence using Eq. (2.10), the expected total time to generate $\mathcal{R}$ is determined by,

$$\begin{aligned}\mathbb{E}[\sum_{R\in\mathcal{R}} w(R)] &= \mathbb{E}[|\mathcal{R}|] \cdot EPT \\ &= O((\bar{b}+\ell')(n+m) \log\ n \cdot \epsilon^{-2}).\end{aligned} \tag{2.11}$$

Notice that generating RR set from scratch for the final node selection, following the fix of [33], only adds a multiplicative factor of 2. Hence the overall asymptotic running time to generate $\mathcal{R}$ remains unaffected. Thus intuitively, there are two changes in PRIMA's running time. The budget k of a single item of IMM is replaced with $\bar{b}$, the maximum budget of any item. Secondly, by applying union bound on every individual item's failure probability, a factor of $\log_n |\vec{b}|$ is added to the sample complexity. Using Lemma 9 and Eq. (2.11) we now prove the correctness and the running time result of PRIMA.

Theorem 4. *PRIMA is prefix preserving and it returns a* $(1-1/e-\epsilon)$*-approximate solution to IM with at least* $1-1/n^\ell$ *probability in* $O((\bar{b}+\ell+\log_n|\vec{b}|)(n+m)\log\ n\cdot\epsilon^{-2})$ *expected time.*

Proof. From Lemma 9, we have that PRIMA returns a prefix preserving $(1-1/e-\epsilon)$-approximate solution with at least $1-1/n^\ell$ probability. In that process PRIMA invokes *NodeSelection*, $\log_2\ n-1$ times in the while loop and once to find the final seed set $S_{\bar{b}}$. Note that, we intentionally avoid

	Flixster	Douban-Book	Douban-Movie	Twitter	Orkut
# nodes	7.6K	23.3K	34.9K	41.7M	3.07M
# edges	71.7K	141K	274K	1.47G	234M
avg. degree	9.43	6.5	7.9	70.5	77.5
type	undirected	directed	directed	directed	undirected

Table 2.3: Network Statistics

redundant calls to *NodeSelection* when we switch budgets, which saves $|\vec{b}|$ additional calls to *NodeSelection*.

Let $\mathcal{R}_i$ be the susbset of $\mathcal{R}$ used in the i-th iteration of the loop. Since *NodeSelection* involves one pass over all RR set, on a given input $\mathcal{R}_i$, it takes $O(\sum_{R\in\mathcal{R}_i}|R|)$ time. Recall $|\mathcal{R}_i|$ doubles with every increment of i. Hence it is a geometric sequence with a common ratio of 2. Now from Theorem 3 of [127] and the fact that there is no additional calls to *NodeSelection* during budget switch, we have total cost of invoking all *NodeSelection* is $O(\mathbb{E}[\sum_{R\in\mathcal{R}}|R|])$.

Since $|R| \leq w(R)$, for any $R \in \mathcal{R}$, then using Eq. (2.11) we have,

$$\begin{aligned} O(\mathbb{E}[\sum_{R\in\mathcal{R}}|R|]) &= O(\mathbb{E}[\sum_{R\in\mathcal{R}}w(R)]) \\ &= O((\bar{b}+\ell')(n+m)\log\ n\cdot\epsilon^{-2}) \\ &= O((\bar{b}+\ell+\log_n|\vec{b}|)(n+m)\log\ n\cdot\epsilon^{-2}). \end{aligned}$$

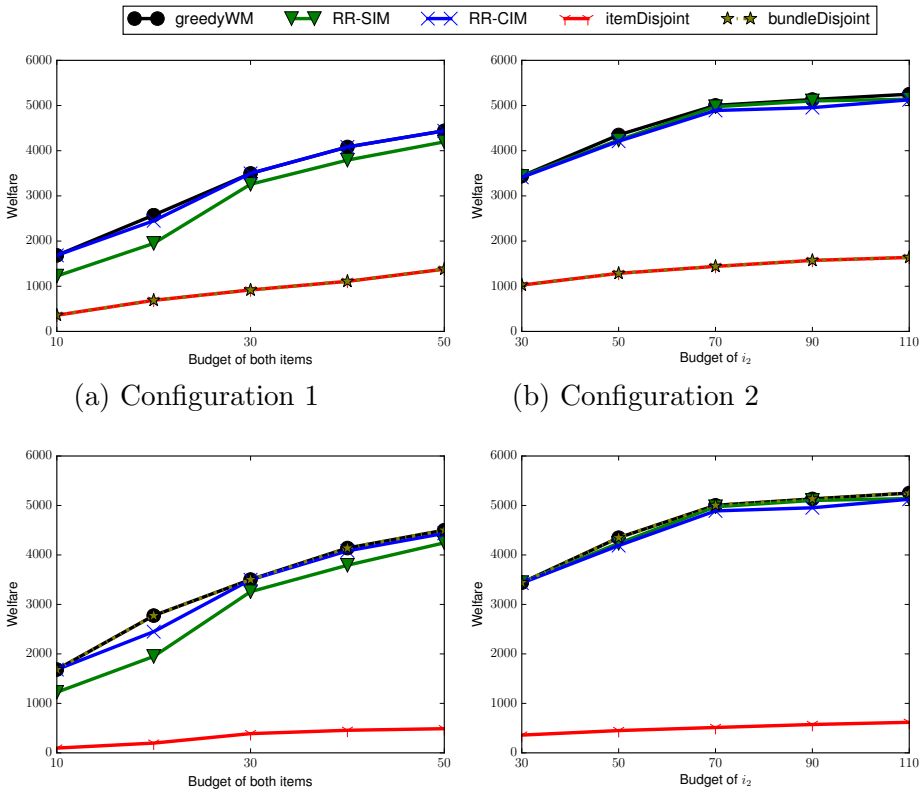
Hence the theorem follows. □

Finally, the combination of Theorems 3 and 4 gives our main Theorem 2.

2.6 Experiments

2.6.1 Experiment Setup

We perform extensive experiments on five real social networks. We first experiment with synthetic utility (value and price) functions. For real utility functions, we learn the value and noise distributions of items from the bidding data in eBay, and obtain item prices from Craigslist and Facebook groups to make them compatible with used items auctioned in eBay. All experiments are performed on a Linux machine with Intel Xeon 2.6 GHz CPU and 128 GB RAM.



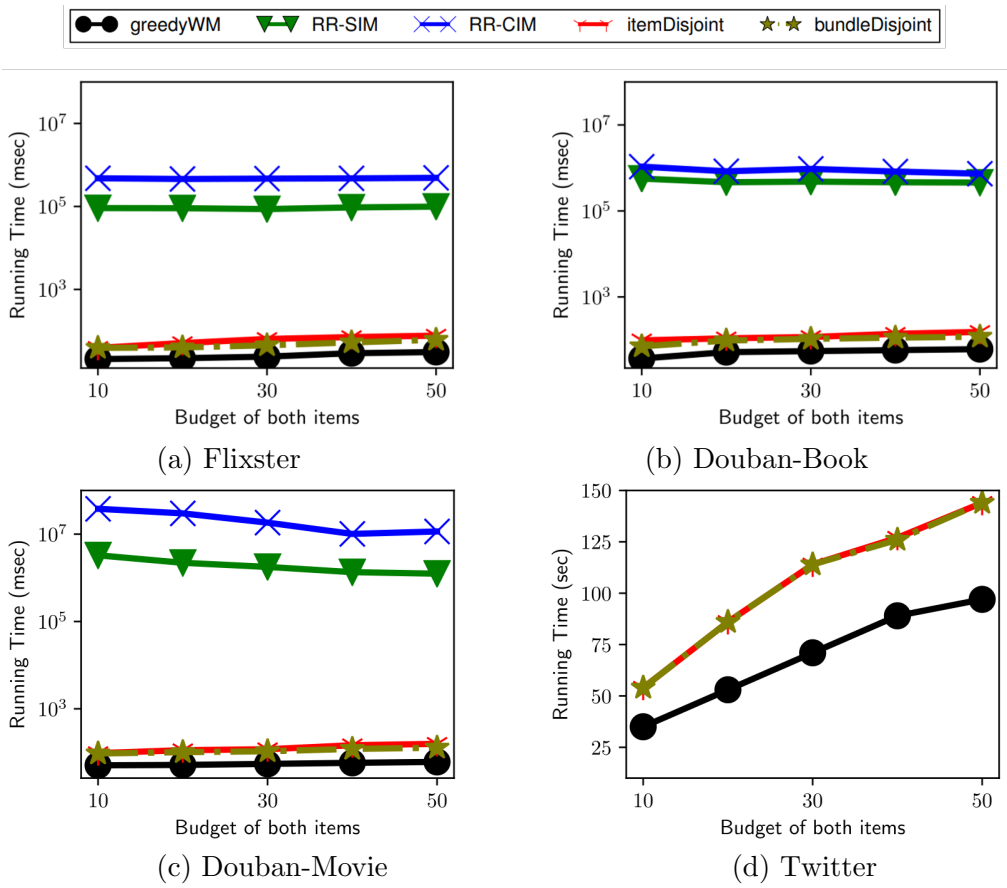
(a) Configuration 1 (b) Configuration 2

(c) Configuration 3 (d) Configuration 4

Figure 2.6: Expected social welfare in four configurations (on the Douban-Movie network)

2.6.1.1 Networks

Table 2.3 summarizes the networks used in the experiments and the characteristics of the networks. Flixster is mined in [97] from a social movie site and a strongly connected component is extracted. Douban is a Chinese social network, where users rate books, movies, music, etc. In [97] all movie and book ratings of the users in the graph are crawled separately to derive two datasets from book and movie ratings: Douban-Book and Douban-Movie. Twitter is one of the largest public network datasets. Finally Orkut is a large social network that we use to test scalability. Both Twitter and Orkut can be obtained from [123].



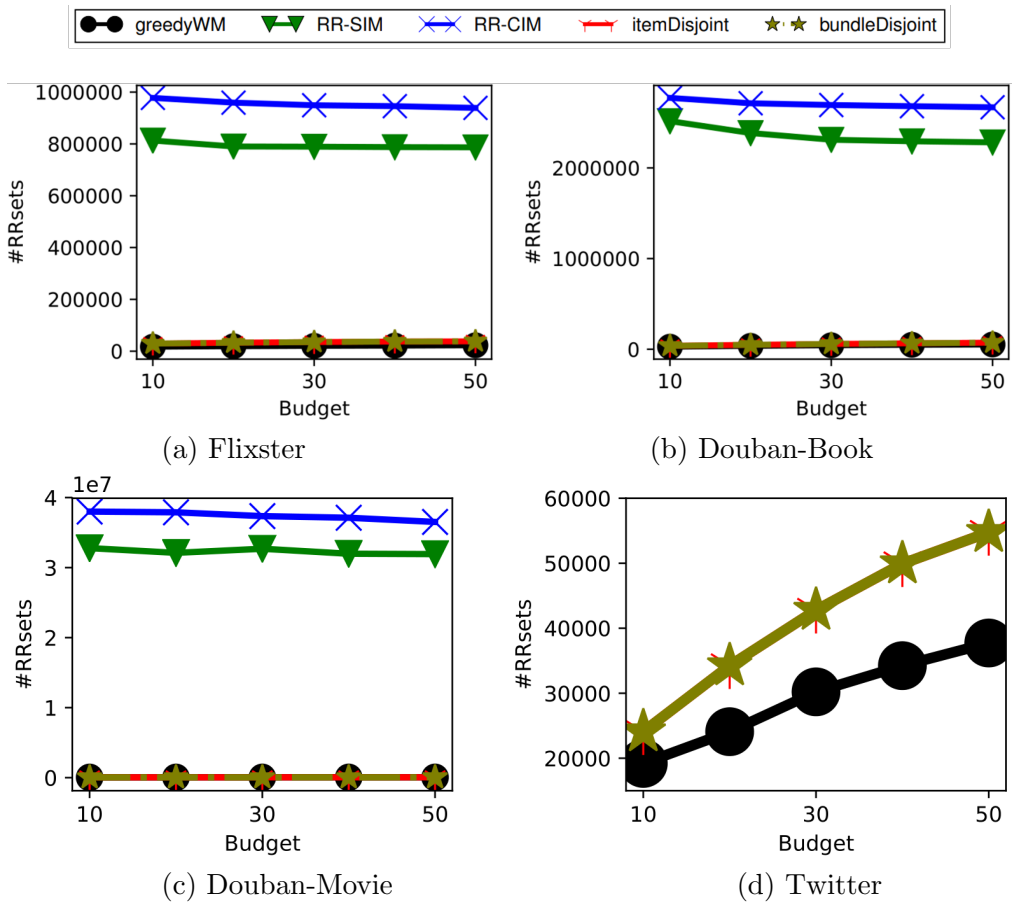
(a) Flixster (b) Douban-Book

(c) Douban-Movie (d) Twitter

Figure 2.7: Running times of bundleGRD, RR-SIM$^+$, RR-CIM, item-disj and bundle-disj (on Configuration 1)

2.6.1.2 Algorithms compared

We compare bundleGRD against six baselines – RR-SIM$^+$, RR-CIM, item-disj, bundle-disj, BDHS-Concave and BDHS-Step. RR-SIM$^+$ and RR-CIM are two state-of-the art algorithms designed for complementary products in the context of IM [97]. However, they work only for two items. Extending the Com-IC framework and the RR-SIM$^+$ and RR-CIM algorithms for more than two items is highly non-trivial as that requires dealing with automata with exponentially many states. Hence in comparing the performance of bundleGRD against RR-SIM$^+$ and RR-CIM, we limit the number of items to two. Later we experiment with more than two items. Below, by deterministic utility of an itemset I, we mean $\mathcal{V}(I) - \mathcal{P}(I)$, i.e., its utility with the noise term



(a) Flixster (b) Douban-Book

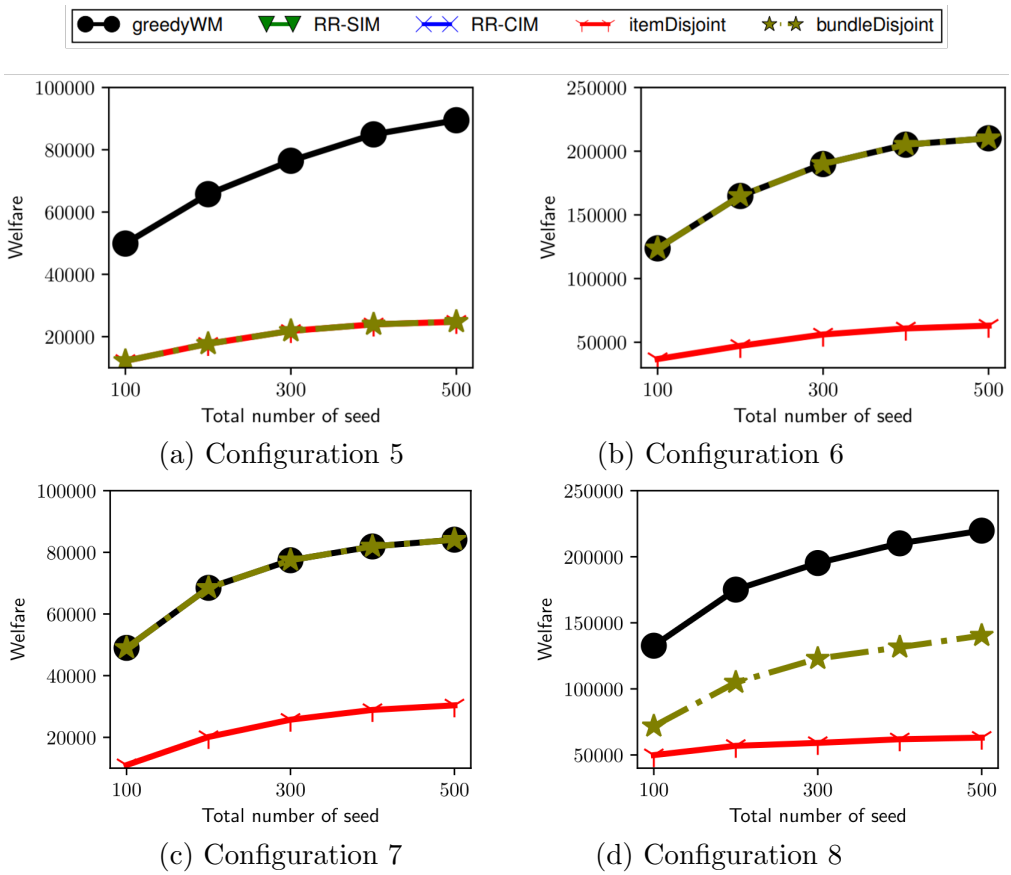
(c) Douban-Movie (d) Twitter

Figure 2.8: Number of RR sets generated by bundleGRD, RR-SIM$^+$, RR-CIM, item-disj and bundle-disj (on Configuration 1)

ignored.

1. **Com-IC baselines.** For two items i_1 and i_2, given seed set of item i_2 (resp. i_1), RR-SIM$^+$ (resp. RR-CIM) finds seed set of item i_1 (resp. i_2) such that expected number of adoptions of i_1 is maximized. Initial seeds of i_2 (resp. i_1) are chosen using IMM [127].

2. **Item-disjoint.** Our next baseline item-disj allocates only one item to every seed node. Given the set of items $\mathbf{I}$, item-disj finds $\sum_{i \in \mathbf{I}} b_i$ nodes, say L, using IMM, where b_i is the budget of item i. Then it visits items in L in non-increasing order of budgets, assigns item i to first b_i nodes and removes those b_i nodes from L. By explicitly assigning



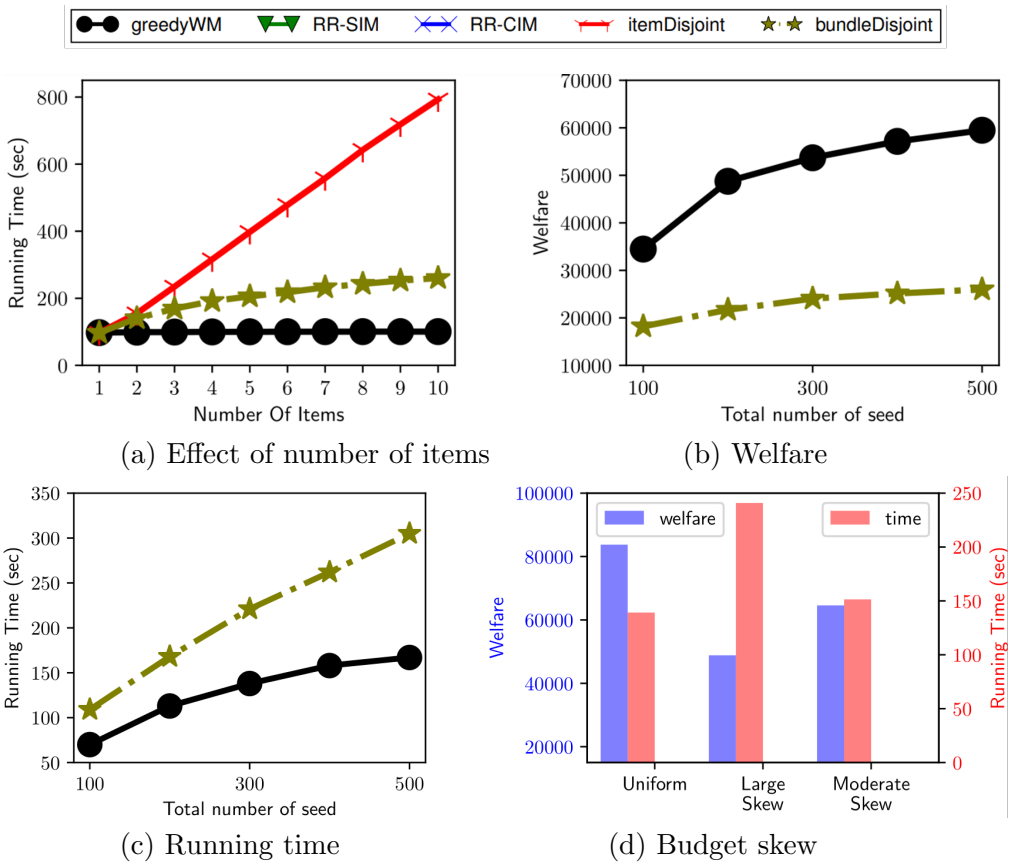
(a) Configuration 5 (b) Configuration 6

(c) Configuration 7 (d) Configuration 8

Figure 2.9: Expected social welfare in four configurations (on the Twitter network)

every item to different seeds, item-disj does not leverage the effect of supermodularity. However it benefits from the *network propagation*: since the utilities are supermodular, if more neighbors of a node adopt some item, it is more likely that the node will also adopt an item. Thus, when individual items have positive utility and hence can be adopted and propagate on their own, by choosing more seeds, item-disj makes use of the network propagation to encourage more adoptions.

3. **Bundle-disjoint.** Baseline bundle-disj, aims to leverage both supermodularity and network propagation. It first orders the items $\mathbf{I}$ in non-increasing budget order and determines successively minimum sized subsets with non-negative deterministic utility, maintaining these sub-



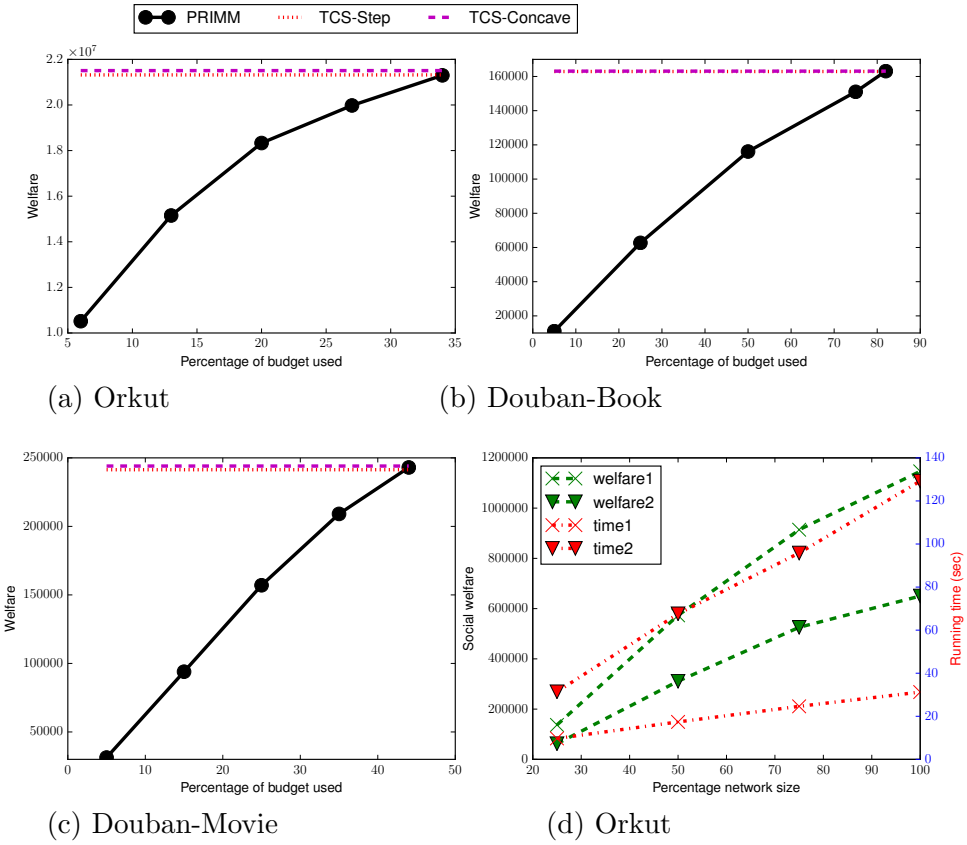
(a) Effect of number of items (b) Welfare

(c) Running time (d) Budget skew

Figure 2.10: (a) Impact of number of items on the running time and (b-d) Experiments using real Param (on the Twitter network)

sets ("bundles") in a list. Items in each bundle B are allocated to a new set of $b_B := min\{b_i \mid i \in B\}$ seed nodes. The budget of each item in B is decremented by b_B, and items with budget 0 are removed. When no more bundles can be found, we revisit each item i with a positive unused budget and repeatedly allocate it to the seeds of the first existing bundle B which does not contain i. If $b_B > b_i$ (where b_i is the current budget of i after all deductions), then the first b_i seeds from the seed set of B are assigned to i. If an item i still has a surplus budget, we select b_i fresh seeds using IMM and assign them to i.

4. **Welfare maximization baselines.** Our last two baselines, BDHS-Concave and BDHS-Step are two state-of-the-art welfare maximization algo-



(a) Orkut (b) Douban-Book

(c) Douban-Movie (d) Orkut

Figure 2.11: (a-c) Comparison against BDHS algorithms and (d) Scalability of bundleGRD

rithms under network externalities [19]. As discussed in Section 2.2, their study has significant differences from our study, but we still make an empirical comparison with their algorithms with the goal to explore what fraction of the budget is needed by our model with network propagation to achieve the same social welfare as their model which has network externality but no network propagation. We defer the details of the comparison method to Section 2.6.4.

2.6.1.3 Default Parameters

Following previous works [74, 110] we set probability of edge $e = (u, v)$ to $1/d_{in}(v)$. Unless otherwise specified, we use $\epsilon = 0.5$ and $\ell = 1$ as our de-

No	Price	Value	Noise	GAP	Budget
1	$i_1 = 3$	$i_1 = 3, i_2 = 4$	$i_1 : N(0,1)$	$q_{i_1\|\emptyset} = 0.5$, $q_{i_2\|\emptyset} = 0.5$	Uniform
2	$i_2 = 4$	$\{i_1, i_2\} = 8$	$i_2 : N(0,1)$	$q_{i_1\|i_2} = 0.84, q_{i_2\|i_1} = 0.84$	Nonuniform
3	$\{i_1, i_2\} = 7$	$i_1 = 3, i_2 = 3$	$\{i_1, i_2\} : N(0,2)$	$q_{i_1\|\emptyset} = 0.5$, $q_{i_2\|\emptyset} = 0.16$	Uniform
4		$\{i_1, i_2\} = 8$		$q_{i_1\|i_2} = 0.98, q_{i_2\|i_1} = 0.84$	Nonuniform

Table 2.4: Two item configurations

fault for all five methods as recommended in [128]. The Com-IC algorithms RR-SIM$^+$ and RR-CIM use adoption probabilities, called GAP parameters [97], to model the interaction between items. The GAP parameters can be simulated within the UIC framework using utilities shown in Eq. (2.12). The derivation follows simple algebra. Here, $q_{i_1|\emptyset}$ (resp., $q_{i_1|i_2}$) denotes the probability that a user adopts item i_1 given that it has adopted nothing (resp., item i_2).

Let i_1 and i_2 be the two items. Suppose the desire set of a node u only has item i_1. The condition that u adopts i_1 is $\mathcal{V}(i_1) - \mathcal{P}(i_1) + \mathcal{N}(i_1) \geq 0$. Thus the GAP parameter $q_{i_1|\emptyset}$ is given by:

$$q_{i_1|\emptyset} = \Pr[\mathcal{V}(i_1) - \mathcal{P}(i_1) + \mathcal{N}(i_1) \geq 0] = \Pr[\mathcal{N}(i_1) \geq \mathcal{P}(i_1) - \mathcal{V}(i_1)].$$

Now suppose i_1 has been adopted by u, and i_2 enters the desire set. The GAP parameter $q_{i_2|i_1}$ is the probability of adopting i_2 given that i_1 has been adopted. So we have

$$\begin{aligned} q_{i_2|i_1} &= \Pr[\mathcal{V}(\{i_1, i_2\}) - \mathcal{P}(i_1) - \mathcal{P}(i_2) + \mathcal{N}(i_1) + \mathcal{N}(i_2) \geq \\ &\qquad \mathcal{V}(i_1) - \mathcal{P}(i_1) + \mathcal{N}(i_1) \mid \mathcal{N}(i_1) \geq \mathcal{P}(i_1) - \mathcal{V}(i_1)] \\ &= \Pr[\mathcal{N}(i_2) \geq \mathcal{P}(i_2) - (\mathcal{V}(\{i_1, i_2\}) - \mathcal{V}(i_1)) \mid \mathcal{N}(i_1) \geq \\ &\quad \mathcal{P}(i_1) - \mathcal{V}(i_1)]. \end{aligned}$$

Since noise $\mathcal{N}(i_2)$ is independent of noise $\mathcal{N}(i_1)$, we can remove the above condition in the conditional probability, and obtain

$$q_{i_2|i_1} = \Pr\{\mathcal{N}(i_2) \geq \mathcal{P}(i_2) - (\mathcal{V}(\{i_1, i_2\}) - \mathcal{V}(i_1))\}.$$

The other two GAP parameters, $q_{i_2|\emptyset}$ and $q_{i_1|i_2}$ can be obtained similarly. To summarize, we have

$$\begin{aligned} q_{i_1|\emptyset} &= \Pr[\mathcal{N}(i_1) \geq \mathcal{P}(i_1) - \mathcal{V}(i_1)], \\ q_{i_1|i_2} &= \Pr[\mathcal{N}(i_1) \geq \mathcal{P}(i_1) - (\mathcal{V}(\{i_1, i_2\}) - \mathcal{V}(i_2))], \\ q_{i_2|\emptyset} &= \Pr[\mathcal{N}(i_2) \geq \mathcal{P}(i_2) - \mathcal{V}(i_2)], \\ q_{i_2|i_1} &= \Pr[\mathcal{N}(i_2) \geq \mathcal{P}(i_2) - (\mathcal{V}(\{i_1, i_2\}) - \mathcal{V}(i_1))]. \end{aligned} \tag{2.12}$$

2.6.2 Experiments on two items

We explore four different configurations corresponding to the choice of the values, prices, noise distribution parameters, and item budgets (see Table 2.4). While UIC does not assume any specific distribution for noise, in our experiments we use a Gaussian distribution for illustration.

In Configurations 1 and 2, individual items have non-negative deterministic utility. In this setting item-disj and bundle-disj are equivalent. In Configurations 3 and 4 one item has a negative deterministic utility while the other item has a non-negative one. In this setting, however, bundleGRD and bundle-disj are equivalent. One may also consider configurations where every individual item has negative deterministic utility. In such a setting, item-disj produces 0 welfare, which makes the comparison degenerates.

For every parameter setting, we consider two budget settings, namely *uniform* (e.g., Configuration 1) and *non-uniform* (resp. Configuration 2). In case of uniform budget, both items have the same budget k, where k is varied from 10 to 50 in steps of 10. For non-uniform budget, i_1's budget is fixed at 70, and i_2's budget is varied from 30 to 110 in steps of 20.

2.6.2.1 Social Welfare

We compare the expected social welfare achieved by all algorithms on all four configurations (Fig. 2.6). We show the results only for Douban-Movie, since the trend of the results is similar on other networks. In terms of social welfare, bundleGRD achieves an expected social welfare upto 5 times higher than item-disj (Fig. 2.6(d)). A similar remark applies when bundle-disj and bundleGRD are not equivalent (e.g., Fig. 2.6(b)). Further, notice that RR-SIM$^+$ and RR-CIM produce welfare similar to bundleGRD. It follows from Table 4 of [97] (full arxiv version) that under this configuration, RR-SIM$^+$ and RR-CIM end up copying the seeds of the other item. Hence their allocations are similar to bundleGRD. However, as shown next, bundleGRD is much more efficient than the other two algorithms, and easily supports more than two items, which makes bundleGRD more suitable in practice for multiple items over large networks.

2.6.2.2 Running time

We study the running time of all algorithms using Configuration 1 as a representative case. The results are shown in Fig. 2.7. As can be seen, bundleGRD and bundle-disj are equivalent and hence have the same running time. However, bundleGRD significantly outperforms all other baselines on

every dataset. RR-SIM$^+$ and RR-CIM are particularly slow. In fact, on the large Twitter network, they could not finish even after our timeout after 6 hours (hence they are omitted from Fig. 2.7(d)). In comparison with the baselines, bundleGRD is upto 5 orders of magnitude (resp. 1.5 times) faster than RR-CIM (resp. item-disj). Running times on other configurations show a similar trend, and are omitted.

2.6.2.3 Memory

Lastly we study the memory required by all algorithms using Configuration 1. The results are shown in Fig. 2.8. Since the amount of memory required is directly related to the number of RR sets each algorithm produces, we show the RR set numbers in the plots. RR-SIM$^+$ and RR-CIM are based on TIM, whereas the other three algorithms leverage IMM, which generates much less number of RR sets than TIM. Further for the comic algorithms, the two separate pass involving forward and backward simulations also results in more RR sets generation.

2.6.3 More than two items

We use the largest dataset Twitter for tests in this subsection.

2.6.3.1 The configurations

Having established the superiority of bundleGRD for two items, we now consider more than two items. Recall that RR-SIM$^+$ and RR-CIM cannot work with more than two items, so we confine our comparison to item-disj and bundle-disj. We gauge the performance of the algorithms on social welfare and running time. We also study the effect of budget distribution on social welfare. We design four configurations corresponding to the choice of budget and utility (see Table 2.5). For all configurations, we sample noise terms from $N(0, 1)$. Price and value are set in such a way as to achieve certain shapes for the set of itemsets in the lattice that have a positive utility (see below).

- **Configurations 5-7.** Configuration 5 is the simplest: every item has the same budget; price and value are set such that every item has the same utility of 1 and utility is additive. Thus, by design, this configuration gives minimal advantage to any algorithm that tries to leverage supermodularity. The next two configurations (6 and 7) model the situation where a single "core" item is necessary in order

No	Value	Budget
5	Additive	Uniform
6	Cone-max	Non-uniform
7	Cone-min	Non-uniform
8	Level-wise	Uniform

Table 2.5: Multiple item configurations

to make an itemset's utility positive. E.g., a smartphone may be a core item, without which its accessories do not have a positive utility. We set the core item's utility to 5. The addition of any other item increases the utility by 2. Thus, all supersets of the core item have a positive utility, while all other subsets have a negative utility. Hence, the set of subsets with positive utility forms a "cone" in the itemset lattice. In Configuration 6 (resp. 7), the core item is the item with maximum (resp. minimum) budget. Finally, we design a more general configuration where the set of itemsets with positive utility forms an arbitrary shape (see Configuration 8 below).

- **Configuration 8.** We consider the itemset lattice, with level t having subsets of size t. We randomly set the prices and values of items in level 1 such that a random subset of items have a non-negative utility. Let A_t be any itemset at level $t > 1$ and $i \in A_t$ any item. We choose a value uniformly at random, $\epsilon \sim U[1, 5]$, and define

$$\mathcal{V}(i|A_t \setminus \{i\}) = max_{B \in \mathbb{P}(A_t \setminus \{i\}, t-2)} \{\mathcal{V}(i|B) + \epsilon\} \tag{2.13}$$

where $\mathbb{P}(A, q)$ denotes the set of subsets of A of size q. That is, the marginal gain of an item i w.r.t. $A_t \setminus \{i\}$ is set to be the maximum marginal gain of i w.r.t. subsets of A_t of size $t-2$, plus a randomly chosen boost (ϵ). E.g., let $A_4 = \{i, j, k, l\}$, $t = 4$ then, $\mathcal{V}(i|\{j, k, l\}) = max\{\mathcal{V}(i|\{j, k\}), \mathcal{V}(i|\{k, l\}), \mathcal{V}(i|\{j, l\})\} + \epsilon$.

Recall that the value computation proceeds level-wise starting from level $t = 0$. Thus, for any itemset A_t in Eq.(2.13), $\mathcal{V}(i|B)$ for subsets B is already defined.

Finally, we set $\mathcal{V}(A_t) = max_{i \in A_t} \{\mathcal{V}(A_t \setminus \{i\}) + \mathcal{V}(i|A_t \setminus \{i\})\}$. Now we show that this way of assigning values ensures that the value function is well-defined and supermodular.

Lemma 10. *The value function of Configuration 8 is supermodular.*

Proof. First we show that for an itemset A_t at level t, and an item $i \notin A_t$, $\mathcal{V}(i \mid A_t) \geq \mathcal{V}(i \mid B)$, where $B \subset A_t$ is any subset of A_t. We prove this claim by induction on level.
Base Case: Let $t = 1$ and A_1 be any singleton itemset. Then $\mathcal{V}(i \mid A_1) = \mathcal{V}(i \mid \emptyset) + \epsilon \geq \mathcal{V}(i \mid \emptyset)$.
Induction: Suppose the claim is true for all levels $t \leq l$. We show it holds for $t = l + 1$. From our method of assigning values we have, $\mathcal{V}(i \mid A_{l+1}) = \max_{B_l \in \mathbb{P}(\mathbf{I},l)}\{\mathcal{V}(i \mid B_l)\} + \epsilon$, where $\mathbb{P}(\mathbf{I}, l)$ is the set of all itemsets at level l. Thus $\mathcal{V}(i \mid A_{l+1}) \geq \mathcal{V}(i \mid B_l)$. By induction hypothesis, $\mathcal{V}(i \mid B_l) \geq \mathcal{V}(i \mid B)$, for any subset $B \subset B_l$, and thus $\mathcal{V}(i \mid A_{l+1}) \geq \mathcal{V}(i \mid B)$.

It then follows that for any itemsets $B \subset A \subset \mathbf{I}$ and item $i \in \mathbf{I} \setminus A$, $\mathcal{V}(i \mid A) \geq \mathcal{V}(i \mid B)$. Hence value is supermodular. □

Lemma 11. *The value function of Configuration 8 is well defined.*

Proof. We show that for an itemset A_t at level t, $\mathcal{V}(i \mid A_t \setminus \{i\}) + \mathcal{V}(i) = \mathcal{V}(j \mid A_t \setminus \{j\}) + \mathcal{V}(j)$, for any $i, j \in A_t$.

Let $\mathcal{V}(A_t) = max_{k \in A_t}\{\mathcal{V}(A_t \setminus \{k\}) + \mathcal{V}(k | A_t \setminus \{k\})\} = m$. Then according to our configuration $\mathcal{V}(i \mid A_t \setminus \{i\}) = m - \mathcal{V}(i)$. Similarly $\mathcal{V}(j \mid A_t \setminus \{j\}) = m - \mathcal{V}(j)$. Hence $\mathcal{V}(i \mid A_t \setminus \{i\}) + \mathcal{V}(i) = m - \mathcal{V}(i) + \mathcal{V}(i) = m - \mathcal{V}(j) + \mathcal{V}(j) = \mathcal{V}(j \mid A_t \setminus \{j\}) + \mathcal{V}(j)$. □

2.6.3.2 Social welfare

First, we study the social welfare achieved by the algorithms, in each of the above configurations, with the total budget varying from 500 to 1000 in steps of 100. For Configurations 7 and 10, we set the budget uniformly for every item. For other configurations, the max budget is set to 20% of the total budget, min budget to 2%, and the remaining budget is split uniformly. The results of the experiment on Twitter network are shown in Fig. 2.9. Under Configurations 8 and 9, bundleGRD and bundle-disj produces the same allocation, hence the welfare is the same. However in general bundleGRD outperforms every baseline in all the four configurations by producing welfare up to 4 times higher than baselines.

2.6.3.3 Running time vs number of items

Next, we study the effect of the number of items on the running time of the algorithms. For this experiment, we use Configuration 5. We set the budget of every item to $k = 50$ and vary the number of items s, from 1 to 10. Fig. 2.10(a) shows the running times on the Twitter dataset. As the number of

items increases the number of seed nodes to be selected for item-disj and bundle-disj increases. Notice both item-disj and bundle-disj select the same number of seeds, which is $k \times s$. item-disj selects it by one invocation of IMM, with budget ks, while bundle-disj invokes IMM s times with budget k for every invocation. So their overall running times differ. By contrast, the running time of bundleGRD only depends on the maximum budget and is independent of the number of items. E.g., when number of items is 10, bundleGRD is about 8 times faster than bundle-disj and 2.5 times faster than item-disj.

2.6.4 Experiment with real value, price, and noise parameters

In this section, we conduct experiments on parameters (value, price, and noise) learned from real data. We consider the following 5 items: (1) Playstation 4, 500 GB console, denoted ps, (2) Controller of the Playstation, denoted c, and (3-5) Three different games compatible with ps, denoted g_1, g_2 and g_3 respectively. We next describe the method by which we learn their parameters from real data.

2.6.4.1 Learning the value, price, and noise

Predicting a user's bid in an auction is a widely studied problem in auction theory. Jiang et al. [76] showed that learning user's valuations of items improves the prediction accuracy. Given the bidding history of an item, their method learns a value distribution of the item, by taking into account hidden/unobserved bids. We use it to learn the values of itemsets from bidding histories. Recall that in our model value is not random, instead noise models the randomness in valuations. Hence we take the mean of the learned distribution to be the value and the noise is set to have 0 mean and the same variance as the learnt distribution. While UIC does not assume specific noise distributions, for concreteness, we fit a Gaussian distribution to noise. We take $10,000$ independent random samples from the learnt distribution to fit the gaussian.

We mine the bidding histories of different itemsets from eBay. To match the used products bidden in eBay, we use prices for the used products on Craigslist and Facebook groups. Since the items bidden in eBay are typically used products, to match them with the right price information, we use Craigslist and Facebook groups where the exact same old product is sold.

The price obtained is C$260 for ps, C$20 for c, and C$5 each for g_1, g_2

Itemset	Price	Value	Noise	eBay bidding link
$\{ps\}$	260	213	$N(0,4)$	https://ebay.to/2ym9Ioj
$\{ps, c\}$	280	220	$N(0,6)$	https://ebay.to/2Escb68
$\{ps, g_1, g_2, g_3\}$	275	258	$N(0,4)$	https://ebay.to/2QYpmxh
$\{ps, g_1, g_2, c\}$	290	292.5	$N(0,5)$	https://ebay.to/2ClEnF2
$\{ps, g_1, g_2, g_3, c\}$	295	302	$N(0,7)$	https://ebay.to/2P60y99

Table 2.6: Learned parameters

and g_3. For some of the itemsets, we show the learned parameters and the links to the corresponding eBay bidding histories used in the learning, in Table 2.6. The rest of the itemsets are omitted from the table for brevity. We describe the parameters of those omitted itemsets here. Firstly, any of c, g_1, g_2, g_3, without the core item ps, is useless. Hence values of those items are set to 0. Secondly, we did not find any bidding record for an itemset consisting of ps, c and a single game. This is perhaps because typically owners of ps own multiple games and while selling they sell all the games together with ps. Hence, we consider the itemset with ps, c and a single game to have negative deterministic utility. However, as the table shows, itemsets with ps, c and two games have non-negative deterministic utility. Finding the bidding history for the exact same games is difficult, so since games g_1–g_3 are priced similarly and valued similarly by users, we assume that any itemset with ps, c and any two games has the same utility as that shown in the fourth row of Table 2.6. From the value column, we can see that the items indeed follow supermodular valuation, confirming that in practice complementarity arises naturally. Lastly, the only itemsets that have positive deterministic utility are itemsets with ps, c and at least two games. All other itemsets including the singleton items, have negative deterministic utility. Consequently, we know that the allocation produced by item-disj will have 0 expected social welfare, so we omit item-disj from our experiments, discussed next.

2.6.4.2 Effect of total budget size

We compare bundleGRD with bundle-disj on the Twitter dataset with different sizes of total budgets. Given a total budget of items, we assign $30\%, 30\%, 20\%, 10\%$, and 10% of that to ps, c, g_1, g_2, g_3 respectively. Then we vary the total budget from 100 to 500 in steps of 100. Fig. 2.10(b) shows the welfare: as can be seen, bundleGRD outperforms bundle-disj in both high and low budgets. In fact with higher budget, bundleGRD produces welfare

more than 2 times that of bundle-disj. Next we report the running time of the two algorithms in Fig. 2.10(c). Since bundle-disj makes multiple calls to IMM, its running time is 1.5 times higher than bundleGRD.

2.6.4.3 Effect of different item budget given the same total budget

Our next experiment studies the following question. Suppose we have a fixed total budget which we must be divided up among various items. How would the social welfare and running time vary for different splits? Since we have seen that in terms of social welfare bundleGRD dominates all baselines, we use it to measure the welfare. Given a total budget of 500, we split it across 5 items following three different budget distributions, namely (i) Uniform: each item has the same budget 100, (ii) Large skew: one item, ps has 82% of the total budget and the remaining 18% is divided evenly among the remaining 4 items; and (iii) Moderate skew: Budgets of the 5 items, $[ps, c, g_1, g_2, g_3]$, are given by the budget vector $\vec{b} = [150, 150, 100, 50, 50]$.

Fig. 2.10(d) shows the expected social welfare and the running time of bundleGRD under the three budget distributions on the Twitter dataset. The welfare is the highest under uniform and worst under large skew, with moderate skew in between. Running time shows consistent trend, with uniform being the fastest and large skew being the slowest. The findings are consistent with the observation that with large skew, the number of seeds to be selected increases and the allocation cannot take full advantage of supermodularity.

2.6.4.4 Effect of propagation vs. network externality

We next compare our bundleGRD against the other two baselines, namely, BDHS-Concave and BDHS-Step (referred to as BDHS algorithms for simplicity). BDHS-Concave and BDHS-Step correspond to the concave and step externality algorithms respectively (i.e. Alg 1 and 3 of [19]). Our overall approach is, despite the differences between our model and BDHS model as highlighted in Section 2.2, we try to convert our model in a reasonable way to their model by means of restriction, and use their algorithms to find the total social welfare that they can achieve. Then we gradually increase the budget of items in our model to see at which budget the social welfare achieved by our solution reaches the social welfare achieved by their solution that has no budget and assigns items to every node directly. This would demonstrate the budget savings due to our consideration of network

propagation.

We now describe how we convert our model to their model. First, our model uses network propagation with the UIC model while their model uses network externality without propagation. To align the two models, we try two alternatives. The first alternative is to sample 10,000 live-edge graphs, and the propagation on one live-edge graph bears similarity with the 1-step function, and thus we use 1-step externality function on each live-edge graph to compute the total social welfare and then average over all live-edge graphs. We refer to this alternative BDHS-Step. The second alternative works when we restrict our UIC model such that every edge has the same propbability p. In this case, the activation probability of a node v is $1-(1-p)^k$, where k is the number of active neighbors of v which is at most the size s of its 2-neighborhood support set. This resembles the concave function case in the BDHS model, and thus we use the concave function $1-(1-p)^s$ in their 2-hop model. We refer to this alternative BDHS-Concave.

Second, to align their unit demand model with our model, we treat each item subset as a virtual item in their model, so that they can assign item subsets as one virtual item to the nodes. Finally, their model has no budget, so they are free to assign all item subsets to all nodes. We use this as a benchmark of the total social welfare they can achieve, and see at what fraction of the budget we can achieve the same social welfare due to the network propagation effect.

We used the Orkut as one of the large networks in this study, which also enables the study of the performance of bundleGRD on a large network other than Twitter (which is already used in Figure 2.7(d), 2.9, and 2.10).Fig. 2.11(a-c) shows the results on Orkut, Douban-Book and Douban-Movie networks respectively. The x axis shows the fraction of the budget needed by bundleGRD, where 100% corresponds to a budget of n, i.e., #nodes in the network, which corresponds to the setting of [19]. As can be seen, for dense networks like Orkut, bundleGRD needs less than 35% as the budget. We found a similar result on *Flixster*, not included here for the lack of space. For a sparse graph like Douban-Book it needs 82%, which is still less than the budget of BDHS. Further, since propagation has a submodular growth, much of the budget is used to increase the latter half of the welfare. E.g., even on Douban-Book, 75% of BDHS' welfare is obtained by only using 50% budget. This test clearly demonstrates that our bundleGRD could leverage the power of propagation, compared to the BDHS approach that only considers externality.

Budget distribution	bundleGRD	MAX_IMM	IMM_MAX
Uniform	37719	37719	37719
Large skew	144328	144328	144328
Moderate skew	50839	50839	50839

Table 2.7: The number of RR sets generated

2.6.4.5 Scalability test

Our next experiment shows the impact of network size on bundleGRD using Orkut with two types of edge probabilities: (1) $1/d_{in}(v)$ and (2) fixed 0.01. We use a uniform budget of 50 for all items. We then use breadth-first-search to progressively increase the network size such that it includes a certain percentage of the total nodes. The results are shown in Fig. 8(d). With increasing network size, the running time in both cases roughly has a linear increase, whereas the welfare depicts a sublinear growth. It is worth noticing that even for the entire million-sized network and fixed probability, bundleGRD requires mere 129 (time 2) seconds to complete, which again attests to its scalability.

2.6.4.6 Memory usage

Lastly we assess the memory usage of bundleGRD. Since the main memory usage is on the RR set storage, we evaluate the number of RR sets bundleGRD generates in comparison to IMM for the three aforementioned budget distributions. Since IMM works only with a single item (i.e., one budget), we consider two variants. In the first variant IMM is invoked with maximum budget, called IMM_MAX. The second variant iterates over all budgets and reports the budget that generates the maximum number of RR sets, called MAX_IMM. Notice IMM_MAX and MAX_IMM are not equivalent because the number of RR sets generated by IMM is not monotone in budget. The results are shown in Table 2.7. In all three budget configurations the *numbers* of RR sets generated by the three algorithms are exactly the same, from which we can conclude that bundleGRD has a similar memory requirement as IMM.

2.7 Summary & Discussion

We propose a novel model combining influence diffusion with utility-driven item adoption, which supports any mix of competing and complementary items. Focusing on complementary items, we study the problem of optimizing expected social welfare. Our objective function is monotone, but neither submodular nor supermodular. Yet, we show that a simple greedy allocation guarantees a $(1 - 1/e - \epsilon)$-approximation to the optimum. Based on this, we develop a scalable approximation algorithm bundleGRD, which satisfies an interesting prefix preserving property. With extensive experiments, we show that our algorithm outperforms the state of the art baselines.

Our results and techniques carry over unchanged to any triggering propagation model [83]. We assumed that price is additive and valuations are supermodular. If we use submodular prices, that would further favor item bundling. In this case, utility remains supermodular and our results remain intact. Further a user specific budget constraint can also be considered, which is left as a future work.

Independently of the extensions mentioned above, UIC can also be studied for competing items using a submodular value function. This study is conducted in the next chapter of the thesis.

Chapter 3

Maximizing social welfare in a utility driven, competitive diffusion model

3.1 Introduction

Influence maximization (IM) on social and information networks is a well-studied problem that has gained a lot of traction since it was introduced by Kempe et al. [83]. Given a network, modeled as a probabilistic graph where users are represented by nodes and their connections by edges, the problem is to identify a small set of k seed nodes, such that by starting a campaign from those nodes, the expected number of users who will be influenced by the campaign, termed influence spread, is maximized. Here, the expectation is w.r.t. an underlying stochastic diffusion model that governs how the influence propagates from one node to another. The "item" being promoted by the campaign may be a product, a digital good, an innovative idea, or an opinion.

Existing works on IM typically focus on two types of diffusion models – *single item* diffusion and diffusion of *multiple items under pure competition*. The studies on multiple-item diffusion mostly focus on two items in pure competition [35, 96, 116, 138], that is, every node would only adopt at most one item, never both. The typical objective is to select seeds for the second item (the follower item) to maximize its number of adoptions, or minimize the spread of the first item [35].

There are a number of key issues on multiple item diffusion that are not satisfactorily addressed in most prior studies. First, most propagation models are purely stochastic, in which if a node v is influenced by a neighboring node u on certain item, it will either deterministically or probabilistically adopt the item, without any consideration of the utility of that item for the node. This fails to incorporate economic incentives into the user adoption behavior. Second, most studies focus on *pure* competition, where each

node adopts at most one item, and ignore the possibility of nodes adopting multiple items. For instance, when items are involved in a partial competition, their combined utility may still be more than the individual utility, although it may be less than the sum of their utilities. Third, most studies on competition focus on the objective of maximizing the influence of one item given other items, or minimizing the influence of existing items, and do not consider maximizing the overall welfare caused by all item adoptions.

The study of Chapter 2 is unique in addressing the above issues. It proposes the utility-based independent cascade model UIC, in which: (a) each item has a utility determined by its value, price and a noise term, and each node selects the best item or itemset that offers the highest utility among all items that the node becomes aware of thanks to its neighbors' influence; and (b) the utility-based adoption naturally models the adoption of multiple items, in a framework that allows arbitrary interactions between items, based on chosen value functions. The previous chapter studies the maximization of expected social welfare, defined as the the total sum of the utilities of items adopted by all network nodes, in expectation. However, the study is confined to the *complementary* item scenario, where item utilities increase when bundled together.

In this chapter, we complement the study of Chapter 2 by considering the *social welfare maximization* problem in the UIC model when items are purely or partially *competitive*. Partial (pure) competition means adopting an item makes a user less likely (resp., impossible) to adopt another item. To motivate the problem, we note that for a social network platform owner (also called the *host*), one natural objective might be to optimize the advertising revenue, as studied by Chalermsook et al. [31], or a proxy thereof, such as expected number of item adoptions. On the other hand, one of the key assets of a network host is the loyalty and engagement of its user base, on which the host relies for its revenue from advertising and other means. Thus, while launching campaigns, it is equally natural for the host to take into account users' satisfaction by making users aware of itemsets that increase their utility. Social Welfare, being the sum of utilities of itemsets adopted by users, is directly in line with this objective.

As a real application, consider a music streaming platform such as the Last.fm. Benson et al. [14] using their discrete choice model showed existence of competition across different genres of songs in the Last.fm dataset. In a platform such as Last.fm, the platform owner (i.e., host) completely controls the promotion of songs and the host would like to keep making engaging recommendations to the users. Even when there are multiple competing songs from different genres, the host should recommend based on

users' preferences, i.e., the users' utility. A similar idea extends to different competing products that an e-retailer like Amazon sells directly. Those products are already procured by the e-retailer and it has full control over how it wants to sell them. Once again, in this setting, keeping users' satisfaction from adopting these products high helps maintain a loyal and engaged user base. Thus maximizing the overall social welfare is in line with the goal of the platform. While the previous chapter studies this problem for complementary items, social welfare maximization under competing products is open. Moreover, under pure competition, the bundling algorithm of Section 2.5 would lead to nodes adopting at most one of several competing items, leading to poor social welfare.

Compared to Chapter 2, in this chapter a more flexible setting is considered where the allocation of some items has been fixed (e.g., the items had the seeds selected by the host earlier) and the host is only allocating seeds for the remaining items. Once again, the objective is still to maximize the total social welfare of all users in the network. We call this the CWelMax problem (for Competitive Welfare Maximization).

As it turns out, CWelMax under UIC is *significantly more difficult than the welfare maximization problem in the complementary setting studied in Chapter 2* We show that when treating the allocation as a set of item-node pairs, the welfare objective function is neither monotone nor submodular. Moreover, with a non-trivial reduction, we prove that CWelMax is in general NP-hard to approximate to within any constant factor. In contrast a constant approximation was possible in the setting considered in Chapter 2

Despite all these difficulties, we design several algorithms that either provide an instance-dependent approximation guarantee in the general case, or better (constant) approximation guarantee in some special cases. In particular, we first design algorithm SeqGRD which provides a $\frac{u_{\min}}{u_{\max}}(1-\frac{1}{e}-\epsilon)$-approximation guarantee for the general CWelMax setting, where $u_{\min}$ is the minimum expected utility among all individual items, $u_{\max}$ is the expected maximum utility among all item bundles, and $\epsilon > 0$ is any small positive number. Next, when the fixed itemset is empty, we complement SeqGRD with MaxGRD, which guarantees $\frac{1}{m}(1-\frac{1}{e}-\epsilon)$-approximation, where m is the total number of items. Thus, when SeqGRD and MaxGRD work together, we can guarantee $\max(\frac{u_{\min}}{u_{\max}}, \frac{1}{m})(1-\frac{1}{e}-\epsilon)$-approximation when there are no prior allocated items. We can see that when the utility difference among items is not high or the number of items is small, the above algorithms can achieve a reasonable approximation performance. Finally, in the special case where we have a unique superior item with utility better than all other items, all other items have had their allocations fixed,

and items exhibit pure competition, we design an efficient algorithm that achieves $(1 - \frac{1}{e} - \epsilon)$-approximation.

We extensively test our algorithms against state-of-the-art IM algorithms under seven different utility configurations including both real and synthetic ones, which capture different aspects of competition. Our results on real networks show that our algorithms produce social welfare up to five times higher than the baselines. Furthermore, they easily scale to large networks with millions of nodes and billions of edges. We also empirically test the effect of social welfare maximization on adoption count and show that whereas the overall adoption count remains the same, social welfare is maximized by reducing adoption of just the inferior items. To summarize, the major contributions made in this chapter are as follows:

- We are the first to study the competitive social welfare maximization problem CWelMax under the utility-based UIC model (Section 3.3).
- We show that social welfare is neither monotone, submodular, nor supermodular; furthermore, it is NP-hard to approximate CWelMax within any constant factor, in general (Section 3.4).
- We provide several algorithms that either solve the CWelMax in the general setting with a utility-dependent approximation guarantee, or have better (constant) approximation guarantees in special cases (Section 3.5).
- We conducted an extensive experimental evaluation over several real social networks comparing our algorithms with existing algorithms. Our results show that our algorithms significantly dominate existing algorithms and validate that our algorithms both deliver good quality and scale to large networks (Section 3.6).

Background and related work are discussed in Section 3.2. We conclude the chapter and discuss future work in Section 3.7.

3.2 Background & Related Work

Recall that in the classical IM problem, a directed graph $G = (V, E, p)$ represents a social network with users V and a set of connections (edges) E. The function $p : E \to [0, 1]$ specifies influence probabilities between users. Independent cascade (IC) model is a commonly used discrete time diffusion model [35, 83].

For a seed set $S \subset V$, we use $\sigma(S)$ to denote the *influence spread* of S, i.e., the expected number of active nodes at the end of diffusion from S. For

a seed budget k and a diffusion model, *influence maximization* (IM) problem is to find a seed set $S \subset V$ with $|S| \leq k$ such that the influence spread $\sigma(S)$ under the model is maximized [83].

Works on single item diffusion are reviewed in Chapter 1. In this section, works on multi-item competitive IM are reviewed, that are more relevant to the work of this chapter.

3.2.1 Multiple item competitive IM

More recently, IM has been studied involving independent items [47], and competing items [17, 25, 70, 90, 96, 138]. In [95] authors studied the problem under pure competition, whereas [57] aims to maximize balanced exposure in the network in presence of two competing ideas,and [96] ensured fairness in the adoption of competing items. These works, however, are restricted to specific type of competition. The Com-IC model proposed by Lu et al. [97] can model any arbitrary degree of interaction between a pair of items. Their main study is therefore restricted to the diffusion of two items. [92] looks into different facets of items to compute influence (e.g., topics of documents). However, unlike our work, they do not consider item utility in adoption decisions made by users. Furthermore, their objective function is based on traditional (expected) number of item adoptions. In addition to the above differences, our objective is to maximize the social welfare that none of these papers have studied. For a more comprehensive survey on competitive influence models, see [35, 93].

3.2.2 Social welfare maximization

Utility driven adoptions have been studied in economics [1, 20, 105, 113]. Given items and users, and the utility functions of users for various subsets of items, the problem is to find an allocation of items to users such that the sum of utilities of users, is maximized. Since the problem is intractable, approximation algorithms have been developed [52, 81, 84]. [14] proposed a discrete choice model to learn the utilities of itemsets from the users' adoption logs. Learning utility is complementary to our problem. Moreover none of these works consider a social network and the effect of recursive propagation of item adoptions by its users.

Host's perspective in the context of IM have been studied. [8] directly maximizes the revenue earned by a network host, whereas [9] aimed to minimize the regret of seed selection. These works donot consider the overall social welfare. Utility based adoption decisions of users are also not part of

their formalism. Welfare maximization on social networks has been studied in a few recent papers[19, 124]. Bhattacharya et al. [19] consider item allocations to nodes for welfare maximization in a network with network externalities. Their model does not consider the effect of recursive propagation nor competition. In addition, they do not consider budget constraints. In contrast, the focus of this chapter is on competition, with budget constraints on every item.

Chapter 2 studies welfare maximization under viral marketing using the UIC propagation model that is also used in this chapter. However, the work of the previous chapter focuses strictly on complementary items, with supermodular value functions. As a result, the objective is shown to be monotone, and further satisfies a nice "reachability" property (details in §2.4), which paved the way for efficient approximation. However such complementary only setting fails to model many real world platforms where competing items are also present as highlighted in the introduction. This chapter, instead, focuses on *competing* items. Consequently, the objective becomes not only non-monotone, non-submdular, and non-supermodular, but unlike in Chapter 2, is inapproximable within any constant. In spite of this, utility dependent approximation algorithms as well as a constant approximation algorithm for special cases are developed in this chapter.

In summary, to our knowledge, *study of this chapter is the first to address social welfare maximization in a network with influence propagation, competing items, and budget constraints, where item adoption is driven by utility.*

3.3 UIC Model under competition

In this section, we first briefly review the *utility driven independent cascade* model (UIC for short) proposed in Chapter 2. Then we describe the competitive setting of UIC studied in this chapter and formally state the new problem we address.

3.3.1 Review of UIC Model

UIC integrates utility driven adoption decision of nodes, with item propagation. Every node has two sets of items – *desire set* and *adoption set*. Desire set is the set of items that the node has been informed about (and thus potentially desires), via propagation or seeding. Adoption set, is the subset of the desire set that has the highest utility, and is adopted by the user. The utility of an itemset $I \subseteq \mathbf{I}$ is derived as $\mathcal{U}(I) = \mathcal{V}(I) - \mathcal{P}(I) + \mathcal{N}(I)$, where

$\mathcal{V}(\cdot)$ denotes users' latent valuation for an itemset, $\mathcal{P}(\cdot)$ denotes the price that user needs to pay, and $\mathcal{N}(\cdot)$ is a random noise term that denotes our uncertainty in users' valuation.

Budget vector $\vec{b} = (b_1, ..., b_{|\mathbf{I}|})$ represents the budgets associated with the items, i.e., the number of seed nodes that can be allocated with that item. An *allocation* is a relation $\boldsymbol{\mathcal{S}} \subset V \times \mathbf{I}$ such that $\forall i \in \mathbf{I} : |\{(v, i) \mid v \in V\}| \leq b_i$. $S_i^{\boldsymbol{\mathcal{S}}} := \{v \mid (v, i) \in \boldsymbol{\mathcal{S}}\}$ denotes the *seed nodes* of $\boldsymbol{\mathcal{S}}$ for item i and $S^{\boldsymbol{\mathcal{S}}} := \bigcup_{i \in \mathbf{I}} S_i^{\boldsymbol{\mathcal{S}}}$. When the allocation $\boldsymbol{\mathcal{S}}$ is clear from the context, we write S (resp., S_i) to denote $S^{\boldsymbol{\mathcal{S}}}$ (resp., $S_i^{\boldsymbol{\mathcal{S}}}$).

Before a diffusion begins, the noise terms of all items are sampled, and they are used until the end of that diffusion. The diffusion proceeds in discrete time steps, starting from $t = 1$. $\mathcal{R}^{\boldsymbol{\mathcal{S}}}(v, t)$ and $\mathcal{A}^{\boldsymbol{\mathcal{S}}}(v, t)$ denote the desire and adoption sets of node v at time t. At $t = 1$, the seed nodes have their desire sets initialized according to the allocation $\boldsymbol{\mathcal{S}}$ as, $\mathcal{R}^{\boldsymbol{\mathcal{S}}}(v, 1) = \{i \mid (v, i) \in \boldsymbol{\mathcal{S}}\}$, $\forall v \in S^{\boldsymbol{\mathcal{S}}}$. These seed nodes then adopt the subset of items from the desire set that maximizes the utility. The propagation then unfolds recursively for $t \geq 2$ in the following way. Once a node u' adopts an item i at time $t - 1$, it influences its out-neighbor u with probability $p_{u'u}$, and if it succeeds, then i is added to the desire set of u at time t. Subsequently u adopts the subset of items from the desire set of u that maximizes the utility. Adoption is progressive, i.e., once a node adopts an item, it cannot unadopt it later. Thus $\mathcal{A}^{\boldsymbol{\mathcal{S}}}(u, t) = \arg\max_{T \subseteq \mathcal{R}^{\boldsymbol{\mathcal{S}}}(u,t)} \{\mathcal{U}(T) \mid T \supseteq \mathcal{A}^{\boldsymbol{\mathcal{S}}}(u, t-1) \ \wedge \ \mathcal{U}(T) \geq 0\}$. The propagation converges when there is no new adoption in the network. For more details readers are referred to Section 2.3.

3.3.2 Social welfare maximization

Social welfare maximization relative to a fixed seed set: Let $G = (V, E, p)$ be a social network, $\mathbf{I}$ the universe of items under consideration. We consider a utility-based objective called *social welfare*, which is the sum of all users' utilities of itemsets adopted by them after the propagation converges. Formally, $\mathbb{E}[\mathcal{U}(\mathcal{A}^{\boldsymbol{\mathcal{S}}}(u))]$ is the expected utility that a user u attains for a seed allocation $\boldsymbol{\mathcal{S}}$ after the propagation ends. The *expected social welfare* for $\boldsymbol{\mathcal{S}}$, is $\rho(\boldsymbol{\mathcal{S}}) = \sum_{u \in V} \mathbb{E}[\mathcal{U}(\mathcal{A}^{\boldsymbol{\mathcal{S}}}(u))]$, where the expectation is over both the randomness of propagation and randomness of noise terms $\mathcal{N}(.)$

In a social network, a campaign may often be launched on top of other existing campaigns, where the seeds for some items $I_1 \subset \mathbf{I}$ may already be fixed. Let $\boldsymbol{\mathcal{S}}^P$ be this fixed allocation for items in I_1. Then $I_2 = \mathbf{I} \setminus I_1$ is the set of items for which the seeds are to be selected. We define the problem

of maximizing expected social welfare, on top of a fixed seed allocation as follows.

Welfare maximization under competition: To model competition, we assume that $\mathcal{V}$ is submodular [27], i.e., the marginal value of an item with respect to an itemset $I \subset \mathbf{I}$ decreases as I grows. We assume $\mathcal{V}$ is monotone, since it is a natural property for valuations. We set $\mathcal{V}(\emptyset) = 0$. For $i \in \mathbf{I}$, $\mathcal{N}(i) \sim \mathcal{D}_i$ denotes the noise term associated with item i, where the noise may be drawn from any distribution $\mathcal{D}_i$ having a zero mean. Every item has an independent noise distribution. For a set of items $I \subseteq \mathbf{I}$, we assume the noise and price to be additive. Since noise is drawn from a zero mean distribution, $\mathbb{E}[\mathcal{U}(I)] = \mathcal{V}(I) - \mathcal{P}(I)$. Below, we refer to $\mathcal{V}, \mathcal{P}, \{\mathcal{D}_i\}_{i \in \mathbf{I}}$, as the model parameters and denote them collectively as Param.

We now illustrate using a toy example how our framework models competition.

Example 10. Let's revisit the example items of Example 3. Suppose a user wants to buy a phone and desires (because of influence) an $iPhonePro$ (abbreviated as iPp) and an $iPhoneMini$ (abbreviated as iPm). Since the user does not own a phone yet, she enjoys no valuation, which is captured by $\mathcal{V}(\emptyset) = 0$. However after she adopts one phone, then although the overall value increases by adopting a second phone ($\mathcal{V}(.)$ is monotone), the marginal value gain decreases ($\mathcal{V}(.)$ is submodular). Formally speaking, $\mathcal{V}(\{iPp, iPm\}) = \mathcal{V}(\{iPp\}) + \mathcal{V}(\{iPm\} \mid \{iPp\})$. Since value is monotone and submodular, $0 \leq \mathcal{V}(\{iPm\} \mid \{iPp\}) \leq \mathcal{V}(\{iPm\})$, hence $\mathcal{V}(\{iPp\}) \leq \mathcal{V}(\{iPp, iPm\}) \leq \mathcal{V}(\{iPp\}) + \mathcal{V}(\{iPm\})$.

Finally price determines the exact adoption set of a user. The itemset that offers the highest utility is adopted by the user. If the second phone has a low price (i.e., $\mathcal{P}(\{iPm\}) \leq \mathcal{V}(\{iPm\} \mid \{iPp\})$), then the user may still adopt both the phones (partial competition). Otherwise (i.e., $\mathcal{P}(\{iPm\}) > \mathcal{V}(\{iPm\} \mid \{iPp\})$), a user who has already adopted iPp will not adopt iPm. □

Problem 2 (CWelMax)**.** *Given $G = (V, E, p)$, the set of model parameters* Param*, an existing fixed allocation $\boldsymbol{\mathcal{S}}^P$, and budget vector $\vec{b}$, find a seed allocation $\boldsymbol{\mathcal{S}}^*$ for items I_2, such that $\forall i \in I_2$, $|S_i^{\boldsymbol{\mathcal{S}}^*}| \leq b_i$ and $\boldsymbol{\mathcal{S}}^*$ maximizes the expected social welfare, i.e., $\boldsymbol{\mathcal{S}}^* = \arg\max_{\boldsymbol{\mathcal{S}}} \rho(\boldsymbol{\mathcal{S}} \cup \boldsymbol{\mathcal{S}}^P)$.*

Note that this problem subsumes the typical "fresh campaigns" setting as a special case where $I_1 = \emptyset$ (and hence $\boldsymbol{\mathcal{S}}^p = \emptyset$).

3.3.3 An equivalent possible world model

In Section 2.4.1, an equivalent possible world interpretation of the diffusion under UIC is presented, which will be useful for the analysis of this chapter as well. It is briefly reviewed below. Let $\langle G, \mathsf{Param} \rangle$ be an instance of CWelMax, where $G = (V, E, p)$. A *possible world* $w = (w_1, w_2)$, consists an *edge possible world* (edge world) w_1, and a *noise possible world* (noise world) w_2: w_1 is a deterministic graph sampled from the distribution associated with G, where each edge $(u, v) \in E$ is sampled in with an independent probability of p_{uv}; and w_2 is a sample of noise terms for items in $\mathbf{I}$, drawn from noise distributions in Param. Note that propagation and adoption in w is fully deterministic. In a possible world w, $\mathcal{N}_w(i)$ is the noise for item i and $\mathcal{U}_w(I)$ is the (deterministic) utility of itemset I. The *social welfare* of an allocation $\boldsymbol{\mathcal{S}}$ in w is $\rho_w(\boldsymbol{\mathcal{S}}) := \sum_{v \in V} \mathcal{U}(\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(v))$, where $\mathcal{A}^{\boldsymbol{\mathcal{S}}}_W(v)$ is the adoption set of v at the end of the propagation in world w. The *expected social welfare* of an allocation $\boldsymbol{\mathcal{S}}$ is $\rho(\boldsymbol{\mathcal{S}}) := \mathbb{E}_w[\rho_w(\boldsymbol{\mathcal{S}})] = \mathbb{E}_{w_1}[\mathbb{E}_{w_2}[\rho_w(\boldsymbol{\mathcal{S}})]] = \mathbb{E}_{w_2}[\mathbb{E}_{w_1}[\rho_w(\boldsymbol{\mathcal{S}})]]$.

3.4 Properties of UIC

It is easy to see that CWelMax is NP-hard.

Proposition 2. *CWelMax in the UIC model is NP-hard.*

Sketch. Classic IM is a special case of CWelMax. □

Given the hardness, we examine whether social welfare satisfies monotonicity, submodularity or supermodularity.

3.4.1 Item blocking

Under the *complementary setting* Chapter 2 leveraged the reachability property: if a node v adopts an item i in any possible world w, then all the other nodes that are reachable from v in w will also adopt i. This property does *not* hold under the competitive setting. In fact, adoption of one particular item can block the propagation of another item, making social welfare non-monotone and non-submodular.

Theorem 5. *Expected social welfare is not monotone, and neither submodular nor supermodular, with respect to sets of node-item allocation pairs.*

Proof. We show a counterexample for each of the three properties. Consider a simple network with two nodes u and v, and a directed edge (u, v) with probability 1. Assume that there is no noise, i.e., noise is 0. There are three items in propagation whose utility configuration is shown in Fig. 3.1.

Item	$\mathcal{V}$	$\mathcal{P}$	$\mathcal{U}$
$\emptyset$	0	0	0
i_1	5	1	4
i_2	7	4	3
i_3	5	1	4
i_1, i_2	7	5	2
i_1, i_3	7	2	5
i_2, i_3	7	5	2
i_1, i_2, i_3	7	6	1

Figure 3.1: Utility configurations, used in Theorem 1

Item bundle	$\mathcal{V}$	$\mathcal{P}$	$\mathcal{U}$	Constraints
$\emptyset$	0	0	0	
i_1	$v - d_1$	$p - d_1 - \delta$	u_1	$\delta > 0$
i_2	v	p	u_2	$u_1 = u_2 + \delta$
i_3	v	p	u_3	$u_1 = u_3 + \delta$
i_4	$8v/c$	$4p/c$	u_4	$u_4 > 4(u_2 + u_3)/c$
i_1, i_2	$2v - d_1 - \epsilon$	$2p - d_1 - \delta$	$u_2 - \epsilon + \delta$	$\epsilon > \delta$
i_1, i_3	$2v - d_1 - \epsilon$	$2p - d_1 - \delta$	$u_2 - \epsilon + \delta$	$\epsilon > \delta$
i_1, i_4	$v - d_1 + 8v/c$	$p - d_1 - \delta + 4p/c$	$u_1 + u_4$	
i_2, i_3	$2v$	$2p$	$u_2 + u_3$	
i_2, i_4	$v + 8v/c$	$p + 4p/c$	$u_2 + u_4$	
i_3, i_4	$v + 8v/c$	$p + 4p/c$	$u_3 + u_4$	
i_1, i_2, i_3	$3v - d_1 - \epsilon - \epsilon_1$	$3p - d_1 - \delta$	u_{123}	$u_{123} = u_{12} - \epsilon_1$
i_1, i_2, i_4	$2v + 8v/c - d_1 - \epsilon_3$	$2p - d_1 - \delta + 4p/c$	u_{124}	$\epsilon_3 > \epsilon$
i_1, i_3, i_4	$2v + 8v/c - d_1 - \epsilon_3$	$2p - d_1 - \delta + 4p/c$	u_{134}	$\epsilon_3 > \epsilon$
i_2, i_3, i_4	$2v + \epsilon_4$	$2p + \frac{4p}{c}$	u_{234}	$\epsilon_4 < 4p/c$
i_1, i_2, i_3, i_4	$\max(v_{123}, v_{124}, v_{134}, v_{234})$	$3p - d_1 - \delta + 4p/c$	u_{124}	

Figure 3.2: Utility configurations, used in Theorem 2

Monotonicity. Consider the two allocations $\boldsymbol{\mathcal{S}}^1 = \{(u, i_1)\}$, and $\boldsymbol{\mathcal{S}}^2 = \{(u, i_1), (v, i_2)\}$. Clearly $\boldsymbol{\mathcal{S}}^1 \subset \boldsymbol{\mathcal{S}}^2$. Under $\boldsymbol{\mathcal{S}}^1$, both u and v adopt i_1, thus $\rho(\boldsymbol{\mathcal{S}}^1) = 8$. However under $\boldsymbol{\mathcal{S}}^2$, u adopts i_1 but v adopts i_2. Thus $\rho(\boldsymbol{\mathcal{S}}^2) = 7 < \rho(\boldsymbol{\mathcal{S}}^1)$.

Submodularity. Consider $\boldsymbol{\mathcal{S}}^1 = \{(v, i_2)\}$, $\boldsymbol{\mathcal{S}}^2 = \{(v, i_2), (v, i_3)\}$ and (u, i_1). Clearly $\boldsymbol{\mathcal{S}}^1 \subset \boldsymbol{\mathcal{S}}^2$ and $(u, i_1) \notin \boldsymbol{\mathcal{S}}^2$. Under $\boldsymbol{\mathcal{S}}^1$, only v adopts i_2. Under $\boldsymbol{\mathcal{S}}^1 \cup \{(u, i_1)\}$, u adopts i_1 and v adopts i_2. So $\rho(\boldsymbol{\mathcal{S}}^1 \cup \{(u, i_1)\}) - \rho(\boldsymbol{\mathcal{S}}^1) = 4$. Under $\boldsymbol{\mathcal{S}}^2$, v adopts i_3. Under $\boldsymbol{\mathcal{S}}^2 \cup \{(u, i_1)\}$, u adopts i_1 and v adopts i_1 and i_3. So $\rho(\boldsymbol{\mathcal{S}}^2 \cup \{(u, i_1)\}) - \rho(\boldsymbol{\mathcal{S}}^2) = 5 > \rho(\boldsymbol{\mathcal{S}}^1 \cup \{(u, i_1)\}) - \rho(\boldsymbol{\mathcal{S}}^1)$.

Supermodularity. Consider $\boldsymbol{\mathcal{S}}^1 = \emptyset$, $\boldsymbol{\mathcal{S}}^2 = \{(v, i_2)\}$ and (u, i_1). Clearly $\boldsymbol{\mathcal{S}}^1 \subset \boldsymbol{\mathcal{S}}^2$ and $(u, i_1) \notin \boldsymbol{\mathcal{S}}^2$. Under $\boldsymbol{\mathcal{S}}^1$, there is no adoption by any node.

Under $\boldsymbol{\mathcal{S}}^1 \cup \{(u, i_1)\}$, u and v both adopt i_1. So $\rho(\boldsymbol{\mathcal{S}}^1 \cup \{(u, i_1)\}) - \rho(\boldsymbol{\mathcal{S}}^1) = 8$. Under $\boldsymbol{\mathcal{S}}^2$, v adopts i_2. Under $\boldsymbol{\mathcal{S}}^2 \cup \{(u, i_1)\}$, u adopts i_1 and v adopts i_2 . So $\rho(\boldsymbol{\mathcal{S}}^2 \cup \{(u, i_1)\}) - \rho(\boldsymbol{\mathcal{S}}^2) = 4 < \rho(\boldsymbol{\mathcal{S}}^1 \cup \{(u, i_1)\}) - \rho(\boldsymbol{\mathcal{S}}^1)$. □

The absence of these properties makes CWelMax really hard to approximate, as shown next.

3.4.2 Hardness results

NP-hardness.

We show that Influence maximization under the IC model, an NP hard problem, is a special case of CWelMax.

The result follows from the fact that the IM problem under the IC model is a special case of CWelMax: let $\mathbf{I} = \{i\}$, set $\mathcal{V}(i) = 1$, $\mathcal{P}(i) = 0$ and set the noise term for item i to 0. This makes $\mathcal{U}(i) = 1$ so any influenced node will adopt i. Thus, the expected social welfare is simply the expected spread. We know maximizing expected spread under the IC model is NP-hard [83].

Hardness of approximation.

Theorem 6. *CWelMax in the UIC model is NP-hard. Further there is no PTIME algorithm that can approximate CWelMax within any constant factor c,* $0 < c \leq 1$*, unless P = NP.*

Proof. We prove the theorem by a gap introducing reduction from SET COVER. Suppose there is a PTIME c-approximation algorithm $\mathcal{A}$ for CWelMax, for some $0 < c \leq 1$. Given an instance $\mathcal{I} = (\mathcal{F}, X)$ of SET COVER, where $\mathcal{F} = \{S_1, ..., S_r\}$ is a collection of subsets over a set of ground elements $X = \{g_1, ..., g_n\}$, and a number k ($k < r < n$), the question is whether there exist k subsets from $\mathcal{F}$ that cover all the ground elements, i.e., whether $\exists \mathcal{C} \subset \mathcal{F} : |\mathcal{C}| = k$ and $\cup_{S \in \mathcal{C}}^{k} S = X$. We can transform $\mathcal{I}$ in polynomial time to an instance $\mathcal{J}$ of CWelMax.

As an overview, our reduction will show that for a YES-instance of SET COVER, the optimal expected welfare in the corresponding CWelMax instance is high and for a NO-instance, it is low. More precisely, let x_y^* (resp., x_n^*) be the optimal welfare on the transformed instance $\mathcal{J}$ whenever the given instance $\mathcal{I}$ is a YES-instance (resp., NO-instance). Our reduction ensures that $x_n^* < cx_y^*$. In this case, running $\mathcal{A}$ on $\mathcal{J}$ will clearly allow us to decide if $\mathcal{I}$ is a YES-instance or not, which is impossible unless P = NP.

For the rest of the discussion we assume no noise, i.e., noise distribution has 0 mean and variance. Also all the edge probabilities of the graph are set to 1. The details of the reduction follow.

Value, price and utility: We consider four items – i_1, i_2, i_3 and i_4, with the following utility configuration: i_1 competes with i_2 and i_3, and i_1 has a higher individual utility than both. However $\{i_2, i_3\}$ as a bundle has higher utility than i_1. Item i_4 has a very high utility, much higher than that of any other individual item. A node adopting i_1 adopts i_4 if it arrives later. However if a node adopts the bundle $\{i_2, i_3\}$, then it will not adopt i_4 later. We use this configuration in the following way. For a YES-instance, i_1 blocks i_2 and i_3, consequently allowing a large number of nodes to adopt i_4. For a NO-instance, however, most nodes adopt $\{i_2, i_3\}$, blocking i_4 adoption. Thus by setting $c \cdot \mathcal{U}(i_4) > \mathcal{U}(\{i_2, i_3\})$, the desired gap in the optimal welfare is achieved. Lastly, as we will see later in the proof that we need, $\mathcal{U}(\{i_2, i_3\}) < c/4 \cdot \mathcal{U}(\{i_1, i_4\})$.

Assuming no noise terms, Fig. 3.2 provides an abstract summary of this utility configuration, focusing on the items $\{i_1, i_2, i_3, i_4\}$. Note that in addition to the aforementioned constraints, the value function is monotone and submodular, as required. Further, we give one such complete configuration (over all four items i_1, i_2, i_3, i_4) in Table 3.1, for $c = 0.4$.

The network: The graph instance constructed from the given instance of SET COVER is illustrated in Fig. 3.3(a). We first create a bipartite graph having two partitions of r nodes $\{s_1, ..., s_r\}$ corresponding to the sets S_i and n nodes $\{g_1, ..., g_n\}$ corresponding to the ground elements g_j respectively. There is a directed edge from s_i node to g_j node iff $g_j \in S_i$ in the SET COVER instance. There are also n number of "a", "b", "e", "f" nodes. Node a_i is connected with a directed edge to the corresponding g_i node. For each g_i, there is an incoming (directed) edge from a_i and an outgoing edge to f_i. Each node b_i is connected to f_i with a path of length 2, i.e., $b_i \rightarrow e_i \rightarrow f_i$, where e_i is the intermediate node between b_i and f_i. This construction creates the following behavior. If all the g nodes adopt i_1 then all the f nodes adopt i_1. However if any one of the "g" nodes adopts i_2 and all the "e" nodes adopt i_3, then all the "f" nodes adopt $\{i_2, i_3\}$. The significance of this behavior will be clear in the remaining part of the proof.

For a large $N >> n$ that is a multiple of n, we create nodes $d_1, ..., d_N$. For $1 \leq i \leq n$, we add the edges $(f_i, d_{(i*N/n-N/n)+1})$, ..., $(f_i, d_{i*N/n})$. This gadget helps create the gap in the welfare that we are aiming for.

We create n copies of "j" nodes. Each j_i is connected to o_i by a directed path from j_i to o_i of length 3, where l_i and m_i are the intermediate nodes. Similar to f_i, each o_i is connected to N/n "d" nodes – $d_{(i-1)N/n+1}, d_{iN/n}$. As a preview, the "a" nodes (resp., "b" nodes, "j" nodes) will serve as seeds for item i_2 (resp., item i_3 and item i_4). Note that the length of the paths from "j" nodes (seeds of i_4) to "d" nodes is 4, while the paths from the seeds of

i_2 and i_3 to "d" nodes are of length 3. Thus, if $\{i_2, i_3\}$ are not blocked by i_1, all the "d" nodes will adopt $\{i_2, i_3\}$ and cannot adopt i_4 when it arrives later. This completes one copy of the graph, shown in Fig. 3.3(a). All edge probabilities are set to 1. We will explain the significance of node color and the surrounding box soon.

Budgets and seed allocation: We set the budgets for i_2, i_3, i_4 to n each. The "a" nodes are seeded with i_2, "b" nodes are seeded with i_3 and "j" nodes are seeded with i_4. These seeds are fixed (see Fig. 3.3). The budget for i_1 is set to k and these seeds are to be selected so as to maximize the expected social welfare. We complete the construction of the instance $\mathcal{J}$ of CWelMax by making N copies of the graph described above.

Notice for YES-instance of the set cover, the N number of "d" nodes adopt $\{i_1, i_4\}$. Hence we have,

Claim 1. *Suppose $\mathcal{I} = (\mathcal{F}, X)$ is a YES-instance and $\mathcal{J}'$ the transformed instance of CWelMax corresponding to Fig. 3.3(a) and the seed allocation described above. The optimal welfare on $\mathcal{J}'$ is $x^* > N \times \mathcal{U}(\{i_1, i_4\})$.*

Proof. For a YES-instance, choosing the corresponding s nodes of the SET COVER solution maximizes the welfare. In this case, every "g" node has at least one in-neighbor that adopted i_1 at time $t = 1$. Thus, all "g" nodes adopt i_1 at time $t = 2$. Consequently all the "f" and "d" nodes adopt i_1 at time $t = 3$ and $t = 4$ respectively. Later at $t = 5$ when i_4 arrives, those "d" nodes adopt $\{i_1, i_4\}$. The optimal welfare in this case $x^* > N \times \mathcal{U}(\{i_1, i_4\})$. □

For a NO-instance, if we hypothetically fix the seeds of i_1 nodes to s nodes, then since there are no k "s" nodes that can cover all the g nodes, there will be at least one g_i node that will not have an in-neighbor adopting i_1. Thus that g_i node, at time $t = 2$, will adopt i_2, being influenced by the corresponding a_i node. At $t = 3$, since $\{i_2, i_3\}$ as bundle has a higher utility than i_1, all "f" nodes adopt $\{i_2, i_3\}$, consequently all "d" nodes will also adopt $\{i_2, i_3\}$ and will not be able to adopt i_4. Thus in this case, assuming a very large value of N, $x^* \leq \mathcal{U}(\{i_2, i_3\}) \times N + o(1)$. However, for a NO-instance, the optimal welfare cannot be achieved by choosing "s" nodes as seeds for i_1. Instead, the g nodes should directly be seeded with i_1. In that case, before i_2 arrives, at $t = 1$, those k seeded "g" nodes adopt i_1. At $t = 2$ all "f" nodes also adopt i_1 and at $t = 3$ all "d" nodes adopt i_1. Later these "d" nodes adopt i_4. Thus welfare is similar to that of YES-instance (which is undesirable).

Completing construction of $\mathcal{J}$:

To circumvent the above mentioned problem, we next create N copies of the structure $\mathcal{J}'$ described above as shown in Fig. 3.3(b). Except for the "s", "a", "b" and "j" nodes, all other nodes and their connections are duplicated exactly the same way in each of those N copies. The nodes that are not duplicated are colored in red. The duplicated nodes are connected to non-duplicated nodes in the same way across all the N copies of the structure. E.g., across different copies, the same duplicated g_i nodes are connected to the (non-duplicated) s_i node, depending on whether $g_i \in S_i$ in the SET COVER instance. Similarly j_1 is connected to copies of l_1, i.e., to $l_{11}, ..., l_{1N}$. In other words the network structure of Fig. 3.3(a) i.e. enclosed in the box, is replicated N times, shown using N boxes in 3.3(b). Together with the seed allocation of items i_2, i_3, i_4 above, this completes the construction of instance $\mathcal{J}$ of the problem.

Now there are N^2 number of "d" nodes. Following Claim 1 for a YES-instance of the set cover, "d" nodes adopt $\{i_1, i_4\}$. Hence,

Claim 2. *Suppose $\mathcal{I} = (\mathcal{F}, X)$ is a YES-instance and $\mathcal{J}$ the transformed instance of CWelMax corresponding to Fig. 3.3(b) and the seed allocation described above. The optimal welfare on $\mathcal{J}$ is $x^* > N^2 \times \mathcal{U}(\{i_1, i_4\})$.*

Proof. There are N^2 "d" nodes in all. For a YES instance, the optimal seeds for i_1 are exactly the solution of SET COVER. It follows from Claim 1 that the optimal welfare in that case is

$$x_y^* > N^2 \times \mathcal{U}(\{i_1, i_4\}). \tag{3.1}$$

□

For a NO-instance, maximum number of "d" nodes adopt i_4. The only candidate seeds which could achieve that are "s", "g", "f", "e", and "o" nodes. We show that regardless of which k seeds are chosen for item i_1, the welfare achieved is $x_n^* < cN^2\mathcal{U}(i_4)$. First, observe that choosing k "g" nodes as seeds of item i_1 achieves a welfare no less than that of any other choice of k seeds for i_1. In the k copies where g nodes are seeded with i_1 we have the following adoptions. 1 number of "g" and n number of "f" nodes adopt i_1; $n-1$ number of "g" adopt i_2; n number of "e" adopt i_3; n number of "l", "m" and "o" nodes adopt i_4; and N number of "d" nodes adopt $\{i_1, i_4\}$. For the remaining $N-k$ copies we have: n number of "g" nodes adopt i_2; n number of "e" nodes adopt i_3; $3n$ number of "l", "m" and "o" nodes adopt i_4; N number of "d" nodes and n number of "f nodes adopt $\{i_2, i_3\}$. Lastly from the seeds, n number of "a" nodes adopt i_2; n

number of "b" nodes adopt i_3; and n number of "j" nodes adopt i_4. So the total welfare for a NO-instance is:

$$\begin{aligned}
&k[(n+1)\mathcal{U}(i_1) + (2n-1)\mathcal{U}(i_2) + 2n\mathcal{U}(i_3) + 4n\mathcal{U}(i_4) \\
&+ N\mathcal{U}(\{i_1, i_4\})] + (N-k)[(2n)\mathcal{U}(i_2) + (2n)\mathcal{U}(i_3) + 4n\mathcal{U}(i_4) \\
&+ (N+n)\mathcal{U}(\{i_2, i_3\})) + n(\mathcal{U}(i_2) + \mathcal{U}(i_3) + \mathcal{U}(i_4)]. \\
&= (kn+k)\mathcal{U}(i_1) + (n-k+2Nn)\mathcal{U}(i_2) + (2Nn+n)\mathcal{U}(i_3) \\
&+ (3Nn+n)\mathcal{U}(i_4) + Nk\mathcal{U}(i_1, i_4) + (N-k)(N+n)\mathcal{U}(i_2, i_3) \quad (*)
\end{aligned}$$

Since $\mathcal{U}(\{i_2, i_3\}) > \mathcal{U}(i_1) > \mathcal{U}(i_2) = \mathcal{U}(i_3)$,

$$\begin{aligned}
(*) &< (k + 2n + 5Nn + N^2 - kN)\mathcal{U}(\{i_2, i_3\}) \\
&+ (3Nn+n)\mathcal{U}(i_4) + Nk\mathcal{U}(\{i_1, i_4\}) \\
&= N^2\mathcal{U}(\{i_2, i_3\}) + (k + 2n + 5Nn - kN)\mathcal{U}(\{i_2, i_3\} \\
&+ (3Nn+n)\mathcal{U}(i_4) + Nk\mathcal{U}(\{i_1, i_4\}) \\
&< N^2\mathcal{U}(\{i_2, i_3\}) + (8Nn - kN)\mathcal{U}(\{i_2, i_3\} \\
&+ 4Nn\mathcal{U}(i_4) + Nk\mathcal{U}(\{i_1, i_4\}) \quad (**)
\end{aligned}$$

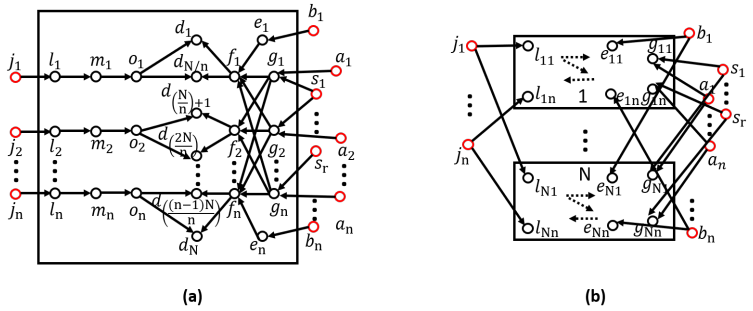


Figure 3.3: Social network: (a) The structure of one copy, $\mathcal{J}'$; (b) Instance $\mathcal{J}$, obtained from N copies of the structure shown on the left side; seeds of i_2: $\{a_1, ..., a_n\}$; seeds of i_3: $\{b_1, ..., b_n\}$; seeds of i_4: $\{j_1, ..., j_n\}$.

We can set the values and prices of items and itemsets such that $\mathcal{U}(\{i_2, i_3\}) < c/4 \cdot \mathcal{U}(\{i_1, i_4\})$ (see Fig. 3.2). Choosing a sufficiently large N: $N > max\{k/c, 8n/c\}$, we can see that each term in the expression $(**)$ above is strictly less than $cN^2\mathcal{U}(\{i1, i4\})$. There are 4 terms in the expression $(**)$ and their sum is $< 4 \times c/4 \times N^2 \times \mathcal{U}(\{i_1, i_4\})$. Thus, the optimal welfare on

a NO-instance is

$$x_n^* = (*) < 4 \times c/4 \times N^2 \times \mathcal{U}(\{i_1, i_4\}) = c \times N^2 \times \mathcal{U}(\{i_1, i_4\}). \tag{3.2}$$

Hence combining this with Claim 2 we get,

Claim 3. *Given a SET COVER instance $\mathcal{I}$, transform it into an instance $\mathcal{J}$ of CWelMax and run the algorithm $\mathcal{A}$ on $\mathcal{J}$. $\mathcal{I}$ is a YES-instance iff the welfare returned by $\mathcal{A}$ is $> cN^2\mathcal{U}(\{i_1, i_4\})$. $\mathcal{I}$ is a NO-instance iff the welfare returned by $\mathcal{A}$ is $< cN^2\mathcal{U}(\{i_1, i_4\})$*

Proof. Suppose $\mathcal{I}$ is a YES-instance. Then, by Claim 2, the optimal welfare of $\mathcal{J}$ is $> N^2\mathcal{U}(\{i_1, i_4\})$, so the welfare returned by Algorithm $\mathcal{A}$ on $\mathcal{J}$ is $> cN^2\mathcal{U}(\{i_1, i_4\})$. Suppose $\mathcal{I}$ is a NO-instance. Then the optimal welfare of $\mathcal{J}$ is $x_n^* < cN^2\mathcal{U}(\{i_1, i_4\})$. Thus, even if Algorithm $\mathcal{A}$ returned the optimal welfare on the NO-instance $\mathcal{J}$, it would be strictly less than the welfare returned on the corresponding YES-instance.

For a NO-instance the optimal welfare is upper bounded by Eq. 3.2. Hence, the claim follows.

□

The theorem follows, $\mathcal{A}$ cannot exist unless $P = NP$. □

3.5 Approximation Algorithms

Since the CWelMax problem cannot be approximated within any constant factor in general, in this section we propose several approximation algorithms that either produce a non-constant approximation guarantee dependent on the problem instance or a constant approximation guarantee for a special case of CWelMax. We first define some important notions.

Truncated utility. For accounting the social welfare of an allocation, we develop the notion of *truncated utility* of an item. Recall that when the noise of an item makes its utility negative, no node adopts the item. Hence what contributes to the final expected social welfare is the set of non-negative contributions to utility. We call this the *truncated utility*, denoted $\mathcal{U}^+(I) := max(0, \mathcal{U}(I))$. Thus for a (node, item) allocation pair (v, i), its expected social welfare (when there are no other allocations) is $\rho(v, i) = \mathbb{E}[\mathcal{U}^+(i)]\sigma(\{v\})$, where $\sigma(\{v\})$ is the influence spread of $\{v\}$.

Minimum and maximum utility bundle. Let, $u_{\min} = \min_{i \in \mathbf{I}} \mathbb{E}[\mathcal{U}^+(i)]$ be the minimum expected truncated utility of any item in $\mathbf{I}$, and $u_{\max} =$

Item bundle	Value	Price	Utility
$\emptyset$	0	0	0
i_1	15.1	10	5.1
i_2	105	100	5
i_3	105	100	5
i_4	101	1	100
i_1, i_2	114.9	110	4.9
i_1, i_3	114.9	110	4.9
i_1, i_4	116.1	11	105.1
i_2, i_3	210	200	10
i_2, i_4	206	101	105
i_3, i_4	206	101	105
i_1, i_2, i_3	214.6	210	4.6
i_1, i_2, i_4	214	111	103
i_1, i_3, i_4	214	111	103
i_2, i_3, i_4	210.5	201	9.5
i_1, i_2, i_3, i_4	214.6	211	3.6

Table 3.1: Utility configuration for different item bundles

$\mathbb{E}[\max_{I \subseteq \mathbf{I}} \mathcal{U}^+(I)]$ be the expected maximum truncated utility of any item *bundle* in $\mathbf{I}$. Note that the definitions of $u_{\min}$ and $u_{\max}$ are not symmetric: (a) $u_{\min}$ takes the minimum of an expectation, while $u_{\max}$ takes the expectation of a maximum; and (b) $u_{\min}$ takes minimum on single items while $u_{\max}$ takes maximum among all bundles. The reason of this asymmetry will be clear in our analysis.

Superior and inferior item. A given itemset $\mathbf{I}$ is said to have a *superior* item i_m, if the least possible utility of i_m is strictly higher than the highest possible utility of any item in $\mathbf{I} \setminus \{i_m\}$. Notice the definition of superior item entails that the noise distribution should be bounded in some way. We discuss a practical way to bound the noise in our experiments (§2.6). Given a superior item, all the other items of the itemset are called *inferior* items.

In what follows, we present three different algorithms with progressively better theoretical guarantees, under progressively stronger assumptions. As a preview, our first algorithm SeqGRD provides a $\frac{u_{\min}}{u_{\max}}(1-\frac{1}{e})$-approximation in the most general case. Our second algorithm, MaxGRD, assumes no prior allocations, i.e., $\boldsymbol{\mathcal{S}}^p = \emptyset$. Under this assumption, it provides a $\frac{1}{m}(1-\frac{1}{e})$-approximation, where m is the number of items. By simply returning the better of the two allocations produced by SeqGRD and MaxGRD, the bound

is improved to $max\{\frac{u_{\min}}{u_{\max}}, \frac{1}{m}\}(1-\frac{1}{e})$, when $\boldsymbol{\mathcal{S}}^p = \emptyset$. Our final algorithm SupGRD assumes that there exists a superior item in the itemset, the allocations for all inferior items are fixed, and that items exhibit pure competition. Under these assumptions, it provides a $(1 - \frac{1}{e})$-approximation.

3.5.1 SeqGrd Algorithm

The pseudocode of our first algorithm SeqGRD is shown in Algorithm 4.

Algorithm 4: SeqGRD$(G, \epsilon, \ell, \boldsymbol{\mathcal{S}}^P, I_2, \vec{b})$

```
1  S^P ← Seed nodes of the allocation 𝒮^P
2  S^Seq ← PRIMA+(G, ε, ℓ, S^P, b⃗, Σ_{i∈I_2} b_i)
3  𝒮^Seq ← ∅
4  Sort I_2 in decreasing order of the expected truncated utility
5  Added ← ∅
6  for i ∈ I_2 do
7  |   S_i^Seq ← top b_i nodes from S^Seq
8  |   if ρ(𝒮^Seq | 𝒮^P) < ρ((𝒮^Seq ∪ (S_i^Seq × {i})) | 𝒮^P) then
9  |   |   𝒮^Seq ← 𝒮^Seq ∪ (S_i^Seq × {i})
10 |   |   Remove those b_i nodes from S^Seq
11 |   |   Added ← Added ∪ {i}
12 for i ∈ I_2 \ Added do
13 |   S_i^Seq ← top b_i nodes from S^Seq
14 |   𝒮^Seq ← 𝒮^Seq ∪ (S_i^Seq × {i})
15 |   Remove those b_i nodes from S^Seq
16 Return 𝒮^Seq
```

Algorithm SeqGRD considers the general setting where a set of items have already been seeded and $\boldsymbol{\mathcal{S}}^P$ corresponds to this partial allocation. Let $S^P := \{v \mid (v, i) \in \boldsymbol{\mathcal{S}}^P\}$ be the seed set allocated in $\boldsymbol{\mathcal{S}}^P$ and let I_2 denote the remaining items which have yet to be allocated. The algorithm takes a graph G, to-be-allocated itemset I_2, item budget vector $\vec{b}$ for the items in I_2, accuracy parameter ϵ, tolerance parameter ℓ, the partial allocation $\boldsymbol{\mathcal{S}}^P$ as input. It first selects a seedset S^{Seq} of size $\overline{b}$, where $\overline{b} := \sum_{i \in I_2} b_i$ [1] (line 2). To select the seeds it uses an algorithm, called PRIMA$^+$, which delivers a set of seeds that are approximately optimal w.r.t. the marginal gain $\sigma(S|S^P)$. We present the PRIMA$^+$ algorithm in §3.5.2.1 and establish its properties.

[1] We assume $I_1 \cap I_2 = \emptyset$.

SeqGRD then sorts the items based on their truncated utility (line 4). Starting from the item i having the highest truncated utility, it tries to allocate the item to the top b_i nodes of S^{Seq}, S_i^{Seq}. If the allocation $S_i^{Seq} \times \{i\}$ yields a positive marginal welfare, it is added to the existing allocation and nodes of S_i^{Seq} are removed for future considerations (Lines 8-11). The items that are not allocated in this iteration are appended following an arbitrary order (lines 12-15) and allocated at the end.

Let $\Gamma_w(S)$ be the set of nodes reachable from a seed set S in the possible world w. We first establish the following lemma.

Lemma 12. *Let $\boldsymbol{\mathcal{S}}$ be an allocation, S be its seedset, let w be a random possible world. Then for any node $v \in V$, we have*

$$u_{\min} \leq \mathbb{E}_w \left[\mathcal{U}_w(\mathcal{A}_w^{\boldsymbol{\mathcal{S}}}(v)) \mid v \in \Gamma_w(S)\right] \leq u_{\max}.$$

Proof. Let $w = (w_1, w_2)$ where w_1 is the edge possible world and w_2 is the noise possible world. Note that, (a) reachable set $\Gamma_w(S)$ is only determined by the edge possible world w_1, so we can use $\Gamma_{w_1}(S)$ to represent it; (b) utility function $\mathcal{U}_w(\cdot)$ is only determined by the noise possible world w_2, so we can use $\mathcal{U}_{w_2}(\cdot)$ to represent it; and (c) adoption set $\mathcal{A}_w^{\boldsymbol{\mathcal{S}}}(v)$ is determined by both w_1 and w_2, so we use $\mathcal{A}_{w_1,w_2}^{\boldsymbol{\mathcal{S}}}(v)$ to represent it.

$$\begin{aligned}&\mathbb{E}_w \left[\mathcal{U}_w(\mathcal{A}_w^{\boldsymbol{\mathcal{S}}}(v)) \mid v \in \Gamma_w(S)\right] \\ &= \mathbb{E}_{w_1} \left[\mathbb{E}_{w_2} \left[\mathcal{U}_{w_2}(\mathcal{A}_{w_1,w_2}^{\boldsymbol{\mathcal{S}}}(v)) \mid v \in \Gamma_{w_1}(S)\right]\right]. \end{aligned} \tag{3.3}$$

We first prove the lower bound $u_{\min}$. To do so, we prove that for any fixed edge possible world w_1 and conditioned on $v \in \Gamma_{w_1}(S)$, we have $\mathbb{E}_{w_2} \left[\mathcal{U}_{w_2}(\mathcal{A}_{w_1,w_2}^{\boldsymbol{\mathcal{S}}}(v))\right] \geq u_{\min} = \min_i \mathbb{E}[\mathcal{U}^+(i)] = \min_i \mathbb{E}_{w_2}[\mathcal{U}_{w_2}^+(i)]$. Once this is proved, from Eq. (3.3), we immediately have $\mathbb{E}_w \left[\mathcal{U}_w(\mathcal{A}_w^{\boldsymbol{\mathcal{S}}}(v)) \mid v \in \Gamma_w(S)\right] \geq \mathbb{E}_{w_1}[u_{\min}] = u_{\min}$.

Consider first a seed $u \in S$. Let $\mathcal{A}_{w_1,w_2}^{\boldsymbol{\mathcal{S}}}(u, 1)$ be the set of items adopted by u initially at time 1 before the propagation starts. Let I^u be the set of items allocated to u in $\boldsymbol{\mathcal{S}}$. Note that I^u is determined purely by the fixed allocation $\boldsymbol{\mathcal{S}}$ and is not affected by the noise or edge possible world. By our model, node u will select the best item bundle in I^u and adopt them as $\mathcal{A}_{w_1,w_2}^{\boldsymbol{\mathcal{S}}}(u, 1)$. Then we know that $\mathcal{U}_{w_2}(\mathcal{A}_{w_1,w_2}^{\boldsymbol{\mathcal{S}}}(u, 1)) \geq \max_{i \in I^u} \mathcal{U}_{w_2}^+(i)$. Therefore, we have

$$\begin{aligned}&\mathbb{E}_{w_2} \left[\mathcal{U}_{w_2}(\mathcal{A}_{w_1,w_2}^{\boldsymbol{\mathcal{S}}}(u, 1))\right] \geq \mathbb{E}_{w_2} \left[\max_{i \in I^u} \mathcal{U}_{w_2}^+(i)\right] \\ &\geq \max_{i \in I^u} \mathbb{E}_{w_2} \left[\mathcal{U}_{w_2}^+(i)\right] \geq \min_{i \in \mathbf{I}} \mathbb{E}_{w_2} \left[\mathcal{U}_{w_2}^+(i)\right] = u_{\min}.\end{aligned}$$

This means that for the initial seed adoption, we have that their expected utility is at least $u_{\min}$. Now for any $v \in \Gamma_{w_1}(S)$, v is reachable from some seed node $u \in S$ via some shortest path in the edge possible world w_1. By the propagation model, then the utility of v's final adoption $\mathcal{U}_{w_2}(\mathcal{A}^{\boldsymbol{\mathcal{S}}}_{w_1,w_2}(v))$ should be at least the utility of u's initial adoption, $\mathcal{U}_{w_2}(\mathcal{A}^{\boldsymbol{\mathcal{S}}}_{w_1,w_2}(u,1))$. Then we have $\mathbb{E}_{w_2}\left[\mathcal{U}_{w_2}(\mathcal{A}^{\boldsymbol{\mathcal{S}}}_{w_1,w_2}(v))\right] \geq \mathbb{E}_{w_2}\left[\mathcal{U}_{w_2}(\mathcal{A}^{\boldsymbol{\mathcal{S}}}_{w_1,w_2}(u,1))\right] \geq u_{\min}$. This concludes the proof.

The proof of the upper bound $u_{\max}$ is obtained using straightforward arithmetic: $\mathbb{E}_w\left[\mathcal{U}_w(\mathcal{A}^{\boldsymbol{\mathcal{S}}}_w(v)) \mid v \in \Gamma_w(S)\right] \leq \mathbb{E}_w\left[\max_{I\subseteq\mathbf{I}}\mathcal{U}^+(I) \mid v \in \Gamma_w(S)\right] = \mathbb{E}_w\left[\max_{I\subseteq\mathbf{I}}\mathcal{U}^+(I)\right] = u_{\max}$. □

Lemma 13. *Let $\boldsymbol{\mathcal{S}}$ be an allocation and S its corresponding seed nodes. Then $u_{\min}\cdot\sigma(S) \leq \rho(\boldsymbol{\mathcal{S}}) \leq u_{\max}\cdot\sigma(S)$.*

Proof. The lower bound is derived below:

$$\begin{aligned}\rho(\boldsymbol{\mathcal{S}}) &= \mathbb{E}_w\left[\sum_v \mathcal{U}_w(\mathcal{A}^{\boldsymbol{\mathcal{S}}}_w(v))\right] = \sum_v \mathbb{E}_w\left[\mathcal{U}_w(\mathcal{A}^{\boldsymbol{\mathcal{S}}}_w(v))\right]\\ &= \sum_v \Pr_w[v\in\Gamma_w(S)]\cdot\mathbb{E}_w\left[\mathcal{U}_w(\mathcal{A}^{\boldsymbol{\mathcal{S}}}_w(v)) \mid v\in\Gamma_w(S)\right]\\ &\geq \sum_v \Pr_w[v\in\Gamma_w(S)]\cdot u_{\min} = u_{\min}\cdot\sigma(S),\end{aligned}$$

where the inequality is by Lemma 12. The upper bound can be shown in a similar way. □

We are now ready to prove the following bound for SeqGRD.

Theorem 7. *Let $\boldsymbol{\mathcal{S}}^{Seq}$ be the allocation returned by the Algorithm* SeqGRD*. Given $\epsilon, \ell > 0$, we have $\rho(\boldsymbol{\mathcal{S}}^{Seq}\cup\boldsymbol{\mathcal{S}}^P) \geq \frac{u_{\min}}{u_{\max}}(1-\frac{1}{e}-\epsilon)\rho(\boldsymbol{\mathcal{S}}^A\cup\boldsymbol{\mathcal{S}}^P)$ w.p. at least $1-\frac{1}{|V|^\ell}$, where $\boldsymbol{\mathcal{S}}^A$ is any arbitrary allocation of items in I_2 respecting the budget constraint.*

Proof. Let S^{Seq}, S^A and S^P be the seed sets of the allocations $\boldsymbol{\mathcal{S}}^{Seq}$, $\boldsymbol{\mathcal{S}}^A$ and $\boldsymbol{\mathcal{S}}^P$ respectively. Then $|S^A| \leq \sum_{i\in I_2} b_i$. By SeqGRD, S^{Seq} exhausts all budgets for items in I_2, so $|S^{Seq}| = \sum_{i\in I_2} b_i$. Since S^{Seq} are the top seeds returned by PRIMA^+, we have that w.p. at least $1-\frac{1}{|V|^\ell}$,

$$\sigma(S^{Seq} \mid S^P) \geq (1-\frac{1}{e}-\epsilon)\sigma(S^A \mid S^P).$$

From this, it follows that

$$\sigma(S^{Seq} \cup S^P) \geq (1 - \frac{1}{e} - \epsilon)\sigma(S^A \cup S^P).$$

Therefore, we have

$$\begin{aligned}\rho(\boldsymbol{\mathcal{S}}^{Seq} \cup \boldsymbol{\mathcal{S}}^P) &\geq u_{\min} \cdot \sigma(S^{Seq} \cup S^P) \\ &\geq u_{\min} \cdot (1 - 1/e - \epsilon) \cdot \sigma(S^A \cup S^P) \\ &\geq \frac{u_{\min}}{u_{\max}}(1 - \frac{1}{e} - \epsilon)\rho(\boldsymbol{\mathcal{S}}^A \cup \boldsymbol{\mathcal{S}}^P),\end{aligned}$$

where the first and the last inequality follow from Lemma 13, while the middle inequality follows from using PRIMA$^+$. □

We note that the property of PRIMA$^+$ that is exploited in the proof above is its ability to select seed nodes S such that they are approximately optimal w.r.t. the marginal gain over an existing seed set S^P. The prefix preserving on marginals property of PRIMA$^+$ is not needed in the above proof. However, our next algorithm MaxGRD relies on the prefix-preserving property.

3.5.1.1 SeqGRD-NM **Algorithm**

The proof of the approximation bound above does not rely on marginal check (Algorithm 4, line 8). We call the version of SeqGRD that does not perform marginal check SeqGRD-NM (No Marginal). Specifically, SeqGRD-NM simply sorts the items based on their truncated utility, allocates item i to the first b_i nodes of S^{Grd}, where S^{Grd} is selected using PRIMA$^+$, and removes those b_i nodes from S^{Grd}.

Computing marginals involves sampling, which takes significant time in large networks. On the other hand, the marginal check avoids the phenomenon of items with lower (truncated) utility blocking those with higher utility, to some extent. Thus even though SeqGRD-NM is faster than SeqGRD and has the same approximation guarantee, under certain utility configurations, the welfare produced by SeqGRD-NM can be worse than that of SeqGRD. We explore this in our experiments in §2.6. On the other hand, we still append all items in the end to exhaust the budget in SeqGRD (lines 12–15). To really discard a certain itemset, we need to exhaustively search through all itemset combinations, which is time-consuming. So we only do a simple marginal check in SeqGRD, and append all items at the end to ensure the theoretical guarantee.

3.5.2 MaxGrd Algorithm

Our next algorithm MaxGRD provides $\frac{1}{m}(1-\frac{1}{e})$-approximation, when $\boldsymbol{\mathcal{S}}^P = \emptyset$, i.e., no prior allocation. The pseudocode is shown in Algorithm 5. Like SeqGRD, MaxGRD also selects its seedset S^{Max} using PRIMA^+, but the size of the seedset is different: $\bar{b} := \max_{i \in I_2} b_i$, i.e., the maximum budget of any unallocated item (line 1). Then for every item $i \in I_2$, it computes the expected marginal social welfare of the allocation $\rho((S_i^{Max} \times \{i\}) \mid \boldsymbol{\mathcal{S}}^P)$, where S_i^{Max} is the set of first b_i nodes of S^{Max}. It returns the allocation with the maximum welfare (line 3).

Algorithm 5: MaxGRD$(G, \epsilon, \ell, \boldsymbol{\mathcal{S}}^P, I_2, \vec{b})$

1 $S^{Max} \leftarrow \mathsf{PRIMA}^+(G, \epsilon, \ell, S^P, \vec{b}, \max_{i \in I_2} b_i)$
2 $S_i^{Max} \leftarrow$ top b_i nodes of S^{Max}, $\forall i \in I_2$
3 $i_{max} \leftarrow \arg\max_{i \in I_2} \{\rho(S_i^{Max} \times \{i\} \mid \boldsymbol{\mathcal{S}}^P)\}$
4 Return $S_{i_{max}}^{Max} \times \{i_{max}\}$

Notice that MaxGRD is applicable even when $S^p \neq \emptyset$, so we have provided the algorithm for this general case. However, it enjoys an approximation bound only for the special case, when $S^p = \emptyset$. We prove the following lemma under this constraint, which is instrumental in the proof of the approximation bound. A key observation is that given a possible world w, the utility function $\mathcal{U}_w(\cdot)$ in that possible world is submodular. This follows from the fact that valuation is submodular and price and noise, being additive are both modular.

Lemma 14. *Let $\boldsymbol{\mathcal{S}} := \cup_{i=1}^m (S_i \times \{i\})$ be an arbitrary allocation, where S_i is the set of seed nodes of item i. Then $\rho(\cup_{i=1}^m (S_i \times \{i\})) \leq \sum_{i=1}^m \rho((S_i \times \{i\}))$.*

Sketch. Consider an arbitrary but fixed possible world w and an arbitrary item $i \in I_2$. Let v be any node that adopts i in w under the allocation $\cup_{i=1}^m (S_i \times \{i\})$. We can show that v must also adopt i in w when the allocation is only $(S_i \times \{i\})$. The lemma follows from this. □

Theorem 8. *Suppose that $\boldsymbol{\mathcal{S}}^P = \emptyset$. Let $\boldsymbol{\mathcal{S}}^{Max}$ be the allocation produced by* MaxGRD. *Given $\epsilon, \ell > 0$, we have $\rho(\boldsymbol{\mathcal{S}}^{Max}) \geq \frac{1}{m}(1 - \frac{1}{e} - \epsilon)\rho(\boldsymbol{\mathcal{S}}^A)$ w.p. at least $1 - \frac{1}{|V|^\ell}$, where $\boldsymbol{\mathcal{S}}^A$ is any arbitrary allocation.*

Proof. Recall that item i has a budget b_i and expected utility u_i. Since in an arbitrary allocation $|S_i^A| \leq b_i$, from the prefix preserving property of

PRIMA$^+$ we have,

$$\sigma(S_i^{Max}) \geq \left(1 - \frac{1}{e} - \epsilon\right) \sigma(S_i^A). \tag{3.4}$$

Let $\mathbb{E}[\mathcal{U}^+(i)]$ be the expected positive utility of item i. We have $\rho(S_i^{Max} \times \{i\}) = \mathbb{E}[\mathcal{U}^+(i)] \cdot \sigma(S_i^{Max})$ and $\rho(S_i^A \times \{i\}) = \mathbb{E}[\mathcal{U}^+(i)] \cdot \sigma(S_i^A)$. Therefore, from Eq.(3.4) we have

$$\rho(S_i^{Max} \times \{i\}) \geq \left(1 - \frac{1}{e} - \epsilon\right) \rho(S_i^A \times \{i\}). \tag{3.5}$$

When $\boldsymbol{\mathcal{S}}^P = \emptyset$, using Eq. 3.5 and Lemma 14, we have

$$\begin{aligned}
&\rho(\boldsymbol{\mathcal{S}}^{Max} \cup \boldsymbol{\mathcal{S}}^P) = \rho(\boldsymbol{\mathcal{S}}^{Max}) = \max_{i \in I_2} \{\rho(S_i^{Max} \times \{i\})\} \\
&\geq \frac{1}{m} \sum_{i=1}^{m} \rho(S_i^{Max} \times \{i\}) \geq \frac{1}{m} \left(1 - \frac{1}{e} - \epsilon\right) \sum_{i=1}^{m} \rho(S_i^A \times \{i\}) \\
&\geq \frac{1}{m} \left(1 - \frac{1}{e} - \epsilon\right) \rho(\cup_{i=1}^{m} (S_i^A \times \{i\})) = \frac{1}{m} \left(1 - \frac{1}{e} - \epsilon\right) \rho(\boldsymbol{\mathcal{S}}^A).
\end{aligned}$$

□

Can MaxGRD **produce better welfare than** SeqGRD**?** Hypothetically, there can be situations where MaxGRD can produce better welfare than SeqGRD. E.g., consider a network with nodes $\{u, v, w, x\}$ and edges $\{(u, v), (v, w), (x, w)\}$ where all edge probabilities are 1. There are two items i, j, with all noise terms being 0. The utilities are $\mathcal{U}(\{i\}) = 10, \mathcal{U}(\{j\}) = 1, \mathcal{U}(\{i, j\}) = 0$ and both items i and j have a budget of 1. Then SeqGRD will yield the allocation $\boldsymbol{\mathcal{S}}^{Seq} = \{(u, i), (x, j)\}$, resulting in a social welfare of $2 \times 10 + 1 \times 2 = 22$. On the other hand, MaxGRD will only allocate u to i, resulting in a social welfare of $3 \times 10 = 30$.

In our experiments, however, we find that situations where MaxGRD dominates SeqGRD are rare. We hypothesize that this is because in a large network, with a number of seeds that is a small fraction of the network size n, blocking caused by the allocation of seeds to additional items by SeqGRD is less likely to occur.

Note that the approximation guarantee of SeqGRD holds also when $\boldsymbol{\mathcal{S}}^p = \emptyset$. Thus running both SeqGRD and MaxGRD individually and returning the allocation with higher welfare would achieve a $max\{\frac{u_{\min}}{u_{\max}}, \frac{1}{m}\}(1 - \frac{1}{e})$-approximation, as a consequence of Theorems 7 and 8.

3.5.2.1 PRIMA$^+$

We now present our PRIMA$^+$ algorithm used by SeqGRD and MaxGRD to select seeds. First, we formally present the property of prefix preservation on marginals.

Definition 2. (Prefix Preservation on Marginals). *Given $G = (V, E, p)$, budget vector $\vec{b}$, the number of seeds to be selected $\bar{b}$ and a fixed seed set S^P, an influence maximization algorithm $\mathbb{A}$ is prefix-preserving on marginals w.r.t. $\vec{b}$ and S^P, if for any $\epsilon > 0$ and $\ell > 0$, $\mathbb{A}$ returns an ordered set S of size $\bar{b}$, such that w.p. at least $1 - \frac{1}{|V|^\ell}$, $\sigma(S \mid S^P) \geq (1 - \frac{1}{e} - \epsilon)$ $OPT_{\bar{b}|S^P}$ and for every $b_i \in \vec{b}$, the first b_i nodes of S, denoted S_i, satisfies $\sigma(S_i \mid S^P) \geq (1 - \frac{1}{e} - \epsilon)$ $OPT_{b_i|S^P}$, where $OPT_{b|S^P}$ is the optimal marginal expected spread of b nodes on top the existing seeds S^P.*

In Section 2.5.3, a seed selection algorithm was developed called PRIMA, that is prefix-preserving in spread, using the Reverse Reachable Sets (RR-sets). Here, we modify the standard RR-set construction slightly to account for the presence of existing seed set S^P: Given an existing allocation $\boldsymbol{\mathcal{S}}^P$, we construct a marginal RR-set as follows. Choose a root node $v \in V$ uniformly at random, add it to R_v and start a BFS from v. Whenever $u \in R_v$, sample each incoming edge (u', u) w.p. $p_{u'u}$ and add it to R_v. Stop when no new nodes are added to R_v; if at any stage R_v overlaps S^P, i.e., if $R_v \cap S^P \neq \emptyset$, then set $R_v := \emptyset$. That is, whenever a generated RR-set "hits" S^P, just set it to $\emptyset$. Algorithm 6 shows the pseudo code of this marginal RR-set sampling process. Given graph G, a number θ denoting how many RR-sets needs to be sampled and a fixed seed nodes S^P, $Marginal\_Sampling$ generates θ number of RR-sets to $\mathcal{R}$ from G, based on the marginal on S^P.

Algorithm 6: $Marginal\_Sampling(G, \mathcal{R}, \theta, S^P)$

```
while |R| ≤ θ do
    Select v from G uniformly at random
    R ← BFS(v)
    if R ∩ S^P ≠ ∅ then
        R ← ∅
    R ← R ∪ R
Return R
```

PRIMA$^+$ using $Marginal\_Sampling$, achieves the property of prefix preservation on marginals. Its pseudo code is shown in Algorithm 7.

Algorithm 7: PRIMA$^+$ $(G, \epsilon, \ell, S^P, \vec{b}, \overline{b})$

Initialize $\mathcal{R} = \emptyset$, $s = 1$, $n = |V|$, $i = 1$, $\epsilon' = \sqrt{2} \cdot \epsilon$, *budgetSwitch* = **false**, $\vec{b} = \vec{b} \cup \overline{b}$
$\ell = \ell + \log 2 / \log n$, $\ell' = \log_n(n^\ell \cdot |\vec{b}|)$
while $i \leq \log_2(n) - 1$ *and* $s \leq |\vec{b}|$ **do**
  $k = b_s$, $LB = 1$
  $x = \frac{n}{2^i}$; $\theta_i = \lambda'_k / x$, where λ'_k is defined in Eq. (3.8)
  $Marginal\_Sampling(G, \mathcal{R}, \theta_i, S^P)$
  if *budgetSwitch* **then**
    S_k = the first k nodes in the ordered set $S_{b_{s-1}}$ returned from the previous call to *NodeSelection*
  else
    $S_k = NodeSelection(\mathcal{R}, k)$
  if $n \cdot F_{\mathcal{R}}(S_k) \geq (1 + \epsilon') \cdot x$ **then**
    $LB = n \cdot F_{\mathcal{R}(S_k)} / (1 + \epsilon')$
    $\theta_k = \lambda^*_k / LB$, where λ^*_k is defined in Eq. (3.6)
    $Marginal\_Sampling(G, \mathcal{R}, \theta_k, S^P)$
    $s = s + 1$; *budgetSwitch* = **true**
  else
    $i = i + 1$; *budgetSwitch* = **false**
if $s \leq |\vec{b}|$ **then**
  $\theta_k = \lambda^*_{b_s} / LB$
$\mathcal{R} = \emptyset$
$Marginal\_Sampling(G, \mathcal{R}, \theta_k, S^P)$
$S_{\overline{b}} = NodeSelection(\mathcal{R}, \overline{b})$
return $S_{\overline{b}}$ as the final seed set;

It runs in time $O((\bar{b}+\ell+\log_n|\vec{b}|)(n+m)\log\ n\cdot\epsilon^{-2})$, where $\bar{b} := \sum_{i\in I_2} b_i$ for SeqGRD and $\bar{b} := \max_{i\in I_2} b_i$, for MaxGRD.

3.5.3 SupGrd Algorithm

Our third algorithm SupGRD provides a constant $(1-\frac{1}{e})$-approximation. The bound holds under more restrictive conditions as given below.

Conditions required for SupGRD **approximation bound.** (i) There exists a superior item (defined in §3.5) i_m in the item set: i.e., under any noise possible world w_2, $\mathcal{U}_{w_2}(i_m) > \mathcal{U}_{w_2}(i), \forall i \in \mathbf{I}\setminus\{i_m\}$. (ii) Seeds for all the inferior items are fixed: that is, $I_2 = \{i_m\}$ is the only item for which an allocation needs to be found; and (iii) There is pure competition between all items: every node can adopt at most one item. Under these conditions, we first show that the social welfare is monotone and submodular.

Lemma 15. *Given $\boldsymbol{\mathcal{S}}^P$ and I_2, let $\boldsymbol{\mathcal{S}}_1$ and $\boldsymbol{\mathcal{S}}_2$ be two allocations over I_2 such that $\boldsymbol{\mathcal{S}}_1 \subseteq \boldsymbol{\mathcal{S}}_2$. Then $\rho(\boldsymbol{\mathcal{S}}_1 \cup \boldsymbol{\mathcal{S}}^P) \leq \rho(\boldsymbol{\mathcal{S}}_2 \cup \boldsymbol{\mathcal{S}}^P)$.*

Proof. In an arbitrary but fixed possible world $w = (w_1, w_2)$, we have, $\mathcal{U}_{w_2}(\mathcal{A}^{\boldsymbol{\mathcal{S}}_1\cup\boldsymbol{\mathcal{S}}^P}_{w_1,w_2}(v)) \leq \mathcal{U}_{w_2}(\mathcal{A}^{\boldsymbol{\mathcal{S}}_2\cup\boldsymbol{\mathcal{S}}^P}_{w_1,w_2}(v))$. This is because if v changes its adoption between the two allocations, then it must change it to i_m since all other inferior item seeds are fixed. Since i_m is the superior item, the claim holds. Since this holds for every w, the lemma follows. □

Lemma 16. *Given $\boldsymbol{\mathcal{S}}^P$ and I_2, let $\boldsymbol{\mathcal{S}}_1$ and $\boldsymbol{\mathcal{S}}_2$ be two allocations over I_2 such that $\boldsymbol{\mathcal{S}}_1 \subseteq \boldsymbol{\mathcal{S}}_2$. Let $s = (u, i_m) \notin \boldsymbol{\mathcal{S}}_2$ be an allocation pair. Then $\rho(s \mid \boldsymbol{\mathcal{S}}_1 \cup \boldsymbol{\mathcal{S}}^P) \geq \rho(s \mid \boldsymbol{\mathcal{S}}_2 \cup \boldsymbol{\mathcal{S}}^P)$.*

Proof. Let $w = (w_1, w_2)$ be a arbitrary but fixed possible world. Let C_i denote the set of all nodes that adopt i_m under allocation s but *not* under $\boldsymbol{\mathcal{S}}_i \cup \boldsymbol{\mathcal{S}}^P$, $i = 1, 2$. Then $C_2 \subseteq C_1$. Thus,

$$\begin{aligned}
\rho_{w_1,w_2}(s \mid \boldsymbol{\mathcal{S}}_2 \cup \boldsymbol{\mathcal{S}}^P) &= \sum_{v\in C_2} (\mathcal{U}_{w_2}(\{i_m\}) - \mathcal{U}_{w_2}(\mathcal{A}^{\boldsymbol{\mathcal{S}}_2\cup\boldsymbol{\mathcal{S}}^P}_{w_1,w_2}(v))) \\
&\leq \sum_{v\in C_1} (\mathcal{U}_{w_2}(\{i_m\}) - \mathcal{U}_{w_2}(\mathcal{A}^{\boldsymbol{\mathcal{S}}_1\cup\boldsymbol{\mathcal{S}}^P}_{w_1,w_2}(v))) \\
&= \rho_{w_1,w_2}(s \mid \boldsymbol{\mathcal{S}}_1 \cup \boldsymbol{\mathcal{S}}^P)
\end{aligned}$$

Since this holds for every w, the lemma follows. □

Since social welfare is monotone and submodular, a standard greedy selection based on the marginal welfare will have $(1-\frac{1}{e})$-approximation.

However since computing spread itself is #P-hard, computing the exact marginal is not feasible. In IM, sampling using RR-sets has been used to achieve state of the art performance. In what follows, by extending IMM [127], we adopt a martingale approach for seed selection in SupGRD. Given ϵ and ℓ, SupGRD returns a seed set that has a $(1 - \frac{1}{e} - \epsilon)$-approximation w.p. at least $1 - \frac{1}{n^\ell}$.

In the classical setting, RR-set samples are used to compute an unbiased estimation of the spread. In our case we need to estimate the *marginal welfare* using the RR-sets. Towards that we define a notion of weight for every RR-set. The weight of an RR-set R_v denotes the marginal gain in the expected social welfare achieved by activating the root v of the RR-set R_v. Thus it is the difference between the expected truncated utility of the item that the root v adopts under the existing partial allocation $\boldsymbol{\mathcal{S}}^P$ and that of i_m. To ensure that the root v indeed adopts i_m, the path from some seed of i_m to v should be no longer than that from any seed of $\boldsymbol{\mathcal{S}}^P$ to v. Thus, a weighted RR-set is constructed as follows.

Definition 3. *(Weighted Reverse Reachable Set). For a given fixed allocation $\boldsymbol{\mathcal{S}}^P$ and a node $v \in G$, a weighted RR-set of v, R_v is obtained by starting with $R_v = \{v\}$ and starting a BFS from v such that: for $u \in R_v$, sample each incoming edge (u', u) w.p. $p_{u',u}$ and add it to R_v; stop when either no new nodes are added or R_v overlaps S^P (so the distance from any node in R_v to v along the reversely generated edges is at most the distance from S^P to v). Then, the weight of R_v is $w(R_v) = \mathcal{U}^+(\{i_m\}) - max_{i \in I^s | s \in S^P \cap R(v)} \mathcal{U}^+(i)$, where I^s denotes the items allocated to node s in the allocation $\boldsymbol{\mathcal{S}}^P$.*

SupGRD samples RR-sets using an early termination as described in Definition 3. This construction ensures that if any member of a weighted RR-set is seeded with i_m, the root of the RR-set, v, will adopt i_m. In what follows, we first establish the connection between marginal social welfare and weighted RR-sets and then present efficient seed selection and RR-set sampling algorithms to maximize the marginal social welfare.

For a node set S, let $\mathbb{I}[.]$ be an indicator function denoting whether S covers the (weighted) RR set R, i.e., $\mathbb{I}(S \cap R \neq \emptyset) = 1$, if $S \cap R_v \neq \emptyset$, 0 otherwise. Also let $\mathcal{L}(G)$ denote the distribution of all the live edge graphs, then extending the result of Borg et al., we now prove the following lemma for weighted RR-sets.

Lemma 17. *For given seed sets S and S^P, we have $\mathbb{E}_{w_1 \sim G}[\rho_{w_1}(S \mid S^P)] = n \cdot \mathbb{E}_{v \sim V, w_1 \sim G}[\mathbb{I}(S \cap R_v \neq \emptyset) \cdot w(R_v)]$ where $n = |V|$ is the number of nodes in G.*

Algorithm 8: $NodeSelection(\mathcal{R}, b')$

```
1 Initialize S_{b'} = ∅, i = 1
2 while i ≤ b' do
3 |   Select v ∈ V \ S_{b'} which has the highest M_R(S_{b'} ∪ v) − M_R(S_{b'})
4 |   S_{b'} ← S_{b'} ∪ {v}
5 |   Remove R from R if v ∈ R
6 return S_{b'} as the final seed set
```

Proof.

$$\begin{aligned}\mathbb{E}_{w_1\sim\mathcal{L}(G)}\rho_{w_1}(S \mid S^P) &= \mathbb{E}_{w_1\sim\mathcal{L}(G)}\left[\sum_{v\in V}\mathbb{I}(S\cap R_v \neq \emptyset)\cdot w(R_v)\right] \\ &= n\cdot\mathbb{E}_{v\sim V, w_1\sim\mathcal{L}(G)}[\mathbb{I}(S\cap R_v = 1)\cdot w(R_v)]\end{aligned}$$

□

We now extend the RR-set based efficient approximation IM algorithm, IMM, for maximizing welfare. Similar to IMM, our algorithm SupGRD has two key phases, namely, *NodeSelection* and *Sampling*. The *NodeSelection* phase is similar to that of IMM, except we consider the weight of RR-sets while selecting seed nodes. For a node set S and a collection of weighted RR-sets $\mathcal{R}$, define $M_{\mathcal{R}}(S) := \sum_{R\in\mathcal{R}}\mathbb{I}[S\cap R\neq\emptyset]\cdot w(R)$. Let $b' = b_{i_m}$ be the budget of the superior item i_m. Given a set $\mathcal{R}$, *NodeSelection* selects b' seeds that maximizes $M_{\mathcal{R}}$ (Algorithm 8).

Next, the goal of the *Sampling* phase (Algorithm 9) is to generate $\mathcal{R}$ such that $|\mathcal{R}| \geq \lambda/OPT$, where OPT is the optimal welfare, and λ is defined as follows,

$$\lambda = 2n\cdot((1-1/e)\cdot\alpha+\beta)^2\cdot\epsilon^{-2}, \tag{3.6}$$

where, $\alpha = \sqrt{\ell\log n + \log 2}$ and
$\beta = \sqrt{(1-1/e)\cdot(\log\binom{n}{b'} + \ell\log\ n + \log 2)}$.

Since OPT is unknown, the *Sampling* first ensures that it finds a lower bound to OPT w.h.p. For that it deploys a statistical test using a binary search on the range of OPT. The maximum possible value of the welfare OPT is $UB = n\times u_{\max}$, when every node in the network adopts the superior item i_m, $u_{\max}$ is the utility of i_m. Thus the binary search ranges from 1 to UB (Line 2).

A good lower bound is found when the condition of Line 5 is satisfied. Different from IMM, this condition directly operates on welfare. This is a

Algorithm 9: $Sampling(G, k, \epsilon, \ell)$

Initialize $\mathcal{R} = \emptyset$, $LB = 1$, $UB = |V| \times u_{\max}$, $i = 1$, $\epsilon' = \sqrt{2} \cdot \epsilon$, $\ell = \ell + log\ 2/log\ n$
for $i = 1$ *to* $log_2 UB - 1$ **do**
  $x = n/2^i$, $\theta_i = \lambda'/x$ **while** $|\mathcal{R}| \leq \theta_i$ **do**
    Add a random RR set to $\mathcal{R}$
  $S_i = NodeSelection(\mathcal{R}, k)$ **if** $\frac{n}{\theta} \cdot M_{\mathcal{R}}(S_i) \geq (1 + \epsilon') \cdot x$ **then**
    $LB = \frac{n}{\theta} \cdot M_{\mathcal{R}}/(1 + \epsilon')$
    Break;
$\mathcal{R} = \emptyset$ **while** $|\mathcal{R}| \leq \lambda/LB$ **do**
  Add a random RR set to $\mathcal{R}$

key step in the correctness of the algorithm, hence we prove it explicitly in Lemma 18.

Lemma 18. *Let $x \in [1, UB], \epsilon'$ and $\delta \in (0, 1)$, then if we invoke $NodeSelection$ with $|\mathcal{R}| = \theta$, where*

$$\theta \geq \frac{(2 + \frac{2}{3}\epsilon') \cdot (log\binom{n}{b'} + log(1/\delta))}{\epsilon'^2} \cdot \frac{n}{x} \tag{3.7}$$

and S is the output $NodeSelection$ returns, then if $OPT < x$, $\frac{n}{\theta} \cdot M_{\mathcal{R}}(S) < (1 + \epsilon') \cdot x$, w.p. at least $(1 - \delta)$.

Proof. Let x_i be a random variable for each $R_i \in \mathcal{R}$ defined as, $x_i = \frac{w(R_i) \cdot \mathbb{I}(S \cap R_i \neq \emptyset)}{w_{max}}$, where w_{max} is the maximum weight possible for any RR set. Thus, $0 \leq x_i \leq 1$, which ensures the martingle property. Now let $F_{\mathcal{R}}(S) = \frac{M_{\mathcal{R}}(S)}{w_{max}}$, $p = \mathbb{E}[F_{\mathcal{R}}(S)]$ and $\alpha = \frac{(1+\epsilon') \cdot x}{np \cdot w_{max}} - 1$, Using Lemma 17 and linearity of expectation,

$$\begin{aligned} p = \mathbb{E}[F_{\mathcal{R}}(S)] = \mathbb{E}[\frac{M_{\mathcal{R}}(S)}{w_{max}}] = \mathbb{E}[\rho(S \mid S^p)]/(w_{max}) \\ \leq OPT/(w_{max} \cdot n) \leq x/(w_{max} \cdot n) \end{aligned}$$

Consequently $\alpha > \epsilon' \cdot x/(np)$ and from Lemma 6 of [127],

$$Pr[\frac{n}{\theta} \cdot M_{\mathcal{R}}(S) \geq (1 + \epsilon') \cdot x] \leq \delta/\binom{n}{b'}$$

Finally by applying union bound we get $\frac{n}{\theta} \cdot M_{\mathcal{R}}(S) < (1 + \epsilon') \cdot x$, w.p. at least $(1 - \delta)$. □

	NetHEPT	Douban-Book	Douban-Movie	Orkut	Twitter
# nodes	15.2K	23.3K	34.9K	3.07M	41.7M
# edges	31.4K	141K	274K	117M	1.47G
avg. deg.	4.13	6.5	7.9	77.5	70.5
type	undirected	directed	directed	undirected	directed

Table 3.2: Network Statistics

Thus by setting λ' using Eq. (3.8), we get Theorem 2 of [127]

$$\lambda' = \frac{(2 + \frac{2}{3}\epsilon') \cdot (\log \binom{n}{b'} + \ell' \cdot \log\ n + \log\log_2\ n) \cdot n}{\epsilon'^2}, \tag{3.8}$$

The rest of the proof is similar to that of IMM, which gives us the following result.

Theorem 9. *Let $\boldsymbol{\mathcal{S}}^P$ be a partial allocation on the inferior items. Let $\boldsymbol{\mathcal{S}}^{Grd}$ be the allocation of the superior item produced by* SupGrd. *Given $\epsilon, \ell > 0$, we have $\rho(\boldsymbol{\mathcal{S}}^{Grd} \cup \boldsymbol{\mathcal{S}}^P) \geq (1 - \frac{1}{e} - \epsilon)\rho(\boldsymbol{\mathcal{S}}^A \cup \boldsymbol{\mathcal{S}}^P)$ w.p. at least $1 - \frac{1}{|V|^\ell}$, where $\boldsymbol{\mathcal{S}}^A$ is any arbitrary allocation.*

Running time: Let w_{min} be the minumum weight of an RR set. Then using Lemma 9 of [127], the expected total time to generate $\mathcal{R}$ is determined by,

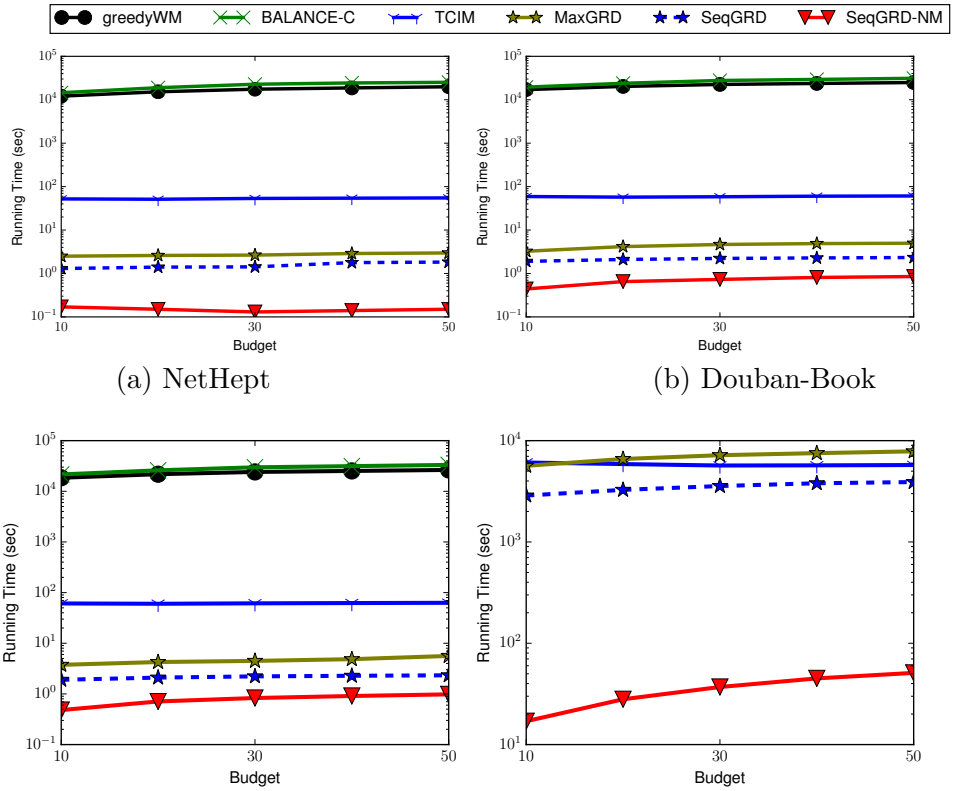
$$\begin{aligned}\mathbb{E}[\sum_{R \in \mathcal{R}} wid(R)] &= \mathbb{E}[|\mathcal{R}|] \cdot EPT \\ &\leq O((b' + \ell) n \log\ n \cdot \epsilon^{-2})/OPT \cdot \frac{m}{n} OPT/w_{min} \\ &= O((b' + \ell)(n + m) \log\ n \cdot \epsilon^{-2}/w_{min})\end{aligned}$$

Notice that generating an RR-set from scratch for the final node selection (line 8), following the fix of [33], only adds a multiplicative factor of 2. Hence the overall asymptotic running time to generate $\mathcal{R}$ remains unaffected.

3.6 Experiments

3.6.1 Experiment Setup

All our experiments are run on a Linux machine with Intel Xeon 2.6 GHz CPU and 128 GB RAM.



(a) NetHept (b) Douban-Book

(c) Douban-Movie (d) Orkut

Figure 3.4: Running times of greedyWM, Balance-C, TCIM, MaxGRD, SeqGRD **and** SeqGRD-NM **(on Configuration** 1)

3.6.1.1 Networks

Our experiments were conducted on five real social networks: NetHEPT, Douban-Book, Douban-Movie, Twitter, and Orkut, whose characteristics are summarized in Table 3.2. Of these, NetHEPT, Douban-Book, and Douban-Movie are benchmarks in IM literature [97], while Twitter and Orkut are two of the largest public networks available at [123].

3.6.1.2 Algorithms compared

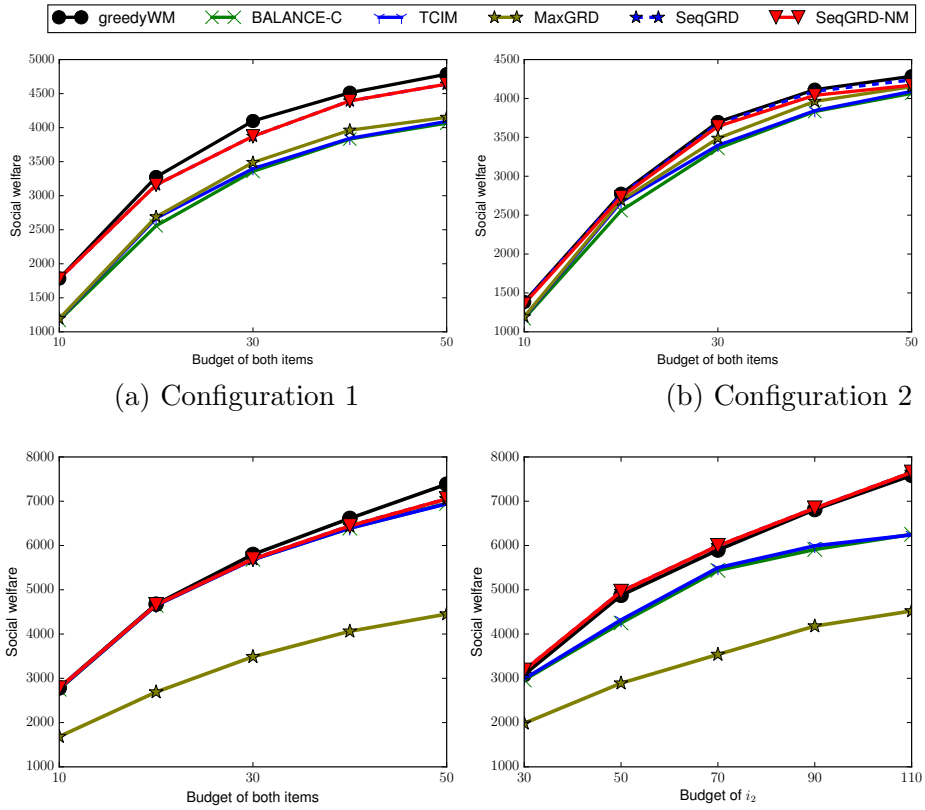
In the experiments our four algorithms – SeqGRD, SeqGRD-NM, MaxGRD, and SupGRD are compared against three baselines – TCIM, Balance-C and greedyWM. There is no previous work that can deal with both arbitrary degree of competition and multiple items in propagation. Our first two baselines each covers one aspect. TCIM [95] in particular assumes a propagation model which is an extension of the IC model under pure competition. It can, however, handle more than two items. Given fixed seed sets of other competing items, TCIM selects seeds of an item under a budget constraint, such that the number of adoptions of that item is maximized. When we run TCIM for multiple items, we select seeds for each of the items one by one, while keeping the seeds of other items fixed and then report the allocation that produces the maximum welfare.

In contrast, Balance-C [57] does not assume pure competition, but it works only when number of items in propagation is two. Given an initial seed placement of the two items, Balance-C chooses the remaining seeds such that at the end of the propagation, the number of nodes seeing either both the items or none, is maximized. Thus for competing ideas, Balance-C ensures that there is a balanced exposure of the two ideas to the most number of nodes. It is non-trivial to extend Balance-C for more than two items hence we compare against it only in two item set up.

Both TCIM and Balance-C aim to maximize adoption count, not social welfare. Our third baseline greedyWM maximizes the social welfare directly. It greedily selects iteratively the (node, item) pair that maximizes the marginal social welfare, till the budgets are exhausted. Below, by deterministic utility of an itemset I, we mean $\mathcal{V}(I) - \mathcal{P}(I)$, i.e., its utility with the noise term ignored.

3.6.1.3 Default parameters

Following previous works [74, 110] we set probability of edge $e = (u, v)$ to $1/d_{in}(v)$, where $d_{in}(v)$ is the in-degree of node v. Unless otherwise specified,



(a) Configuration 1 (b) Configuration 2

(c) Configuration 3 (d) Configuration 4

Figure 3.5: Expected social welfare in four configurations (on the Douban-Movie network)

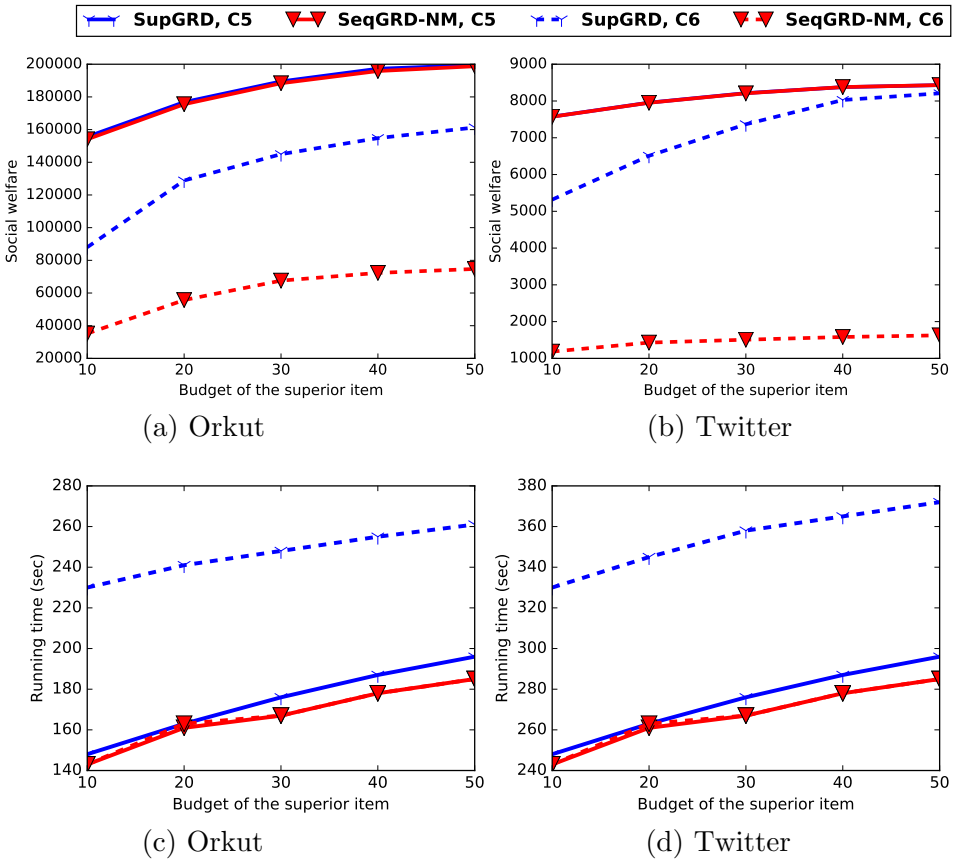
we use $\epsilon = 0.5$ and $\ell = 1$ as our default in all the algorithms that use these parameters. We test the algorithms across a wide variety of utility configurations to cover different aspects of competition. We will describe the configurations as we present the corresponding experiments. Whenever marginal gains are required, we run 5000 simulations and take the average result.

3.6.2 Experiments with two items

For our first set of experiments we restrict the number of items to two so that we can compare against all of the mentioned baselines. We also consider four different configurations to capture different kinds of competition. The details of the configurations are given in Table 3.3. In configurations C1 and C2, the items exhibit pure competition. In C1, items have comparable individual utility. In C2, the difference between individual utility is high: i's deterministic utility is 1, 10 times higher than that of j. C3 and C4 exhibit soft competition. Except for C4, in all configurations we consider the same budget for both items (uniform); budget is varied from 10 to 50 in steps of 10. In C4, we fix the budget of i to 50 and vary j's budget (non-uniform) from 30 to 100 in steps of 20. We assume $\boldsymbol{\mathcal{S}}^p = \emptyset$ in these configurations. Since it does not meet constraints required by SupGRD, we defer the comparison until Section 3.6.2.3.

3.6.2.1 Running time

First we compare the running time of the algorithms using C1 as a representative case. Fig. 3.4 shows the result on four networks. SeqGRD-NM is orders of magnitude faster than other algorithms in every network. The reason is that SeqGRD-NM does not compute any marginal. Each marginal computation requires iterating over 5000 samples, which significantly increases the running time. For the same reason greedyWM and Balance-C exhibit exorbitantly high running time: they do not in fact complete in 6 hours on a large network like Orkut. Hence they are not included in Fig. 3.4(d). Except for SeqGRD-NM, none of the other algorithms scale to the largest network Twitter. We will compare SeqGRD-NM and SupGRD on Twitter later. Performance on other configurations shows similar trends, and is hence omitted for brevity.



(a) Orkut (b) Twitter

(c) Orkut (d) Twitter

Figure 3.6: Comparison between SupGRD and SeqGRD on C5 and C6 (a-b) Social welfare, (c-d) Running time

3.6.2.2 Social welfare

We now compare the expected social welfare achieved by the algorithms on the four configurations (Fig. 3.5). We show the results only for Douban-Movie , since the trend of the results is similar on other networks. In all configuration SeqGRD, SeqGRD-NM and greedyWM outperform all other algorithms. The difference in welfare is up to $3\times$ higher. MaxGRD in particular allocates just one of the two items. Thus when items exhibit soft competition (C3 and C4), ,MaxGRD performs significantly worse. Balance-C performs comparatively better under soft competition (C3), however for a non-uniform budget again its performance drops. TCIM on the other hand aims to maximize the adoption count of the item being allocated. Thus it also ends up allocating both the items in same seed nodes. This reduces the overall social welfare for configuration such C1, where both Balance-C and TCIM perform comparatively worse. Social welfare produced by greedyWM is consistently good, but its running time is exorbitantly high, which prohibits its applicability on any decently sized network. SeqGRD-NM on the other hand is the fastest algorithm, which produces similar welfare across all these four configurations. However, notice that in none of these four configurations, item blocking is effective. We will show later in Section 3.6.3.2 that in the presence of multiple items, when avoiding item blocking is critical, the performance of SeqGRD-NM deteriorates.

3.6.2.3 Comparison against SupGRD

In this experiment we compare SupGRD and SeqGRD-NM on the two largest networks, Orkut and Twitter. We use utility configurations of C1 and C2, but adopt the seed placements needed to meet the constraints required for SupGRD. Recall that for SupGRD the seeds for the inferior items need to be fixed. Hence we select the top 50 nodes using IMM and set them as seeds of j. Then, the seeds of i are selected using SupGRD and SeqGRD-NM with the budget being varied from 10 to 50 in steps of 10. We call these new configurations C5 and C6 respectively.

Since the top nodes in terms of the spread are given to j, these two cases pose a unique challenge of dealing with arbitrary degree of competition when maximizing welfare. When items' utilities are similar, in C5, new seeds of i should be chosen in a way that minimizes i's overlap with j propagation. Instead in C6, when i has much higher utility, it should be allocated to the top seed nodes. That way, the number of nodes that can be reached by i is much higher and that helps boost the overall social welfare. As can be

No	Price	Value	Noise	Budget
C1	$i = 3$	$i = 4$, $j = 4.9$ $\{i, j\} = 4.9$	$i : N(0, 1)$	Uniform
C2	$j = 4$	$i = 4, j = 4.1$ $\{i, j\} = 4.1$		Uniform
C3	$\{i, j\} = 7$	$i = 4, j = 4.9$ $\{i, j\} = 8.7$	$j : N(0, 1)$	Uniform
C4				Nonuniform

Table 3.3: Two item configurations

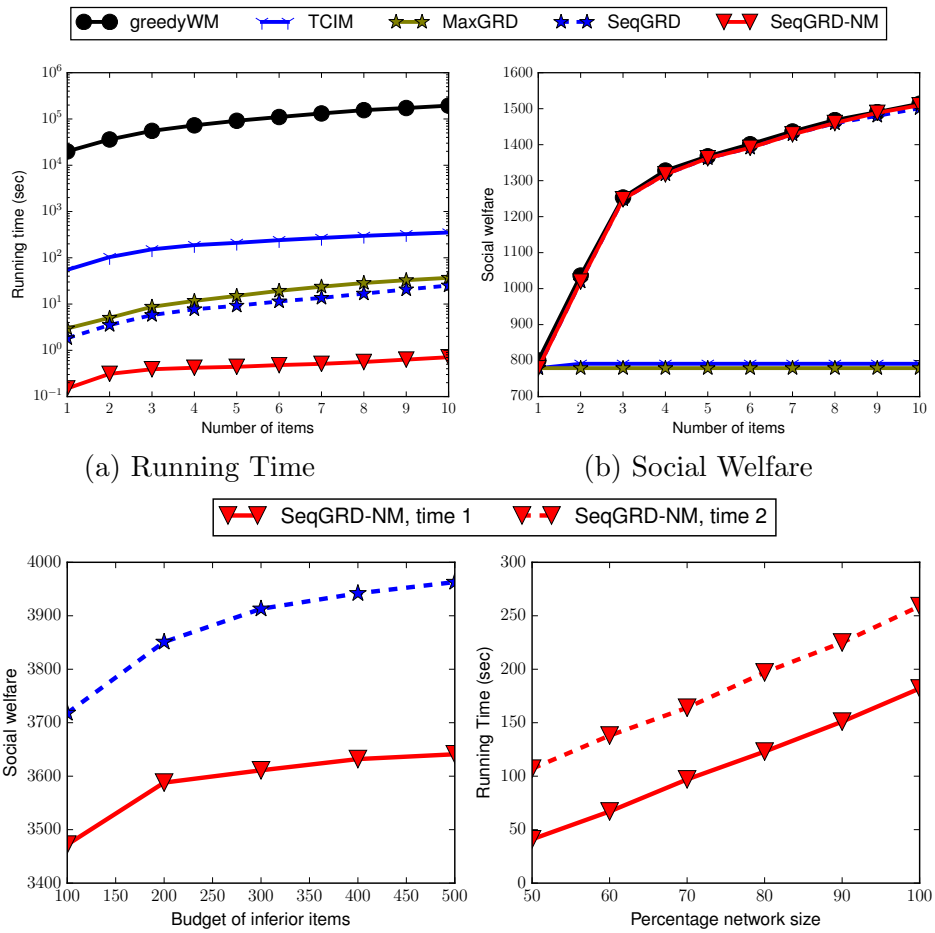
seen from our results next, that SupGRD can navigate through these varied "strategies", while SeqGRD-NM cannot.

Fig. 3.6 (a) and (b) shows the result on the expected social welfare on Orkut and Twitter respectively. "SeqGRD-NM-C5" (resp. "SupGRD-C5") refers to SeqGRD-NM (resp. SupGRD) on C5 and "SeqGRD-NM-C6" (resp. "SupGRD-C6") on C6. Notice that in C5 the welfare produced by the two algorithms are comparable. However in C6, where the gap between the individual utilities of the two items is higher, difference between the welfare of SupGRD and SeqGRD-NM is also larger. The reason for that is as follows. SeqGRD-NM uses PRIMA^+ to select the seeds of i. Consequently to maximize the marginal gain in spread, it minimizes the overlap in the spread of i and j and hence allocates i to lower ranked nodes in terms of spread. However i is the superior item, so allocating lower ranked nodes to i decreases the overall welfare.

Fig. 3.6(c) and (d) compares the running time of the two algorithms on Orkut and Twitter. Both the algorithms scale on these large networks. Unlike SeqGRD-NM, running time SupGRD depends on the utility configurations as well. As our running time analysis (Section 3.5.3) suggests, when the minimum utility of an item is lower, the running time of SupGRD is higher. However as can be seen, even on large networks, the difference in the running times is not very high: e.g., in configuration C6, the running time of SupGRD is only a $2\times$ that of SeqGRD-NM, whereas in C5 the running times are similar. To summarize, SupGRD addresses this unique challenge of dealing with an arbitrary degree of competition, with a slightly higher running time.

3.6.3 More than two items

Except for Balance-C, all the algorithms can deal with multiple items. In this section, we study their performances when the number of items is more



(a) Running Time (b) Social Welfare

(c) Social Welfare (d) Running Time

Figure 3.7: Multi-item experiments: Impact of number of items on (a) Running time, (b) Social welfare on NetHept. (c) Comparing performance of SeqGRD **and** SeqGRD-NM **on NetHept. (d) Scalability on Orkut**

$\mathcal{U}(i) = 2$	$\mathcal{U}(\{i,j\})<0$
$\mathcal{U}(\{j\}) = 0.11$	$\mathcal{U}(\{j,k\})<0$
$\mathcal{U}(\{k\}) = 0.1$	$\mathcal{U}(\{i,j,k\})$
$\mathcal{U}(\{i,k\}) = 2.1$	<0

Table 3.4: Three item configuration

than two. First, we show the impact of increasing the number of items on the running time and social welfare produced. Then we study how the algorithms behave under some challenging configurations designed using multiple items.

3.6.3.1 Impact of number of items

For this experiment, the configuration we test is as follows. Each individual item has expected utility of 1 and the items exhibit pure competition. Every item has budget 50 and $\boldsymbol{\mathcal{S}}^p = \emptyset$.

Fig. 3.7(a) and (b) show respectively, the running time and social welfare produced by the five algorithms. Since Balance-C cannot run on more than two items, it is omitted. Running time of algorithms greedyWM, TCIM, MaxGRD, and SeqGRD increases significantly w.r.t the number of items. As the number of items increase, the number of times marginal check is needed for these algorithms, also increases. The marginal check is the most time consuming portion in their running time. SeqGRD-NM on the other hand relies solely on RR-sets and does not do any marginal checks. Hence the growth in running time is not high. With higher number items, the difference between the running time of SeqGRD-NM and other algorithms, increases.

In terms of social welfare, TCIM and MaxGRD perform worse than the other algorithms. MaxGRD selects only one item in the final allocation, hence it misses out on the additional welfare that could come from allocating the remaining items. Similarly TCIM tries to maximize the spread of the last allocated item, at cost of propagation of other items. Thus their welfare does not increase with more items, unlike the other algorithms.

3.6.3.2 Effect of marginal check

In our experiments so far, social welfare of SeqGRD-NM has been similar to other algorithms that perform marginal checks. One exception being SupGRD (Section 3.6.2.3), but SupGRD assumes specific constraints that

item	p	q	$\mathcal{U}_D$
$\{indie\}$	0.107	na	7.0
$\{rock\}$	0.091	na	6.8
$\{industrial\}$	0.015	na	5.0
$\{progressive\_metal\}$	0.011	na	4.7

Table 3.5: Learned parameters

are not general. By not performing the marginal check, SeqGRD-NM runs much faster compared to other algorithms. This begs the question if there is any advantage of using the marginal check altogether. In this experiment we show how marginal check helps avoid item-blocking that SeqGRD-NM fails to circumvent.

For this experiment, we consider three items in the propagation. Their expected utilities are specified in Table 3.4. i has the highest expected utility, followed by j and k has the least. i and k exhibit soft competition hence bundle $\{i, k\}$ has a positive utility, but all other item bundles have negative utilities, exhibiting pure competition. We set the budget of i to 500, and increase the budget of j and k from 100 to 500 each in steps of 100 and study the effect on the welfare produced by SeqGRD-NM and SeqGRD.

Fig. 3.7(c) shows the result on the NetHept network. Both algorithms first allocate i as it has the highest individual utility. Then SeqGRD-NM allocates j next, however this allocation is "adjacent" to i since NetHept is small, and blocks propagation of i more. Since the utility of i is significantly higher than j, allocating j this way in fact causes a negative marginal. SeqGRD, using marginal check, postpones allocation of j. After i, it instead allocates k. Although k also has a low individual utility, because of soft competition, it does not block propagation of i and the marginal is non-negative. It later allocates j, which is now further apart from i, hence cannot block i's propagation. Thus SeqGRD produces a social welfare which is higher than that of SeqGRD-NM. Further, as the budget of j increases, the amount of blocking also increases, hence the welfare difference between the two algorithms also goes up.

3.6.3.3 Scalability of SeqGRD-NM

Our next experiment shows the impact of network size on SeqGRD-NM using Orkut with two types of edge probabilities: (1) $1/d_{in}(v)$ and (2) fixed 0.01. We use a uniform budget of 50 for all three items. Instead of using the full network, we use breadth-first-search to progressively increase the

network size so that it includes a certain percentage of the total nodes in the network. At 100%, the full network is used. Fig. 3.7(d) shows the results. "SeqGRD-NM, time 1" and "SeqGRD-NM, time 2" depict the running time of SeqGRD-NM on the two types of edge probabilites respectively. As the network size increases, the running time in both cases roughly has a linear increase.

3.6.4 Real item experiments

In this section, we learn the utilities of items from real dataset instead of the synthetic utilities used in earlier experiments. The dataset used is the LastfmGenres generated from the listening behavior of users of the music streaming service Last.fm [30, 85]. This dataset was used in [14] to learn the adoption probabilities of different items, where each genre is treated as an item. This dataset also echos our first motivating example in the introduction.

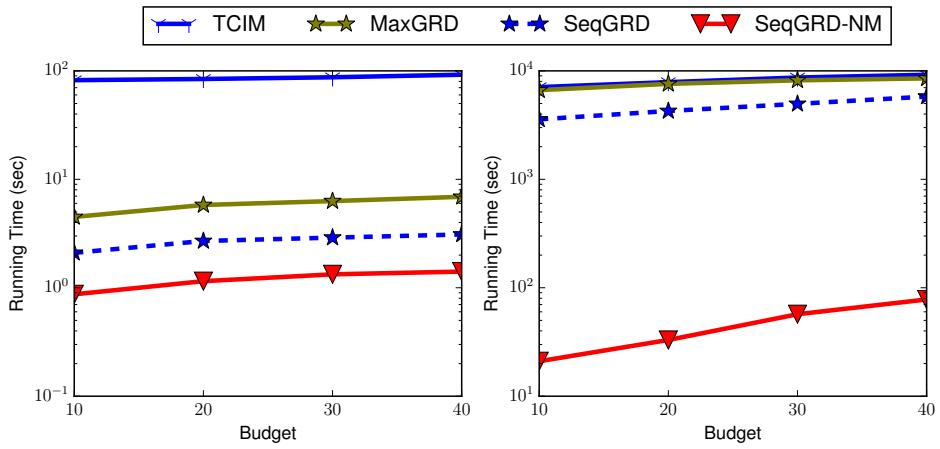
We next establish the connection between the adoption probabilities and the utilities, which enables us to learn the parameters using [14].

3.6.4.1 Learning the utilities

In [14], every item i is associated with an adoption probability p_i. Adoption probability of an itemset $I = \{i, ..., k\}$ is $p_I = \gamma_{|I|} \prod_{j \in I} p_j + q_I$, where q_I is a correction received depending on the way items in I interact with each other: if the items are complementary, then the correction is positive, if competing then it is negative, and 0 if the items are independent. These probabilities and corrections are learnt in [14] from the dataset of how frequently items are selected together by the users.

According to Observation 2.2 of [14], $p_i = e^{v_i} / \sum_j e^{v_j}$, where v_i is the expected utility of item i as per our utility model. Given a set of learnt p_i, we first set $\sum_j e^{v_j} = 10000$. Then for every i, we set $v_i = \ln(10000 \cdot p_i)$. We choose the number 10000 to ensure that the corresponding utilities are positive. Finally we set the expected utility of item i, $\mathcal{U}(i) = v_i$. Next for an itemset $I = \{i, ..., k\}$, [14] learns two parameters $\gamma_{|I|}$ and q'_I. By using $q_I = \gamma \cdot q'_I$ the probability of adopting the bundle, p_I is derived. The expected utility of the bundle is similarly set to be $\mathcal{U}(I) = \ln(10000 \cdot p_I)$. Notice that the exact values of utilities are not as important as the relative order of utilities of different itemsets. The way utilities are learnt is in correspondence with the adoption probabilities learned in [7].

Table 3.5 shows the utilities of four different items (i.e., genres) in the



(a) NetHept (b) Orkut

(c) Nethept (d) Orkut

Figure 3.8: Performance of TCIM, MaxGRD, SeqGRD **and** SeqGRD-NM **on real utility configurations (Table 3.5)**

dataset: *rock, indie, industrial and progressive_metal*, learned using the above described method. Larger bundles are either not present in the dataset or have smaller learned utilities compared to the individual items in the bundle, suggesting that items are in pure competition in our utility model.

Network	Budget	Algorithm	Real Utility Configuration (as shown in Table 3.5)					Synthetic Utility Configuration (as shown in Table 3.4)			
			indie	rock	industrial	progressive_metal	welfare	i	j	k	welfare
NetHEPT	10	RR	203	217	191	196	4473.08	277	244	234	513.2
		Snake	204(+0.005)	201(-0.081)	207(+0.083)	195(-0.005)	4458.64(-0.003)	258(-0.068)	246(+0.008)	261(+0.115)	478.4(-0.068)
		SGRD-NM	255(+0.252)	199(-0.082)	188(-0.016)	**165(-0.158)**	4951.8 (+0.112)	306(+0.105)	220(-0.098)	227(-0.030)	577.4(+0.125)
NetHEPT	40	RR	496	493	491	475	10795.3	667	576	645	1227.3
		Snake	483(-0.026)	496(+0.006)	488(-0.004)	488(+0.027)	10758.2(-0.003)	648(-0.028)	581(+0.009)	669(+0.037)	1194.5(-0.027)
		SGRD-NM	673(+0.357)	499(+0.012)	419(-0.147)	**365(-0.189)**	11264.5(+0.043)	800(+0.199)	510(-0.114)	514(-0.203)	1510.6(+0.230)
Orkut	10	RR	37790	38888	38331	34711	828368.2	69151	49730	67405	110032.5
		Snake	38241(+0.012)	37401(-0.038)	39818(+0.039)	34260(-0.013)	828235.4(-0.002)	67648(-0.021)	50511(+0.016)	68510(+0.016)	107227.7(-0.026)
		SGRD-NM	50800(+0.344)	40837(+0.050)	31189(-0.186)	26895(-0.225)	864154.3(+0.040)	76784(+0.110)	50219(+0.010)	57199(-0.151)	124210.9(+0.129)
Orkut	40	RR	58142	58586	59939	54607	1276650.6	119039	83291	113359	183853.2
		Snake	57211(-0.016)	56922(-0.028)	61603(+0.028)	55538(+0.017)	1272190.7(-0.035)	117454(-0.013)	82937(-0.043)	115338(+0.018)	180269.4(-0.020)
		SGRD-NM	106876(+0.838)	54909(-0.063)	42218(-0.296)	**27272(-0.501)**	1397770.8(+0.095)	150926(+0.268)	**63577(-0.237)**	87480(-0.228)	**253427.9(+0.378)**

Table 3.6: Comparison of adoption counts of different items and the overall social welfare

3.6.4.2 Results using real parameters

We use the learned utility configuration to compare the social welfare produced by the algorithms on two networks, namely NetHEPT and Orkut. For the experiment we set uniform budget for all the four items, which varies from 10 to 40 in steps of 10. The algorithms compared in this section are, TCIM, MaxGRD, SeqGRD and SeqGRD-NM. The results are shown in Fig 3.8.

In terms of running time the results are similar to our previous experiments (Fig 3.8(a)-(b)). SeqGRD-NM outperforms the other algorithms by orders of magnitude, since it does not require the time consuming marginal gain computation. For social welfare, notice that the real utility configuration exhibits pure competition. As noted earlier, under pure competition, social welfare produced by SeqGRD and SeqGRD-NM coincide. MaxGRD and TCIM on the other hand typically encourage the adoption of one single item. Hence the difference in social welfare produced by these algorithms compared to SeqGRD-NM is higher since the number of items are also more than the previous configurations we used.

3.6.4.3 Social welfare vs adoption

Our final set of experiments compare the relationship between the social welfare and item adoptions. In particular, we want to investigate whether maximizing welfare for competing items could result in a significant drop in the number of item adoptions. For this experiment, we focus on two utility configurations – (i) Real utility of Table 3.5, which exhibits pure competition and (ii) Synthetic utility of Table 3.4, which exhibits a mix of partial and pure competition. NetHEPT and Orkut are the two networks used and each item's budget is set to two different values, 10 and 40.

We compare our algorithm SeqGRD-NM against two baselines. After selecting the seed nodes, the first baseline allocates items to the nodes in a round robin manner, hence it is called $Round-robin$. The second baseline, called *Snake*, is similar to Round-robin, but it flips the order for every successive sequence of allocations. To illustrate, if there are 4 seed nodes $s_1, ..., s_4$, in order, and two items i, j, SeqGRD-NM allocates as $s_1 : i, s_2 : i, s_3 : j, s_4 : j$, Round-robin allocates as $s_1 : i, s_2 : j, s_3 : i, s_4 : j$ and Snake allocates as $s_1 : i, s_2 : j, s_3 : j, s_4 : i$. Table 3.6 shows the adoption count of each item and the social welfare produced by these algorithms under different configurations.

In terms of the social welfare objective, SeqGRD-NM dominates across

all different configurations. Round-robin produces the next highest welfare. Hence we report the fractional change (+ denotes increase and − denotes decrease), in comparison to Round-robin, next to each entry of the table. The entries that deserve more attention are highlighted in green.

As can be seen, the total number of adoptions of all the items remains the same across all three algorithms. However, SeqGRD-NM generally increases the adoption of the superior product to increase the welfare, while reducing the adoption of the inferior item. On NetHEPT, for budget 10, the maximum drop in adoptions happens for the most inferior item (progressive_metal), by 15.8%. For a higher budget, the drop increases (to 18.9%), because when budget increases for the superior item, SeqGRD-NM allocates lower ranked seeds for the inferior item.

The highest drop in adoption i.e., 50.1%, also happens for the item progressive_metal for budget 40 on Orkut. This is because when items are purely competing, number of items is high and each item has a large budget, the inferior items' seeds are in fact much lower ranked. However, if it exhibits partial competition with a superior item, then leveraging it the adoption does not decrease that much. That is why in Orkut even when the budget is 40, for the synthetic utility configuration, the highest drop in adoptions for the inferior items is only 23.7%. Also notice SeqGRD-NM produces significantly higher social welfare compared to the baselines, the increase being up to 37.8%. In summary, we see that our welfare maximization algorithm provide more adoptions to the superior items and fewer adoptions to the inferior items, but the amount of change is not too drastic. We argue that this is the "price" of enhancing the overall user satisfaction; also the drop in the adoptions of the inferior items is exactly because they are not as competitive.

To conclude this section, we generally observe that: (a) when the conditions required by SupGRD are met, it is the best option providing the best social welfare and competitive running time; (b) in the general case, SeqGRD-NM performs well in most cases and has the best running time, but when item blocking is significant, its marginal-checking version SeqGRD could provide better social welfare, at the cost of higher running time; (c) MaxGRD could be used to enhance the theoretical guarantee when the utility difference is high, but its superiority is not typically observed in large networks; (d) our algorithms outperform all baselines on social welfare and running time and scale to large networks. Our algorithms achieve superior welfare at the expense of a reasonable drop in the adoption count of inferior items, keeping the total adoption count unchanged.

3.7 Conclusions and future work

In this chapter, we studied the problem of maximizing social welfare over competing items under the UIC model. The problem is not only NP-hard but is also NP-hard to approximate within a constant factor. Further we find that due to conflicting requirements, it is challenging to design a single algorithm that can work effectively for all different utility configurations. Yet we propose a cohort of efficient algorithms that not only provide approximation guarantees but also scale well to real large networks, and their performance is validated through extensive experiments on real-world networks.

Although welfare maximization under competition ensures that users' total utility from adoptions is maximized, it does not directly ensure fairness. For a campaigner who often pays for advertising, ensuring that her item is seen at least by a certain number of users is critical. While fairness in IM has been studied recently, incorporating fairness in social welfare maximization will be an interesting challenge.

Further, this chapter and Chapter 2 studied competition and complementarity in isolation. Designing algorithms for an arbitrary mix of competing and complementary items is an intriguing problem. The next chapter demonstrates that in the context of mitigating the filter bubble problem, a utility configuration, combining competition and complementarity, naturally arises. It proposes algorithms under such configuration.

Chapter 4

Mitigating the filter bubble problem using a utility driven diffusion model

4.1 Introduction

With the proliferation of social networks, new ways have emerged to provide users with an abundance of information [29] and engage them in the propagation of the information [3, 18]. Although access to information has never been easier, social media have also led to increased societal polarization [53]. Two common polarization effects that have been widely observed in the social networks are, (i) echo chambers [10, 59], where users are found to be confined to information only from like-minded individuals, and (ii) filter bubbles [112, 115], where, in an attempt improve user's engagement, algorithms present the user only those type of information that aligns with the user's viewpoint. Existence of such echo chambers and filter bubbles, impedes natural and fair opinion formation [103]. This can in turn inhibit free and open discourse among people with different viewpoints and can lead to one-sided policy decisions [117], and potentially lead to reduced social trustworthiness [108]. Attempts to address these issues have resulted in research across several dimensions. Earlier works focused on measuring the extent of polarization [2, 4, 68, 99], identifying the groups of users to whom "counter-information" could be propagated in order to balance their exposure to opposing viewpoints, and also what kind of counter-information needs to be propagated [50, 61, 131]. By creating new links among users of opposing viewpoints, researchers have attempted to mitigate the echo chamber problem [5, 56, 103, 137]. To address the filter bubble problem, a recent body of research has effectively used the influence propagation paradigm in social networks [57, 98, 132].

Influence propagation is extensively studied in the context of the influence maximization (IM) problem. Starting from a small set of users,

called seeds, in a given social network, influence cascades unfold following a stochastic diffusion model which specifies how influence propagates from one user to another in the network. While the classical IM objective is to select k seeds to maximize the expected number of influenced users, works related to the filter bubble problem aimed at developing methods for balancing [57, 132] and diversifying [98] information exposure, while the information spreads through the network following the same stochastic information propagation model used in the classical IM literature. As a result, there are several limitations of these works.

Many studies on the filter bubbles problem have reported that items containing opinion-challenging information spread less extensively than other items [59, 130], i.e., from the propagation point of view, some items are competing by nature. As an example, suppose that a user has been exposed to article A arguing for relaxing gun control. Assume that the user is swayed by the article and has adopted the viewpoint that it promotes. Consider an article B arguing that gun ownership should be more restricted citing studies revealing a strong correlation between gun ownership and violent crime. If the user is then exposed to article B, she may not readily "adopt" the viewpoint of article B at the same time! Thus, she may not propagate information about article B to her social peers. Earlier works completely ignore this aspect. Instead, they tacitly assume that a user, once influenced by a peer, and exposed to two opposing viewpoints, will adopt both of them. Thus, they focus on maximizing the objective of balancing the exposure of users to opposing viewpoints, assuming no competition between their adoption. Therefore, in order to truly capture the effect of countering a filter bubble, it is necessary to model two seemingly conflicting requirements: (i) in terms of propagation, the items (as in opposing viewpoints on an issue) need to be competing, and (ii) the objective function measuring the effectiveness of a strategy for countering the filter bubble needs to treat the two items as complementary in that the reward for a user adopting both items should be significantly greater than the reward for adopting any one item. *To our knowledge, none of the existing works is able to capture this.* In addition to lacking competition, the configurations used in the prior works have other unnatural assumptions that do not capture the true essence of the filter bubble. It was noted in [132] that the objective of [57] is not natural as it rewards a strategy for users who do not adopt any item! Similarly, [98] uses an objective where the score improves even when users adopt more items that are with the same or similar (political) leaning. In [132] seeds are constrained to be disjoint, as a result influential nodes that are selected as seeds are forced to stay in the filter bubbles.

In Chapter 2 this thesis introduced a propagation model, called utility-driven independent cascade (UIC), where influence is decoupled from item adoption and users' adoption decision is driven by the utility of itemsets. In this chapter, we use the expressive power of UIC to meet the needs of the filter bubble problem. We set the utility function to be a combination of complementary and competitive aspects. The first component of the utility is a complementary reward function that awards a reward for a user adopting two items from opposite viewpoints, that is significantly higher than the user adopting any one item. However, the second component is a competition parameter which controls the probability of a user adopting the second item when she has already adopted the first item. That is, user's adoption decision captures the inherent competition between opposing viewpoints. The objective is to maximize the social welfare, i.e., the sum of utilities of all the users at the end of the propagation. Although UIC helps to realistically model the requirements of the filter bubble problem, it poses some unique technical challenges.

In the previous two chapters, studies on UIC were restricted either to only complementary utility (Chapter 2) or only competing utility (Chapter 3). Even in the case of utility function for only competing items, it was shown that welfare maximization is NP-hard to approximate within any constant factor (Section 3.4). Hence it is no surprise that for a utility which is a combination of competing and complementary functions, the optimization problem is difficult. Specifically, the objective function is neither monotone nor submodular, hence approximation algorithm cannot be obtained by leveraging these properties. In fact, unlike Chapter 3, in this Chapter it is shown that the objective function remains non-monotone and non-submodular even under several simplifying assumptions.

We therefore develop two instance-dependent approximation algorithms for the general problem. We show that the bound provided by our first algorithm, SpreadGRD, is a tight bound. Our second algorithm, SandwichGRD, leverages sandwich approximation [97] after bounding the non-submodular objective function. However, none of these algorithms explicitly optimizes for the social welfare objective. To that end, we design a non-trivial heuristic – WelfareGRD using the RR set machinery which is extensively used by the state-of-the-art IM algorithms [110, 127]. WelfareGRD owes its efficiency to the fact that it directly extends RR sets for the welfare maximization problem.

In summary, we make the following contribution in this chapter:

- We extend the UIC model for the filter bubble problem. In Section

4.3, we introduce competition parameter and reward parameters which drive the adoption and show how they interact with the underlying propagation model. Then we formally develop the FBWelMax problem to mitigate the filter bubble problem.

- In Section 4.4 we show that the objective of FBWelMax is neither monotone nor submodular for the general case and even under simplifying assumptions. Therefore it is difficult to design a constant approximation algorithm for our problem.
- Since it is difficult to get a constant approximation, in Section 4.5 we devise two instance-dependent approximation algorithms for FBWelMax, namely, SpreadGRD and SandwichGRD. We also show that the bound of SpreadGRD is a tight bound.
- Later in Section 4.5, we also develop an effective heuristic, called WelfareGRD. WelfareGRD makes non-trivial extensions to RR sets such that RR sets can be used for the welfare maximization objective of FBWelMax.
- We conduct extensive experiments on five real-world networks in Section 4.6. We test our algorithms against two state-of-the-art algorithms for countering filter bubbles under several different configurations. Our results demonstrate the efficacy and efficiency of our algorithms over the baselines in all those configurations.

Related work is discussed in Section 4.2 and we conclude this chapter in Section 4.7.

4.2 Background & Related Work

As discussed in the earlier chapters, in IM problem, a social network is represented using a directed graph $G = (V, E, p)$ where users are nodes of V, their connections are edges of E, and function $p : E \to [0, 1]$ is used to denote the influence probabilities between nodes. Influence propagates following a diffusion model; one such diffusion model that is widely used as a discrete-time diffusion model is Independent Cascade (IC) [35, 83]. Diffusion under IC unfolds in discrete time steps as follows. Given a seed set $S \subset V$, at time $t = 0$, only the seed nodes in S are active. For $t > 0$, if a node u becomes active at $t - 1$, then it makes one attempt to activate its every inactive out-neighbor v, with success probability $p_{uv} := p(u, v)$. The diffusion stops when no more nodes can become active.

The expected number of active nodes at the end of diffusion, i.e., the *influence spread* for a given seed set $S \subset V$, is denoted by $\sigma(S)$. Under a budget constraint k, *influence maximization* (IM) problem is to find a seed set $S \subset V$ with $|S| \leq k$ such that the influence spread $\sigma(S)$ under the specified diffusion model is maximized. More details related to the classical IM problem is discussed in Chapter 1. Remainder of this section focuses on works related to the echo chamber and welfare maximization problems.

4.2.1 Echo chamber and filter bubble

A number of recent studies characterized the presence of echo chambers and filter bubbles in social networks. Echo chambers form when users only interact with like-minded individuals and get exposed to information only from them [10, 59]. When search and recommendation algorithms present personalized content to users in an effort to improve accuracy/relevance based on the content the user has consumed so far, users tend to get exposed to a narrow world view, as a result of which filter bubbles are formed [10, 115]. Both echo chambers and filter bubbles lead to polarization in discourse. Numerous research papers have measured how strongly these phenomena manifest themselves on social networks by quantifying polarization in the network [2, 4, 41, 43, 58, 68, 99].

Mitigating filter bubbles and more generally polarization is important. The mitigation task poses multifaceted research questions such as which users to target, what viewpoints to promote, or how best to present possibly opposing viewpoints to users [94]. Works such as [5, 56, 103, 137] attempted to solve the echo chamber problem by building connections between users from groups of opposing viewpoints. These works use the opinion-formation model and assume that the underlying graph can be tweaked by adding or removing edges.

In contrast, this chapter leverages the power of influence cascade using a propagation model, to solve the filter bubble problem. A recent body of work [57, 98, 132], which is most similar to the work of this chapter, aims to tackle the filter bubble problem using an influence propagation setting; they aim to ensure a balanced exposure of conflicting opinions to the maximum number of users of the network. However these works do not differentiate between exposure and adoption: a user being exposed to opposing viewpoints is not guaranteed to adopt both view points and certainly not share both of them on with her social peers. One significant aspect lacking in the propagation model of the above body of works is the presence of competition among the different opinions that are spreading. Put differently,

they assume that when exposed to different viewpoints, a user will share all of them with their peers! There are several studies on filter bubbles that have established that items containing opinion-challenging information spread less than other items [59, 60, 130]. The UIC propagation model enables to adequately capture the competition effect, by separating awareness from adoption. Further, the objectives studied in the previous papers do not align well with mitigating the filter bubbles problem. E.g., the framework of Matakos et al. [98] is based on an exposure quality function which tracks how balanced information exposure is. Surprisingly, it yields a higher quality score even when a user is exposed to more items with the same (e.g., political) leaning! Exposure to more items of the same leaning should intuitively make the balance lower as it increases the skew. This chapter addresses these concerns by using reward parameters to more accurately capture the needs of the filter bubble problem.

Orthogonal to these works, the fairness aspect of IM has also received significant attention [50, 61, 131]. In these studies, there are no competing items and the primary goal is to identify different node communities in the network and ensure that all the different communities are well represented in the influence coverage.

4.2.2 Social welfare maximization

Several works in economics have characterized utility-driven auction design and allocations [1, 20, 105, 113]. These works focus on finding an allocation of items to users for a given itemset and set of users, and the utility functions of users for various subsets of items, such that the sum of utilities of users, is maximized. Since the problem is intractable, approximation algorithms have been developed [52, 81, 84]. However, none of these works consider a social network and the effect of recursive propagation of item adoptions by its users.

Sun et al. [124] study participation maximization where an item is a discussion topic, and adopting an item means posting or replying on the topic. They, however, assume that items are independent, and also their budget constrains the number of items each seed node can be allocated with, rather than the number of seeds each item can be allocated to as studied in our model. As a result any number of nodes can be selected as seed nodes in [124].

Considering the host's perspective, [8] directly maximizes the revenue earned by a network host, whereas [9] aims to minimize the regret of seed selection. These works do not consider the overall social welfare. Utility-based

adoption decisions of users are also not part of their formalism. Welfare maximization on social networks has been studied in a few recent papers[19, 124].

In [19], Bhattacharya et al. consider item allocations to nodes for welfare maximization in a network with network externalities. A user's valuation of an item is affected by the network externality, i.e., the number of her direct one- or two-hop neighbors in the network adopting the same item. Their model does not consider the effect of recursive propagation nor competition. In addition, they do not consider budget constraints.

The previous two chapters of this thesis, namely Chapter 2 and Chapter 3, study welfare maximization under viral marketing using the UIC propagation model. However, the works of those chapters, require items to be complementary only or competing only. In contrast, as we highlighted in the introduction, mitigating filter bubbles requires a combination of both functionalities. As a result, the problem is more difficult to solve; even for only competing items the problem is shown to be NP-hard to approximate in Section 3.4. However, when items are only competing, under some simplifying assumptions, algorithm having constant approximation can be designed 9. The problem studied in this chapter remains difficult to approximate even after several simplifying assumptions. Still, we propose an instance-dependent approximation bound and prove that our bound is tight.

In summary, to our knowledge, *the study of this chapter is the first to use the power of a influence propagation model that have user decision engines, for holistically solving the filter bubble problem, where the challenges of the filter bubble problem is more accurately modeled using a combination of competition and complementary functions.*

4.3 UIC-FB Model for filter bubble

In this section, we first present the adaptation of *utility driven independent cascade* model (UIC for short) proposed in [12], for the filter bubble problem, which we call the UIC-FB model. We use a and b to denote the two polarized opinions (items) being propagated. After describing the propagation model, we formally state the new problem that we study under the UIC-FB diffusion model.

4.3.1 The UIC-FB Model

UIC amends item propagation using independent cascade with utility driven adoption decision of nodes. Each node maintains two sets of items, namely

awareness set and *adoption set*. Propagation or seeding populates the awareness set of a node i.e., the set of items that the node has been informed about. The node then adopts a subset of the awareness set which constitutes the adoption set. UIC-FB inherits the concept of awareness and adoption sets. Additionally, it uses *competition parameter* and *reward parameters* using which a node computes its utility. Utility governs the subset of items from the awareness set that a node adopts, i.e., it regulates whether a node adopts only the first item it is made aware of or both items it is made aware of. We describe the competition parameter and reward parameters next.

Competition parameter.

A node is said to be *polarized* when it has adopted exactly one of the two propagating items. If a node is not polarized, i.e., has not adopted any item yet, then it always adopts the first item it becomes aware of. However, if the node has already adopted an item (and therefore polarized), then it adopts the second item with probability c, where $0 \leq c \leq 1$, where c is a competition parameter. The competition parameter indicates the degree of competition between the two items, a lower value of c indicating that it is more difficult to penetrate an already polarized node with the second item, i.e., opposite viewpoint.

Reward parameters.

Our reward parameters capture the complementary aspect – a node that adopts both the items earns a higher reward than one adopting one item, which in turn is higher than adopting no item. Notice that this is in spite of the two items being competitive. We argue that this is a necessary requirement for faithfully capturing the effect of polarization and filter bubble. In particular, the reward for adopting no item is 0, for adopting one item is δ, and for adopting both items is $\delta + \Delta$, where $\Delta > \delta > 0$. We will explain our rationale for choosing such reward values in our design choices in Section 4.3.4 later.

Propagation model.

Budget vector $\vec{k} = (k_a, k_b)$ represents the budgets associated with the items, i.e., the number of seed nodes that can be allocated with each item. An *allocation* is a relation $\boldsymbol{\mathcal{S}} \subset V \times \{a, b\}$ such that $\forall i \in \{a, b\} : |\{(v, i) \in \boldsymbol{\mathcal{S}} \mid v \in V\}| \leq k_i$. Let $S_i^{\boldsymbol{\mathcal{S}}} := \{v \mid (v, i) \in \boldsymbol{\mathcal{S}}\}$ denote the *seed nodes* of $\boldsymbol{\mathcal{S}}$ for item i and $S^{\boldsymbol{\mathcal{S}}} := \bigcup_{i \in \{a,b\}} S_i^{\boldsymbol{\mathcal{S}}}$.

When the allocation $\boldsymbol{\mathcal{S}}$ is clear from the context, we write S (resp., S_i) to denote $S^{\boldsymbol{\mathcal{S}}}$ (resp., $S_i^{\boldsymbol{\mathcal{S}}}$). Further $\boldsymbol{\mathcal{S}}_a \subset \boldsymbol{\mathcal{S}}$ (resp. $\boldsymbol{\mathcal{S}}_b$), denotes the allocation involving only item a (resp. b).

The diffusion proceeds in discrete time steps, starting from $t = 1$. $\mathcal{R}^{\boldsymbol{\mathcal{S}}}(v, t)$

and $\mathcal{A}^{\boldsymbol{\mathcal{S}}}(v,t)$ denote the awareness and adoption sets of node v at time t. At $t = 1$, the seed nodes have their awareness sets initialized according to the allocation $\boldsymbol{\mathcal{S}}$ as, $\mathcal{R}^{\boldsymbol{\mathcal{S}}}(v,1) = \{i \mid (v,i) \in \boldsymbol{\mathcal{S}}\}$, $\forall v \in S^{\boldsymbol{\mathcal{S}}}$. Awareness sets of non-seed nodes are initially empty.

These seed nodes then adopt the items from the awareness set following the competition parameter. If a seed node becomes aware of the two items simultaneously, then it breaks the tie arbitrarily to decide which one of the two it adopts first. The propagation then unfolds recursively for $t \geq 2$ in the following way.

Once a node v adopts an item i at time $t-1$, it influences its out-neighbor u with probability p_{vu}, and if it succeeds and if i is not in u's awareness set, then i is added to the awareness set of u at time t. If u has not adopted any item yet, then u adopts i, w.p. 1, else w.p. c. If a node u is influenced by two in-neighbors at the same time t, then a random permutation o_v of u's in-neighbors is generated. Then, u is tested with each in-neighbor's adopted item following o_v to decide which item (a or b) u adopts first. If there is an in-neighbor of u, v such that v adopted both a and b at $t-1$, then u considers the items according to the same order of adoption used by v. There is no time delay between becoming aware and adopting an item. Adoption is progressive, i.e., once a node adopts an item, it cannot unadopt it later. The propagation converges when there is no new adoption in the network.

4.3.2 Utility of a node

We define utility to be the stochastic reward a node earns at the end of the propagation. The utility of node u is denoted as $\mathcal{U}(\mathcal{A}(u))$ where its adoption set is $\mathcal{A}(u)$. Recall that when $|\mathcal{A}(u)| = 0$, then the reward is 0, if $|\mathcal{A}(u)| = 1$, then reward is δ, and if $|\mathcal{A}(u)| > 1$, then the reward is $\delta + \Delta$, where $\Delta > \delta > 0$. However, note that when a node is aware of two items, it does not necessarily adopt both due the presence of the competition parameter c. This accurately reflects the challenge of filter bubble, where users are less likely to adopt item (opinion) from the opposite spectrum. Different from the deterministic reward parameter, utility takes into account the effect of c. Therefore, accounting for the competition parameter, the maximum possible stochastic utility of a node is $c(\delta+\Delta)+(1-c)\delta = \delta+c\Delta$, which is less than the maximum possible reward of $\delta + \Delta$.

4.3.3 Welfare maximization to mitigate filter bubble

Given a social network $G = (V, E, p)$ and a seed allocation $\boldsymbol{\mathcal{S}}$, we consider a utility-based objective called *social welfare*, which is the sum of all nodes' expected rewards of itemsets adopted by them after the propagation converges. Formally, $\mathbb{E}[\mathcal{U}(\mathcal{A}^{\boldsymbol{\mathcal{S}}}(u))]$ is the expected reward that a user u attains for a seed allocation $\boldsymbol{\mathcal{S}}$ after the propagation ends. The *expected social welfare* for $\boldsymbol{\mathcal{S}}$, is $\rho(\boldsymbol{\mathcal{S}}) = \sum_{u \in V} \mathbb{E}[\mathcal{U}(\mathcal{A}^{\boldsymbol{\mathcal{S}}}(u))]$, where the expectation is over both the randomness of propagation and competition parameter c.

In a social network, information bubbles are formed because of existing campaigns. To capture this we assume that the seed set of item a, S_a is fixed. Seeds of item b are to be selected so as to maximize the social welfare. We define the problem of maximizing expected social welfare, as follows.

Problem 3 (FBWelMax)**.** *Given $G = (V, E, p)$, competition parameter c, reward values δ and Δ, a fixed allocation of item a, $\boldsymbol{\mathcal{S}}_a$, and a budget k_b for item b, find a seed allocation $\boldsymbol{\mathcal{S}}_b^*$ for item b, such that $|\boldsymbol{\mathcal{S}}_b^*| \leq k_b$ and $\boldsymbol{\mathcal{S}}_b^* = \arg\max_{\boldsymbol{\mathcal{S}}_b} \rho(\boldsymbol{\mathcal{S}}_a \cup \boldsymbol{\mathcal{S}}_b)$.*

4.3.4 Design choices

The utility is higher when a node adopts both items. Therefore, our framework primarily incentivizes to select seeds in a way so that more nodes are informed about both the items, hence encouraging an algorithm to try to reduce the effect of a filter bubble. Additionally, utility is 0, when a node does not adopt any item. It avoids the unnatural scenario that some previous works [57] are susceptible to, whereby nodes are encouraged to adopt no item!

Also, note that our problem formulation reflects the "follower allocation" found in competitive IM literature [17, 95, 138], where, given a fixed seed set for one item, seeds of other item is selected. However, unlike classic competitive IM with a follower perspective, we are not trying to neutralize the propagation of any item. This is a property of the utility function we are using.

At this juncture, we would like to highlight another important design decision we take. Suppose that the given fixed a seeds are of low quality in terms of spread. Then the b seed selection objective has two possibilities to consider - (i) selecting b seeds that only maximizes co-adoptions, and therefore forced to be of inferior quality in terms of spread or (ii) selecting good quality b seeds that do not necessarily lead to many co-adoptions.

Note, if $\delta = 0$ then the second possibility does not arise, and the objective is then just maximizing co-adoption that one previous study has used [132]. In our work, we let our welfare-maximizing algorithms decide based on social welfare. The rationale behind our decision is as follows. Firstly, from a business perspective, where maximizing influence is the primary goal, having no reward for influencing with one item is unnatural. Thus setting $\delta > 0$ is a more natural choice. Further, in our attempt to minimize filter bubbles, we should not stop the spread of information altogether. Doing so prevents access to information which is a basic need. We show the effect of this decision empirically in Section 4.6.4.

4.3.5 An equivalent possible world model

For our analysis later, it will be useful to have a possible world interpretation of the UIC-FB propagation model.

Let $\langle G, M \rangle$ be an instance of FBWelMax, where $G = (V, E, p)$, and $M = (c, \delta, \Delta)$ denotes the set of model parameters.

A *possible world* $w = (w_1, w_2)$, consists an *edge possible world* (edge world) w_1, and an *adoption possible world* (adoption world) w_2. w_1 is obtained by sampling a deterministic graph from the distribution associated with G, where each edge $(u, v) \in E$ is sampled with an independent probability of p_{uv}. w_2 is obtained by establishing a random but fixed order o_v for overall the in-neighbors of node v and then sampling a value t_v, where $0 \leq t_v \leq 1$. When v is influenced by more than one in-neighbors at the same timestep t, it selects a neighbor following the order o_v and adopts the corresponding item as the first item. v adopts the second item after becoming aware of it only when $t_v \leq c$.

Note that propagation and adoption in w is fully deterministic. The *social welfare* of a given seed allocation $\boldsymbol{\mathcal{S}}$ in w is $\rho_w(\boldsymbol{\mathcal{S}}) := \sum_{v \in V} \mathcal{U}(\mathcal{A}_w^{\boldsymbol{\mathcal{S}}}(v))$, where $\mathcal{A}_w^{\boldsymbol{\mathcal{S}}}(v)$ is the adoption set of v at the end of the propagation in world w. The *expected social welfare* of an allocation $\boldsymbol{\mathcal{S}}$ is $\rho(\boldsymbol{\mathcal{S}}) := \mathbb{E}_w[\rho_w(\boldsymbol{\mathcal{S}})] = \mathbb{E}_{w_1}[\mathbb{E}_{w_2}[\rho_w(\boldsymbol{\mathcal{S}})]] = \mathbb{E}_{w_2}[\mathbb{E}_{w_1}[\rho_w(\boldsymbol{\mathcal{S}})]]$.

Further given a seed set S and an edge world w, we use $\phi_w(S)$ to denote the nodes reachable from S in w. Note that to realize the set $\phi_w(S)$, we do not require adoption world sampled. Therefore the reward of the nodes are still random.

4.4 Properties of UIC-FB

It is easy to see that FBWelMax is NP-hard.

Proposition 3. *FBWelMax in the UIC-FB model is NP-hard.*

Proof. We show that Influence Maximization under the IC model, a well-known NP hard problem [83], is a special case of FBWelMax: let $\boldsymbol{\mathcal{S}}_a = \emptyset$, set $\delta = 1$.[2] Hence, there will be only one item, b, in propagation. Additionally, since $\delta = 1$, an allocation of item b maximizes the expected social welfare iff the corresponding seed set maximizes the expected spread. □

Given the hardness, we examine whether social welfare satisfies monotonicity or submodularity.

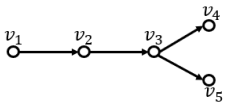


Figure 4.1: Non-monotone and non-submodular example

Theorem 10. *Given a fixed a allocation $\boldsymbol{\mathcal{S}}_a$, expected social welfare is neither monotone nor submodular, with respect to b allocation $\boldsymbol{\mathcal{S}}_b$ under the propagation model UIC-FB.*

Proof. We show a counterexample for each of the two properties. Consider the network shown in Figure 4.1. The edge probabilities are all 1.

Monotonicity. Consider the fixed a allocation $\boldsymbol{\mathcal{S}}_a = \{(v_1, a)\}$. Let $\boldsymbol{\mathcal{S}}_b^1 = \{(v_4, b), (v_5, b)\}$ and $\boldsymbol{\mathcal{S}}_b^2 = \{(v_3, b), (v_4, b), (v_5, b)\}$. Clearly $\boldsymbol{\mathcal{S}}_b^1 \subset \boldsymbol{\mathcal{S}}_b^2$. Let $\boldsymbol{\mathcal{S}}^i = \boldsymbol{\mathcal{S}}_a \cup \boldsymbol{\mathcal{S}}_b^i$, for $i \in \{1, 2\}$. Under $\boldsymbol{\mathcal{S}}^1$, v_1, v_2 and v_3 adopt a w.p. 1; both v_4 and v_5 adopt b w.p. 1, and later when a arrive via v_3, they adopt a w.p. c. thus $\rho(\boldsymbol{\mathcal{S}}^1) = 3\delta + 2(\delta + c\Delta)$. However under $\boldsymbol{\mathcal{S}}^2$, v_1 and v_2 adopt a w.p. 1; v_3 adopts b w.p. 1 and then a w.p. c; v_4 and v_5 adopt b w.p. 1, and a w.p. c^2 (because they can adopt a only when v_3 also adopts a). Thus $\rho(\boldsymbol{\mathcal{S}}^2) = 2\delta + (\delta + c\Delta) + 2(\delta + c^2\Delta)$. $\rho(\boldsymbol{\mathcal{S}}^1) - \rho(\boldsymbol{\mathcal{S}}^2) = c\Delta(1 - 2c) > 0$, for any $c < \frac{1}{2}$ which violates monotonicity.

Submodularity. Again consider the fixed a allocations $\boldsymbol{\mathcal{S}}_a = \{(v_1, a)\}$. Let $\boldsymbol{\mathcal{S}}_b^1 = \{(v_4, b), (v_5, b)\}$ and $\boldsymbol{\mathcal{S}}_b^2 = \{(v_2, b), (v_4, b), (v_5, b)\}$. Clearly $\boldsymbol{\mathcal{S}}_b^1 \subset \boldsymbol{\mathcal{S}}_b^2$. Let $x = \{(v_3, b)\} \notin \boldsymbol{\mathcal{S}}_b^2$. Note that adding x to $\boldsymbol{\mathcal{S}}^2$ does not change the adoption of any node. Therefore, $\rho(x \mid \boldsymbol{\mathcal{S}}^2) = 0$. Whereas, from the counterexample to monotonicity above, we have $\rho(x \mid \boldsymbol{\mathcal{S}}^1) < 0 = \rho(x \mid \boldsymbol{\mathcal{S}}^2)$, (when $c < \frac{1}{2}$). This breaks submodularity.

□

[2] The choice of Δ is immaterial.

The absence of these properties makes FBWelMax hard to approximate. In our attempts to alleviate that we inspect the effect of using a surrogate propagation model and a surrogate objective next.

4.4.1 Sequential propagation model UIC-FB-sequential

We first study a slightly modified propagation model called UIC-FB-sequential. Under UIC-FB-sequential, we assume that the propagation of one item completes before the propagation of the second item begins. In particular, we assume propagation of b starts after a's propagation has ended. We next show that monotonicity and submodularity hold for UIC-FB-sequential.

Theorem 11. *Given a fixed a allocation $\boldsymbol{\mathcal{S}}_a$, the expected social welfare is monotone and submodular, with respect to b allocation $\boldsymbol{\mathcal{S}}_b$ under the propagation model UIC-FB-sequential.*

Proof. We show that monotonicity and submodularity hold for any arbitrary but fixed possible world w. Recall in w both edges and adoption are deterministic.

Monotonicity. Consider any fixed a allocations $\boldsymbol{\mathcal{S}}_a$. Let $\boldsymbol{\mathcal{S}}_b^1$ and $\boldsymbol{\mathcal{S}}_b^2$ be two b allocations where $\boldsymbol{\mathcal{S}}_b^1 \subset \boldsymbol{\mathcal{S}}_b^2$. We show that $\rho_w(\boldsymbol{\mathcal{S}}^1) \leq \rho_w(\boldsymbol{\mathcal{S}}^2)$. For that we argue that for any node v, $\mathcal{A}_w^{\boldsymbol{\mathcal{S}}^1}(v) \subseteq \mathcal{A}_w^{\boldsymbol{\mathcal{S}}^2}(v)$.

If $a \in \mathcal{A}_w^{\boldsymbol{\mathcal{S}}^1}(v)$, then $v \in \phi_w(S_a)$, hence $a \in \mathcal{A}_w^{\boldsymbol{\mathcal{S}}^2}(v)$ because S_a is fixed and UIC-FB-sequential lets a propagates first. That is, propagation of a in UIC-FB-sequential is monotone. If $b \in \mathcal{A}_w^{\boldsymbol{\mathcal{S}}^1}(v)$, then $v \in \phi_w(S_b^1)$. From monotonicity of spread we know $v \in \phi_w(S_b^2)$ must be true. Also if the same node v adopted a before, that implies $t_v \leq c$ in w. Since w remains fixed, $b \in \mathcal{A}_w^{\boldsymbol{\mathcal{S}}^2}(v)$ must be true as well. Therefore, monotonicity follows.

Submodularity. As before let fixed a allocation be $\boldsymbol{\mathcal{S}}_a$ and let $\boldsymbol{\mathcal{S}}_b^1$ and $\boldsymbol{\mathcal{S}}_b^2$ be two b allocations where $\boldsymbol{\mathcal{S}}_b^1 \subset \boldsymbol{\mathcal{S}}_b^2$. Let $x = (u, b) \notin \boldsymbol{\mathcal{S}}^2$ be an additional pair. To prove submodularity we show that $\rho_w(x \mid \boldsymbol{\mathcal{S}}^2) \leq \rho_w(x \mid \boldsymbol{\mathcal{S}}^1)$. Consider any node $v \in \phi_w(x \mid S_b^2)$, by submodularity of spread, $v \in \phi_w(x \mid S_b^1)$ must be true. Therefore, if v adopts only b under $\{x\} \cup \boldsymbol{\mathcal{S}}^2$, v will adopt b under $\{x\} \cup \boldsymbol{\mathcal{S}}^1$ as well. If v also adopted a under $\boldsymbol{\mathcal{S}}^2$, since $\boldsymbol{\mathcal{S}}_a$ is fixed, v would adopt a under $\boldsymbol{\mathcal{S}}^1$, and also since $t_v \leq c$, v will adopt b as well. Hence submodularity holds. □

Since the expected social welfare is monotone and submodular under UIC-FB-sequential, a simple greedy algorithm achieves $(1-\frac{1}{e})$-approximation. While this is encouraging, it still does not provide any guarantee for the original UIC-FB model because the welfare under UIC-FB is neither an upper

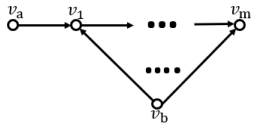


Figure 4.2: Example network showing UIC-FB is worse than UIC-FB-sequential

bound nor a lower bound of welfare under UIC-FB. In fact, it can be arbitrarily worse in both directions compared to the welfare of UIC-FB-sequential as shown in the following lemma.

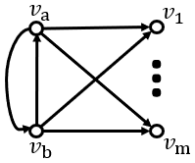


Figure 4.3: Example network showing UIC-FB-sequential is worse than UIC-FB

Lemma 19. *Given an allocation $\boldsymbol{\mathcal{S}}$, $\rho(\boldsymbol{\mathcal{S}})$ under UIC-FB can be arbitrarily worse than $\rho(\boldsymbol{\mathcal{S}})$ under UIC-FB-sequential and vice versa.*

Proof. **UIC-FB arbitrarily worse than UIC-FB-sequential.** Consider the network shown in Figure 4.2, where all edge probabilities are 1. Let $\boldsymbol{\mathcal{S}}_a = \{(v_a, a)\}$ and $\boldsymbol{\mathcal{S}}_b = \{v_b, b\}$.

Under UIC-FB-sequential a propagates first. Therefore, all $m+1$ nodes $v_a, v_1, ..., v_m$ adopt a. Later when propagation of b starts from v_b, $v_1, ..., v_m$ adopt it with probability c. Therefore the total welfare $2\delta + m(\delta + c\Delta)$.

Under UIC-FB a and b start propagating at the same time. They arrive only at v_1 at the same time, but for all the other nodes $v_2, ..., v_m$, b arrives first. The total welfare in this case $2\delta + \sum_{i=1}^{m}(\delta + c^i\Delta)$. As m increases it becomes arbitrarily worse compared to welfare of UIC-FB-sequential.

UIC-FB-sequential arbitrarily worse than model. We now show the opposite is true using the network of Figure 4.3. Again $\boldsymbol{\mathcal{S}}_a = \{(v_a, a)\}$ and $\boldsymbol{\mathcal{S}}_b = \{v_b, b\}$ and all edge probabilities are 1.

Under UIC-FB-sequential when a propagates first, all the nodes in the network, including v_b, adopt a. Later when propagation of b begins, at node v_b, the utility is $\delta + c\Delta$. Every other node earns a utility of $\delta + c^2\Delta$. Therefore the total utility is $(\delta + c\Delta) + (m+1)(\delta + c^2\Delta)$.

Under UIC-FB at time $t = 1$ v_a adopts a and v_b adopts b, at $t = 1$ all nodes experience first level competition. Therefore the utility is $(m+2)(\delta + c\Delta)$. Thus for a large m and small c it is arbitrarily better than welfare produced under UIC-FB-sequential.

□

4.4.2 Surrogate objective of maximizing first level competition

Next, we study how UIC-FB behaves under a surrogate objective. For this part of the discussion, we assume that there exists an arbitrary but fixed edge world, w_1. In a fixed w_1, the maximum utility any node can achieve is $\delta + c\Delta$. The surrogate objective aims to maximize the number of such nodes. Given a seed allocation $\boldsymbol{\mathcal{S}}$, the first level competition nodes are a set of nodes that are reached by both the items, and also every node in that set adopts the second item with a probability of at least c. Let $\psi(\boldsymbol{\mathcal{S}})$ denote the set of first-level nodes. Our surrogate objective aims to maximize the size of the set $\psi(\boldsymbol{\mathcal{S}})$. Note that for the first level node, the expected utility is $\delta + c\Delta$, and this is the maximum utility that any node can achieve under UIC. Therefore maximizing the size of $\psi(\boldsymbol{\mathcal{S}})$ indeed helps maximize the overall utility.

However, as it turns out $\psi(\boldsymbol{\mathcal{S}})$ is not monotone or submodular either.

Theorem 12. *Given a fixed a seedset $\boldsymbol{\mathcal{S}}_a$, the expected number of first level nodes, $E_w[|\psi_w(\boldsymbol{\mathcal{S}})|]$, is neither monotone nor submodular with respect to the b seedset $\boldsymbol{\mathcal{S}}_b$ under the propagation model UIC-FB.*

Proof. We use the same examples of Theorem 10 as the counterexamples here.

Monotnicity. Recall the graph is as shown in Figure 4.1. $\boldsymbol{\mathcal{S}}_a = \{(v_1, a)\}$, $\boldsymbol{\mathcal{S}}_b^1 = \{(v_4, b), (v_5, b)\}$ and $\boldsymbol{\mathcal{S}}_b^2 = \{(v_3, b), (v_4, b), (v_5, b)\}$, where $\boldsymbol{\mathcal{S}}_b^1 \subset \boldsymbol{\mathcal{S}}_b^2$. Note under allocation $\boldsymbol{\mathcal{S}}^1$, both v_4 and v_5 are the first level competition nodes. Therefore $\psi(\boldsymbol{\mathcal{S}}^1) = \{v_4, v_5\}$. Under $\boldsymbol{\mathcal{S}}^2$, $\psi(\boldsymbol{\mathcal{S}}^2) = \{v_3\}$. Therefore it is not monotone as $|\psi(\boldsymbol{\mathcal{S}}^2)| < |\psi(\boldsymbol{\mathcal{S}}^1)|$.

Submodularity. In this example, $\boldsymbol{\mathcal{S}}_a = \{(v_1, a)\}$, $\boldsymbol{\mathcal{S}}_b^1 = \{(v_4, b), (v_5, b)\}$, $\boldsymbol{\mathcal{S}}_b^2 = \{(v_2, b), (v_4, b), (v_5, b)\}$ and $x = (v_3, b)$. $\psi(\boldsymbol{\mathcal{S}}^2) = \psi(\{x\} \cup \boldsymbol{\mathcal{S}}^2) = \{v_2\}$. $\psi(\boldsymbol{\mathcal{S}}^1) = \{v_4, v_5\}$, but $\psi(\{x\} \cup \boldsymbol{\mathcal{S}}^1) = \{v_3\}$. Hence $|\psi(\{x\} \cup \boldsymbol{\mathcal{S}}^2)| - |\psi(\boldsymbol{\mathcal{S}}^2)| = 0 > |\psi(\{x\} \cup \boldsymbol{\mathcal{S}}^1)| - |\psi(\boldsymbol{\mathcal{S}}^1)| = -1$, which violates submodularity.

□

The results of this section demonstrate how difficult it is to approximate the general version of FBWelMax problem, even under simplifying

assumptions. In the following section, we develop instance dependent approximation algorithm and an effective heuristic algorithm to tackle the problem.

4.5 Algorithms

Since the FBWelMax problem under UIC-FB model is difficult to approximate, in this section we propose several algorithms that either produce a non-constant approximation guarantee which is dependent on the problem instance or heuristic that is later shown to have empirical prowess in Section 4.6. In particular, we propose three algorithms in the subsequent sections. Given a fixed $\boldsymbol{\mathcal{S}}_a$, our first algorithm, SpreadGRD, produces a $(\frac{\delta}{c\Delta+\delta}(1-\frac{1}{e}-\epsilon)\rho(\boldsymbol{\mathcal{S}}_a \cup \boldsymbol{\mathcal{S}}_b^*) + (\frac{1}{e}+\epsilon)\rho(\boldsymbol{\mathcal{S}}_a))$-guarantee, where $\boldsymbol{\mathcal{S}}_b^*$ is the optimum b allocation for the given budget k. The algorithm is presented in Section 4.5.1 where we also show that the guarantee is tight. Our second algorithm, SandwichGRD, described in Section 4.5.2, leverages the sandwich approximation proposed in [97]. Lastly, in Section 4.5.3 we present our final algorithm, WelfareGRD, which is an effective heuristic. WelfareGRD is built extending the RR-set machinery that is used in the state-of-the-art IM literature [74, 110, 126], for our filter bubble problem.

4.5.1 SpreadGRD Algorithm

Our first algorithm is a greedy algorithm based on the spread, called SpreadGRD. The algorithm takes a graph G, fixed a allocation $\boldsymbol{\mathcal{S}}_a$, budget of item b k_b, accuracy parameter ϵ, tolerance parameter ℓ.

The pseudocode is shown in Algorithm 10. The idea is simple. It selects seeds S_b of size k_b using an algorithm, called $MIMM$ (Marginal Influence Maximization Method), which delivers a set of seeds that are approximately optimal w.r.t. the marginal gain $\sigma(S_b|S_a)$ (Line 1). I.e., $\sigma(S_b|S_a) \geq (1 - \frac{1}{e} - \epsilon)\sigma(S'_b|S_a)$, where S'_b is an arbitrary b allocation of size k_b. We formally present the $MIMM$ algorithm in Section 4.5.1.1 and establish its properties. SpreadGRD return the final b-allocation as $S_b \times \{b\}$.

Algorithm 10: $SpreadGRD(G, \boldsymbol{\mathcal{S}}_a, k_b, \epsilon, \ell)$

1 $S_b \leftarrow MIMM(G, S_a, k_b, \epsilon, \ell)$
2 Return $S_b \times \{b\}$

We now analyze the performance of SpreadGRD. Before proving its ap-

proximation guarantee, we first show a bound on the welfare for any arbitrary allocation $\boldsymbol{\mathcal{S}}$, where S is the set of corresponding seed nodes of allocation $\boldsymbol{\mathcal{S}}$.

Lemma 20. *Given allocation $\boldsymbol{\mathcal{S}}$, its expected welfare has the following bound,* $\delta\sigma(S) \leq \rho(\boldsymbol{\mathcal{S}}) \leq (\delta + c\Delta)\sigma(S)$

Proof. Consider an arbitrary but fixed edge possible world w. Recall that in w $\phi_w(S)$ is a deterministic set. Further the minimum utility for any node $v \in \phi_w(S)$ is δ, because v must adopt at least one item. The maximum utility for v is $\delta + c\Delta$, when the v is a first level node. Thus for $v \in \phi_w(S)$, $\delta \leq \mathcal{A}_w^{\boldsymbol{\mathcal{S}}}(v) \leq (\delta + c\Delta)$.

From Section 4.3.5, we know,

$$\begin{aligned}\rho_w(\boldsymbol{\mathcal{S}}) &= \sum_{v \in V} \mathcal{U}(\mathcal{A}_w^{\boldsymbol{\mathcal{S}}}(v)) \\ &= \sum_{v \in \phi_w(S)} \mathcal{U}(\mathcal{A}_w^{\boldsymbol{\mathcal{S}}}(v)), \text{since, } \forall v \in V \setminus \phi_w(S), \mathcal{A}_w^{\boldsymbol{\mathcal{S}}}(v) = 0 \\ &= \sigma_w(S)\mathcal{U}(\mathcal{A}_w^{\boldsymbol{\mathcal{S}}}(v))\end{aligned}$$

Using the bound of $\mathcal{U}(\mathcal{A}_w^{\boldsymbol{\mathcal{S}}}(v))$, we get, $\delta\sigma_w(S) \leq \rho_w(\boldsymbol{\mathcal{S}}) \leq (\delta + c\Delta)\sigma_w(S)$. Since it holds for every possible w, the lemma follows. □

Let $\boldsymbol{\mathcal{S}}_b$ and $\boldsymbol{\mathcal{S}}'_b$ be an arbitrary b allocation. We now show the following guarantee for SpreadGRD

Theorem 13. *Let $\boldsymbol{\mathcal{S}}_b$ be the allocation returned by SpreadGRD. Given fixed $\boldsymbol{\mathcal{S}}_a$ and $\epsilon, \ell > 0$, we have, $\rho(\boldsymbol{\mathcal{S}}_a \cup \boldsymbol{\mathcal{S}}_b) \geq \frac{\delta}{\delta + c\Delta}(1 - \frac{1}{e} - \epsilon)\rho(\boldsymbol{\mathcal{S}}_a \cup \boldsymbol{\mathcal{S}}'_b) + (\frac{1}{e} + \epsilon)\rho(\boldsymbol{\mathcal{S}}_a)$, w.p. at least $1 - \frac{1}{|V|^\ell}$, where $\boldsymbol{\mathcal{S}}'_b$ is an arbitrary b-allocation of size k_b.*

Proof. Using the upper bound of Lemma 20, for the arbitrary allocation $\boldsymbol{\mathcal{S}}'_b$, we know that,

$$\begin{aligned}\rho(\boldsymbol{\mathcal{S}}_a \cup \boldsymbol{\mathcal{S}}'_b) &\leq (\delta + c\Delta)\sigma(S_a \cup S'_b) \\ \rho(\boldsymbol{\mathcal{S}}'_b \mid \boldsymbol{\mathcal{S}}_a) + \rho(\boldsymbol{\mathcal{S}}_a) &\leq (\delta + c\Delta)(\sigma(S'_b \mid S_a) + \sigma(S_a))\end{aligned}$$

By rearranging the above we get,

$$\sigma(S'_b \mid S_a) \geq \frac{\rho(\boldsymbol{\mathcal{S}}'_b \mid \boldsymbol{\mathcal{S}}_a) + \rho(\boldsymbol{\mathcal{S}}_a)}{\delta + c\Delta} - \sigma(S_a) \tag{4.1}$$

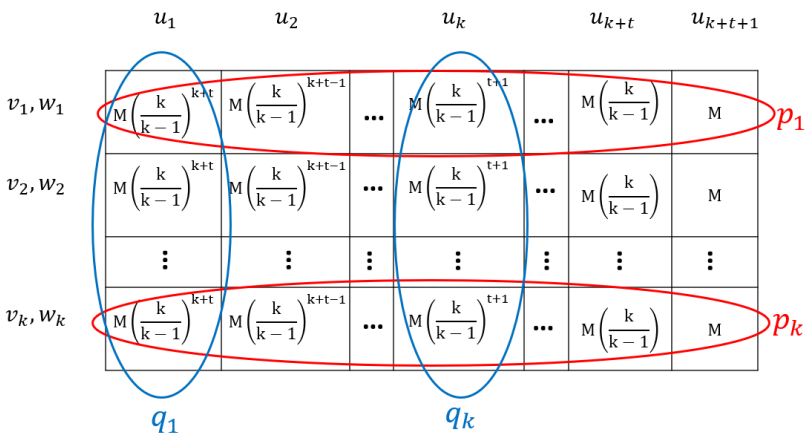


Figure 4.4: Node clusters

Now for allocation $\boldsymbol{\mathcal{S}}_b$ we have,

$$\begin{aligned}
\rho(\boldsymbol{\mathcal{S}}_a \cup \boldsymbol{\mathcal{S}}_b) &= \rho(\boldsymbol{\mathcal{S}}_b \mid \boldsymbol{\mathcal{S}}_a) + \rho(\boldsymbol{\mathcal{S}}_a) \\
&\geq \delta\sigma(S_b \mid S_a) + \rho(\boldsymbol{\mathcal{S}}_a), \text{ from the lower bound of Lemma 20} \\
&\geq \delta(1 - \frac{1}{e} - \epsilon)\sigma(S'_b \mid S_a) + \rho(\boldsymbol{\mathcal{S}}_a), \text{ from MIMM's guarantee} \\
&\geq \delta(1 - \frac{1}{e} - \epsilon)(\frac{\rho(\boldsymbol{\mathcal{S}}'_b \mid \boldsymbol{\mathcal{S}}_a) + \rho(\boldsymbol{\mathcal{S}}_a)}{\delta + c\Delta} - \sigma(S_a)) + \rho(\boldsymbol{\mathcal{S}}_a), \text{ from Equation 4.1} \\
&= \frac{\delta}{\delta + c\Delta}(1 - \frac{1}{e} - \epsilon)\rho(\boldsymbol{\mathcal{S}}'_b \mid \boldsymbol{\mathcal{S}}_a) + \rho(\boldsymbol{\mathcal{S}}_a)(1 - \frac{(1 - \frac{1}{e} - \epsilon)c\Delta}{\delta + c\Delta}) \\
&= \frac{\delta}{\delta + c\Delta}(1 - \frac{1}{e} - \epsilon)\rho(\boldsymbol{\mathcal{S}}_a \cup \boldsymbol{\mathcal{S}}'_b) + (\frac{1}{e} + \epsilon)\rho(\boldsymbol{\mathcal{S}}_a)
\end{aligned}$$

Hence the theorem follows □

We now prove that the above bound is tight.

Lemma 21. *Bound of Theorem 13 is a tight bound.*

Proof. Consider a network of nodes shown in Figure 4.4. Each v_i and w_i node is directly connected to all the nodes of row i and there are p_i number of nodes in row i. Let $p = \sum_{i \in [k]} p_i$. Likewise, each u_j node is directly connected to all the nodes of column j and there are q_j number of nodes in row j (these nodes are different from row nodes). Let $q = \sum_{j \in [k]} q_j$. Clearly, $\frac{q}{p} = 1 - \frac{1}{e}$.

Now let $k_a = k$, $S_a = \{v_1, ..., v_k\}$. Therefore $\rho(\boldsymbol{\mathcal{S}}_a) = p\delta$. Also let $k_b = k$. Therefore if $S'_b = \{w_1, ..., w_k\}$, then $\rho(\boldsymbol{\mathcal{S}}_a \cup \boldsymbol{\mathcal{S}}'_b) = p(\delta + c\Delta)$.

SpreadGRD selects $\{u_1, .., u_k\}$ as the b seeds. The welfare in this case,

$$\begin{aligned}
\rho(\boldsymbol{\mathcal{S}}_a \cup \boldsymbol{\mathcal{S}}_b) &= \delta(p+q) \\
&= q\delta + p\delta \\
&= p(1 - \frac{1}{e})\delta + p\delta \\
&= \frac{\delta}{\delta + c\Delta}(1 - \frac{1}{e})p(\delta + c\Delta) + p\delta \\
&= \frac{\delta}{\delta + c\Delta}(1 - \frac{1}{e})\rho(\boldsymbol{\mathcal{S}}_a \cup \boldsymbol{\mathcal{S}}'_b) + o(1)(\frac{1}{e})\rho(\boldsymbol{\mathcal{S}}_a)
\end{aligned}$$

Hence the bound is tight. □

4.5.1.1 MIMM Algorithm

In this section, we present the MIMM algorithm used by SpreadGRD to select the seeds.

Algorithm 11: $Marginal\_Sampling(G, \mathcal{R}, \theta, S^P)$

```
while |R| ≤ θ do
    Select v from G uniformly at random
    R ← BFS(v)
    if R ∩ S^P ≠ ∅ then
        R ← ∅
    R ← R ∪ R
Return R
```

In [22], the authors proposed an efficient seed selection algorithm that used the Reverse Reachable Sets (RR-sets) samples. We adopt the standard RR-set construction slightly to account for the presence of existing seed set S_a using marginal RR-sets. Given an existing allocation $\boldsymbol{\mathcal{S}}_a$, we construct a marginal RR-set as follows. A root node $v \in V$ is chosen uniformly at random, then v is added to R_v and a BFS is originated from v. Whenever $u \in R_v$, sample each incoming edge (u', u) w.p. $p_{u'u}$ and add it to R_v. The process stops when there are no new nodes that can be added to R_v. If R_v overlaps S_a at any stage, i.e., if $R_v \cap S_a \neq \emptyset$, then set $R_v := \emptyset$. In other words, if a generated RR-set "hits" S_a, we discard it by setting it to $\emptyset$. Algorithm 11 shows the pseudo-code of this marginal RR-set sampling

process. Given graph G, a number θ denoting how many RR-sets needs to be sampled and a fixed seed nodes S_a, $Marginal\_Sampling$ generates θ number of RR-sets to $\mathcal{R}$ from G, based on the marginal on S_a.

$MIMM$ using $Marginal\_Sampling$, achieves the approximation guarantee on marginals. The pseudo-code of $MIMM$ is shown in Algorithm 12.

Algorithm 12: $MIMM$ $(G, \epsilon, \ell, S_a, k_b)$

Initialize $\mathcal{R} = \emptyset$, $n = |V|$, $i, LB = 1$, $\epsilon' = \sqrt{2} \cdot \epsilon$
while $i \leq \log_2(n) - 1$ **do**
    $x = \frac{n}{2^i}$; $\theta_i = \lambda'/x$, where λ' is defined in Eq.9 of [127]
    $Marginal\_Sampling(G, \mathcal{R}, \theta_i, S_a)$
    $S_k = NodeSelection(\mathcal{R}, k)$
    if $n \cdot F_{\mathcal{R}}(S_k) \geq (1 + \epsilon') \cdot x$ **then**
        $LB = n \cdot F_{\mathcal{R}(S_k)}/(1 + \epsilon')$
        break
$\theta = \frac{\lambda^*}{LB}$, where λ^* is deined in Eq.6 of [127]
$\mathcal{R} = \emptyset$
$Marginal\_Sampling(G, \mathcal{R}, \theta, S_a)$
$S_b = NodeSelection(\mathcal{R}, k_b)$
return S_b as the final seed set

This idea is similar to the marginal sampling used in Section 3.5.2.1. As the analysis of the Section 3.5.2.1 shows, the sampling runs in time $O(k_a + k_b + \ell)(n + m) \log\ n \cdot \epsilon^{-2})$.

4.5.2 Sandwich Approximation Algorithm

The sandwich strategy was proposed in [97] to provide a data-dependent approximation guarantee for non-submodular maximization under a cardinality constraint. Since our objective function $\rho(\cdot)$ is also non-submodular, we leverage the sandwich to have a similar data-dependent guarantee.

The main result of [97] is as follows. Given a nonsubmodular objective function $f : 2^V \rightarrow \mathcal{R}_{\geq 0}$, on a ground set V, let f_l and f_u be functions on V such that, $f_l(I) \leq f(I) \leq f_u(I)$, $\forall I \subseteq V$, and both f_l and f_u are monotone and submodular. Then run the greedy algorithm on f_l, f, and f_u and let S_l, S, and S_u be the corresponding greedy solutions. Set $S_{sand} = \arg\max_{T \in \{S_l, S, S_u\}} f(T)$, then we have the following theorem from [97],

Theorem 14. *Applying sandwich approximation we get,*

$$f(S_{sand}) \geq max\{\frac{f(S_u)}{f_u(S_u)}, \frac{f_l(S^*)}{f(S^*)}\}(1-\frac{1}{e})f(S^*)$$

where S^* is the optimal solution of maximizing f.

Not that the above approximation bound is better as we get tighter f_l and f_u. Hence we try to find such tight bounds for our objective ρ

4.5.2.1 Establishing upper bound

To find an upper bound, we slightly tweak our propagation model; note that this new propagation model is used only for technical interest. Recall that, according to our original model, UIC-FB, if a node u has adopted item a (or b) first, then if it is influenced by b (or a) later, then it adopts b (a) with a probability c. W.p. $(1-c)$ does not adopt the second item. When u does not adopt an item, in UIC-FB, it blocks the propagation by not propagating the item any further. In the tweaked propagation model, called UIC-Tattler, a node u propagates all the items it is influenced by, even in the case when u does not adopt all. We keep the reward function unchanged. Thus if u is influenced by both a and b and adopts one item, the utility at u is δ, but u propagates both the items.

We denote $\rho_T(\cdot)$ to denote the objective under UIC-Tattler. We now show that $\rho_T(\cdot)$ is monotone and submodular under UIC-Tattler. Towards that, we first prove the following lemma.

Lemma 22. *Given a possible world w, a seed nodes S_a and b seed nodes S_b, every node that is reachable from a seed node, is influenced by the corresponding item.*

Proof. We show it by induction on the hop count t from seed nodes. As a base case, when $t = 0$, the claim holds. Let this be true for a node u which is t hops away from a seed node. An out-neighbor of u, node v, will also be influenced via u irrespective of whether v adopts the item. Hence for the node v that is $t+1$ hops away, the claim holds. □

We are now ready to prove the monotonicity.

Lemma 23. *Given a fixed a seedset S_a, $\rho(\boldsymbol{\mathcal{S}})$ is monotone in S_b, i.e., for S_b^1 and S_b^2, where $S_b^1 \subseteq S_b^2$, $\rho(\boldsymbol{\mathcal{S}}^1) \leq \rho(\boldsymbol{\mathcal{S}}^2)$, where $\boldsymbol{\mathcal{S}}^i = \{S_a \times a\} \cup \{S_b^i \times b\}$ for $i \in \{1, 2\}$.*

Proof. We show that the claim holds for any arbitrary but fixed edge possible world w. Note that if a node is influenced by one item only, then the utility of that node is δ, whereas if it is influenced by both the items then the utility is $(c\Delta + \delta) > \delta$.

Let $\phi_w(S_a)$ be the set of nodes reached by a when the seed nodeset is S_a. From Lemma 22, $\phi_w(S_a)$ remains the same, when the allocation is changed to $\boldsymbol{\mathcal{S}}^2$. Also, $\phi_w(S_b^1) \subseteq \phi_w(S_b^2)$, from monotonicity of spread. Consequently, $\phi_w(S_a) \cap \phi_w(S_b^1) \subseteq \phi_w(S_a) \cap \phi_w(S_b^2)$.

Therefore, if a node is influenced by both the items under $\boldsymbol{\mathcal{S}}^1$, the node will be influenced by two items under $\boldsymbol{\mathcal{S}}^2$ as well. Further, as a direct consequence of Lemma 22, if a node is influenced by one item under $\boldsymbol{\mathcal{S}}^1$, it will be influenced by at least one item under $\boldsymbol{\mathcal{S}}^2$. Hence the welfare is monotone.

□

We now prove the submodularity.

Lemma 24. *Given a fixed a seedset S_a, $\rho(\boldsymbol{\mathcal{S}})$ is submodular in S_b, i.e., for S_b^1 and S_b^2, where $S_b^1 \subseteq S_b^2$, and $x = (b, v)$, $\rho(x \mid \boldsymbol{\mathcal{S}}^2) \leq \rho(x \mid \boldsymbol{\mathcal{S}}^1)$, where $\boldsymbol{\mathcal{S}}^i = \{S_a \times a\} \cup \{S_b^i \times b\}$, for $i \in \{1, 2\}$.*

Proof. We show that the claim holds for any arbitrary but fixed edge possible world w.

Consider a node $v \in \phi_w(x \mid S_b^2)$. Using monotonicity of reachability, $v \in \phi_w(x \mid S_b^1)$ must be true. Further since S_a is fixed, if $v \in \phi_w(x \mid S_a)$, it will be so under both $\boldsymbol{\mathcal{S}}^1$ and $\boldsymbol{\mathcal{S}}^2$. Therefore for any v which is in the reachable node set of $x \mid S_b^2$, its utility under $x \mid \boldsymbol{\mathcal{S}}^1$ is at least that of under $x \mid \boldsymbol{\mathcal{S}}^2$. Hence submodularity holds.

□

Therefore using a greedy algorithm we get $1 - \frac{1}{e}$ approximation under UIC-Tattler.

We now show that the welfare under the UIC-Tattler model is an upper bound on the welfare produced by our model, UIC-FB.

Lemma 25. *Given seed allocation $\boldsymbol{\mathcal{S}}$, $\rho_T(\boldsymbol{\mathcal{S}}) \geq \rho(\boldsymbol{\mathcal{S}})$.*

Proof. We show this holds in any arbitrary edge possible world w.

In our model, if a node v in w has utility greater than δ, then v definitely is reachable from S_a and S_b. Therefore under tattler model v will have a utility of $\delta + c\Delta$.

Alternatively if v has utility δ in our model, then v is reachable from at least one node of $S_a \cup S_b$. Therefore under the tattler model, v will have a utility of at least δ.

Since this is true for every node v in w, the claim follows. □

4.5.2.2 Establishing a lower bound

We first prove a useful property regarding the effect of c on the objective $\rho(\cdot)$ under the UIC propagation model.

Lemma 26. *$\rho(\cdot)$ is monotone w.r.t. competition parameter c for a given seed allocation $\boldsymbol{\mathcal{S}}$.*

Proof. We show it holds in arbitrary but fixed possible world w. Recall that in w, the graph is deterministic and each node v in the deterministic graph samples a fixed value t_v. For a given c, v adopts the second item if $t_v \leq c$. Let $\rho_w(\boldsymbol{\mathcal{S}}, c_1)$ denote the welfare of allocation c in w when the competition parameter is c. We want to show that $\rho_w(\boldsymbol{\mathcal{S}}, c_1) \leq \rho_w(\boldsymbol{\mathcal{S}}, c_2)$, for any $c_1 \leq c_2$.

Towards that we first establish the following claim. Consider a node v in w. Let $A^t(v, c_1)$ and $A^t(v, c_2)$ be the two item subsets that v adopts at time t under allocation $\boldsymbol{\mathcal{S}}$ for c_1 and c_2 respectively. Then we show by induction that $A^t(v, c_1) \subseteq A^t(v, c_2)$. Let be $t = 1$, then v is a seed node. If v adopts a single item under c_1, then clearly v as a seed node will adopt at least one item under c_2 as well. Otherwise if v adopts two items then v must be seeded with both items and $t_v \leq c_1$. Since $c_1 \leq c_2$, $t_v \leq c_2$, hence v will adopt both the items under c_2 as well.

Let this be true for time until $t-1$, we then show that it holds for t. If $|A^t(v, c_1)| = 2$, then there is some neighbor of v that adopted item a and b by $t-1$, let the nodes be v_1 and v_2 (it is possible to have $v_1 = v_2$). Clearly since w is the same v_1 and v_2 will remain to be neighbors of v. Further using IH, they will adopt a and b respectively under c_2. Also we know $t_v \leq c_1$. Hence following $c_1 \leq c_2$, $t_v \leq c_2$. Therefore the claim follows.

Therefore we can conclude that the lemma holds.

□

A direct consequence of Lemma 26 is that by setting $c = 0$ we get a lower bound for $\rho(\cdot)$, i.e., for any allocation $\boldsymbol{\mathcal{S}}$, $\rho(\boldsymbol{\mathcal{S}}, 0) \leq \rho(\boldsymbol{\mathcal{S}}, c)$. Further when $c = 0$, every node can adopt at most one item, hence maximizing welfare is same as maximizing the spread. Hence $\rho(\boldsymbol{\mathcal{S}}, 0)$ is monotone and submodular w.r.t. $\boldsymbol{\mathcal{S}}$, hence a greedy solution has $1 - \frac{1}{e}$ approximation.

Let S_T be the greedy solution on ρ_T, then from Theorem 14 we get,

$$\rho(\boldsymbol{\mathcal{S}}) \geq max\{\frac{\rho(S_T)}{\rho_T(S_T)}, \frac{\rho(S^*, 0)}{\rho(S^*)}\}(1 - \frac{1}{e})\rho(S^*)$$

where S^* is the optimum b seeds for $\rho(\cdot)$.

4.5.3 WelfareGRD

Our previous algorithms, although have approximation guarantees, do not attempt to maximize our welfare objective directly. Since maximizing welfare is difficult to approximate, in this section, we propose a non-trivial heuristic, called WelfareGRD (Welfare Greedy) to that effect.

WelfareGRD uses RR-sets that are used in all state-of-the-art IM algorithms in recent times. However, there are a few key differences that we make to the standard RR set machinery, to adopt it for our problem. First, in standard RR set construction there are two random steps - (a) the root of the RR set is a node chosen at random from the node-set, and (b) the edges are then sampled according to the edge probability as the RR set grows from the root. In addition to these, for our case, we also fix the randomness associated with the competition parameter c and the order of adoption o_v. Next, once the RR sets are constructed, in the standard IM papers, the nodes are greedily selected by solving the set-cover problem. We instead require solving a weighted set cover problem, where the weight of a node in the RR set denotes its contribution to welfare if selected as a seed. Lastly, we restrict an RR set to be a tree to enable an efficient weight computation of the nodes. We will show that using a recursive rule, the node weights can be computed by only using a linear pass when the RR set is a tree.

In what follows, we first formally describe the RR tree construction process in Section 4.5.3.1. We then present the recursive formula to compute node weights of the RR tree in Section 4.5.3.2. Finally in Section 4.5.3.3 we present WelfareGRD that uses node weights of the RR trees to greedily select the seed nodes.

4.5.3.1 RR tree construction

As mentioned earlier, in the classical setting, RR-set samples are used to compute an unbiased estimation of the spread. In our case we need to estimate the marginal welfare using the RR-sets. Towards that we define a notion of weight for each node in every RR-set. To compute these weights efficiently we restrict an RR set as a tree. Therefore our RR tree construction

works as follows. Given a graph $G = (V, E, p)$ and S_a, we randomly select a node $v \in V$ as the root of the RR tree RT_v. Set $RT_v = \{v\}$, then start a BFS such that: for $u \in RT_v$, sample each incoming edge (u', u) w.p. $p(u', u)$ and add it to RT_v if $u' \notin RT_v$. Stop when no new nodes can be added. For each $u \in RT_v$, set - (i) $t_u = 2$ w.p. c, or $t_u = 1$ w.p. $(1 - c)$; (ii) o_u as a random order of its in-neighbors; (iii) w_u as the weight of u that denotes the welfare contribution of node u to the root v when selected as a b seed. We next elaborate how to compute the weight w_u.

4.5.3.2 Node weight assignment

Recall in the traditional RR set, any node u in the RR set activates the root, when selected as seed. Therefore every node has a uniform contribution of 1 towards the spread. In our setting, each node can contribute one of the three possible values among $0, \delta$, and Δ. Hence we weigh each node differently based on how much the node would contribute; for a node v its weight can thus be 0, or δ, or Δ. We describe the recursive process of computing the weight; Algorithm 13 shows the pseudo-code of it.

Algorithm 13: $WelfareWeight(RT_v, S_a, v)$

```
if RT_v ∩ S_a = ∅ then
    setWeight(v, δ)
    Return
if t_v = 1 then
    setWeight(v, 0)
    Return
setACounts(RT_v, S_a)
Set r = a(v)
WelfareWeight-helper(v, r, S_a)
```

Algorithm 14: $setACounts(RT_v, S_a)$

```
for Node u ∈ RT_v do
    Set a(u) = 0
for Node s ∈ S_a do
    Set P = path(s, v) for u ∈ P do
        a(u) = a(u) + 1
```

Algorithm 15: $WelfareWeight\text{-}helper(u, r, S_a)$

```
1 Set w_u = Δ
2 for u' ∈ v.children do
3 |   if t_{u'} = 1 and a(u') = r then
4 |   |   setWeight(u', 0)
5 |   if t_{u'} = 1 and a(u') < r then
6 |   |   setABasedWeight(RT_{u'}, S_a)
7 |   if t_{u'} = 2 then
8 |   |   WelfareWeight-helper(u', r, S_a)
```

Algorithm 16: $setWeight(u, w)$

```
1 Set w_u = w
2 for u' ∈ v.children do
3 |   setWeight(u', w)
```

Suppose $RT_v \cap S_a = \emptyset$, then any node $u \in RT_v$ when selected as a b seed, results in a b adoption at the root v. Hence for every such u, $w_u = \delta$ (Line 2). This is done using the *setWeight* method, that takes a node v and a weight w as input, and sets weights of every node rooted at the subtree v as w as shown in Algorithm 16. Now suppose $RT_v \cap S_a \neq \emptyset$ and $t_v = 1$ for root node v. Since $t_v = 1$, v can never adopt two items. Also $RT_v \cap S_a \neq \emptyset$, v adopts a already and obtains a utility of δ which is the maximum utility v can have. Therefore in this case, for every $u \in RT_v$, we set $w_u = 0$ as shown in Line 5.

Now lastly for the case when $RT_v \cap S_a \neq \emptyset$ and $t_v = 2$, for a node $u \in RT_v$, w_u can be either 0 or Δ. This is because without any b seed v definitely adopts a as $RT_v \cap S_a \neq \emptyset$. If a b seed causes it to adopt both a and b then the marginal gain is Δ, otherwise it is 0. Root v adopts one item if our choice of b seed blocks propagation of a, or the existing a seed blocks b to reach v. Hence in this case, we need to know the exiting a propagation paths. We first make a pass on the tree to compute the number of a paths that passes through each u, let $a(u)$ denote the count (Line 7). Since RT_v is a tree, there is a unique path from each $u_a \in S_a$ to any u; $a(u)$ is the count of total number of such paths to u. Note $a(u)$ for every $u \in RT_v$ can be computed by visiting root v from each $u_a \in S_a$ as depicted in Algorithm 14. Running time is $O(n_v)$, where n_v is the number of nodes in RT_v. Let r be the total number a paths that reaches root v (Line 8). We

Algorithm 17: $setABasedWeight(RT_u, S_a)$

```
1  Set u = root(RT_u)
2  if a(u) = 0 then
3  |  setweight(u, Δ)
4  |  Return
5  Set Q = ∅
6  enqueue(u) while Q.notEmpty() do
7  |  u' = Q.dequeue
8  |  S = u'.sibblings ∩ S_a
9  |  if S = ∅ then
10 |  |  w_u' = Δ
11 |  |  for u'' ∈ u'.children do
12 |  |  |  enqueue(u'')
13 |  else
14 |  |  for u'' ∈ u'.children do
15 |  |  |  setWeight(u'', 0)
16 |  |  else
17 |  |  |  w_u' = Δ
18 |  |  |  P_b = path(u', u)
19 |  |  |  for s ∈ S do
20 |  |  |  |  P_a = path(s, u)
21 |  |  |  |  Nodes = P_b ∩ P_a
22 |  |  |  |  for node ∈ Nodes do
23 |  |  |  |  |  if t_node = 1 and o_first = {a, b} then
24 |  |  |  |  |  |  w_u' = 0
25 |  |  |  |  |  |  break
26 |  |  |  |  if w_u' = 0 then
27 |  |  |  |  |  break
```

then begin computing the weights for this case using the recursive function $WelfareWeight\text{-}helper$ as shown in Algorithm 15.

$WelfareWeight\text{-}helper$ recurses on the children of the nodes. Each child is categorized into one of the following three subcases.
(i) Consider a child u such that $t_u = 1$ and $a(u) = r$. This means that all the a paths to root passes via u and u can adopt only one item. Then for all the nodes u', of the subtree rooted at u (including u), $w_{u'} = 0$ (Line 4). This holds because if for any b seed selected in this subtree u adopts b, then v cannot adopt a as there is no other path via which v can adopt a. Thus v only adopts b in this case. Otherwise, if u does not adopt b, then v only adopts a.
(ii) Now suppose $t_u = 1$ and $a(u) < r$ (note $a(u) > r$ is impossible). In this case there is at least one a path that reaches v but not via u. Hence all the nodes in the subtree rooted at u, RT_u, that as a seed makes u adopt b, has weight Δ. All other nodes of the subtree RT_u has weight 0. We set this weights by doing a level-order traversal on RT_u in Line 6. The pseudo code of the traversal in shown in Algorithm 17. Clearly when there is no a seed in RT_u (therefore $a(u) = 0$), any node when selected as b seed leads to b adoption at u, hence they all have weight Δ (Line 3). If there is an a seed, then any node that is at a level below the level of first a seed, has a weight 0, because a would reach u before b can reach from any such node (Line 15). For a node u' that is at the same level of the first a seed s, we compute the path from u' and s to root of RT_u. If there is a node where these paths intersect then at that node both a and b arrive together. If there is node in that intersecting path that adopts only one item, and the first node of the intersecting point has a first in its order o_{first}, it will block propagation of b, resulting in a weight of 0 for u' (Line 24). If no such blocking exists then u' has weight Δ (Line 17).
(iii) Lastly when, $t_u = 2$ (note this the case for root v as well), $w_u = \Delta$ (Line 1). In this case u adopts both items, hence v will do so. Further for the subtree rooted at this node we can recursively encounter one of the three subcases again. Hence we invoke $WelfareWeight\text{-}helper$ again for this case (Line 8).

We now present the seed selection algorithm, WelfareGRD.

4.5.3.3 Seed selection

Pseudocode of WelfareGRD is shown in Algorithm 18. Similar to classical IM algorithm, IMM [127], WelfareGRD first samples a set of RR trees $\mathcal{R}$, where $|\mathcal{R}| = \theta$, and θ is provided as a input parameter. Then for every node

Algorithm 18: $WelfareGRD(G = (V, E, p), \theta, S_a, k_b)$

```
1  Sample θ number of RR trees from G in R
2  for v ∈ V do
3  |  Set w(v) = 0
4  Set S_b = ∅
5  for RT ∈ R do
6  |  for v ∈ RT do
7  |  |  w(v) = w(v) + WelfareWeight(RT, S_a, v)
8  for i = 1 to k_b do
9  |  v_max = arg max_{v∈V} w(v)
10 |  if w(v_max) ≤ 0 then
11 |  |  break
12 |  S_b = S_b ∪ v_max
13 |  for RT ∈ R do
14 |  |  if v_max ∈ RT then
15 |  |  |  for v ∈ RT do
16 |  |  |  |  w(v) = w(v) − WelfareWeight(RT, S_a, v)
17 Return S_b
```

$v \in \mathcal{R}$, it computes the sum of its weight contribution in each RR trees (Line 7). After that, it greedily selects the node that has the highest weight contribution (Line 9). If the weight contribution is greater than 0 then it is selected as a b seed. Every time a new node is selected as a seed, the RR trees, where the node is present, are discarded from the future consideration (Line 16). This process is repeated k_b times, where k_b is the budget of item b.

4.6 Experiments

4.6.1 Experiment Setup

Our experiments are performed on a Linux machine with Intel Xeon 2.6 GHz CPU and 128 GB RAM.

4.6.1.1 Networks

We conduct our experiments of this section on four real social networks: NetHEPT, Douban-Book, Douban-Movie, and Orkut; properties of these

	NetHEPT	Douban-Book	Douban-Movie	Orkut
# nodes	15.2K	23.3K	34.9K	3.07M
# edges	31.4K	141K	274K	117M
avg. deg.	4.13	6.5	7.9	77.5
type	undirected	directed	directed	undirected

Table 4.1: Network Statistics

networks are summarized in Table 4.1. Among these networks, NetHEPT, Douban-Book, and Douban-Movie are benchmark datasets in the IM literature [97]. Orkut is a large publicly available network that is made available at [123].

4.6.1.2 Algorithms compared

We compare the three algorithms we developed , namely SpreadGRD, SandwichGRD, and WelfareGRD against two baselines – TDEM [98], Balance-C [57].

Similar to our work, Balance-C [57], considers propagation of two items. Given an initial (and partial) seed placement of both the items, the remaining seeds are chosen such that at the end of the propagation, the number of nodes influenced by either both the items or none, is maximized. Thus by ensuring that most nodes are adopting both items or none, Balance-C manages to achieve a balanced exposure of the two items to the most number of nodes.

TDEM, in contrast, relies on the "leaning scores" of nodes (users) and items, provided as input. An exposure quality function $g(\cdot)$ is defined based on the leaning scores of the items and that of the node which is influenced by those items. The goal is to maximize the sum of exposure qualities over all the nodes in the network. We will establish a connection between this g function and our reward parameters later in Section 4.6.1.3. It is worth noting at this point that no existing work, including Balance-C and TDEM, distinguishes awareness from adoption, and consequently lacks a competition parameter such as c. In other words, they consider $c = 1$.

4.6.1.3 Default configuration

In this section we describe the default configuration we use for the algorithms; Unless explicitly stated, all the parameters are set to the default values as mentioned in this section.

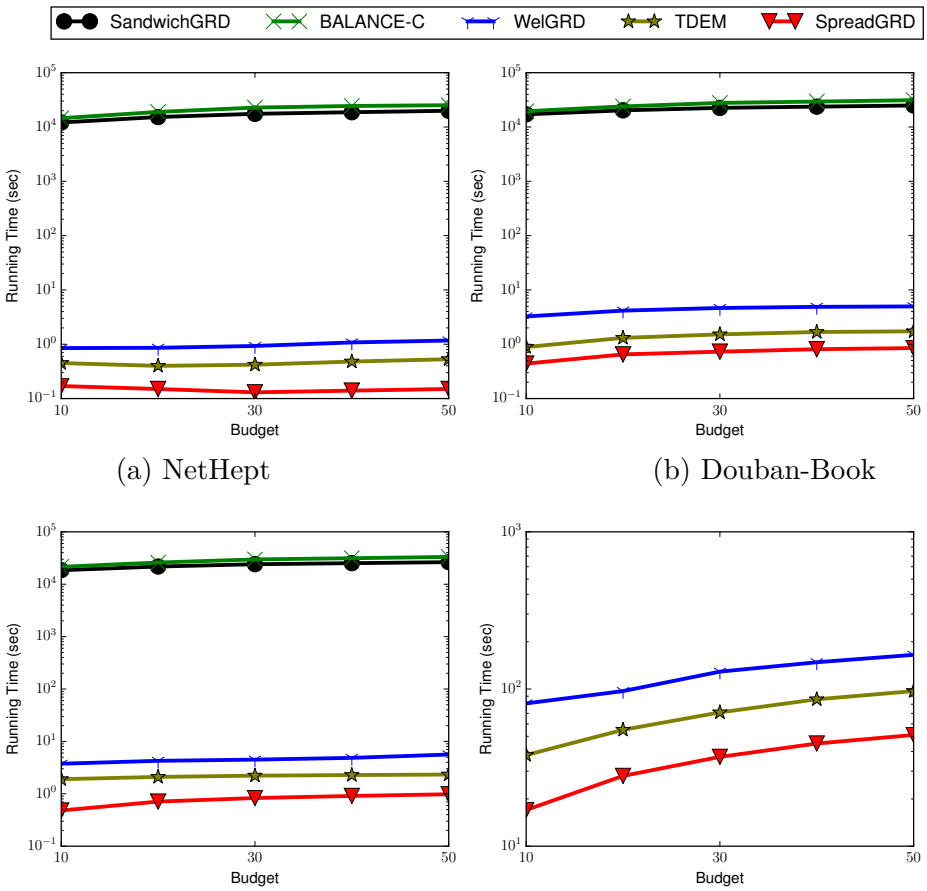
Following previous works [74, 110] we set probability of edge $e = (u, v)$ to $1/d_{in}(v)$, where $d_{in}(v)$ is the in-degree of node v. We use $\epsilon = 0.5$ and $\ell = 1$ as our default in all the algorithms that use these parameters.

For TDEM we require to set leaning scores for the nodes and items. As required by TDEM, the leaning score should be in the range $[-1, 1]$, where -1 and 1 denote the two extreme leaning and 0 denote neutral. Therefore, for each node $v \in V$, we $l(v) = 0$, considering each node as a neutral node. Further the exposure quality is noted to be highest for TDEM, when the items' leaning scores are evenly spaced out in $[-1, 0, 1]$ ([98]). Hence we select the leaning scores a and b as -0.5 and 0.5 respectively, to provide TDEM with the best possible configuration. Using the exposure quality function $g(\cdot)$, that takes leaning scores as input, we set δ and Δ to be the exposure quality of adopting one item and both. Thus $\delta = g(\{-1, -0.5, 0, 1\}) = g(\{-1, 0, 0.5, 1\}) = 0.625$ and $\Delta = g(\{-1, -0.5, 0, 0.5, 1\}) = 0.8125$. Additionally, for our algorithms, we set competition parameter as $c = 1-(\Delta-\delta)$. This captures the effect that when the rewards are further away, i.e., $\Delta >> \delta$, it is more difficult to make the user adopt both, although the reward, in that case, will be high, and vice versa. The default value is therefore, $c = 1 - (0.8125 - 0.625) = 0.8125$. Notice that the default value of c being high favors the baselines that implicitly assume $c = 1$ as they do not differentiate between awareness and adoption. However, even under this relatively favorable setting, we show that our algorithms outperform the baselines.

Unless specified otherwise, budget of an item i, $k_i = 50$, where $i \in \{a, b\}$. Whenever marginal gains are required, we run 5000 simulations and take the average result. If an algorithm does not complete its execution in six hours, it is omitted from the comparison.

4.6.2 Scalability and quality

For our first set of experiments, we compare the running time of the algorithms in the four networks. For this experiment, the seeds of item a are kept fixed. Given the default budget $k_a = 50$, S_a is set to be the seeds returned by IMM for k_a. Budget of item b ranges from 10 to 50 with a step size of 10. The results are presented in Figure 4.5. When the network size increases or when the budget of b increases, the running times of all the algorithms increase. However, it is seen that the running times of SpreadGRD, TDEM, and WelfareGRD are orders of magnitude faster than Balance-C and SandwichGRD. This is because Balance-C and SandwichGRD use costly MC simulations, in comparison SpreadGRD, TDEM, and WelfareGRD use a much faster alternative of RR sets for sampling. In fact,



(a) NetHept (b) Douban-Book

(c) Douban-Movie (d) Orkut

Figure 4.5: Running times of *TDEM*, Balance-C, *SpreadGRD*, *SandwichGRD*, **and** *WelfareGRD*

Balance-C and SandwichGRD do not complete after six hours on Orkut, hence they are excluded in the plot.

Using the same setup, we next compare the quality of the seeds selected by the algorithms in terms of the social welfare produced. The results are shown in Figure 4.6. SpreadGRD produces the lowest welfare because it selects the b seeds that maximize the marginal spread and hence the number of nodes co-adopting both is the minimum. WelfareGRD dominates in all the networks, in a certain network, such as NetHept, WelfareGRD produces 50% more welfare than the closest baseline TDEM. Notice that, since the value of c under the default configuration is set to be high, performance

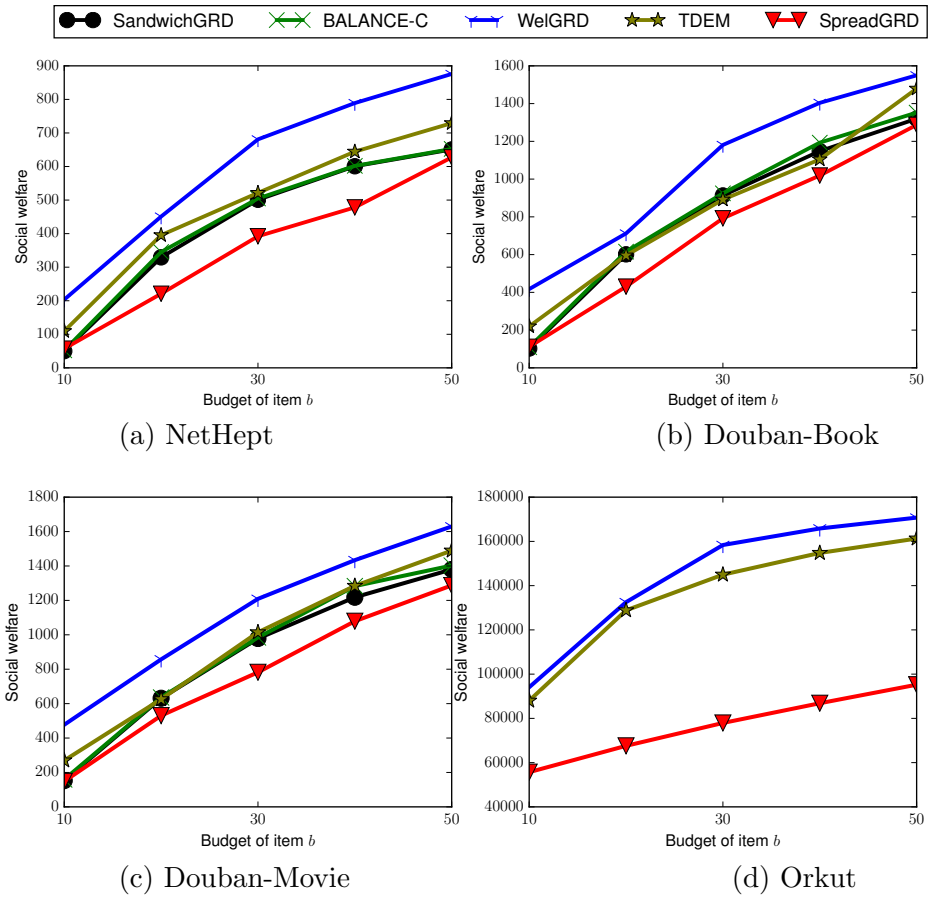


Figure 4.6: Expected social welfare of $TDEM$**,** Balance-C, $SpreadGRD$**,** $SandwichGRD$**, and** $WelfareGRD$

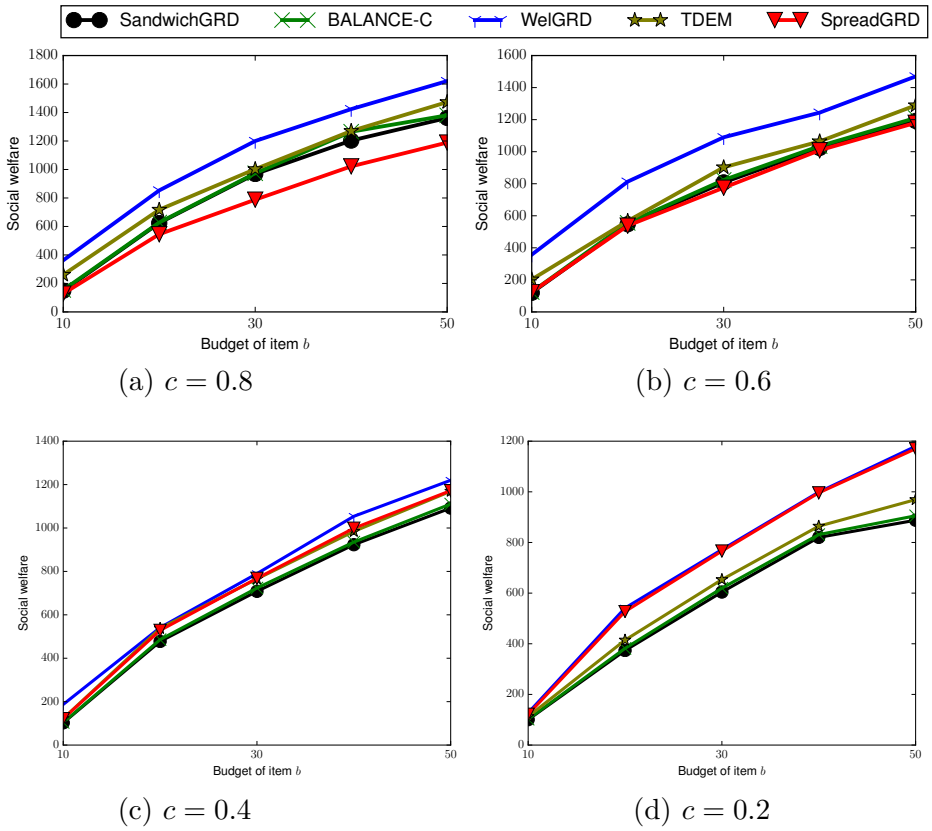


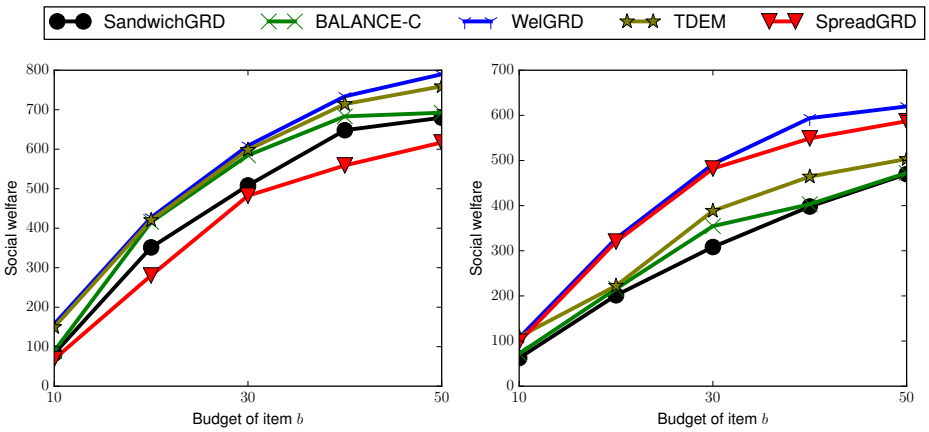
Figure 4.7: Expected social welfare under different values of competition parameter c

the competition-agnostic baselines are comparable with WelfareGRD. This is because to maximize co-adoption these algorithms select the same seeds for a and b. We will show next in Section 4.6.3, that for a lower value of c, selecting the same seeds for both the items do not work well, and hence the difference between the baselines and WelfareGRD increases.

4.6.3 Impact of the competition parameter

For this subset of experiments, we hold all the parameters, except c, to the same value as described in the section before. Value of c is set as one of the four values - $0.8, 0.6, 0.4$ and 0.2. The results are shown in Figure 4.7.

As c decreases, competition increases, and consequently welfare produced by any algorithm decreases. However, the drop for SpreadGRD is



(a) Random (b) Low degree

Figure 4.8: Expected social welfare under different initial allocation of a

the minimum because SpreadGRD optimizes marginal spread, hence increasing competition has very little impact on its performance. Note that, since the baselines are not designed for competition, their performance deteriorates fast for a low value of c. In fact for a value of $c = 0.4$, they perform worse than SpreadGRD. WelfareGRD also experience a drop in the overall welfare. However, unlike the baselines, it does not blindly co-allocate the seeds. Hence for a low value of $c = 0.2$, its performance is similar to that of SpreadGRD.

4.6.4 Different fixed a allocations

In the previous sections, a seeds are chosen to be the approximately optimal nodes of the network in terms of the spread. Therefore many nodes of the network were guaranteed to be exposed to at least one item, i.e., a. In this section, we see the impact of choosing spread-wise inferior nodes as a seeds. In particular, we use two alternative approaches of finding S_a - (i) Random: In this case, k_a nodes are chosen at random to be the a seeds and (ii) Low degree: In this case, k_a number of nodes having the lowest out-degrees are chosen to be the a seeds. Ties are broken arbitrarily. The results corresponding to these new a allocations are shown in Figure 4.8.

Generally speaking, given an inferior a allocation, the welfare produced by any algorithm reduces. However, notice when a seeds are really bad, SpreadGRD significantly outperforms baselines TDEM and Balance-C. This is because, as the baselines co-allocate the seed nodes, they cannot produce

high welfare if the given seed node of one item, i.e., item a, is of low quality. SpreadGRD on the other hand will still select spread-wise high-quality b seeds. Similarly, WelfareGRD also does the same, because it can choose a candidate seed that activates a really high number of nodes, say n_1, with reward δ over another candidate that activates a tiny number of nodes, say n_2 with reward Δ, i.e., $n_1\delta >> n_2\Delta$.

We would also like to revisit the question we raised in Section 4.3.4 - when the potential of co-adoption is low, should the second propagation sacrifice the potential to influence new nodes? As stated before, we think the answer is no, which we motivated by arguing the business objective and right of access to information, i.e., our quest for balanced information spread should not stop the spread of information altogether. Results that we obtain in this section, particularly for the low degree setting, demonstrate the necessity.

4.7 Conclusions

Our work takes the first step of realistically modeling the effect of the filter bubble, where the two opposing information exhibit competition among them, whereas the objective encourages co-adoption of the information that results in a complementary objective. We show that designing a constant algorithm, under such conflicting needs, is difficult. Hence, we resort to instance-dependent approximation and heuristic for our optimization objective. Breaking away from the co-allocation that the existing baselines end up doing, our algorithms are shown to outperform the baselines in real networks.

There are still some interesting research questions that remain to be addressed. Our work is limited to a propagation involving two opinions, however, in the real-world, multiple views propagate concurrently. Further, it is assumed that the competition parameters are global i.e., they have the same value for all the users. In reality, users have individual perceptions that lead to non-uniform parameters, which follow-up works need to look into.

Chapter 5

Summary and discussions

In this thesis, we have proposed a novel utility-driven decision engine to augment the classical independent cascade (IC) propagation model; the new model is called UIC. The utility function is powerful enough to capture the various kinds of relationships among the propagating items, such as complementary, competition, and a mix of the two. As a result, UIC can be studied under such configurations. Further, it also enables the study of the filter bubble problem, where the requirement involves a mix of complementary and competing behavior.

Additionally, UIC opens up the study of a novel optimization objective in the context of multi-item influence maximization, called welfare maximization. We show that by incorporating the item relationships, higher social welfare, i.e., the users' satisfaction from adopting various items, can be achieved. This, in turn, results in user retention in a social network and also provides a balanced overall exposure which can help mitigate filter bubbles.

In Chapter 2 we introduced UIC and studied it under a setting where items are only of complementary type. In a first-of-a-kind result, we show that even when the objective is not submodular, a simple greedy achieves $1 - \frac{1}{e}$ approximation guarantee. We also devise an efficient prefix preserving seed selection algorithm along the way. This chapter showcases how UIC can effectively solve the influence maximization problem for complementary items, which the traditional influence maximization papers have overlooked.

The greedy algorithm also uses a prefix preserving seed selection algorithm. We devised an efficient way to extend the state-of-the-art reverse influence sampling-based seed selection algorithm, IMM, to make it prefix preserving. As a result, our algorithm is shown to scale for real networks having millions of nodes.

Chapter 3 extends the study of the UIC framework for competing items. We show that the objective is not monotone, not submodular. In fact, by using a non-trivial gap preserving reduction, we show that welfare can not be approximated within any constant unless $P = NP$.

We, therefore, propose algorithms that have either instance-dependent

approximation or constant approximation under reasonable assumptions. Our analysis shows that with stronger assumptions, the approximation quality improves. This chapter also demonstrates the power of UIC to model a spectrum of competition among the propagating items, which those existing models could not do.

Finally, in Chapter 4, in the context of mitigating the filter bubble problem, UIC is used for an objective that involves both complementarity and competition. While it models the real world more realistically, it makes the optimization problem significantly more difficult. Even under reasonable assumptions, a constant approximation could not be provided, hence we design instance-dependent approximation algorithms and effective heuristics. We hope that our work in this chapter provides a more realistic paradigm, using a combination of competition and complementary behavior, which is needed for tackling the filter bubble problem.

In summary, we show that awareness does not necessarily lead to adoption, and hence separating the two is an important step to achieving a more powerful and realistic influence propagation model. We formalize the adoption decision using the utility theory of economics, whereas awareness comes via influence propagation. We believe that our principled approach can benefit the ongoing influence maximization research on several fronts. To that end, we propose some future extensions of our work presented in this thesis.

- **Local model parameters:** Throughout this thesis we assumed that model parameters are not user-specific. Therefore we used a global noise distribution in Chapters 2 and 3, and in Chapter 4, global competition and reward parameters. However, several works on item adoption have shown that the decision-making process is individual [19, 73, 80], therefore having a noise distribution specific to individual users, will be more realistic. However, having such general noise makes the problem more difficult to tract. A feasible middle ground can be clustering users into communities and then having community-specific noise distributions.

  Further, the noise distribution assumed in the thesis is a zero-mean Gaussian distribution; therefore it is a symmetric distribution around the mean. In reality, however, utility is far from being symmetric. Instead, it would have a long tail representing the skew in users' valuations. Users' sentiments such as brand loyalty often cause such skews. Therefore, extending the results for asymmetric distributions will be an interesting option for future research.

- **Modeling users' decision making process:** Throughout the thesis it is assumed that the value function is learned at the beginning of the propagation and then it is used to model how users adopt and propagate items. In contrast, users' valuations can be learned by receiving feedback in the form of actions they take in the network. It requires extending our current model using a learning paradigm such as reinforcement learning [125].

  Further, our current adoption model assumes that users' adoption decisions are rational and are governed only by the absolute utility of the items. Later works in economics have shown that the concept of relative utility plays a major role in real-world decision-making related to adoptions by the users. Therefore alternatives such as prospect theory [79] have emerged as a more realistic way to model the adoption decisions. Extending our UIC model for prospect theory-based adoption decisions will be an interesting future research to consider.

- **Alternatives to the greedy approach:** The algorithms devised in different chapters of the thesis, primarily use a greedy algorithm for the seed selection. Some of the key advantages of using greedy are as follows - (a) greedy is shown to achieve a $(1-\frac{1}{e})$ approximation bound, which is the best possible approximation guarantee for the underlying IM problem [51]; (b) Using a reverse influence sampling technique, greedy can effectively scale for billion-sized real social networks. However the greedy algorithm needs to store all the samples, and as a result, it suffers from high memory requirement [114].

  Attempts to address this issue either use a simplified problem instance that in turn degrades the solution quality [114], or they rely on properties specific to a diffusion process [109], while still using an overall greedy strategy. Departing from the greedy approach, a recent work [16] has proposed the use of a fractional approach to solving the IM problem that addresses the memory issue while providing state-of-the-art solution quality. Other optimization techniques such as branch-and-bound [86], can be considered as alternatives. However, more research is needed to ensure that such algorithms also scale for large social networks and conjoining them with users' adoption decisions.

- **Approximation algorithms for non-monotone, non-submodular functions:** As we move closer to reality, the objective function loses

properties, such as monotonicity or submodularity, that are the cornerstones of having an approximation guarantee for the greedy algorithm. In our work, we show that under certain conditions it is possible to achieve a constant $1-\frac{1}{e}$ approximation even when the objective is non-submodular. Further research is needed to generalize this observation, which we believe would provide more insight into having constant approximation for non-submodular objectives in general.

- **Multi-item filter bubble:** In Chapter 4, the filter bubble problem is studied in the context of two item propagation. A natural extension is to study it for more than two items. In a real network, propagations of such nature are observed quite often. To illustrate, consider the topic of abortion, where the opinions can be pro-abortion, anti-abortion, abortion only under certain circumstances such as age, health condition developed, etc. Under multiple items, designing the right objective is even more challenging, as it is not clear to what extent an item would counter the effect of the filter bubble created by some other item. Signed embedding of graphs has been proposed recently to measure the effect of polarization in a network where multiple opinions can propagate [75]. Extending such a model that can capture the relations of multiple items and then mitigate the effect of filter bubbles, is an interesting research direction to pursue.

- **Considering the effect of time:** Time is an important parameter to consider in real propagations. As time progresses, new nodes and edges may be added to the network as more users join and new connections are created. Likewise, some nodes and edges get deleted as time proceeds. Further, the influence weights of the existing edges can also change over time. Therefore a dynamic graph concerning the time parameter can be considered to accurately model these effects. The thesis assumes a static graph in contrast. Hence the algorithms introduced in this thesis, need to be extended for such dynamic graphs.

  Additionally, time also plays a critical role for a particular campaign to be effective. E.g., in the context of a filter bubble, if there is a significant delay between the two propagations, then users' stance after seeing the earlier opinion could get hardened. Consequently, it would be more challenging to convince such users with the opposite opinion. There is a significant research opportunity to extend UIC under such time-sensitive influence propagation.

- **Other application areas:** IM is currently used for various applications besides viral marketing. We have studied one such, filter bubble in the thesis. Outbreak detection [89], detecting the source of an infection [78] are other such applications to name a few. As seen with the filter bubble problem, the "utility" changes as the application changes. Therefore designing new utility functions and studying their impact on the welfare objective would be interesting.

- **Alternative diffusion models** This thesis uses discrete time-independent cascade (IC) as the only propagation model. Several other alternatives are studied in the literature such as linear threshold (LT) [116] and continuous-time models [64]. UIC needs to be extended for those propagation models in further research attempts.

Bibliography

[1] Ben Abramowitz and Elliot Anshelevich. Utilitarians without utilities: Maximizing social welfare for graph problems using only ordinal preferences. In *AAAI*, pages 894–901, 2018. → pages 24, 76, 124

[2] Leman Akoglu. Quantifying political polarity based on bipartite opinion networks. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 8, 2014. → pages 119, 123

[3] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36, 2017. → page 119

[4] Victor Amelkin, Petko Bogdanov, and Ambuj K Singh. A distance measure for the analysis of polar opinion dynamics in social networks. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13(4):1–34, 2019. → pages 119, 123

[5] Victor Amelkin and Ambuj K Singh. Fighting opinion control in social networks via link recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 677–685, 2019. → pages 119, 123

[6] Roy M Anderson and Robert M May. *Infectious diseases of humans: dynamics and control*. Oxford university press, 1992. → page 3

[7] Nicolas M Anspach. The new personal influence: How our facebook friends influence the news we read. *Political Communication*, 34(4):590–606, 2017. → page 1

[8] Cigdem Aslay, Francesco Bonchi Laks VS Lakshmanan, and Wei Lu. Revenue maximization in incentivized social advertising. *Proceedings of the VLDB Endowment*, 10(11), 2017. → pages 76, 124

[9] Cigdem Aslay, Wei Lu, Francesco Bonchi, Amit Goyal, and Laks VS Lakshmanan. Viral marketing meets social advertising: Ad allocation

with minimum regret. *Proceedings of the VLDB Endowment*, 8(7), 2015. → pages 76, 124

[10] Eytan Bakshy, Solomon Messing, and Lada A Adamic. Exposure to ideologically diverse news and opinion on facebook. *Science*, 348(6239):1130–1132, 2015. → pages 119, 123

[11] Robert F Bales. Interaction process analysis; a method for the study of small groups. 1950. → page 1

[12] Prithu Banerjee, Wei Chen, and Laks VS Lakshmanan. Maximizing welfare in social networks under a utility driven influence diffusion model. In *Proceedings of the 2019 International Conference on Management of Data*, pages 1078–1095, 2019. → pages v, 125

[13] Prithu Banerjee, Wei Chen, and Laks VS Lakshmanan. Maximizing social welfare in a competitive diffusion model. *Proceedings of the VLDB Endowment*, 14(4):613–625, 2020. → page v

[14] Austin R Benson, Ravi Kumar, and Andrew Tomkins. A discrete choice model for subset selection. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 37–45, 2018. → pages v, 8, 73, 76, 112

[15] Joseph Berger, Susan J Rosenholtz, and Morris Zelditch Jr. Status organizing processes. *Annual review of sociology*, 6(1):479–508, 1980. → page 1

[16] Glenn S Bevilacqua and Laks VS Lakshmanan. A fractional memory-efficient approach for online continuous-time influence maximization. *The VLDB Journal*, 31(2):403–429, 2022. → page 157

[17] Shishir Bharathi, David Kempe, and Mahyar Salek. Competitive influence maximization in social networks. In *International Workshop on Web and Internet Economics*, pages 306–311. Springer, 2007. → pages 7, 15, 22, 76, 128

[18] Devipsita Bhattacharya and Sudha Ram. Sharing news articles using 140 characters: A diffusion analysis on twitter. In *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 966–971. IEEE, 2012. → page 119

[19] Sayan Bhattacharya, Wolfgang Dvořák, Monika Henzinger, and Martin Starnberger. Welfare maximization with friends-of-friends network externalities. *Theory of Computing Systems*, 61(4):948–986, 2017. → pages 24, 60, 68, 69, 77, 125, 156

[20] Robin W Boadway and Neil Bruce. *Welfare economics*. B. Blackwell New York, 1984. → pages 30, 76, 124

[21] Robert M Bond, Christopher J Fariss, Jason J Jones, Adam DI Kramer, Cameron Marlow, Jaime E Settle, and James H Fowler. A 61-million-person experiment in social influence and political mobilization. *Nature*, 489(7415):295–298, 2012. → page 1

[22] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. Maximizing social influence in nearly optimal time. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 946–957. SIAM, 2014. → pages 5, 15, 21, 49, 137

[23] Allan Borodin, Yuval Filmus, and Joel Oren. Threshold models for competitive influence in social networks. In *International Workshop on Internet and Network Economics*, pages 539–550. Springer, 2010. → pages 7, 22

[24] Guy Bresler, Frederic Koehler, and Ankur Moitra. Learning restricted boltzmann machines via influence maximization. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 828–839, 2019. → page 1

[25] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th international conference on World wide web*, pages 665–674. ACM, 2011. → pages 7, 15, 22, 76

[26] John T Cacioppo, Richard E Pett, and Cal D Stoltenberg. Processes of social influence: The elaboration likelihood model of persuasion. 1985. → page 1

[27] Robert Carbaugh. *Contemporary Economics: An Applications Approach*. Routledge, 8th edition, 2016. → pages 32, 79

[28] Tim Carnes, Chandrashekhar Nagarajan, Stefan M Wild, and Anke Van Zuylen. Maximizing influence in a competitive social network: a

follower's perspective. In *Proceedings of the ninth international conference on Electronic commerce*, pages 351–360. ACM, 2007. → pages 7, 22

[29] Manuel Castells. The information age. *Media Studies: A Reader*, 2(7):152, 2010. → page 119

[30] O. Celma. Last.fm Dataset – 1K users. `http://www.dtic.upf.edu/~ocelma/MusicRecommendationDataset/lastfm-1K.html`, 2010. → page 112

[31] Parinya Chalermsook, Atish Das Sarma, Ashwin Lall, and Danupon Nanongkai. Social network monetization via sponsored viral marketing. In *ACM SIGMETRICS Performance Evaluation Review*, volume 43, pages 259–270. ACM, 2015. → pages 15, 73

[32] Lena Chang and William B Fairley. Pricing automobile insurance under multivariate classification of risks: additive versus multiplicative. *Journal of Risk and Insurance*, 1979. → page 30

[33] Wei Chen. An issue in the martingale analysis of the influence maximization algorithm imm. *arXiv preprint arXiv:1808.09363*, 2018. → pages 51, 52, 53, 101

[34] Wei Chen, Alex Collins, Rachel Cummings, Te Ke, Zhenming Liu, David Rincon, Xiaorui Sun, Yajun Wang, Wei Wei, and Yifei Yuan. Influence maximization in social networks when negative opinions may emerge and propagate. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 379–390. SIAM, 2011. → pages 7, 22

[35] Wei Chen, Laks VS Lakshmanan, and Carlos Castillo. Information and influence propagation in social networks. *Synthesis Lectures on Data Management*, 5(4):1–177, 2013. → pages 2, 5, 22, 72, 75, 76, 122

[36] Wei Chen, Fu Li, Tian Lin, and Aviad Rubinstein. Combining traditional marketing and viral marketing with amphibious influence maximization. In *EC*, 2015. → page 21

[37] Wei Chen, Wei Lu, and Ning Zhang. Time-critical influence maximization in social networks with time-delayed diffusion process. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012. → page 5

[38] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1029–1038. ACM, 2010. → pages 1, 4, 15

[39] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM, 2009. → page 5

[40] Wei Chen, Yifei Yuan, and Li Zhang. Scalable influence maximization in social networks under the linear threshold model. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 88–97. IEEE, 2010. → pages 5, 15

[41] Xi Chen, Jefrey Lijffijt, and Tijl De Bie. Quantifying and minimizing risk of conflict in social networks. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1197–1205, 2018. → page 123

[42] Edith Cohen, Daniel Delling, Thomas Pajor, and Renato F Werneck. Sketch-based influence maximization and computation: Scaling up with guarantees. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 629–638. ACM, 2014. → pages 15, 21

[43] Michael D Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. Political polarization on twitter. In *Fifth international AAAI conference on weblogs and social media*, 2011. → page 123

[44] Federico Corò, Emilio Cruciani, Gianlorenzo D'Angelo, and Stefano Ponziani. Vote for me! election control via social influence in arbitrary scoring rule voting systems. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1895–1897, 2019. → page 1

[45] Peter Cramton, Yoav Shoham, and Richard Steinberg. Combinatorial auctions. pages 1–33, 2006. → pages 11, 23

[46] Gianlorenzo D'Angelo, Debashmita Poddar, and Cosimo Vinci. Better bounds on the adaptivity gap of influence maximization under

full-adoption feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12069–12077, 2021. → page 1

[47] Samik Datta, Anirban Majumder, and Nisheeth Shrivastava. Viral marketing for multiple products. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 118–127. IEEE, 2010. → pages 7, 22, 76

[48] Morton Deutsch and Harold B Gerard. A study of normative and informational social influences upon individual judgment. *The journal of abnormal and social psychology*, 51(3):629, 1955. → page 1

[49] David Easley, Jon Kleinberg, et al. *Networks, crowds, and markets*, volume 8. Cambridge university press Cambridge, 2010. → page 1

[50] Golnoosh Farnad, Behrouz Babaki, and Michel Gendreau. A unifying framework for fairness-aware influence maximization. In *Companion Proceedings of the Web Conference 2020*, pages 714–722, 2020. → pages 119, 124

[51] Uriel Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998. → page 157

[52] Uriel Feige and Jan Vondrák. The submodular welfare problem with demand queries. *Theory of Computing*, 6(1):247–290, 2010. → pages 9, 16, 23, 25, 38, 76, 124

[53] Seth Flaxman, Sharad Goel, and Justin M Rao. Filter bubbles, echo chambers, and online news consumption. *Public opinion quarterly*, 80(S1):298–320, 2016. → page 119

[54] John R French, Bertram Raven, and Dorwin Cartwright. The bases of social power. *Classics of organization theory*, 7:311–320, 1959. → page 1

[55] Sainyam Galhotra, Akhil Arora, and Shourya Roy. Holistic influence maximization: Combining scalability and efficiency with opinion-aware models. In *Proceedings of the 2016 International Conference on Management of Data*, pages 743–758, 2016. → page 5

[56] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Reducing controversy by connecting opposing views. In *Proceedings of the Tenth ACM International Confer-*

ence on Web Search and Data Mining, pages 81–90, 2017. → pages 119, 123

[57] Kiran Garimella, Aristides Gionis, Nikos Parotsidis, and Nikolaj Tatti. Balancing information exposure in social networks. In *Advances in Neural Information Processing Systems*, pages 4663–4671, 2017. → pages 7, 76, 103, 119, 120, 123, 128, 148

[58] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Quantifying controversy on social media. *ACM Transactions on Social Computing*, 1(1):1–27, 2018. → page 123

[59] R Kelly Garrett. Echo chambers online?: Politically motivated selective exposure among internet news users. *Journal of computer-mediated communication*, 14(2):265–285, 2009. → pages 119, 120, 123, 124

[60] Matthew Gentzkow and Jesse M Shapiro. Ideological segregation online and offline. *The Quarterly Journal of Economics*, 126(4):1799–1839, 2011. → page 124

[61] Shay Gershtein, Tova Milo, Brit Youngmann, and Gal Zeevi. Im balanced: influence maximization under balance constraints. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 1919–1922, 2018. → pages 119, 124

[62] Jacob Goldenberg, Barak Libai, and Eitan Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3):211–223, 2001. → pages 2, 3

[63] Jacob Goldenberg, Barak Libai, and Eitan Muller. Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata. *Academy of Marketing Science Review*, 9(3):1–18, 2001. → page 3

[64] Manuel Gomez-Rodriguez, Le Song, Nan Du, Hongyuan Zha, and Bernhard Schölkopf. Influence estimation and maximization in continuous-time diffusion networks. *ACM Transactions on Information Systems (TOIS)*, 34(2):1–33, 2016. → page 159

[65] Amit Goyal, Wei Lu, and Laks VS Lakshmanan. Celf++ optimizing the greedy algorithm for influence maximization in social networks. In

Proceedings of the 20th international conference companion on World wide web, pages 47–48, 2011. → page 5

[66] Amit Goyal, Wei Lu, and Laks VS Lakshmanan. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 211–220. IEEE, 2011. → pages 5, 15

[67] Mark Granovetter. Threshold models of collective behavior. *American journal of sociology*, 83(6):1420–1443, 1978. → page 2

[68] Pedro Calais Guerra, Wagner Meira Jr, Claire Cardie, and Robert Kleinberg. A measure of polarization on social media networks based on community boundaries. In *Seventh international AAAI conference on weblogs and social media*, 2013. → pages 119, 123

[69] Qiang He, Xingwei Wang, Zhencheng Lei, Min Huang, Yuliang Cai, and Lianbo Ma. Tifim: A two-stage iterative framework for influence maximization in social networks. *Applied Mathematics and Computation*, 354:338–352, 2019. → page 1

[70] Xinran He, Guojie Song, Wei Chen, and Qingye Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, pages 463–474. SIAM, 2012. → pages 7, 15, 22, 76

[71] Sabrina Helm. Viral marketing-establishing customer relationships by'word-of-mouse'. *Electronic markets*, 10(3):158–161, 2000. → page 1

[72] Jack Hirshleifer. The private and social value of information and the reward to inventive activity. In *Uncertainty in Economics*, pages 541–556. Elsevier, 1978. → pages 23, 26

[73] Ming Hu, Joseph Milner, and Jiahua Wu. Liking and following and the newsvendor: Operations and marketing policies under social influence. *Management Science*, 62(3):867–879, 2016. → pages 1, 156

[74] Keke Huang, Sibo Wang, Glenn Bevilacqua, Xiaokui Xiao, and Laks V. S. Lakshmanan. Revisiting the stop-and-stare algorithms for influence maximization. *Proc. VLDB Endow.*, 10(9):913–924, 2017. → pages 5, 21, 60, 103, 134, 149

[75] Zexi Huang, Arlei Silva, and Ambuj Singh. Pole: Polarized embedding for signed networks. *arXiv preprint arXiv:2110.09899*, 2021. → page 158

[76] Albert Xin Jiang and Kevin Leyton-Brown. Bidding agents for online auctions with hidden bids. *Machine Learning*, 67(1-2):117–143, 2007. → pages 26, 66

[77] Jiaojiao Jiang, Sheng Wen, Shui Yu, Yang Xiang, and Wanlei Zhou. Identifying propagation sources in networks: State-of-the-art and comparative studies. *IEEE Communications Surveys & Tutorials*, 19(1):465–481, 2016. → page 3

[78] Jiaojiao Jiang, Sheng Wen, Shui Yu, Yang Xiang, and Wanlei Zhou. Identifying propagation sources in networks: State-of-the-art and comparative studies. *IEEE Communications Surveys Tutorials*, 19(1):465–481, 2017. → page 159

[79] Daniel Kahneman and Amos Tversky. Prospect theory: An analysis of decision under risk. In *Handbook of the fundamentals of financial decision making: Part I*, pages 99–127. World Scientific, 2013. → page 157

[80] Shlomo Kalish. A new product adoption model with price, advertising, and uncertainty. *Management science*, 31(12):1569–1585, 1985. → page 156

[81] Michael Kapralov, Ian Post, and Jan Vondrák. Online submodular welfare maximization: Greedy is optimal. In *Proceedings of the Twenty-fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 1216–1225, 2013. → pages 9, 16, 23, 38, 76, 124

[82] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972. → page 4

[83] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003. → pages 1, 2, 3, 4, 9, 15, 16, 21, 30, 31, 71, 72, 75, 76, 82, 122, 130

[84] Nitish Korula, Vahab Mirrokni, and Morteza Zadimoghaddam. Online submodular welfare maximization: Greedy beats 1/2 in random order. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 889–898. ACM, 2015. → pages 9, 16, 23, 38, 76, 124

[85] Paul Lamere. LastFM-ArtistTags2007 dataset. `http://musicmachinery.com/2010/11/10/lastfm-artisttags2007/`, 2008. → page 112

[86] Ailsa H Land and Alison G Doig. An automatic method for solving discrete programming problems. In *50 Years of Integer Programming 1958-2008*, pages 105–132. Springer, 2010. → page 157

[87] Bibb Latané. The psychology of social impact. *American psychologist*, 36(4):343, 1981. → page 1

[88] Bibb Latané. Dynamic social impact: The creation of culture by communication. *Journal of communication*, 46(4):13–25, 1996. → page 1

[89] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429, 2007. → pages 5, 159

[90] Hui Li, Sourav S Bhowmick, Jiangtao Cui, Yunjun Gao, and Jianfeng Ma. Getreal: Towards realistic selection of influence maximization strategies in competitive networks. In *Proceedings of the 2015 ACM SIGMOD international conference on management of data*, pages 1525–1537, 2015. → pages 7, 76

[91] Qiang Li, Wei Chen, Xiaoming Sun, and Jialin Zhang. Influence maximization with ϵ-almost submodular threshold functions. In *NIPS*, 2017. → pages 21, 22

[92] Yuchen Li, Ju Fan, George Ovchinnikov, and Panagiotis Karras. Maximizing multifaceted network influence. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 446–457. IEEE, 2019. → page 76

[93] Yuchen Li, Ju Fan, Yanhao Wang, and Kian-Lee Tan. Influence maximization on social graphs: A survey. *TKDE*, 2018. → pages 15, 76

[94] Q Vera Liao and Wai-Tat Fu. Can you hear me now? mitigating the echo chamber effect by source position indicators. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 184–196, 2014. → page 123

[95] Yishi Lin and John CS Lui. Analyzing competitive influence maximization problems with partial information: An approximation algorithmic framework. *Performance Evaluation*, 91:187–204, 2015. → pages 9, 76, 103, 128

[96] Wei Lu, Francesco Bonchi, Amit Goyal, and Laks VS Lakshmanan. The bang for the buck: fair competitive viral marketing from the host perspective. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 928–936. ACM, 2013. → pages 7, 9, 15, 22, 72, 76

[97] Wei Lu, Wei Chen, and Laks VS Lakshmanan. From competition to complementarity: comparative influence diffusion and maximization. *Proceedings of the VLDB Endowment*, 9(2):60–71, 2015. → pages 7, 8, 15, 16, 17, 21, 22, 38, 55, 56, 61, 62, 76, 103, 121, 134, 138, 148

[98] Antonis Matakos, Cigdem Aslay, Esther Galbrun, and Aristides Gionis. Maximizing the diversity of exposure in a social network. *IEEE Transactions on Knowledge and Data Engineering*, 2020. → pages 7, 119, 120, 123, 124, 148, 149

[99] Antonis Matakos, Evimaria Terzi, and Panayiotis Tsaparas. Measuring and moderating opinion polarization in social networks. *Data Mining and Knowledge Discovery*, 31(5):1480–1505, 2017. → pages 119, 123

[100] Julian McAuley, Rahul Pandey, and Jure Leskovec. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2015. → page 23

[101] Paul Milgrom and John Roberts. Complementarities and fit strategy, structure, and organizational change in manufacturing. *Journal of accounting and economics*, 1995. → page 30

[102] Elchanan Mossel and Sebastien Roch. On the submodularity of influence in social networks. In *Proceedings of the thirty-ninth annual*

ACM symposium on Theory of computing, pages 128–134, 2007. → pages 1, 4

[103] Cameron Musco, Christopher Musco, and Charalampos E Tsourakakis. Minimizing polarization and disagreement in social networks. In *Proceedings of the 2018 World Wide Web Conference*, pages 369–378, 2018. → pages 119, 123

[104] Seth A Myers and Jure Leskovec. Clash of the contagions: Cooperation and competition in information diffusion. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pages 539–548. IEEE, 2012. → pages 8, 23

[105] Roger B Myerson. Optimal auction design. *Mathematics of operations research*, 6(1):58–73, 1981. → pages 9, 11, 16, 23, 25, 76, 124

[106] Ramasuri Narayanam and Amit A Nanavati. Viral marketing for product cross-sell through social networks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 581–596. Springer, 2012. → pages 7, 22

[107] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978. → page 4

[108] C. Thi Nguyen. Echo chambers and epistemic bubbles. *Episteme*, 17(2):141–161, 2020. → page 119

[109] Hung T Nguyen, Tri P Nguyen, NhatHai Phan, and Thang N Dinh. Importance sketching of influence dynamics in billion-scale networks. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 337–346. IEEE, 2017. → page 157

[110] Hung T Nguyen, My T Thai, and Thang N Dinh. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *Proceedings of the 2016 International Conference on Management of Data*, pages 695–710. ACM, 2016. → pages 1, 5, 21, 48, 60, 103, 121, 134, 149

[111] Walter Nicholson and Christopher Snyder. *Microeconomic theory: Basic principles and extensions*. Nelson Education, 2011. → pages 11, 23

[112] Dimitar Nikolov, Diego FM Oliveira, Alessandro Flammini, and Filippo Menczer. Measuring online social bubbles. *PeerJ computer science*, 1:e38, 2015. → page 119

[113] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic game theory*. Cambridge university press, 2007. → pages 9, 11, 16, 23, 25, 76, 124

[114] Naoto Ohsaka, Tomohiro Sonobe, Sumio Fujita, and Ken-ichi Kawarabayashi. Coarsening massive influence networks for scalable diffusion analysis. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 635–650, 2017. → page 157

[115] Eli Pariser. *The filter bubble: What the Internet is hiding from you*. Penguin UK, 2011. → pages 119, 123

[116] Nishith Pathak, Arindam Banerjee, and Jaideep Srivastava. A generalized linear threshold model for multiple cascades. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 965–970. IEEE, 2010. → pages 7, 9, 15, 22, 72, 159

[117] Matthew Perrone and Ben Wieder. Pro-painkiller echo chamber shaped policy amid drug epidemic. *The Center for Public Integrity*, 2016. → page 119

[118] Eric Rasmusen and Basil Blackwell. Games and information. *Cambridge, MA*, 15, 1994. → page 23

[119] Hans Risselada, Peter C Verhoef, and Tammo HA Bijmolt. Dynamic effects of social influence and direct marketing on the adoption of high-technology products. *Journal of Marketing*, 78(2):52–68, 2014. → page 1

[120] George Ritzer et al. *The Blackwell encyclopedia of sociology*, volume 1479. Blackwell Publishing New York, 2007. → page 1

[121] Grant Schoenebeck and Biaoshuai Tao. Beyond worst-case (in)approximability of nonsubmodular influence maximization. In *WINE*, 2017. → pages 21, 22

[122] Aybike ŞİMŞEK and KARA Resul. Using swarm intelligence algorithms to detect influential individuals for influence maximization in social networks. *Expert Systems with Applications*, 114:224–236, 2018. → page 1

[123] Twitter dataset. `https://snap.stanford.edu/data/`. Accessed: 2018-05-30. → pages 55, 103, 148

[124] Tao Sun, Wei Chen, Zhenming Liu, Yajun Wang, Xiaorui Sun, Ming Zhang, and Chin-Yew Lin. Participation maximization based on social influence in online discussion forums. In *ICWSM*, 2011. → pages 24, 77, 124, 125

[125] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018. → page 157

[126] Jing Tang, Xueyan Tang, Xiaokui Xiao, and Junsong Yuan. Online processing algorithms for influence maximization. In *SIGMOD*, 2018. → pages 1, 48, 134

[127] Youze Tang, Yanchen Shi, and Xiaokui Xiao. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1539–1554. ACM, 2015. → pages 5, 15, 19, 21, 32, 48, 49, 50, 51, 52, 54, 57, 98, 100, 101, 121, 138, 146

[128] Youze Tang, Xiaokui Xiao, and Yanchen Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 75–86. ACM, 2014. → pages 5, 21, 61

[129] Donald M Topkis. *Supermodularity and Complementarity*. Princeton University Press, 1998. → pages 30, 32

[130] Petter Törnberg. Echo chambers and viral misinformation: Modeling fake news as complex contagion. *PLoS one*, 13(9):e0203958, 2018. → pages 120, 124

[131] Alan Tsang, Bryan Wilder, Eric Rice, Milind Tambe, and Yair Zick. Group-fairness in influence maximization. In *IJCAI*, 2019. → pages 119, 124

[132] Sijing Tu, Cigdem Aslay, and Aristides Gionis. Co-exposure maximization in online social networks. *Advances in Neural Information Processing Systems*, 33, 2020. → pages 7, 119, 120, 123, 129

[133] Leslie G Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979. → page 4

[134] Zheng Wen, Branislav Kveton, Michal Valko, and Sharan Vaswani. Online influence maximization under independent cascade model with semi-bandit feedback. *Advances in neural information processing systems*, 30, 2017. → page 1

[135] Mateusz Wilinski and Andrey Lokhov. Prediction-centric learning of independent cascade dynamics from partial observations. In *International Conference on Machine Learning*, pages 11182–11192. PMLR, 2021. → page 1

[136] Stephan Winter, Caroline Brückner, and Nicole C Krämer. They came, they liked, they commented: Social influence on facebook news channels. *Cyberpsychology, Behavior, and Social Networking*, 18(8):431–436, 2015. → page 1

[137] Liwang Zhu, Qi Bao, and Zhongzhi Zhang. Minimizing polarization and disagreement in social networks via link recommendation. *Advances in Neural Information Processing Systems*, 34, 2021. → pages 119, 123

[138] Yuqing Zhu, Deying Li, and Zhao Zhang. Minimum cost seed set for competitive social influence. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016. → pages 9, 72, 76, 128