Computational Social Influence:
Models, Algorithms, and Applications

by

Wei Lu

B.Sc., Simon Fraser University, 2010
B.Eng., Zhejiang University, 2010

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)

July 2016

© Wei Lu, 2016
Abstract

Social influence is a ubiquitous phenomenon in human life. Fueled by the extreme popularity of online social networks and social media, computational social influence has emerged as a subfield of data mining whose goal is to analyze and understand social influence using computational frameworks such as theoretical modeling and algorithm design. It also entails substantial application potentials for viral marketing, recommender systems, social media analysis, etc. In this dissertation, we present our research achievements that take significant steps toward bridging the gap between elegant theories in computational social influence and the needs of two real-world applications: viral marketing and recommender systems. In Chapter 2, we extend the classic Linear Thresholds model to incorporate price and valuation to model the diffusion process of new product adoption; we design a greedy-style algorithm that finds influential users from a social network as well as their corresponding personalized discounts to maximize the expected total profit of the advertiser. In Chapter 3, we propose a novel business model for online social network companies to sell viral marketing as a service to competing advertisers, for which we tackle two optimization problems: maximizing total influence spread of all advertisers and allocating seeds to advertisers in a fair manner. In Chapter 4, we design a highly expressive diffusion model that can capture arbitrary relationship between two propagating entities to arbitrary degrees. We then study the influence maximization problem in a novel setting consisting of two complementary entities and design efficient approximation algorithms. Next, in Chapter 5, we apply social influence into recommender systems. We model the dynamics of user interest evolution using social influence, as well as attraction and aversion effects. As a result, making effective recommendations are substantially more challenging and we apply semi-definite programming techniques to achieve near-optimal solutions. Chapter 6 concludes the dissertation and outlines possible future research directions.
Preface

This dissertation is the result of collaborative research with several other computer scientists. All work were done under the supervision of Prof. Laks V.S. Lakshmanan.

Chapter 2 is based on a publication [109] in the 2012 IEEE International Conference on Data Mining, a joint work with Laks V.S. Lakshmanan. I developed the theory and conducted the experiments under his guidance.

Chapter 3 is based on a publication [106] in the 2013 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, a joint work with Francesco Bonchi, Amit Goyal, and Laks V.S. Lakshmanan. I developed the theory and conducted the experiments under their guidance. Dr. Goyal refined the idea of the K-LT model that eventually led to submodularity. Prof. Lakshmanan and Dr. Goyal gave the proof of NP-hardness for the Fair Seed Allocation problem.

Chapter 4 is based on a publication in the Proceedings of VLDB Endowment (volume 9, issue 2) [107], a joint work with Wei Chen at Microsoft Research Asia and Laks V.S. Lakshmanan. I proposed the current ComIC model, studied its properties, and conducted the experiments under their guidance. Wei Chen proposed the possible world definition, GeneralTIM framework, and the RR-ComIC algorithm.

Chapter 5 is based on a publication in the 2014 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining [108], a joint work with Stratis Ioannidis, Smriti Bhagat, and Laks V.S. Lakshmanan. I developed the theory and conducted experiments under their guidance. The semi-definite relaxation formulation was developed by Stratis Ioannidis.
# Table of Contents

Abstract ............................................................... ii
Preface ................................................................. iii
Table of Contents ....................................................... iv
List of Tables .......................................................... viii
List of Figures .......................................................... x
Acknowledgments ....................................................... xiii
Dedication ............................................................... xiv

1 Introduction ......................................................... 1
   1.1 Influence Maximization in Social Networks ................. 3
   1.2 Challenges and Key Contributions .......................... 7
       1.2.1 Viral Marketing ........................................ 8
       1.2.2 Competition and Complementarity: A Unified Influence Propagation Model .................. 9
       1.2.3 Applying Social Influence in Recommender Systems . 10
   1.3 Outline .......................................................... 11

2 Influence-Driven Profit Maximization in Social Networks . 12
   2.1 Introduction .................................................... 12
   2.2 Related Work ................................................... 14
   2.3 Linear Threshold Model with User Valuations .............. 16
       2.3.1 Model and Problem Definition ....................... 16
   2.4 Special Case with Fixed Valuations ......................... 17
   2.5 General Properties of the LT-V Model ..................... 21
4.5 Properties of ComIC Model ........................................ 88
4.5.1 Monotonicity .................................................... 88
4.5.2 Submodularity in Complementary Setting ................. 92
4.6 Scalable Approximation Algorithms .............................. 95
4.6.1 A General Framework Extending TIM ....................... 96
4.6.2 Generating RR-Sets for Influence Maximization with ComIC ........................................... 99
4.7 The Sandwich Approximation Strategy ......................... 107
4.7.1 Methodology ..................................................... 107
4.7.2 Applying Sandwich Approximation to Influence Maximi-
        zation ............................................................ 109
4.8 Experiments .......................................................... 109
4.8.1 Experiments with Synthetic Adoption Probabilities .... 110
4.8.2 Learning Global Adoption Probabilities .................... 113
4.8.3 Experimental Settings with Learned Adoption Prob-
        abilities .......................................................... 115
4.8.4 Results and Analysis ........................................... 117
4.9 Discussion and Future Work ....................................... 120
5 Recommendations with Attraction, Aversion, and Social
        Influence Diffusion ........................................... 122
5.1 Introduction ......................................................... 122
5.2 Related Work ....................................................... 126
5.3 Problem Formulation ............................................... 127
5.3.1 Overview ......................................................... 127
5.3.2 Recommender System and User Utilities .................. 128
5.3.3 Interest Evolution .............................................. 129
5.3.4 Recommended Item Distribution ............................. 131
5.3.5 Recommendation Objective ................................... 132
5.4 Algorithm Design ................................................... 132
5.4.1 Steady State Social Welfare .................................. 132
5.4.2 SDP Relaxation .................................................. 134
5.4.3 Solvable Special Cases ........................................ 138
5.4.4 Finite Catalog ................................................... 139
5.5 Parameter Learning ................................................ 139
5.6 Experiments .......................................................... 141
5.6.1 Evaluation of Parameter Learning ........................... 142
5.6.2 Social Welfare Performance ................................. 145
5.7 Discussion and Future Work ....................................... 150
List of Tables

Table 2.1 Dataset statistics ........................................ 30
Table 2.2 Running time, in hours (WD weights, $c_a = 0.1$) ...... 36
Table 2.3 Running time, in hours (TV weights, $c_a = 0.1$) ...... 37

Table 3.1 Cyclic behaviors of $C_1$ and $C_2$, assuming $c < 91/3$. The rightmost column represents the best response by a company. E.g., $(C_1, 28)$ means that company $C_1$ will change its bid to 28 in the next round. Note that the round 7 is identical to round 1, indicating a cyclic trend of the strategic behaviors. .................................................. 62
Table 3.2 Datasets statistics ........................................... 65
Table 3.3 Test cases with varying budget distribution .......... 65
Table 3.4 Comparing Needy-Greedy and Integer Linear Programming: For each dataset, we show the largest deviation between Needy-Greedy’s outcome and Integer Linear Programming’s outcome, in percentage, among all instances where Integer Linear Programming finished. ................................. 69

Table 4.1 Frequently used acronyms. ................................. 77
Table 4.2 Statistics of graph data (all directed) ................. 110
Table 4.3 Percentage improvement of GeneralTIM with RR-ComIC over VanillaIC & Copying, where the fixed $B$-seed set is chosen to be the 101st to 200th ones from the VanillaIC order . 111
Table 4.4 Percentage improvement of GeneralTIM with RR-ComIC over VanillaIC & Copying, where the fixed $B$-seed set is randomly chosen ................................................. 111
Table 4.5 Percentage improvement of GeneralTIM with RR-ComIC over VanillaIC & Copying, where the fixed $B$-seed set is chosen to be the top-100 nodes by VanillaIC ............... 112
Table 4.6 Selected GAPs learned for pairs of movies in Flixster . . . 114
Table 4.7 Selected GAPs learned for pairs of books in Douban-Book 114
Table 4.8 Selected GAPs learned for pairs of movies in Douban-Movie 114
Table 4.9 Sandwich approximation factor: $\sigma(S_\nu)/\nu(S_\nu)$ . . . . . . . 118

Table 5.1 Datasets statistics . . . . . . . . . . . . . . . . . . . . . . . . . . . 141
List of Figures

Figure 2.1  Node states in the LT-V model ............... 16
Figure 2.2  An example graph .......................... 26
Figure 2.3  Distribution of influence weights in Flixster ...... 32
Figure 2.4  A review for Canon EOS 300D camera on Epinions.com.
             At the end of the review, the user mentioned the price – $999. ....... 33
Figure 2.5  Statistics of Valuations (Epinions.com) ............ 34
Figure 2.6  Expected profit achieved (Y-axis) on Epinions graphs
             w.r.t. |S| (X-axis). (N)/(U) denotes normal/uniform distribution. ......... 34
Figure 2.7  Expected profit achieved (Y-axis) on Flixster graphs w.r.t. |S| (X-axis). (N)/(U) denotes normal/uniform distribution. ......... 35
Figure 2.8  Expected profit achieved (Y-axis) on NetHEPT graphs
             w.r.t. |S| (X-axis). (N)/(U) denotes normal/uniform distribution. ......... 35
Figure 2.9  Price assigned to seeds (Y-axis) w.r.t. |S| (X-axis) on Epinions-TV with $\mathcal{N}(0.53, 0.14^2)$. .......... 36
Figure 3.1  Sample graph accompanying Example 1 ........... 46
Figure 3.2  An example graph for illustrating adjusted marginal gain 52
Figure 3.3  Adjusted marginal gains on the four datasets. On the X-axis, the seeds are ranked in the order in which they were chosen by the greedy algorithm. .......... 66
Figure 3.4  Minimum amplification factors (higher is better) .... 67
Figure 3.5  Maximum amplification factors (lower bar is better) ... 68
Figure 3.6  Empirical variance of amplification factors for Needy-Greedy, Round-Robin, and Random Allocation. ........ 70
Figure 3.7 Running time comparisons. Bars touching the top of the Y-axis means that the algorithm did not finish within one week. 71
Figure 3.8 Scalability tests of Needy-Greedy 72
Figure 4.1 ComIC model: Node-level automaton for product A. 79
Figure 4.2 ComIC model: Diffusion dynamics 83
Figure 4.3 The graph for Example 3 88
Figure 4.4 The graph for Example 4 92
Figure 4.5 Complementary effects learned from data: The histogram of all \((q_{A|B} - q_{A|\emptyset})\) and \((q_{B|A} - q_{B|\emptyset})\) values on Flixster, Douban-Book, and Douban-Movie (10000 pairs of items each) 114
Figure 4.6 Effects of \(\varepsilon\) on the running time and influence spread of RR-ComIC and RR-ComIC++, on all four datasets. The influence spread achieved by RR-ComIC and RR-ComIC++ were almost identical in all cases, and thus we only drew one line using the spread of RR-ComIC++. 116
Figure 4.7 Influence spread vs. seed set size 118
Figure 4.8 Running time comparisons on real networks: GeneralTIM with RR-ComIC, GeneralTIM with RR-ComIC++, and Greedy with Monte Carlo simulations 119
Figure 4.9 Running time comparisons on synthetic power-law random graphs up to one million nodes: GeneralTIM with RR-ComIC versus GeneralTIM with RR-ComIC++ 119
Figure 5.1 Illustration of aversion and attraction in MovieLens, and gains from accounting for them in optimization. 125
Figure 5.2 The decreasing trend of Test RMSE, RMSE_\(\alpha\),RMSE_\(\gamma\), and RMSE_\(\delta\) 142
Figure 5.3 Interest evolution probabilities learned on synthetic data, compared against the generated ground-truth values for accuracy 143
Figure 5.4 Learned values of \(\alpha\) on three real-world datasets 144
Figure 5.5 Values of \(\gamma, \delta\) on three real-world datasets 144
Figure 5.6 Test RMSE comparisons between our extended MF model and the standard MF model 145
Figure 5.7 Relative increase in social welfare by GRA over MF-Local on synthetic datasets: Varying network size \((n)\) 147
Figure 5.8 Relative increase in social welfare by GRA over MF-Local on synthetic datasets: Varying $\beta$ . . . . . . . . . . . . . . . . . . 147
Figure 5.9 Relative increase in social welfare by GRA over MF-Local on synthetic datasets: Varying $\gamma_i - \delta_i$ . . . . . . . . . . . . 148
Figure 5.10 Social welfare, where FX(0.1) and FX(0.5) denote Flixster with $\beta = 0.1$ and 0.5 respective; FT denotes FilmTipSet and ML denotes MovieLens. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 149
I wish to express my deepest gratitude to my supervisor Prof. Laks V.S. Lakshmanan. This dissertation would not have been possible without his supervision and continuous guidance and encouragement. Throughout the years I have learned so much from him, and his unparalleled passion for research will always motivate me to pursue excellence.

I sincerely thank my supervisory committee members Kevin Leyton-Brown and Raymond T. Ng for their generous time and valuable feedback on the dissertation. It is certainly worth mentioning that the valuable suggestions from Kevin has led to significant improvements to the dissertation, in particular Chapter 3. I am also very thankful to Sathish Gopalakrishnan, Jiawen Han (University of Illinois at Urbana-Champaign), Ruben H. Zamar for examining my dissertation and providing very helpful comments. Bill Aiello chaired my proposal defense and William Dunford chaired the final defense; I am very thankful for their help and time.

I am grateful to Amit Goyal, Wei Chen, Francesco Bonchi, Stratis Ioannidis, Smriti Bhagat, Martin Ester, and Mitul Tiwari for being great mentors and research collaborators in various stages of my Ph.D. studies. It is my great pleasure collaborating with them.

The acknowledgements would be barely complete without thanking all of my friends for making this journey joyful. Special thanks go to DMM buddies Min Xie, Pei Li, Rui Chen, and Xin Huang; I should definitely mention Zhilong Fang and Chenyi Zhang, my high school classmates reunited in Vancouver. I will not attempt to list any more names to avoid inevitable omissions, but buddies Xin Wang, Kan Wu, Weilu Liu, and Ning Tang must be called out. Together we went from Zhejiang University to Simon Fraser University, we grew in Vancouver, and have had so much fun.

Most importantly, I am greatly indebted to my family and especially my parents, Hanquan Lu and Ping Huo. Their unconditional love and support means everything to me. This dissertation is devoted to them.
To my parents.
Chapter 1

Introduction

Over the past decade, fueled by the rapid growth of online social networking websites (e.g., Facebook and LinkedIn) and social media (e.g., Twitter and Instagram), Computational Social Influence, and in particular, the dynamics of social influence propagation (or diffusion, or cascade)\(^1\) have attracted substantial interests from computer scientists in the fields of data mining, machine learning, algorithmic game theory, theoretical computer science, etc [35,52,67].

Social influence occurs when an individual’s beliefs, opinions, or behaviors change as a result of interactions with other individuals [126]. Social influence theory has been studied extensively by sociologists and psychologists [11, 15, 30, 47, 57, 95, 96], dating back several decades. As our modern world becomes increasingly connected, it is now well understood that the actions of individuals (e.g., adopting a new technology, sharing a video, or retweeting a tweet) may impact their friends, families, colleagues, and as a result, trigger a viral effect over the entire social network. For instance, a large-scale voting mobilization experiment carried out on Facebook during the 2010 United States congressional voting reported statistically significant results indicating messages with social cues (that promote and encourage voting) did impact the behaviors of millions of people [21].

In computational social influence, a fundamental algorithmic problem is the Influence Maximization problem proposed by Kempe et al. [86]. An instance of Influence Maximization is defined by (i) a directed graph \(G = (V,E)\), where each node \(v \in V\) represents an individual (or user) in the

\(^1\)Throughout the dissertation, we use the three terms “propagation”, “diffusion”, and “cascade” interchangeably to refer to the processes in which a behavior or an action virally spreads from one user to another following links in a social network.
social network and each edge \((u,v)\) represents a relationship between user \(u\) and user \(v\); \((ii)\) a positive integer \(k < |V|\) as budget\(^2\), and \((iii)\) a stochastic diffusion model \(M\) which describes the random process of how a certain behavior would spread from one node to its neighbors in the network. The task is to find a subset \(S^* \subset V\) satisfying the cardinality constraint \(|S^*| \leq k\), so that by targeting the nodes in \(S^*\) as early-adopters, the expected number of users who end up adopting the (propagating) behavior is maximized by the end of a propagation process that proceeds according to probabilistic rules specified by model \(M\). Influence Maximization is NP-hard under many stochastic diffusion models [35, 86]. However, a constant-factor greedy approximation algorithm can be obtained as long as model \(M\) satisfies certain desirable properties.

Following this seminal work by Kempe et al., the research on computational social influence began to take off, benefiting numerous applications such as viral marketing (a.k.a. viral advertising or word-of-mouth advertising) [8, 50, 77, 86, 98, 106, 128], community detection [13, 135], recommender systems [19, 70, 108, 144], outbreak detection [100], social media and blogosphere analysis [10, 76, 81, 101, 117], etc. Here, at a high level, we classify existing research into three major categories, inspired by [33]:

**Influence Modeling.** Often times, the forefront challenge is how to model influence cascades over social networks. Only after a satisfactory model is given, one can study worthy optimization problems and design optimization algorithms. There are two common desiderata: First, a good model should reflect human interactions in real world scenarios as much as possible; Second, a good model should be mathematically sound and ideally, allow tractable solutions for natural optimization problems. Notable work in this category has successfully modeled various aspects in real-life social influence, e.g., the spread of both positive and negative opinions [16, 29, 34, 79], competitions of multiple diffusions [18, 23, 24, 31, 106, 123], cooperations or complementarity between different diffusions [107, 117, 118, 149], time-delay in propagation [36, 102], non-progressive dynamics, where a user may switch between “on” (influenced) and “off” (not influenced) states [105], as well as continuous-time propagation [51, 65] as opposed to discrete-time.

\(^2\)Following the convention [23, 69, 71, 86], through the thesis we use the term “budget” to refer to the maximum number of influential users (seeds) to be selected in the influence maximization problem. Note that this is not a monetary value and should not be confused with such. If a monetary budget is meant, it will be explicitly mentioned.
Influence Optimization. With a well-defined diffusion model, we can then optimize a certain desirable objective under this model. Among such, Influence Maximization is the most fundamental problem. Typically, optimization objectives are tailored to models. For instance, in models with negative opinions, a common formulation is that given the set of “negative” seeds, find an optimal set of “positive” seeds to block the spread of negative opinions to the maximum extent possible [29, 79]. As another example, in models that focus on product adoptions, the typical objective is to maximize adoption, profit, or some other related business objectives instead of pure influence spread (as those models explicitly distinguish between the state of merely being influence and the state of actual adoption) [16, 107, 109].

Many of the influence optimization problems, including the seminal Influence Maximization, are NP-hard [86], and moreover for a large family of stochastic diffusion models, it is #P-hard to compute the exact influence spread (in expectation) of any given seed set [37, 39]. Therefore, lots of work focuses on devising efficient and effective approximation algorithms or heuristics that scale to massive social networks [22, 37–40, 71, 72, 84, 87, 100, 136, 137].

Influence Learning. Almost all stochastic diffusion models assumes the complete knowledge of pairwise influence strength (in the form of probabilities or weights) between friends in social networks. This assumption can be unrealistic, as those data is almost never explicitly available. Work has been done to develop principled methodologies for learning influence strength from user interaction data. Representative work can be further divided into two kinds: some assume that network structures are known and thus the main task is to infer influence strength [68, 129, 141], while others learn network structures and influence strength at the same time [42, 64, 122].

This dissertation concentrates more on the first two aspects, namely modeling and optimization. In the experiments where influence probabilities or influence weights are required, we draw on existing work on influence learning to compute such quantities.

1.1 Influence Maximization in Social Networks
Domingos and Richardson [50, 128] first posed influence maximization as an algorithmic problem to the data mining community. They used Markov
Random Field (MRF) techniques to model and study the problem of finding an optimal set of individuals (seeds) on which a company should perform marketing actions, so that the expected increase in profit is maximized. They propose a greedy-style hill-climbing heuristic to solve the problem.

However, the MRF-based formulation in [50, 128] did not gain nearly as much popularity as the discrete optimization formulation in Kempe et al [86], which we re-state here for convenience: Given a directed graph \( G = (V, E) \) in which nodes representing individuals in a social network and edges representing the links or relationships between individuals, as well as a positive integer \( k \), the task of influence maximization (Influence Maximization) is to find a node-set \( S \) of size no more than \( k \), such that by targeting them initially for early activation, the expected number of activated nodes over the entire social network is maximized, under a certain stochastic diffusion model. Note that it is the diffusion model that specifies probabilistic rules under which the dynamics of a propagation/diffusion/cascade process unfold.

The targeted set \( S \) of nodes is often referred to as the seed set. Let \( \sigma : 2^V \rightarrow \mathbb{R}_{\geq 0} \) denote the influence spread function, such that \( \sigma(S) \) is the expected number of active nodes by the end of diffusion, given that \( S \) is activated at the beginning. As a convention, \( \sigma(S) \) is commonly referred to as the influence spread of \( S \).

Two classic and fundamental stochastic diffusion models are the Independent Cascade (IC) model [63] and the Linear Thresholds (LT) model [73]. In both models, there are two possible states for nodes: \{inactive, active\}. Both models are progressive, meaning that once a node becomes active, it will never revert back to inactive. Intuitively, the IC model describes individual interactions between friends in social networks, while the LT model characterizes thresholding behaviours often found in life experiences: e.g., a person may finally decide to purchase an iPhone if an overwhelming majority of her friends have already done so.

**Independent Cascade (IC).** An instance of the IC model has a directed graph \( G = (V, E) \) and an influence probability function \( p : E \rightarrow [0, 1] \). For each edge \((u, v) \in E\), let \( p_{u,v} = \text{def} \ p(u, v) \) denote the probability that \( u \) activates \( v \). Initially, all nodes are inactive. At time step 0, all seeds become active and the propagation starts to proceed in discrete time steps. At time \( t \), every node \( u \) that became active at \( t - 1 \) makes one attempt to activate each of its inactive out-neighbors \( v \in N_{\text{out}}(u) \), succeeding with probability \( p_{u,v} \), which is independent of the diffusion history thus far. The diffusion process terminates when no more nodes can be activated.
Algorithm 1: Greedy — The Greedy Hill-Climbing Algorithm

Data: graph $G = (V, E)$, cardinality constraint $k$
Result: seed set $S$ such that $|S| \leq k$

begin
2 $S \leftarrow \emptyset$
3 for $i \leftarrow 1$ to $k$ do
4 $w \leftarrow \arg\max_{u \in V \setminus S} [f(S \cup \{u\}) - f(S)]$
5 $S \leftarrow S \cup \{w\}$

Linear Thresholds (LT). In this model, each node $v$ has an activation threshold $\theta_v$ uniformly distributed in the interval $[0, 1]$, which represents the minimum weighted fraction of active in-neighbors that are needed to activate $u_i$. Each edge $(u, v) \in E$ is associated with an influence weight $p_{u,v}$, and for all $v \in V$, the sum of incoming weights does not exceed 1:

$$\sum_{u \in N^\text{in}(v)} p_{u,v} \leq 1,$$

where $N^\text{in}(v)$ denotes the set of in-neighbors of $v$.

Influence propagation also proceeds in discrete time steps. At time step 0, a seed set $S$ is activated. At any time step $t \geq 1$, any inactive $v$ becomes active if the total influence weight from its active in-neighbors reaches or exceeds $\theta_v$:

$$\sum_{\text{active } u \in N^\text{in}(v)} p_{u,v} \geq \theta_v.$$

The diffusion process terminates when no more nodes can be activated.

Hardness and Approximation. Influence Maximization is NP-hard under both models [86], and furthermore, it is #P-hard to compute the exact value of $\sigma(S)$ for seed set $S$ [37, 39]. To obtain approximation algorithms, Kempe et al. [86] show that the influence spread function $\sigma(S)$ is monotone and submodular w.r.t. the seed set $S$.

Intuitively, submodularity captures the law of diminishing marginal return. Let $X$ be a finite ground set, and let $f : 2^X \rightarrow \mathbb{R}_{\geq 0}$ denote a set function that maps elements in $X$ to nonnegative real values. The function $f$ is submodular iff for all $S \subseteq T \subseteq X$ and all $x \in X \setminus T$, we have $f(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T)$. Monotonicity simply means the function value is non-decreasing as the set grows: $f(S) \leq f(T)$ whenever $S \subseteq T \subseteq X$. 5
Thanks to these two properties, an approximation algorithm for Influence Maximization can be obtained by applying a seminal result in Nemhauser et al. [119]: the problem of maximizing a monotone submodular function subject to a cardinality constraint can be approximated within a factor of $1 - 1/e$, by the following greedy algorithm (pseudo-code in Algorithm 1): It starts with $S = \emptyset$, and add one element at a time to $S$: the element that has the largest incremental function value w.r.t. $S$. The algorithm terminates after the cardinality of $S$ reaches $k$.

To circumvent the \#P-hardness of computing $\sigma(S)$, a common practice is to estimate the spread using Monte-Carlo (MC) simulations, in which case the approximation ratio of Greedy drops from $1 - 1/e$ to $1 - 1/e - \epsilon$, for any $\epsilon > 0$. The value of $\epsilon$ depends on the number of MC iterations (which is typically 10000). Unless otherwise noted, hereafter whenever we mention Greedy, MC simulations are used jointly.

The aforementioned approximation result is not just applicable to IC and LT. In fact, this elegant framework works for a large class of discrete diffusion models, all of which can be seen as special cases of the General Thresholds model (GT). This model specifies that each node $v \in V$ is associated with a threshold function $f_v: 2^V \rightarrow [0,1]$ that is monotone w.r.t. the set of active in-neighbors of $v$. Mossel and Roch [116] show that whenever the threshold function at every node is monotone and submodular, the influence spread function $\sigma(\cdot)$ is also monotone and submodular (a conjecture posed in [86]). This enables the approximation results of Greedy to apply.

Efficient Influence Maximization Algorithms. Greedy suffers from serious efficiency issues because of the MC simulations. It may take days or weeks to finish mining 50 or 100 seeds on graphs with merely thousands of nodes [35]. By exploiting submodularity, Leskovec et al. [100] devised the cost-effective lazy forward (CELF) technique, which improves the running time of Greedy by up to 700 times. Goyal et al [71] proposed CELF++, which further improve the efficiency of CELF by intelligent look-ahead computations of marginal gains. CELF++ was able to gain up to 61% is running time compared to CELF.

More recently, Tang et al. [137] proposed a randomized algorithm called Two-phase Influence Maximization (TIM), which produces a $(1 - 1/e - \epsilon)$-approximation with at least $1 - |V|^{-\ell}$ probability in $O((k + \ell)(|E| + |V|) \log |V|/\epsilon^2)$ expected running time. TIM is based on the concept of Reverse-Reachable sets (RR-sets) [22], and is applicable to the Triggering Model [86] that generalizes both IC and LT. This algorithm is orders of magnitude faster than Greedy with CLEF++, while still yielding approximation
solutions with high probability. Very recently, Tang et al. also proposed a new improvement [136], which significantly reduced the number of RR-sets generated using martingale analysis. The new IMM algorithm (for Influence Maximization with Martingale) allows dependent RR-sets, in contrast with TIM that requires all RR-sets generated must be independent. IMM has the same approximation guarantee and expected running time complexity, but is empirically shown orders of magnitude faster than the fastest version of TIM [136]. In Section 4.6.1, we present a generalized solution framework that extends TIM to work for an arbitrary stochastic diffusion model that satisfies submodularity and monotonicity.

**Efficient and Effective Heuristics.** In addition to searching for scalable approximation algorithms, a number of heuristics have been proposed in the literature (especially before the invention of TIM and IMM) [37, 39, 72, 84]; they are not only orders of magnitude faster than Greedy but are also comparable in terms of seed set quality as evaluated in terms of the influence spread achieved.

The common intuition is to leverage mathematical properties of the diffusion model and then devise efficient schemes to accurately estimate influence spread to replace the prohibitive MC simulations. For example, the Maximum Influence Arborescence (MIA) algorithm in [37] (for IC) and the Local Directed Acyclic Graph (LDAG) [39] algorithm (for LT) restrict the computation of influence spread within a simpler structure such as trees and DAGs that are local to seeds. The SimPath algorithm (for LT) [72] intelligently enumerates simple paths originating from seeds and leverages the fact that the probability of a path quickly diminishes as the length increases. Thus, inexpensive enumeration can be done to estimate influence spread accurately. For more details, we refer the reader to the original papers or the monograph by Chen et al. [35].

### 1.2 Challenges and Key Contributions

Despite extensive prior work, there are still plenty of unknowns to explore in this field. The goal of this dissertation to advance the research in this field by studying some of the important new problems (described in more details below) and tackle challenges that arise from those problems using novel modeling ideas and algorithm design techniques.

Our overall approaches and objectives are (i) to draw on social sciences to design more expressive, more realistic, yet still tractable influence diffusion models; (ii) to pose well-motivated and challenging optimization
problems and design approximation algorithms and/or effective heuristics to solve them; (iii) to recognize the important role of social influence in recommender systems and devise cascade-aware models and algorithms to improve the quality of recommendations. In what follows, we motivate the problems studied in this dissertation and summarize key contributions.

1.2.1 Viral Marketing

_Viral marketing_ [26] is a cost-effective advertising strategy that leverages word-of-mouth effects to promote ideas or products through a social network. It is often seen as the primary application of computational social influence [35, 37, 50, 86], due to natural connections: In principle, techniques developed for influence maximization is well-suited for running viral marketing campaigns as their objectives align. However, applying influence maximization theory as is to viral marketing may be problematic, as we elaborate below.

**Influence-Driven Profit Maximization.** The ultimate goal of viral marketing is to convince consumers to adopt (purchase) a product, and such decisions undoubtedly involve _monetary_ considerations, which are important missing links between social influence theory and viral marketing in practice. Various challenges arise in this context. For instance, how can we incorporate the monetary nature of the product adoption dynamics into the modeling and optimization of social influence propagations? And what should be the best price of the propagating product to extract as much profit as possible from the viral marketing campaigns?

We address these challenges in Chapter 2. There are two major factors in play: the _price_ of the product being advertised and the _valuation_ that users have toward the product, which is defined as the maximum amount of money that a user is willing to pay. Economics theory suggests that only when the valuation is at least as high as the price, an adoption will be made [85]. We extend the LT model by incorporating both price and buyer valuation into the adoption diffusion process. We study the problem of finding an optimal marketing strategy consisting of a seed set and a price vector, such that the expected total profit at the end of propagation is maximized.

One of the biggest challenges in the above problem is _pricing_, and more specifically, how to strike the balance between the profit earned on the seeds and the profit earned on follow-up adoptions: if a large discount is offered, then the seller has a higher chance to convince the seed to adopt, so that her influence can be leveraged; but on the other hand, the loss of profit from this particular seed is larger due to deeper discount. We come up with a price-
aware greedy algorithm that not only identifies seeds from a social networks, but also dynamically adjusts the optimal personalized discounts for each seed chosen. Our experiments on three real-world datasets demonstrate that the price-aware greedy algorithm achieves the best performance both in the expected profit achieved and in running time, compared to several intuitive baselines.

**Viral Marketing from the Host Perspective.** Another important yet often overlooked question regarding viral marketing is: Can the advertisers simply take the liberty of approaching social network users and promoting their products? The answer is likely no, as such unsolicited actions would be viewed as spamming and may not gain much trust from users [104]. On the other hand, many service providers of social networks (a.k.a. hosts) have established their own platforms and protocols for advertising and monetization. Thus, a viable way to conduct viral advertising is to do it through the host. However, to the best of our knowledge, prior work cannot offer suitable solutions for this task.

To bridge this gap, we propose a novel business model for the hosts to sell viral marketing as a service (Chapter 3). When competing advertisers request to run viral marketing campaigns, the host is responsible for selecting and allocating seeds, for which it charges advertisers. There are two main desiderata: First, the collective influence spread over all advertisers should be maximized, which in turn maximizes the host’s revenue via commission; Second, the allocation of seeds must be fair, in such a way that the expected gain (influence spread achieved) for each advertiser is in proportion to its investment in the campaign. We extend the LT model to characterize competitive influence diffusion. Then, by exploiting the mathematical properties derived from the diffusion model, we tackle both influence maximization and fair allocation problems using greedy and dynamic programming algorithms. We run experiments on three real-world social networks, showing that our proposed allocation algorithms are both efficient and effective.

1.2.2 Competition and Complementarity: A Unified Influence Propagation Model

Products or technologies often face fierce competitions (e.g., iPhone vs. Android phones), and such scenarios can be modeled by competitive influence propagation models [18, 23, 24, 31, 106, 123]. However, consider the following...

---

hypothesised example: Suppose Apple wants to launch a joint advertising campaign on iPhone 6S and the Apple Watch. Unfortunately, none of the competitive diffusion models is suitable for this job. This is because iPhone 6S and Apple Watch are simply not competitors, but rather complementary to each other, yet all competitive models assume that users adopt at most one product. Therefore, the main challenge is how to carefully model influence cascades of complementary goods, especially when the degree of complementarity is asymmetric: adopting an iPhone without Apple Watch is a rational decision, while the opposite action is not, due to the fact that almost all functionalities of Apple Watch relies on a pairing iPhone.

In Chapter 4, we propose a new diffusion model called *Comparative Independent Cascade* (ComIC) which is capable of covering the full spectrum of entity interactions from competition to complementarity. Users' adoption decisions depend not only on edge-level information propagation, but also on a node-level automaton whose behavior is governed by a set of model parameters, enabling our model to capture not only competition, but also complementarity, to any possible degree. We study Influence Maximization in a novel setting with complementary entities, where the objective is to find seeds of product $A$ (e.g., Apple Watch), given the seeds of its complementary product $B$ (e.g., iPhone), such that the influence spread for $A$ is maximized. We devise effective approximation algorithms via non-trivial techniques based on reverse-reachable sets and a novel “sandwich approximation” strategy. The applicability of both techniques extends beyond our model and problems. Our experiments show that the proposed algorithms consistently outperform intuitive baselines in four real-world social networks, often by a significant margin. In addition, we learn model parameters from real user action logs.

### 1.2.3 Applying Social Influence in Recommender Systems

Computational social influence has application values in a variety of data mining tasks. One primary example is the task of personalized recommendations. Typically, a recommender system extracts user feedback from historical transactions, ratings, and reviews, to build a model that profiles users and items, and computes a ranked list of items for each user as recommendations [3, 127]. One of the key challenges is data sparsity: given an arbitrary user, we may not have enough past ratings to accurately profile her and predict her preferences. Previous work [82, 112, 152] leverages social influence to address the sparsity issue, as the intuition is that a user tends to adopt an item rated positively by trustworthy or influential friends.
However, to the best of our knowledge, no prior work explicitly models interest cascades that may be triggered jointly by social influence and recommended items. To elaborate, suppose that Alice was recommended the book *The Unbearable Lightness of Being* by Milan Kundera. She really liked it and gave a five-star rating, which may trigger the RecSys to further suggest this book to users whom Alice has strong influence on. Subsequently, a cascade of recommendations and high ratings may start propagating over the entire social network. Thus, a subtle and challenging question is that how recommendations can be designed to ensure (i) the suggested items remain highly relevant to users and (ii) the interest cascades over time can be incorporated to further improve the quality of recommendation.

To this end, in Chapter 5 we model interest evolution through dynamic interest cascades: we consider a scenario where a user’s interests may be affected by (i) the interests of other users in her social circle and (ii) suggestions she receives from a recommender system. In the latter case, we model user reactions through either attraction or aversion towards past suggestions. We study this interest evolution process, and the utility accrued by recommendations, as a function of the system’s recommendation strategy. We show that, in steady state, the optimal strategy can be computed as the solution of a semi-definite program (SDP). Using datasets of user ratings, we provide evidence for the existence of aversion and attraction in real-life data, and show that our optimal strategy can lead to significantly improved recommendations over systems that ignore aversion and attraction.

### 1.3 Outline

In Chapter 2, we incorporate monetary aspects into social influence propagations and study the problem of influence-driven profit maximization for viral marketing on social networks. In Chapter 3, we study viral marketing from the perspective of social networking service providers, and design a viral-marketing-as-a-service business model and corresponding algorithmic frameworks to solve the problem of fair seed allocation. In Chapter 4, we propose a novel influence diffusion model that characterizes both competitive and complementary relationship between two propagating entities, to any degree possible. In Chapter 5, we use social influence propagation as a key building block to model dynamic user interests in social networks, and devise optimal recommendation algorithms. Finally, we summarize this dissertation and discuss future research directions in Chapter 6.
Chapter 2

Influence-Driven Profit Maximization in Social Networks

2.1 Introduction

Although Influence Maximization has been studied extensively, a majority of the previous work has focused on the classical propagation models, namely IC and LT, which do not fully incorporate important monetary aspects in people’s decision-making process of adopting new products. The importance of such aspects is seen in actual scenarios and recognized in the management science literature.

As real-world examples, until recently, Apple’s iPhone has seemingly created bigger buzz in social media than any other smartphones. However, its worldwide market share in 2011 fell behind Nokia, Samsung, and LG\(^1\). This is partly due to the fact that iPhone is pricier in terms of both hardware (if one buys it contract-free and factory-unlocked) and monthly rate plans. On the contrary, the HP TouchPad was shown little interest by the tablet market when it was initially priced at $499 (16GB). However, it was sold out within a few days after HP dropped the price substantially to $99 (16GB)\(^2\).

In management science, the adoption of a new product is characterized as a two-step process [85]. In the first step, “awareness”, an individual gets

\(^1\)IDC Worldwide Mobile Phone Tracker, July 28, 2011.
\(^2\)http://www.pcworld.com/article/237088/hp_drops_touchpad_price_to_spur_sales.html, last accessed on September 7th, 2015.
exposed to the product and becomes familiar with its features. In the second step, “actual adoption”, a person who is aware of the product will purchase it if her valuation outweighs the price. Product awareness is modeled as being propagated through the word-of-mouth of existing adopters, which is indeed articulated by classical propagation models. However, the actual adoption step is not captured in these classical models and is indeed the gap between these models and that in [85].

In a real marketing scenario, viral or otherwise, products are priced and people have their own valuations for owning them, both of which are critical in making adoption decisions. Precisely, the valuation of a person for a certain product is the maximum money she is willing to pay for it; the valuation for not adopting is defined to be zero [132]. Thus, when a company attempts to maximize its expected profit in a viral marketing campaign, such monetary factors need to be taken into account. However, in Influence Maximization, only influence weights and network structures are considered, and the marketing strategies are restricted to binary decisions: for any node in the network, an Influence Maximization algorithm just decides whether or not it should be seeded.

To address the aforementioned limitations, we propose the problem of profit maximization (ProMax) over social networks, by incorporating both prices and valuations. ProMax is the problem of finding an optimal strategy to maximize the expected total profit earned by the end of an influence diffusion process under a given propagation model. We extend the LT model to propose a new propagation model named the Linear Threshold model with user Valuations (LT-V), which explicitly introduces the states influenced and adopting. Every user will be quoted a price by the company, and an influenced user adopts, i.e., transitions to adopting, only if the price does not exceed her valuation.

As pointed out in [90], consumers typically do not want to reveal their valuations before the price is quoted for reasons of trust. Moreover, for privacy concerns, after a price is quoted, they usually only reveal their decision of adoption (i.e., “yes” or “no”), but do not wish to share information about their true valuations. Thus, following the literature [90, 132], we make the Independent Private Value (IPV) assumption, under which the valuation of each user is drawn independently at random from a certain distribution. Such distributions can be learned by a marketing company from historical sales data. Furthermore, our model assumes users to be price-takers who respond myopically to the prices offered to them, solely based on their privately-held valuations and the price offered.
Since prices and valuations should be considered in the optimization, marketing strategies for ProMax require non-binary decisions: for any node in the network, we (i.e., the system) need to decide whether or not to seed it, and what price should be quoted. Given this factor, the objective function to optimize in ProMax, i.e, the expected total profit, is a function of both the seed set and the vector of prices. As we shall show in Secs. 2.3 and 2.6, since discounting may be necessary for seeds, the profit function is in general non-monotone. We shall also show that the profit function maintains submodularity for any fixed vector of prices, regardless of the specific forms of valuation distributions.

In light of the above, ProMax brings about more challenges compared to Influence Maximization, and calls for more sophisticated algorithms for its solution. As the profit function is in the form of the difference between a monotone submodular set function and a linear function, we first design an “unbudgeted” greedy (U-Greedy) framework for seed set selection. In each iteration, it picks the node with the largest expected marginal profit until the total profit starts to decline. We show that for any fixed price vector, U-Greedy provides quality guarantees slightly lower than a \((1 - 1/e - \epsilon)\)-approximation.

To obtain complete profit maximization algorithms, we propose to integrate U-Greedy with three pricing strategies, which leads to three algorithms All-OMP (Optimal Myopic Prices), FFS (Free-For-Seeds), and PAGE (Price-Aware GrEedy). The first two are baselines and choose prices in ad hoc ways, while PAGE dynamically determines the optimal price to be offered to each candidate seed in each round of U-Greedy. Our experimental results on three real-world network datasets illustrate that PAGE outperforms All-OMP and FFS in terms of expected profit achieved and running time, and is more robust against various network structures and valuation distributions.

2.2 Related Work

Bhagat et al. [17] addressed the difference between product adoption and influence in their Linear Thresholds with Colors (LT-C) model. They focus on the possible emergence of negative opinions and the fact that even non-adopting users may spread opinions or information to friends. In the LT-C model, the extent to which a node is influenced by its neighbors depends on two factors: influence weights and the opinions of the neighbors. LT-C also features a “tattle” state for nodes: if an influenced node does not adopt, it may still propagate positive (promote state) or negative influence (inhibit state).
state) to neighbors. The optimization objective is to maximize the spread of positive influence and the algorithmic framework in [86] is applicable. Our work departs from [17] by modeling the monetary aspects (price and valuation) in product adoption and posing a natural, but different problem whose objective is to maximize the expected profit in a viral marketing campaign. Due to the specific properties of both the LT-V model and the formulation of profit maximization, the greedy approximation algorithm [86] no longer applies. In principle, the LT-V and LT-C models capture different aspects of influence diffusion and can be merged into a combined model – We defer detailed discussions to Section 2.8. In this chapter, our focus is to understand the direct impact of price and valuation in solving influence-driven profit maximization, a problem, as we shall see, is already challenging by just extending the canonical LT model.

Considerable work has been done on pricing problems in social networks. Hartline et al. [77] studied optimal marketing for digital goods (zero cost) in social networks and propose the Influence-and-Exploit (IE) framework. In the Influence step, the seller offers free samples to a set of seeds; In the Exploit step, the seller determines a random sequence to visit each of the remaining nodes and offer them a price that would maximize the seller’s expected revenue based on the probability distribution of buyer valuation. The valuation of a user depends on the set of her active neighbors. Notice that this randomized approach effectively bypassed the network structure and furthermore, their approach did not consider the viral diffusion of adoption behaviors, which is a clear distinction from our work.

Arthur et al. [6] adopted the IE framework to study revenue maximization for viral marketing. Given a seed set $S$, their algorithm first computes an approximate max-leaf spanning tree $T$ of the input social network graph $G$ rooted at $S$. All seeds and all internal nodes of $T$ will be offered free samples. Each leaf node will be charged a constant price with a certain probability, or it gets a copy for free as well. There are several key differences between this work and ours: (i) seeds are given as input in [6], whereas in our case, the choice of the seed set is dictated by profit maximization and hence made by algorithms; (ii) our profit maximization algorithm is capable of dynamically finding the best personalized discounts for seeds, however in [6] all seeds get free samples; (iii) unlike ours, their work lacks influence modeling.
2.3 Linear Threshold Model with User Valuations

2.3.1 Model and Problem Definition

In the LT-V model, the social network is modeled as a directed graph \( G = (V,E) \), in which each node \( u_i \in V \) is associated with a valuation \( v_i \in [0,1] \). Recall that in §2.1, we made the IPV assumption under which valuations are drawn independently at random from some continuous probability distribution assumed known to the marketing company. Let \( F_i(x) = \Pr[v_i \leq x] \) be the distribution function of \( v_i \), and \( f_i(x) = \frac{d}{dx} F_i(x) \) be the corresponding density function. The domain of both functions is \([0,1]\) as we assume both prices and valuations are in \([0,1]\). As in the classical LT model, each node \( u_i \) has an influence threshold \( \theta_i \) chosen uniformly at random from \([0,1]\). Each edge \((u_i,u_j) \in E\) has an influence weight \( w_{i,j} \in [0,1]\), such that for each node \( u_j \), \( \sum_{u_i \in N^{in}(u_j)} w_{i,j} \leq 1 \). If \((u_i,u_j) \notin E\), define \( w_{i,j} = 0 \). Following [50, 128], we assume that there is a constant acquisition cost \( c_a \in [0,1) \) for marketing to each seed (e.g., rebates, or costs of mailing ads and coupons).

**Diffusion Dynamics.** Figure 2.1 presents a state diagram for the LT-V model. At any time step, nodes are in one of the three states: inactive, influenced, and adopting. A diffusion under the LT-V model proceeds in discrete time steps. Initially, all nodes are inactive. At time 0, a seed set \( S \) is targeted and becomes influenced. Next, every user \( u_i \) in the network is offered a price \( p_i \) by the system. Let \( p = (p_1, \ldots, p_{|V|}) \in [0,1]^{|V|} \) denote a vector of quoted prices, which remains constant throughout the diffusion. For any \( u_i \in S \), it gets one chance to adopt (enters adopting state) at step 0 if \( p_i \leq v_i \); otherwise it stays influenced.

At any time step \( t \geq 1 \), an inactive node \( u_j \) becomes influenced if the total influence from its adopting in-neighbors reaches its threshold, i.e.,

\[
\sum_{u_i \in N^{in}(u_j) \land u_i \text{ adopting}} w_{i,j} \geq \theta_j.
\]
Then, $u_j$ will transition to adopting at time step $t$ if $p_j \leq v_j$, and will stay influenced otherwise. The model is progressive, meaning that once a node “advances” from one state to the next, e.g., from inactive to influenced, or from influenced to adopting, it will never revert to the previous states. The diffusion ends if no more nodes can change states. Following [85], we assume that only adopting nodes propagate influence, as adopters can release experience-related product features (e.g., durability, usability), making their recommendations more effective in removing doubts of inactive users.

Formally, we define $\pi: 2^V \times [0, 1]^{|V|} \rightarrow \mathbb{R}$ to be the profit function such that $\pi(S, p)$ is the expected (total) profit earned by the end of a diffusion process under the LT-V model, with $S$ as the seed set and $p$ as the vector of prices. The problem studied in the paper is as follows.

**Problem 1** (Profit Maximization (ProMax)). Given an instance of the LT-V model consisting of a graph $G = (V, E)$ with edge weights, find the optimal pair of a seed set $S$ and a price vector $p$ that maximizes the expected profit $\pi(S, p)$.

It is worth emphasizing that users in this problem can be naturally seen as utility-maximizing agents, where the utility of each adopting user $i$ is defined to be the difference between $i$’s valuation toward the product and the price she is going to pay: $v_i - p_i$. The utility of not adopting is simply 0. In our setup, every node is offered a price once and only once, and we assume that there is no reconsideration after a node declines to adopt the product. Hence, it is clear that no user has any incentive to deviate from her truthful valuation. In other words, for all users, the dominant strategy is to compare their truthful valuations to the price.

### 2.4 Special Case with Fixed Valuations

To better understand the properties of the LT-V model and the hardness of ProMax, we first study a special case of the problem. We assume the valuation distribution degenerate to an identical single-point, i.e., for all $u_i \in V$, $v_i = p$ with probability 1, where $p \in (0, 1]$ is a constant. As mentioned in §2.1, this is usually not the case, and the degeneration assumption here is of theoretical interest only.

For simplicity, we also assume that for every $u_i \in S$, the quoted price $p_i = 0$. Strictly speaking, for the sake of maximizing expected total profit, the seller should also charge price $p$ to all seeds, since it is assumed that all users have valuation $p$. In this case, the expected profit function becomes $\hat{\pi}(S) = p \cdot \sigma_L(S) - c_a|S|$, and the non-monotonicity still holds in general.
To see this, consider a social network graph \( G = (V, E) \) where \(|V| = 100\), and also let \( p = 0.5\), \( c_a = 0.1\). Suppose that there is a single node \( v \) that has expected influence spread of 90, which alone would yield a profit of 44.9, while if \( S = V \), the expected profit would be 40, which is less than the case when \( S = \{v\} \). This example illustrates that the profit function is still non-monotone when all users have the same valuation and are offered the same price.

Since valuation is the maximum money one is willing to pay for the product, in this case, the optimal pricing strategy is to set \( p_j = p, \forall u_j \in V \setminus S \). The situation amounts to restricting the marketing strategy to a binary one: free sample (\( p_i = 0 \)) for seeds and full price for non-seeds (\( p_j = p \)). Given this pricing strategy, once a node is influenced, it transitions to adopting with probability 1. Thus, ProMax boils down to a problem to determine a seed set \( S \), and the profit function \( \pi(S, p) \) reduces to a set function \( \hat{\pi}(S) \), since \( p \) is uniquely determined given \( S \):

\[
\hat{\pi}(S) = p \cdot (\sigma_L(S) - |S|) - c_a |S| \\
= p \cdot \sigma_L(S) - (p + c_a) |S|, 
\]

where \( \sigma_L(S) \) is the expected number of adopting nodes under the LT-V model by seeding \( S \).

In general, the degenerated profit function \( \hat{\pi} \) is non-monotone. To see this, let \( u \) be any seed that provides a positive profit. Now, clearly \( \hat{\pi}(\emptyset) = 0 < \hat{\pi}(\{u\}) \) but \( \hat{\pi}(V) \leq 0 < \hat{\pi}(\{u\}) \), as giving free samples to the whole network will result in a loss of \( c_a |V|\) on account of seeding expenses. Since \( \hat{\pi} \) is non-monotone, unlike Influence Maximization, it is natural to not use a budget \( k \) for the number of seeds, but instead ask for a seed set of any size that results in the maximum expected profit. In other words, the number of seeds to be chosen, \( k \), is not preset, but is rather determined by a solution. This restricted case of ProMax is to find \( S = \arg \max_{T \subseteq V} \hat{\pi}(T) \), which we show is NP-hard.

**Theorem 1.** The Restricted ProMax problem (RPM) is NP-hard for the LT-V model.

*Proof.* Given an instance of the NP-hard Minimum Vertex Cover (MVC) problem, we can construct an instance of the ProMax problem, such that an optimal solution to the ProMax problem gives an optimal solution to the MVC problem. Consider an instance of MVC defined by an undirected \( n \)-node graph \( G = (V, E) \); we want to find a set \( S \) such that \(|S| = k\) and \( k \) is the smallest number such that \( G \) has a vertex cover (VC) of size \( k \).
Influence Maximization, the function is non-monotone and we want to find linear combination of two submodular functions. However, unlike for \( \mu \), which leads to the submodularity of \( \nu \) and vice versa; this completes the proof.

\[ x \] which is a contradiction. Hence, \( x \) will consider MVC later). This implies that there exists at least one edge \( e \) such that the node-set of \( e \), denoted by \( u_i \) and \( u_j \), are not in \( S \). From the way in which influence weights and thresholds are set up, we know there are exactly \( j \) nodes in \( V \setminus T \) that are not activated. Let \( J \) be the set containing those \( j \) nodes, and consider the set \( T' = T \cup J \), for which we have \( \hat{\pi}(T') = n \). From the proof of Theorem 2.7 of [86], \( T' \) is a VC of \( G \). But since \( |T'| = |T| + j < |S| \), \( T' \) is a VC with a strictly smaller size than \( S \), which gives a contradiction since \( S \) is a MVC.

\( \iff \). Suppose that \( S = \arg \max_{T \subseteq V} \hat{\pi}(T) \), but \( S \) is not a VC of \( G \) (we will consider MVC later). This implies that there exists at least one edge \( e \in E \) such that both endpoints of \( e \), denoted by \( u_i \) and \( u_j \), are not in \( S \). From the way in which influence weights and thresholds are set up in \( G' \), we know both \( u_i \) and \( u_j \) are not activated. Thus, if we add either one of them, say \( u_i \), into \( S \), \( \sigma_L(S \cup \{u_i\}) \) is at least \( \sigma_L(S) + 2 \), and thus \( \hat{\pi}(S \cup \{u_i\}) - \hat{\pi}(S) > 1 \), which contradicts with the fact that \( S \) optimizes \( \hat{\pi} \). Hence, \( S \) must be a VC of \( G \). Now suppose that in addition \( S \) is not a MVC. Then, there must exist some \( x \in S \) such that the node-set \( S \setminus \{x\} \) is still a VC of \( G \); this means that \( \sigma_L(S \setminus \{x\}) = n \), too. Thus, \( \hat{\pi}(S \setminus \{x\}) = n - |S| + 1 > \hat{\pi}(S) = n - |S| \), which is a contradiction. Hence, \( S \) is indeed a MVC of \( G \).

Now we have shown that an optimal solution to the restricted ProMax problem is an optimal solution to the Minimum Vertex Cover problem, and vice versa; this completes the proof.

Observe that both components of \( \hat{\pi} \), \( \sigma_L(S) \) and \( -|S| \), are submodular, which leads to the submodularity of \( \hat{\pi} \) as it is a non-negative linear combination of two submodular functions. However, unlike for Influence Maximization, the function is non-monotone and we want to find
Algorithm 2: U-Greedy

Data: $G = (V, E)$
Result: seed set $S$

1 begin
2 $S$ $\leftarrow$ $\emptyset$
3 while true do
4     $u$ $\leftarrow$ $\arg \max_{u_i \in V \setminus S} [\hat{\pi}(S \cup \{u_i\}) - \hat{\pi}(S)]$
5     if $\hat{\pi}(S \cup \{u\}) - \hat{\pi}(S) > 0$ then
6         $S$ $\leftarrow$ $S$ $\cup$ $\{u\}$
7     else
8         break
9 end

a set $S$ of any size that maximizes $\hat{\pi}(S)$, so the standard Greedy is not applicable here.

Feige et al. [55] give a randomized local search (2/5-approximation) for maximizing non-monotone submodular functions. This is applicable to $\hat{\pi}$, but have time complexity $O(|V|^3|E|/\epsilon)$, where $(1 + \epsilon/|V|^2)$ is the per-step improvement factor in the search. In contrast, the function $\hat{\pi}$ is the difference between a monotone submodular function and a linear function, we propose a greedy approach (Algorithm 2 U-Greedy) with time complexity $O(|V|^2|E|)$ and a better approximation ratio, which is slightly lower than $1 - 1/\epsilon - \epsilon$. U-Greedy grows the seed set $S$ in a greedy fashion similar to Greedy, and terminates when no node can provide positive marginal gain w.r.t. $S$.

Theorem 2. Given an instance of the restricted ProMAX problem under the LT-V model consisting of a graph $G = (V, E)$ with edge weights and objective function $\hat{\pi}$, let $S_g \subseteq V$ be the seed set returned by Algorithm 2, and $S^* \subseteq V$ be the optimal solution. Then,

$$\hat{\pi}(S_g) \geq (1 - 1/\epsilon - \epsilon) \cdot \hat{\pi}(S^*) - \Theta(\max\{|S_g|, |S^*|\}). \quad (2.2)$$

Proof. Case (i). If $|S^*| \leq |S_g|$, then since $\sigma_L$ is monotone and submodular, we have

$$\sigma_L(S_g) \geq (1 - 1/\epsilon - \epsilon) \cdot \sigma_L(S^*).$$
Thus, by the definition of $\hat{\pi}$, we have

$$\hat{\pi}(S_g) = p \cdot \sigma_L(S_g) - (p + c_o) |S_g|$$

$$\geq p(1 - 1/e - \epsilon) \cdot \sigma_L(S^*) - (p + c_o) |S_g|$$

$$= (1 - 1/e - \epsilon) \cdot \hat{\pi}(S^*) - (p + c_o) |S_g| + (1 - 1/e - \epsilon)(p + c_o) |S^*|$$

$$= (1 - 1/e - \epsilon) \cdot \hat{\pi}(S^*) - \Theta(S_g).$$

Case (ii). If $|S^*| > |S_g|$, consider a set $S'_g$ obtained by running U-Greedy until $|S'_g| = |S^*|$. Clearly, from case (i), we have

$$\hat{\pi}(S'_g) \geq (1 - 1/e - \epsilon) \cdot \hat{\pi}(S^*) - \Theta(|S'_g|).$$

Due to the fact that $|S^*| = |S'_g| > |S_g|$, and $S_g$ is obtained by running U-Greedy until no node can provide positive marginal profit, we have

$$\hat{\pi}(S_g) \geq \hat{\pi}(S'_g) \geq (1 - 1/e - \epsilon) \cdot \hat{\pi}(S^*) - \Theta(|S^*|).$$

Combining the above two cases gives Equation (2.2).

Theorem 2 indicates that the gap between the U-Greedy solution and a $(1 - 1/e - \epsilon)$-approximation grows linearly w.r.t. the cardinality of the seed set. Since this cardinality is typically much smaller than the expected spread, U-Greedy can provide quality guarantees for restricted ProMax with objective function $\hat{\pi}$.

### 2.5 General Properties of the LT-V Model

Theorem 1 shows that in a restricted setting where exact valuations are known and the optimal pricing strategy is trivial, ProMax is still NP-hard. Now we consider the general ProMax described in §2.3.1, and show that for any fixed price vector, the general profit function maintains submodularity (w.r.t. the seed set) regardless of the specific forms of the valuation distributions.

Given a seed set $S$ and a price vector $p$, let $ap(u_i|S, p)$ denote $u_i$’s adoption probability, defined as the probability that $u_i$ adopts the product by the end of the diffusion started with seed set $S$ and price vector $p$. Similarly, let $ip(u_i|S, p_{-i})$ denote $u_i$’s probability of getting influenced under the same initial conditions, where $p_{-i} \in [0, 1]|V|^{-1}$ is the vector of all prices excluding $p_i$. Also, let $\pi^{(i)}(S, p)$ be the expected profit earned from $u_i$. 


By model definition, for any $u_i \in V \setminus S$, we have
\[ ap(u_i|S, p) = ip(u_i|S, p_{-i}) \cdot (1 - F_i(p_i)) \]
and
\[ \pi^{(i)}(S, p) = p_i \cdot ap(u_i|S, p). \]

If $u_i \in S$, then we have
\[ ip(u_i|S, p_{-i}) = 1 \]
and
\[ \pi^{(i)}(S, p) = p_i \cdot (1 - F_i(p_i)) - c_a. \]

By linearity of expectations, we have
\[ \pi(S, p) = \sum_{u_i \in V} \pi^{(i)}(S, p). \]

Hence, to analyze the profit function, we just need to focus on the adoption probability, in which the factor $(1 - F_i(p_i))$ does not depend on $S$, but $ip(u_i|S, p_{-i})$ requires careful analysis, which we shall present in the proof of Theorem 3.

Let $v = (v_1, \ldots, v_{|V|}) \in [0, 1]^{|V|}$ be a vector of user valuations, corresponding to random samples drawn from the various user valuation distributions. We now have:

**Theorem 3 (Submodularity).** *Given an instance of the LT-$V$ model, for any fixed vector $p \in [0, 1]^{|V|}$ of prices, the profit function $\pi(S, p)$ is submodular w.r.t. $S$, for an arbitrary vector $v$ of valuation samples.*

The proof of submodularity of the influence spread function $\sigma$ in the classical LT model [86] relies on establishing an equivalence between the LT model and reachability in a family of random graphs generated as follows: for each node $u_i \in V$, select at most one of its incoming edges at random, such that $(u_j, u_i)$ is selected with probability $w_{j,i}$, and no edge is selected with probability $1 - \sum_{u_j \in N^+(u_i)} w_{j,i}$. We will use a similar approach in the proof of Theorem 3.
Proof of Theorem 3. By linearity of expectation as well as the above analysis on adoption probabilities,

\[
p(S, p) = \sum_{u_i \in V} p(i)(S, p) = \sum_{u_i \in S} [p_i(1 - F_i(p_i)) - c_u] + \sum_{u_i \in S} p_i(1 - F_i(p_i)) \cdot ip(u_i|S, p_\neg i).
\]

Since the first sum is linear in \(S\), it suffices to show that \(ip(u_i|S, p_\neg i)\) is submodular in \(S\), whenever \(u_i \notin S\).

To encode random events of the LT-V model using the possible world semantics, we do the following. First, we run a node coloring process on \(G\): for each node \(u_i\), if \(p_i \leq v_i\), color it black; otherwise color it white. Meanwhile, we run a live-edge selection process following the aforementioned protocol [86]. Note that the two processes are orthogonal and independent of each other. Combining the results of both leads to a colored live-edge graph, which we call a possible world \(X\). Let \(X\) be the probability space in which each sample point specifies one such possible world \(X\).

Next, we define the notion of “black-reachability”. In any possible world \(X\), a node \(u_i\) is black-reachable from a node set \(S\) if and only if there exists a black node \(s \in S\) such that \(u_i\) is reachable from \(s\) via a path consisting entirely of black nodes, except possibly for \(u_i\) (even if \(u_i\) is white, it is still considered black-reachable since here we are interested in the probability of being influenced, not adopting). From the same argument in the proof of Claim 2.6 of [86], on any black-white colored graph, the following two distributions over the sets of nodes are the same: (i) the distribution over sets of influenced nodes obtained by running the LT-V process to completion starting from \(S\); (ii) the distribution over sets of nodes that are black-reachable from \(S\), under the live-edge selection protocol.

Let \(I_X(u_i|S)\) be the indicator set function such that it is 1 if \(u_i\) is black-reachable from \(S\), and 0 otherwise. Consider two sets \(S\) and \(T\) with \(S \subseteq T \subseteq V\), and a node \(x \in V \setminus T\). Consider some \(u_i\) that is black-reachable from \(T \cup \{x\}\) but not from \(T\). This implies (i) \(u_i\) is not black-reachable from \(S\) either (otherwise, \(u_i\) would also be black-reachable from \(T\), which is a contradiction); (ii) the source of the path that “black-reaches” \(u_i\) must be \(x\). Hence, \(u_i\) is black-reachable from \(S \cup \{x\}\), but not from \(S\), which implies \(I_X(u_i|S \cup \{x\}) = I_X(u_i|T \cup \{x\}) - I_X(u_i|T)\). Thus, \(I_X(u_i|S)\) is submodular. Since \(ip(u_i|S, p_\neg i) = \sum_{X \in \mathcal{X}} \Pr[X] \cdot I_X(u_i|S)\) is a
nonnegative linear combination of submodular functions, it is also submodular w.r.t. $S$. This completes the proof.

We also remark that in general graphs, given any $S$ and $p$, it is #P-hard to compute the exact value of $\pi(S, p)$ for the LT-V model, just as in the case of computing the exact expected spread of influence for the LT model. This can be shown similarly to the proof of Theorem 1 in [39].

2.6 Algorithm Design and Analysis

For ProMax, since the expected profit is a function of both the seed set and the vector of prices, a ProMax algorithm should determine both the seed set and an assignment of prices to nodes to optimize the expected profit. Accordingly, it has two components: (i) a seed selection procedure that determines $S$, and (ii) a pricing strategy that determines $p$. Due to acquisition costs and the possible need for seed-discounting (details later), $\pi(S, p)$ is still non-monotone in $S$ and is in the form of the difference between a monotone submodular function and a linear function. Hence, inspired by the restricted ProMax studied in 2.4, we propose to use U-Greedy for seed set selection.

We then propose three pricing strategies and integrate them with U-Greedy to obtain three ProMax algorithms. The first two, All-OMP and FFS, are baselines with simple strategies that set prices of seeds without considering the network structure and influence spread, while the third one, PAGE, computes optimal discounts for candidate seeds based on their “profit potential”. Intuitively, it “rewards” seeds with higher influence spread by giving them a deeper discount to boost their adoption probabilities, and in turn the adoption probabilities of nodes that may be influenced directly or indirectly by such seeds.

Notice that taking valuations into account when modeling the diffusion process of product adoption makes a difference for a marketing company. A pricing strategy that does not consider valuations is limited: either it charges everyone full price (or at best gives full discount to the seeds), or it uses an ad-hoc discount policy which is necessarily suboptimal. By contrast, PAGE makes full use of valuation information to determine the best discounts.

2.6.1 Two Baseline Algorithms

Recall that in our model, users in the social network are price-takers who myopically respond to the price offered to them. Thus, given a distribution
Algorithm 3: All-OMP — Optimal Myopic Price for All Users

Data: graph $G = (V, E)$, CDFs $F_i(\cdot)$ for all $u_i \in V$

Result: seed set $S$, price vector $p^m$

begin

1. $S \leftarrow \emptyset$
2. $p^m \leftarrow 0$

3. foreach $u_i \in V$ do
4. 
5.   $p^m[i] \leftarrow p^m_i = \arg\max_{p \in [0, 1]} p \cdot (1 - F_i(p))$
6. while true do
7.   $u \leftarrow \arg\max_{u_i \in V \setminus S} [\pi(S \cup \{u_i\}, p^m) - \pi(S, p^m)]$
8. if $\pi(S \cup \{u\}, p^m) - \pi(S, p^m) > 0$ then
9.   $S \leftarrow S \cup \{u\}$
10. else
11.   break

end

function $F_i$ of valuation $v_i$, the optimal myopic price (OMP) [77] can be calculated by:

$$p^m_i = \arg\max_{p \in [0, 1]} p \cdot (1 - F_i(p)).$$  \hspace{1cm} (2.3)

2.6.1.1 OMP For All Users

Offering OMP to a single influenced node ensures that the expected profit earned solely from that node is the maximum. This gives our first PROMAX algorithm, All-OMP, which offers OMP to all nodes regardless of whether a node is a seed or how influential it is. First, for each $u_i \in V$, it calculates $p^m_i$ using Equation (2.3), and populates all OMPs to form the price vector $p^m = (p^m_1, \ldots, p^m_{|V|})$. Then, treating $p^m$ fixed, it essentially runs U-Greedy (Algorithm 2) to select the seeds. When the algorithm cannot find a node of which the marginal profit is positive, it stops.

Notice that Equation (2.3) overlooks the network structure and ignores the profit potential of seeds. This may lead to the sub-optimality of All-OMP in general. Figure 2.2 illustrates this with an example.

Suppose that all valuations are distributed uniformly in $[0, 1]$ and the acquisition cost $c_a = 0.001$. Hence, $p^m = (1/2, \ldots, 1/2)$. Consider seeding node 1: it adopts w.p. 0.5, giving a profit of $0.5 + 5 \times 0.5^3 - 0.001 = 1.124$; it does not adopt w.p. 0.5, resulting in a profit of $-0.001$. Thus, the expected profit

25
\( \pi(\{1\}, p^m) = 0.5615 \). However, when \( p_1 = 3/16 \), \( \pi(\{1\}, p^m_{-1} \oplus (3/16)) = 0.661 \). Here we have used \( p_{-i} \oplus x \) to denote a vector sharing all values with \( p \) except that the \( i \)-th coordinate is replaced by \( x \), e.g., if \( p = (0.2, 0.3, 0.4) \), then \( p_{-1} \oplus 0.5 = (0.5, 0.3, 0.4) \). This shows that for high-influence networks and low acquisition cost, the profit earned by running \textit{All-OMP} can be improved by seed-discounting, i.e., lowering prices for seeds so as to boost their adoption probabilities and thus better leverage their influence over the network. The intuition is that the profit loss over seeds (stemming from the discount) can potentially be compensated and even surpassed by the profit gain over non-seeds: more seeds may adopt as a result of the discount and the probabilities of non-seeds getting influenced will go up as more seeds adopt.

### 2.6.1.2 Free Samples for Seeds

Generally speaking, there exists a trade-off between the immediate (myopic) profit earned from seeds and the potentially more profit earned from non-seeds. Favoring the latter, we propose our second algorithm \textit{FFS} (Free-For-Seeds) which gives a full discount to seeds and charges non-seeds the OMP. \textit{FFS} first calculates \( p^m = (p^m_1, \ldots, p^m_{|V|}) \) using Equation (2.3). Then it runs \textit{U-Greedy}: in each iteration, it adds to \( S \) the node which provides the largest marginal profit when a full discount (i.e., price 0) is given. For all seeds added, their prices remain 0; the algorithm ends when no node can provide positive marginal profit.

Since \textit{FFS} has a completely opposite attitude towards seed-discounting compared to \textit{All-OMP}, intuitively, it should be suitable for high-influence networks and low acquisition costs, but it may be overly aggressive for low-influence networks and high acquisition costs. For example, in Fig 2.2, the \textit{FFS} profit by seeding node 1 is 0.625, better than the \textit{All-OMP} profit 0.5615.
Algorithm 4: FFS — Free-For-Seeds

Data: graph $G = (V, E)$, CDFs $F_i(\cdot)$ for all $u_i \in V$

Result: seed set $S$, price vector $p_f$

1 begin
2 $S \leftarrow \emptyset$
3 $p_f \leftarrow 0$
4 foreach $u_i \in V$ do
5 \hspace{1em} $p_f[i] \leftarrow p_i^m = \arg \max_{p \in [0,1]} p \cdot (1 - F_i(p))$
6 while true do
7 \hspace{1em} $u \leftarrow \arg \max_{u_i \in V \setminus S} \pi(S \cup \{u_i\}, p_f^j_u \oplus 0) - \pi(S, p_f^j)$
8 \hspace{2em} if $\pi(S \cup \{u\}, p_f^j_u \oplus 0) - \pi(S, p_f^j) > 0$ then
9 \hspace{3em} $S \leftarrow S \cup \{u\}; p_f \leftarrow p_f^j_u \oplus 0$
10 \hspace{2em} else
11 \hspace{3em} break

But if all influence weights are 0.01 instead of 0.5, and $c_a = 0.01$, All-OMP gives a profit of 0.246, while FFS gives only 0.0025.

2.6.2 The Price-Aware Greedy Algorithm

Both All-OMP and FFS are easy for marketing companies to operate, but they are not balanced and are not robust against different input instances as illustrated above by examples. To achieve more balance, we propose the PAGE (for Price-Aware GrEedy) algorithm (Algorithm 5). PAGE also employs U-Greedy to select seeds. It initializes all seed prices to their OMP values (Step 3). In each round, it calculates the best price for each candidate seed such that its marginal profit (MP) w.r.t. the chosen $S$ and $p$ is maximized (Step 7); then it picks the node with the largest maximum MP (Step 8). It stops when it cannot find a seed with a positive MP (Step 11). For all non-seed nodes, PAGE still charges OMP. We next explain the details of determining the best price for a candidate seed.

Given a seed set $S$, consider an arbitrary candidate seed $u_i \in V \setminus S$, with its price $p_i$ to be determined. The marginal profit (MP) that $u_i$ provides w.r.t. $S$ with $p_i$ is

$$MP(u_i) = \pi(S \cup \{u_i\}, p_{-i} \oplus p_i) - \pi(S, p_{-i} \oplus p_i^m),$$
Algorithm 5: PAGE — Price Aware Greedy Algorithm

Data: graph $G = (V, E)$, CDFs $F_i(\cdot)$ for all $u_i \in V$

Result: seed set $S$, price vector $p$

1 begin
2 $S \leftarrow \emptyset$
3 $p \leftarrow 0$
4 foreach $u_i \in V$ do
5 | $p[i] \leftarrow p_i^{\text{na}} = \arg \max_{p \in [0, 1]} p \cdot (1 - F_i(p))$
6 while true do
7 | foreach $u_i \in V \setminus S$ do
8 | | Estimate the value of $Y_0$ and $Y_1$ by MC simulations
9 | | $p_i^* \leftarrow \arg \max_{p_i \in [0, 1]} g_i(p_i)$; normalize if needed
10 | | $u \leftarrow \arg \max_{u_i \in V \setminus S} g_i(p_i^*)$
11 | if $\pi(S \cup \{u_i\}, p_i \oplus p_i^{\text{na}}) - \pi(S, p_i \oplus p_i^{\text{na}}) > 0$ then
12 | S $\leftarrow S \cup \{u_i\}$; $p \leftarrow p_i \oplus p_i^{\text{na}}$
13 | else
14 | | break

where $p_i$ is fixed. The key task in PAGE is to find $p_i$ such that $MP(u_i)$ is maximized. Since $\pi(S, p_i \oplus p_i^{\text{na}})$ does not involve $u_i$ and $p_i$, it suffices to find $p_i$ that maximizes this quantity:

$$\pi(S \cup \{u_i\}, p_i \oplus p_i^{\text{na}})$$

Seeding $u_i$ at a certain price $p_i$ results in two possible worlds: world $X_1^{(i)}$ with $\Pr[X_1^{(i)}] = 1 - F_i(p_i)$, in which $u_i$ adopts, and world $X_0^{(i)}$ with $\Pr[X_0^{(i)}] = F_i(p_i)$, in which $u_i$ does not adopt. In world $X_1^{(i)}$, the profit earned from $u_i$ is $p_i - c_a$ and let the expected profit earned from other nodes be $Y_1$. Similarly, in world $X_0^{(i)}$, the profit from $u_i$ is $-c_a$ and let the expected profit from other nodes be $Y_0$. Notice that $Y_1$ depends on the influence of $u_i$ but $Y_0$ does not. Putting it all together, the quantity of $\pi(S \cup \{u_i\}, p_i \oplus p_i^{\text{na}})$ can be expressed as a function of $p_i$ as follows:

$$g_i(p_i) = (1 - F_i(p_i)) \cdot (p_i + Y_1) + F_i(p_i) \cdot Y_0 - c_a. \quad (2.4)$$

Similarly to the expected spread of influence in Influence Maximization, the exact values of $Y_1$ and $Y_0$ cannot be computed in PTIME (due to $\#P$-
hardness [39]), but sufficiently accurate estimations can be obtained by Monte Carlo (MC) simulations.

Finding \( p^*_i = \arg \max_{p_i \in [0,1]} g_i(p_i) \) now depends on the specific form of the distribution function \( F_i \). We consider two kinds of distributions: the normal distribution, for which \( v_i \sim \mathcal{N}(\mu, \sigma^2) \), \( \forall u_i \in V \), and the uniform distribution, for which \( v_i \sim \mathcal{U}(0,1) \), \( \forall u_i \in V \). The choice of the normal distribution is supported by evidence from real-world data from Epinions.com (see §2.7), and also work in [83]. When sales data are not available, it is common to consider the uniform distribution with support \([0,1]\) to account for our complete lack of knowledge [20,132].

**Normal Distribution.** For normal distribution, assume that \( v_i \sim \mathcal{N}(\mu, \sigma^2) \) for some \( \mu \) and \( \sigma \), then \( \forall p_i \in [0,1] \),

\[
F_i(p_i) = \frac{1}{2} \left[ 1 + \text{erf} \left( \frac{p_i - \mu}{\sqrt{2\sigma}} \right) \right],
\]

where erf(\( x \)) is the error function, defined as

\[
\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} \, dt.
\]

Plugging \( F_i(\cdot) \) back into Equation (2.4), one cannot obtain an analytical solution for \( p^*_i \), as erf(\( x \)) has no closed-form expression. Thus, we turn to numerical methods to approximately find \( p^*_i \). Specifically, we use the golden section search algorithm, a technique that finds the extremum of a unimodal function by iteratively shrinking the interval inside which the extremum is known to exist [61]. In our case, the search algorithm starts with the interval \([0,1]\), and we set the stopping criteria to be that the size of the interval which contains \( p_i \) is strictly smaller than \( 10^{-8} \).

**Uniform Distribution.** The uniform distribution has easier calculations and analytical solutions. If \( v_i \sim \mathcal{U}(0,1) \), then \( \forall p_i \in [0,1] \), \( F_i(p_i) = p_i \), and plugging it back to Equation (2.4) gives

\[
g_i(p_i) = -p_i^2 + (1 - Y_1 + Y_0) \cdot p_i + Y_1 - c_a.
\]

Hence, the optimal price

\[
p^*_i = \frac{(1 + Y_1 - Y_0)}{2}.
\]
<table>
<thead>
<tr>
<th></th>
<th>Epinions</th>
<th>Flixster</th>
<th>NetHEPT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>11K</td>
<td>7.6K</td>
<td>15K</td>
</tr>
<tr>
<td>Number of edges</td>
<td>119K</td>
<td>50K</td>
<td>62K</td>
</tr>
<tr>
<td>Average out-degree</td>
<td>10.7</td>
<td>6.5</td>
<td>4.12</td>
</tr>
<tr>
<td>Maximum out-degree</td>
<td>1208</td>
<td>197</td>
<td>64</td>
</tr>
<tr>
<td>#Connected components</td>
<td>4603</td>
<td>761</td>
<td>1781</td>
</tr>
<tr>
<td>Largest component size</td>
<td>5933</td>
<td>2861</td>
<td>6794</td>
</tr>
</tbody>
</table>

Table 2.1: Dataset statistics

For both normal and uniform distributions, if $p_i^* > 1$ or $p_i^* < 0$, it is normalized back to 1 or 0, respectively. Also note that the above solution framework applies to any probability distribution that $v_i$ may follow, as long as an analytical or numerical solution can be found for $p_i^*$.

Lines 7-10 in Algorithm 5 (and also the U-Greedy seed selection procedure in All-OMP and FFS) can be accelerated by the CELF optimization [100], or the more recent CELF++ [71]. The adaptation is straightforward and the details can be found in [71,100].

### 2.7 Experiments

We conducted experiments on real-world network datasets to evaluate our proposed baselines and the PAGE algorithm. In all these algorithms, a key step is to compute the marginal profit of a candidate seed. As mentioned in Section 2.3, computing the exact expected profit is intractable for the LT-V model. Thus, we estimated the expected profit with Monte Carlo (MC) simulations. Following [86], we ran 10,000 simulations for this purpose. This is an expensive step and as for Influence Maximization, it limits the size of networks on which we can run these simulations. For the same reason, the CELF optimization was used in all algorithms as a heuristic.

All implementations were done in C++ and all experiments were run on a server with 2.50GHz eight-core Intel Xeon E5420 CPU, 16GB RAM, and Windows Server 2008 R2.

#### 2.7.1 Data

We used three network datasets whose statistics are summarized in Table 3.2: (i) Epinions [128], a who-trust-whom network extracted from review site Epinions.com: an edge $(u_i, u_j)$ was drawn if $u_j$ has expressed her trust
in $u_i$’s reviews; (ii) Flixster\textsuperscript{3}, a friendship network from social movie site Flixster.com: if $u_i$ and $u_j$ are friends, we drew edges in both directions; (iii) NetHEPT (a standard dataset that is widely adopted in the literature of social influence \cite{37,39,72,86})\textsuperscript{4}, a co-authorship network extracted from the High Energy Physics Theory section of arXiv.org: if $u_i$ and $u_j$ have co-authored papers. Edges are bidirectional in this dataset.

The raw data of Epinions and Flixster contain 76K users, 509K edges and 1M users, 28M edges, respectively. We used the METIS graph partition software\textsuperscript{5} to extract a subgraph for both networks, to ensure that MC simulations can finish within a reasonable amount of time.

### 2.7.1.1 Computing Influence Weights

We used two methods, Weighted Distribution (WD) and Trivalency (TV), to assign influence weights to edges. For WD, $w_{i,j} = A_{i,j} / N_j$, where $A_{i,j}$ is the number of actions $u_i$ and $u_j$ both perform, and $N_j$ is a normalization factor, i.e., the number of actions performed by $u_j$, to ensure $\sum_{u_i \in N^{in}(u_j)} w_{i,j} \leq 1$. In Flixster, $A_{i,j}$ is the number of movies $u_j$ rated after $u_i$; in NetHEPT, $A_{i,j}$ is the number of papers $u_i$ and $u_j$ co-authored; in Epinions, since no action data is available, we used $w_{i,j} = 1/d^{in}(u_j)$ as an approximation. For TV, $w_{i,j}$ was selected uniformly at random from $\{0.001, 0.01, 0.1\}$, and was normalized to ensure $\sum_{u_i \in N^{in}(u_j)} w_{i,j} \leq 1$. Figure 2.3 illustrates the distribution of weights for Flixster, and shows that influence strength is higher in WD graphs.

### 2.7.1.2 Learning Valuation Distributions

As mentioned in Section 2.1, valuations are difficult to obtain directly from users, and we had to estimate the distribution using historical sales data. On Epinions.com, a user typically wrote a review, gave an integer rating from 1 to 5, and might also report the paid in US dollars: Figure 2.4 shows an example of such reviews. If a review contains both price and rating, we can combine them to approximately estimate the valuation of that user, as in such systems, ratings are seen as people’s utility for a good, and utility is the difference of valuation and price \cite{132}.

We observed that most products have only a limited number of reviews (typically no more than 100), and thus a single product may not provide

\textsuperscript{3}http://www2.cs.sfu.ca/~sja25/personal/datasets/, last accessed on September 7th, 2015. All ratings in this dataset contain timestamps.


\textsuperscript{5}http://glaros.dtc.umn.edu/gkhome/views/metis, last accessed on September 7th, 2015.
enough samples. To circumvent this difficulty, we acquired all reviews for
the popular Canon EOS 300D, 350D, and 400D cameras. Given that these
cameras belong to the same product line of Canon (entry-level Digital
Single-Lens Reflex camera, or DSLR), we assume that they had similar monetary values to consumers at their respective release time. Further, we assume most users opted to buy (and hence reviews) the latest model when multiple
exist on the market. This allows us to treat the three cameras as a “unified”
product to obtain sufficient data points.

After removing reviews without prices reported, we were left with 276
samples. Next, we transformed prices and ratings to obtain estimated valu-
ations as follows:

\[
\text{valuation} = \text{price} \times \left(1 + \frac{\text{rating}}{5}\right).
\]

We then normalized the results into [0, 1] and fit the data to a normal
distribution with mean 0.53 and variance 0.14 estimated by maximum
likelihood estimation (MLE). Figure 2.5a plots the histogram of the normal-
ized valuations; Figure 2.5b presents the CDFs of our empirical data and
\(\mathcal{N}(0.53, 0.14^2)\). To test the goodness of fit, we computed the Kolmogorov-
Smirnov (K-S) statistic [66] of the two distributions, which is defined as the
maximum difference between the two CDFs; in our case, the K-S statistic is
0.1064. As can be seen from Figure 2.5b, \(\mathcal{N}(0.53, 0.14^2)\) is a reasonable fit
for the estimated valuations of the three Canon EOS cameras.

Since there were no price data available to collect in Flixster and
NetHEPT, we use \(\mathcal{N}(0.53, 0.14^2)\) in the simulations for all three datasets.
For a comprehensive comparison among different algorithms, we also tested

Figure 2.3: Distribution of influence weights in Flixster
Figure 2.4: A review for Canon EOS 300D camera on Epinions.com. At the end of the review, the user mentioned the price – $999.

the uniform distribution over $[0, 1]$, i.e., $\mathcal{U}(0, 1)$, as it has been commonly assumed and used in the literature [20, 132].

2.7.2 Results and Analysis

We now compare PAGE, All-OMP, and FFS in terms of the expected profit achieved, price assignments, and running time. Although all algorithms employ U-Greedy which does not terminate until the marginal profit starts decreasing. For uniformity, we report simulation results up to 100 seeds.

Expected Profit Achieved. The quality of outputs (seed sets and price vectors) of All-OMP, FFS, and PAGE for general PROMAX were evaluated based on the expected profit achieved. Figures 2.6, 2.7, and 2.8 illustrate the results on Epinions, Flixster, and NetHEPT, respectively. In each dataset, both valuation distributions were tested in four settings: WD weights with $c_a = 0.1$ and 0.001; TV weights with $c_a = 0.1$ and 0.001. As prices and
valuations are in \([0,1]\), we used \(0.1\) to simulate high acquisition costs and \(0.001\) for low costs. Except for NETHEPT-TV with \(c_a = 0.1\) (Figure 2.8(c)) and \(0.001\) (Figure 2.8(d)), FFS was better than All-OMP; this indicates that only in NetHEPT-TV, influence strength is low enough so that giving free samples blindly to all seeds would negatively impact profits.
Figure 2.7: Expected profit achieved (Y-axis) on Flixster graphs w.r.t. \(|S|\) (X-axis). (N)/(U) denotes normal/uniform distribution.

(a) WD with \(c_a = 0.1\)  
(b) WD with \(c_a = 0.001\)  
(c) TV with \(c_a = 0.1\)  
(d) TV with \(c_a = 0.001\)

Figure 2.8: Expected profit achieved (Y-axis) on NetHEPT graphs w.r.t. \(|S|\) (X-axis). (N)/(U) denotes normal/uniform distribution.
In all test cases, PAGE performed consistently better than FFS and All-OMP. The margin between PAGE and FFS is higher in TV graphs (by, e.g., 15% on Epinions-TV with $\mathcal{N}(0.53, 0.14^2)$, $c_a = 0.1$) than that in WD graphs (by, e.g., 2.1% on Epinions-WD with $\mathcal{N}(0.53, 0.14^2)$, $c_a = 0.1$), as higher influence in WD graphs can potentially bring more compensations for profit loss in seeds for FFS. Also, the expected profit of all algorithms under $\mathcal{N}(0.53, 0.14^2)$ is higher than that under $\mathcal{U}(0, 1)$, since adoption probabilities under $\mathcal{N}(0.53, 0.14^2)$ are higher.

**Price Assignments.** For $\mathcal{N}(0.53, 0.14^2)$ and $\mathcal{U}(0, 1)$, the OMP is 0.41 and 0.5, respectively. Figure 2.9 demonstrates the prices offered to each seed by All-OMP, FFS, and PAGE on Epinions-TV with $\mathcal{N}(0.53, 0.14^2)$. All-OMP and FFS assigns 0.41 and 0 for all seeds, respectively. For PAGE, as the seed set grows, price tends to increase, reflecting the intuition that discount is proportional to the influence (profit potential) of seeds, as they are added in a greedy fashion and those added later have diminishing profit potential.

<table>
<thead>
<tr>
<th></th>
<th>Epinions-WD</th>
<th>Flixster-WD</th>
<th>NetHEPT-WD</th>
</tr>
</thead>
<tbody>
<tr>
<td>All-OMP</td>
<td>$\mathcal{N}$</td>
<td>$\mathcal{U}$</td>
<td>$\mathcal{N}$</td>
</tr>
<tr>
<td></td>
<td>6.7</td>
<td>2.3</td>
<td>3.0</td>
</tr>
<tr>
<td>FFS</td>
<td>6.3</td>
<td>2.1</td>
<td>2.8</td>
</tr>
<tr>
<td>PAGE</td>
<td>4.8</td>
<td>1.3</td>
<td>2.3</td>
</tr>
</tbody>
</table>

Table 2.2: Running time, in hours (WD weights, $c_a = 0.1$)
<table>
<thead>
<tr>
<th></th>
<th>Epinions-TV</th>
<th>Flixster-TV</th>
<th>NetHEPT-TV</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\mathcal{N}$</td>
<td>$\mathcal{U}$</td>
<td>$\mathcal{N}$</td>
</tr>
<tr>
<td>All-OMP</td>
<td>5.1</td>
<td>2.4</td>
<td>1.4</td>
</tr>
<tr>
<td>FFS</td>
<td>5.5</td>
<td>2.5</td>
<td>1.5</td>
</tr>
<tr>
<td>PAGE</td>
<td>4.0</td>
<td>1.0</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Table 2.3: Running time, in hours (TV weights, $c_a = 0.1$)

Running Time. Tables 2.2 and 2.3 present the running time of all algorithms on the three networks with WD weights and TV weights, respectively. As adoption probabilities under $\mathcal{N}(0.53, 0.14^2)$ are higher, all algorithms ran longer with the normal distribution on all graphs. Similarly, as influence in WD graphs are higher, the running time on them is longer than that on TV graphs.

All-OMP and FFS had roughly the same running time. More interestingly, PAGE was faster than both baselines in all cases. The observation is that in each round of U-Greedy, PAGE maximizes the marginal profit for each candidate seed in the priority queue maintained by CELF. Thus, heuristically, the lazy-forward procedure in CELF (see [100]) may have a better chance to return the best candidate seed sooner for PAGE. All-OMP and FFS also benefited from CELF, but since the marginal profits of candidate seeds are often suboptimal, elements in the CELF queue tend to be clustered, and thus the lazy-forward is not as effective. Besides, for PAGE under $\mathcal{N}(0.53, 0.14^2)$, the golden section search usually converges in less than 40 iterations with stopping criteria $10^{-8}$ (defined in Section 2.6.2); thus the extra overhead it brought is negligible compared to MC simulations.

To conclude, our empirical results on three real-world datasets with two different valuation distributions have shown that the PAGE algorithm consistently outperforms baselines All-OMP and FFS in both expected profit achieved and running time. It was also the most robust one (against various inputs) among all algorithms.

2.8 Discussion and Future Work

The current algorithms for ProMax cannot scale to graphs with more than tens of thousands of nodes due to the expensive MC simulations. To achieve scalability, one extend fast heuristics developed specifically for LT, e.g., LDAG [39] and SimPath [72], so that they are suitable for computing

---

6 The results for $c_a = 0.001$ are similar, which are omitted here.
expected profit accurately under our LT-V model. Another possible extension is to consider users’ spontaneous interests in product adoption, and incorporate it into the LT-V model for profit maximization. Due to personal demand, a user may have spontaneous interests in a certain product even when no neighbor in the network has adopted. To model this, each node $u_i$ is associated with a “network-less” probability $\delta_i$ [50]. An inactive node becomes influenced when the sum of $\delta_i$ and the total influence from its adopting neighbors are at least $\theta_i$. A marketing company can thus wait for spontaneous adopters to emerge first and propagate their adoption (for $\ell$ time steps, where $\ell$ is the diameter of $G$), and then deploy a viral marketing campaign to maximize the expected profit. Our analysis and solution framework can be naturally applied to this setting. In addition, it is interesting to look into more sophisticated methodologies to acquire knowledge on user valuations, e.g., by leveraging users’ full previous transaction history, as well as look at real datasets besides Epinions.

In Section 2.2, we discussed the connections and differences between the LT-V model proposed in this chapter and the LT-C model studied in [17]. As they model different aspects in social influence and offer orthogonal extensions to the canonical LT model, it is interesting to incorporate the new elements of both models into a combined one. Note that a common property between the two models is that they both distinguish between the state of a node being influenced and the state of actually adopting the product. Although the extensions model different aspects of social influence diffusions, in principle the two models may be integrated together to obtain a mixed model. In the integrated model, termed LT-VC, a user $i$ transits from the influenced state to adopting with probability $1 - F_i(p_i)$, namely the probability that the valuation of $i$ is at least as high as the price offered to $i$. If $i$ does not adopt the product, then she would “tattle”, i.e., still spreading her opinions on the product to neighbors: She enters the “promote” state with probability $\alpha_i \cdot F_i(p_i)$ to spread positive opinions and the “inhibit” state with probability $(1 - \alpha_i) \cdot F_i(p_i)$ to spread negative opinions.

Challenges exist for solving ProMax under the new LT-CV model. First, is the expected total profit still a submodular function of the seed set given any fixed price vector, especially considering the presence of tattling behaviors as well as the existence of both positive and negative opinions. Second, will the approximation guarantee of the U-Greedy algorithm be preserved under this model? Third, is it possible (and if yes, how) to solve the PAGE algorithm to tackle ProMax for this model?
Chapter 3

Competitive Viral Marketing: The Host Perspective

In this chapter, we propose a “viral-marketing-as-a-service” business model for social network hosts, and study viral marketing from a novel angle: the host’s perspective. In particular, we solve the problem of how a social network service provider should select and allocate seeds to competing advertisers, such that first, the collective spread of influence over all advertisers are maximized, and second, seed allocations are fair: expected reward for each advertiser is in proportion to its investment in the marketing campaign.

3.1 Introduction

The bulk of research on viral marketing and influence maximization assumes that there is one company, introducing one product in the market. In other words, there is no competition. However, in the real world, typically multiple players compete with comparable products over the same market. For example, consider consumer technologies such as videogame consoles (X-Box vs. Playstation), digital SLR cameras (Canon vs. Nikon) or smartphones (Android vs. iPhone): since the adoption of these consumer technologies is not free, it is very unlikely that an average consumer will adopt more than one of the competing products. Recognizing this, there has been some recent work on competitive viral marketing, where two or more players compete with similar products for the same market. The majority of these studies focus on the best strategy for one of the players [18, 24, 29, 31, 79, 94].

Our motivating observation is that social network platforms are owned by third party such as Facebook and LinkedIn. The owner keeps the proprietary
social graph confidential\footnote{http://techcrunch.com/2013/01/24/my-precious-social-graph/; last accessed on September 7th, 2015.} for obvious reasons of the company benefits, as well as due to privacy legislation. We call the owner the host. Companies that want to run viral campaigns are the host’s clients. The clients typically do not have direct access to the network and thus cannot choose seeds for their campaign on their own. Any campaign would need the host’s permission and privilege to run. Take Facebook as an example, business owners can set up a Facebook Page and create display ads or promoted posts to reach users\footnote{https://www.facebook.com/business; last accessed on September 7th, 2015.}, but they are not able to effectively implement a viral marketing campaign which directly reaches individual users, due to the lack of access to the network graph and privacy concerns.

Motivated by this observation, we propose and study the novel problem of competitive viral marketing from the host perspective. We consider a new business model where the host offers viral marketing as a service, for a price. It allows the clients to run campaigns by specifying a seed budget, i.e., number of seeds desired. The host controls the selection of seeds and their allocation to companies. Once seeds are allocated, companies compete for adopters of their products on the common network.

In classical non-competitive influence maximization, the objective is to choose the seeds so as to maximize the expected number of adopters. However, in a competitive setting, from the host’s perspective, it is important to not only choose the seeds to maximize the collective expected number of adoptions across all companies, but also allocate seeds to companies in such a way that guarantees the “bang for the buck” for all companies is nearly the same.

More specifically, the bang for the buck for a company is the ratio between the expected number of adopters of its product and the number of seeds. We call this the amplification factor, as it reflects how the investments in a small number of seeds get amplified by the network effect through propagation of social influence. If the host allocates the seeds carelessly to its clients, it may result in a wide variance in the amplification factors, leading to resentful clients. Consider the following hypothetical scenario. Suppose Canon and Nikon are two clients with seed limit 20 and 30 respectively, and Facebook, as host, would select 50 seeds in total. If those 50 seeds are allocated in such a way that Canon ends up getting expected spread of 400 (“bang for the buck” = 20), while Nikon gets 300 (“bang for the buck” = 10), this allocation is unfair and may lead to Nikon to feel resentful.
Motivated by the aforementioned, we propose a new propagation model
called $K$-LT by extending the classical Linear Threshold (LT) model [86] to
capture the competitive aspect in viral marketing. Intuitively, propagation
in our model consists of two phases. A node (user) is in one of three states:
inactive, influenced, or active. It adopts a product only in the active state.
In the first phase, inactive nodes may become influenced (to adopt a prod-
uct) as a result of influence coming in from their neighbors. In the second
phase, an influenced node makes its choice to adopt one of the products (i.e.,
becomes active) based on the relative strengths of incoming influence for dif-
ferent products. The model is intuitive and retains the desired properties of
monotonicity and submodularity.

We then define the fair seed allocation problem whose goal is to allocate
seeds to the companies such that their amplification factors are as close
to each other as possible, while the total expected number of adoptions
over all companies is maximized. The problem is NP-hard and we devise an
efficient and effective greedy heuristic to tackle it. To summarize, we make
the following contributions:

- We study competitive viral marketing from a campaign host’s perspec-
tive. We propose the $K$-LT propagation model and show that in our
model, expected influence spread for any individual competing product
is monotone and submodular (Section 3.4.1).

- We define the problem of Fair Seed Allocation (FSA) and discuss a
number of options for formalizing it. As a case study, we focus on
minimizing the maximum amplification factor offered to companies
(Section 3.3.2).

- We show that FSA under $K$-LT model is NP-hard. To tackle this prob-
lem, we develop two exact algorithms based on dynamic programing
and integer linear programing. We also propose an efficient heuristic
algorithm, Needy Greedy, a natural adaptation of the classic greedy
algorithm for machine scheduling [89] (Section 3.5.4).

- We conduct extensive experiments on four real-world network datasets.

3.2 Related Work

Historically, competitions between two products have largely been ad-
dressed in economics. For example, in Arthur [7] and David [44], network-
independent properties were employed to model the propagation of two tech-
nologies through a market. Tomochi et al. [139] offered a game-theoretic approach which relies on the network for spatial coordination games. However, they did not address the problem of how to effectively take advantage of social networks and viral influence propagation when introducing a new technology into a market.

Recent studies on competitive viral marketing mainly aimed to extend the IC or LT model. Almost all of them focused on the client perspective as opposed to the host perspective (which is the case for our work in Section 3). In addition, some of them were restricted to just two entities. Two early papers to tackle influence maximization in a competitive setting are [18, 31] and both studied the problem from the “follower’s perspective”. The follower is the player trying to introduce a new product into an environment where a competing product already exists. Both studies showed that the problem for the follower maintains the desired properties of monotonicity and submodularity and the approximation result in [119] is applicable.

In the model described in Bharathi and Kempe [18], when a node $u$ adopts a product at time $t$ it tries to activate each currently inactive neighbor $v$. If the activation attempt from $u$ on $v$ succeeds, $v$ will become active, adopting the same product of $u$, at time $t+T_{uv}$, where $T_{uv}$’s are mutually independent, exponentially distributed random variables, named activation time. They are needed in order to avoid tie-breaking for simultaneous activation attempts.

Carnes et al. [31] proposed two models. In the distance based model, each edge is also given a “length”: this might be for instance the inverse of the probability of seeing influence travelling through that edge. In each moment, edges on which influence has travelled are called active. Let us consider two competitive products $A$ and $B$, and the two sets of initial adopters for $A$ and $B$, denoted $I_A$ and $I_B$ respectively. For a given node $u$ we consider the shortest path distance $d$ along active edges to any initiator. Let $u^d_A$ denote the number of initiators nodes in $I_A$ which are at distance $d$, along active edges, from $u$ (and similarly for $u^d_B$). Then $u$ adopts product $A$ with probability $u^d_A/(u^d_A + u^d_B)$ (and similarly for product $B$). In the wave propagation model propagation happens in discrete steps. In step $d$, all nodes that are at distance at most $d-1$ from some node in either $I_A$ or $I_B$, and all nodes for which the closest initial node is farther than $d-1$ do not have taken a product yet (where the distance is again with respect to active edges). Both models above reduce to the IC model if there is no competition. For both models, they showed that the corresponding influence function is monotone and submodular, so they can apply the simple greedy algorithm to obtain a $(1 - 1/e - \epsilon)$-approximation to the optimal strategy for the follower.
Kostka et al. [94] studied competitive influence diffusions under a game-theoretic framework and showed that finding the optimal strategy of both the first and second player is \textbf{NP}-complete. Budak et al. [29] and He et al. [79] studied the problem of influence blocking maximization, where one entity tries to block the influence propagation of its competitor as much as possible, under extended IC and LT models, respectively. Pathak et al. [123] proposed an extension of the voter model to study multiple cascades. Chen et al. [34] study influence maximization in the presence of negative opinions. Their model, called IC-N, extends the IC model by further dividing the active node state into two substates: positive and negative. The spread of negative opinions is characterized by a “quality factor” $q$, a parameter that models the quality of the propagating entity, as perceived by users. There is no seed set for the negative opinion, but rather, any active user may become positive or negative with corresponding probabilities. They show that when $q$ is assumed to be uniform across all users, the influence spread of the positive opinion is a monotone submodular function of the seed set. However, when this assumption is lifted, submodularity no longer holds. We note that none of the above work considered the host’s perspective.

Borodin et al. [24] proposed extensions to the LT model to deal with competing products. In Section 3.3 we shall discuss their Weighted-Proportional Competitive LT (WPCLT) model in more details and highlight the differences between WPCLT and our proposed K-LT model. More recently, Borodin et al. [23] studied viral marketing mechanism design from the host perspective. They showed that when the number of competitors is at least three, a mechanism which first uses the greedy algorithm to select seeds and then randomly assigns seeds to companies is truthful. We shall discuss their results in more details in Section 3.5.5.

Last but not the least, Myers and Leskovec analyzed Twitter data to study competitions which took place in real-world information and influence diffusions on Twitter [117].

### 3.3 Models and Problem Definition

In this section we present the propagation model underlying our work and provide the problem statement. We first introduce our extended LT model (dubbed K-LT, for Linear Thresholds with $K$ Competitors) that captures competition, and then provide conceptual justifications of the model. Then we highlight the difference between K-LT and the Weighted-Proportional Competitive LT (WPCLT) model by Borodin et al. [24].
3.3.1 The K-LT Diffusion Model

Let $K$ denote the number of competing companies. Let $C_i$ and $S_i$, $i \in \{1, 2, \ldots, K\}$, denote the $i$-th company and its seed set respectively. Each node $v \in V$ selects an activation threshold $\theta_v$ uniformly at random from $[0, 1]$. Initially, all nodes are inactive. At time step 0, for each company $C_i$, a seed set $S_i$ is targeted. This means that if $u \in S_i$, then $u$ becomes active with respect to $C_i$ at time 0. We also assume that all seed sets are disjoint, i.e., $S_i \cap S_j = \emptyset$ whenever $i \neq j$. Moreover, since it is a competitive model, each node adopts at most one company’s product.

At any subsequent time step $t \geq 1$, the activation of a node takes place in two phases. First, an inactive node $v$ becomes influenced when the total influence weight from its active in-neighbors (regardless of which company) reaches $v$'s threshold:

$$\sum_{\text{active } u \in N^\text{in}(v)} p_{u,v} \geq \theta_v.$$  

Then, in a second phase (still at time $t$), $v$ becomes active by picking one company out of those of its in-neighbors activated at $t - 1$.

Let $A^i_{t-1}$ denote the set of nodes that are active with company $C_i$ at the end of time $t - 1$ and $A_{t-1}$ denote the set of nodes that are active at the end of time $t - 1$, w.r.t. any company. Hence, $v$ becomes active at time $t$ with company $C_i$ with probability

$$\frac{\sum_{u \in A^i_{t-1} \setminus A^j_{t-2}} p_{u,v}}{\sum_{u \in A_{t-1} \setminus A_{t-2}} p_{u,v}}.$$  

Once a node becomes active, it remains so and will not switch company. The diffusion process continues until no more nodes can be activated.

The $K$-LT model reflects several phenomena of competitive influence propagation that match our daily experience as well as studies in the literature. The first phase models the threshold behavior in influence propagation as in the original LT model, and the second phase incorporates the recency effect in the final decision among competing products. Indeed, it has been recognized in various studies that influence decays very quickly in time, and thus customers are more likely to rely on recent information than on old information, when choosing which product to adopt [80, 124, 151].

Next, we introduce a critical invariance property w.r.t. influence spread under this model.

**Influence spread functions.** Let $S = \{S_1, \ldots, S_K\}$ be the set of seeds sets for the various companies, i.e., $S$ corresponds to a seed set allocation. We
use \( S_{-i} \) to denote the set of seed sets for all companies but company \( C_i \), i.e.,
\[
S_{-i} = \{ S_1, \ldots, S_{i-1}, S_{i+1}, \ldots, S_K \}.
\]

**Definition 1** (Expected Spread). For each company \( C_i \), let \( \sigma_i(S_i, S_{-i}) \) denote the expected number of active nodes, or the expected spread, of \( C_i \), given seed set allocation \( S \). We define the overall expected spread, denoted \( \sigma_{\text{all}} = \sum_{i=1}^{K} \sigma_i(S_i, S_{-i}) \), to be the expected number of active nodes w.r.t. any company.

Observe that under both \( K \)-LT and WPCLT models, the first phase of activation follows the same activation condition as in the classic LT model. Therefore, we have the following proposition.

**Proposition 1.** Given a directed graph \( G = (V, E) \) with edge weights, and \( K \) pair-wise disjoint subsets \( S_1, S_2, \ldots, S_K \) of \( V \), then under both the \( K \)-LT model and the WPCLT model, letting \( S = S_1 \cup \ldots \cup S_K \), we have
\[
\sigma_{\text{all}} = \sigma_{LT}(S).
\]

where \( \sigma_{LT} \) is the spread function for the classical LT model.

This implies that once a union seed set \( S \) is given, no matter how it gets partitioned (into \( K \) disjoint subsets), \( \sigma_{\text{all}} \) remains the same. In other words, the total influence spread is invariant w.r.t. any \( K \)-partition of \( S \).

**Comparisons with the WPCLT model.** In the WPCLT model proposed by Borodin et al [24], the first phase of activation is exactly the same as in \( K \)-LT. The difference lies in the second phase, i.e., the way in which newly influenced nodes decides the company. In WPCLT, a node \( v \) picks a certain \( C_i \) with probability equal to the ratio between the total weight from the \( C_i \)-active in-neighbors and that from all active in-neighbors. That is, all past exposure are accounted for adoption. Thus, \( v \) becomes active with company \( C_i \) with probability
\[
\frac{\sum_{u \in A_{i-1}} p_{u,v}}{\sum_{u \in A_{i-1}} p_{u,v}}.
\]

As we shall show in Theorem 4, \( \sigma_i(S_i, S_{-i}) \) is monotone and submodular w.r.t. \( S_i \) under the \( K \)-LT model, while this result does not hold in WPCLT, which is somewhat counter-intuitive as noted by the authors of [24]. Indeed, \( \sigma_i(S_i, S_{-i}) \) being non-monotone means that adding a new seed \( x \) to \( S_i \) may cause the spread of \( C_i \) to go down. This counter-intuitive phenomenon can be explained with the possibility that a certain graph structure will allow the seeding of some nodes to trigger multiple “activation attempts” for seeds of a different company, shown in Example 1 below. For more detailed examples that illustrate non-monotonicity and non-submodularity of the WPCLT model, we refer the reader to [24].
**Example 1** (Activation in WPCLT). Consider Figure 3.1. Suppose that there are two companies with seed sets $S_1 = \{u\}$ and $S_2 = \{w\}$. Also suppose that $\theta_u$ and $\theta_x$ fall into the interval $(0.5, 1)$. At time step 1, $v$ becomes active w.r.t. company 2 (as $p_{w,v} = 1 > \theta_v$), while $x$ remains inactive (as $p_{u,x} = 0.5 < \theta_x$). Subsequently, at time step 2, $x$ first gets influenced as the total incoming influence weight is now 1. Then, $x$ will activate w.r.t. company 1 with probability 0.5 and company 2 with probability 0.5.

![Figure 3.1: Sample graph accompanying Example 1.](image)

In this example, although $u$ (in company 1) fails to activate $x$ at time step 1, $x$ may still adopt company 1 under WPCLT. The reason is that $x$ gets additional influence from $v$ which has company 2! Thus, seeding $w$ for company 2 ends up “helping” the competitor company 1: $u$ gets a second chance at activating $x$ after failing at first. However, this phenomenon will not occur in $K$-LT: at time step 2, after getting influenced, $x$ will activate w.r.t. company 2 exclusively, with probability $0.5/0.5 = 1$.

### 3.3.2 Problem Definition

We are ready to provide the formal problem statement of fair competitive viral marketing from the host perspective. We will focus on the $K$-LT model hereinafter, unless otherwise specified. Assume that there are $K$ companies, as clients of the host $H$, competing with similar products (one product each). Before the campaign is run, each company $C_i$ would approach the host, specifying a positive integer $b_i$ as its budget (maximum number of seeds wanted), and we assume that $b_1 + b_2 + \ldots + b_K < |V|$. As its business model, $H$ charges every company a fixed amount of money per requested seed, as well as surcharges proportional to the expected spread achieved. Before defining the problem, we first introduce the important notion of *amplification factor*.

**Definition 2** (Amplification Factor). The amplification factor of $C_i$, denoted $\alpha_i$, is the average influence spread that $C_i$ gets per seed, i.e.,

$$\alpha_i = \frac{\sigma_i(S_i; S_{-i})}{b_i}. \quad (3.2)$$

Intuitively, after receiving budgets from all companies, $H$ will allocate each company $C_i$ a seed set $S_i$, $|S_i| = b_i$, such that

---

46
1. The overall influence spread — \( \sigma_{\text{all}} \) (Definition 1) — is maximized;

2. Given that the overall influence spread is maximized, the allocations of seeds to the \( K \) participating companies should be done in such a way that the expected influence spread across all companies is as “balanced” as possible, i.e., the amplification factor of each company is as close as possible. In other words, the allocation of seeds should be fair.

Note that pursuing the second objective (fair seed allocation) does not contradict with the first objective (maximizing the total influence spread), due to the invariant property on total influence spread under the \( K \)-LT model, as stated in Proposition 1.

Formally, we define the problem of competitive influence maximization from the host’s perspective, which consists of two subproblems, as follows.

**Problem 2** (Overall Influence Maximization). Given a directed graph \( G = (V, E) \) with pair-wise edge weights, numbers \( b_1, b_2, \ldots, b_K \in \mathbb{Z}_+ \) with \( \sum_{i=1}^{K} b_i \leq |V| \), select a seed set \( S \subseteq V \) of size \( \sum_{i=1}^{K} b_i \), such that \( \sigma_{\text{all}} \) is maximized.

By Proposition 1, Problem 2 can be solved by establishing a connection to the classical single-company influence maximization problem for the vanilla LT model. In addition, by Proposition 1, under both \( K \)-LT and WPCLT models, Problem 2 is equivalent to the original influence maximization under the LT model, and hence it is NP-hard.

By the same token, since \( \sigma_{LT}(\cdot) \) is monotone and submodular [86], selecting the set of seeds \( S \) can be done using the classic greedy algorithm outlined in the introduction as for the original LT model, giving a \((1 - 1/e - \varepsilon)\)-approximate solution to the optimum selection of seeds. Formally, we have:

**Corollary 1.** For an arbitrary instance of the \( K \)-LT model, Algorithm 1 provides an \((1 - 1/e - \varepsilon)\)-approximation for Problem 2.

**Corollary 2.** For an arbitrary instance of the WPCLT model, Algorithm 1 provides an \((1 - 1/e - \varepsilon)\)-approximation for Problem 2.

In the rest of this chapter, we assume that the union (global) seed set is selected by the greedy algorithm, and we shall focus on their fair allocation to the \( K \) participating companies.

The goal of our second problem is to allocate seeds among the \( K \) clients such that the amplification factor of all companies is as close as possible, to achieve maximum fairness. We have various options to formalize the notion
of fairness. In the following problem statement and hereinafter we adopt as objective function to maximize the minimum amplification factor. Intuitively, when the minimum amplification factor is maximized, it balances out all the amplification factors. A discussion on other alternatives is provided in Section 3.3.3.

**Problem 3 (Fair Seed Allocation (FSA)).** Given a directed graph $G = (V, E)$ with pair-wise edge weights, numbers $b_1, b_2, \ldots, b_K \in \mathbb{Z}_+$, a set $S \subseteq V$ with $|S| = \sum_{i=1}^{K} b_i$, find a partition of $S$ into $K$ disjoint subsets $S_1, S_2, \ldots, S_K \subseteq S$, such that $|S_i| = b_i$, $i \in [1, K]$, and the minimum amplification factor of any color is maximized:

$$\arg \max_{S_1 \ldots S_K} \min_{i=1 \ldots K} \frac{\sigma_i(S_i, S_{-i})}{b_i}.$$ (3.3)

It is worth emphasizing that although the two problems are formulated separately, the host $\mathcal{H}$ must solve them in a sequential order to achieve its goals. In particular, notice that the output of Problem 2, i.e., the union seed set $S$, is given as input for Problem 3.

### 3.3.3 Possible Alternative Objectives

Our goal of partitioning the union seed set $S$ is to make the amplification factors of all advertisers as close as possible, such that the allocation is deemed fair. To achieve this goal, in Problem 3, we defined the objective function as maximizing the minimum amplification factor. One can offer similar alternative objective functions, while trying to achieve the same goal. For instance, we may try to minimize the difference $\alpha_{\text{max}} - \alpha_{\text{min}}$, or the ratio $\alpha_{\text{max}}/\alpha_{\text{min}}$. More sophisticated objective functions can be based on the $L_p$-norm. In general, the objective function based on $L_p$-norm can be defined as

$$\left( \frac{1}{\sum_{i=1}^{K} \left| \frac{\sigma_i(S_i, S_{-i}) - b_i}{\sigma_{\text{all}}} \right|^p } \right)^{1/p},$$ (3.4)

which one may want to minimize. A comprehensive theoretical analysis of these various objective functions would be an interesting exercise, but it is not the focus of this paper. In the experiments section, we show that our algorithm performs well w.r.t. essentially all of these objectives.
3.4 Model Properties

Before we develop algorithms to solve Problem 3, we take a deeper look at the properties of the K-LT model, which will allow us to characterize the complexity of FSA under K-LT and develop efficient and effective seed allocation algorithms.

3.4.1 Submodularity

We first show that the expected spread function for individual colors is monotone and submodular (Theorem 4) in the K-LT model. To prove this result, we employ a plot similar to the one in Kempe et al. [86], by establishing the equivalence between the K-LT model and a competitive version of the “live-edge” model (Definition 3). This, importantly, will in turn help us derive a closed-form expression for the spread function (Theorem 5), which will play a pivotal role in the design of our algorithms and characterizing the complexity of FSA: it is NP-hard in general (Theorem 6), but can be solved in polynomial time for $K = 2$.

We start by introducing the competitive live-edge model, by extending the live-edge model defined in [86].

**Definition 3 (Competitive Live-Edge Model).** Given a directed graph $G = (V, E)$ with edges labeled by influence weights, we can obtain a possible world $X$ as follows. Each node $v$ picks at most one of its incoming edges at random, selecting edge $(u, v)$ with probability $p_{u,v}$ and selecting no edge with probability $1 - \sum_{w \in N^{in}(v)} p_{w,v}$. The selected edges are declared “live”, while others “blocked”. By definition, incoming edges to nodes in the seed set $S$ are blocked. We call a directed path a live-edge path if it consists entirely of live edges.

In a possible world $X$, we say a node is $C_i$-reachable, if there exists a live-edge path from a node in $S_i$ to $v$. Note that a node $v$ has at most one incoming live edge, thus there is at most one live-edge path from $S$ to $v$. Thus, the notion of color rechability is well-defined.

It is easy to see that the spread function under the competitive live-edge model is monotone and submodular. Clearly, each possible world $X$ is a deterministic graph. Let $R_X(\{u\})$ be the set of reachable nodes from a particular node $u$ on live-edge paths, in $X$. Then the set of nodes reachable from $S_i$ is $R_X(S_i) = \cup_{u \in S_i} R_X(\{u\})$. The function $|R_X(S_i)|$ is clearly monotone and submodular. Finally, the expected number of $C_i$-reachable nodes according to the live-edge model, $\sum_X \Pr[X] \cdot |R_X(S_i)|$, is a non-negative linear combination of monotone submodular functions, and thus is monotone.
and submodular (in \( S_i \)). Here, \( \Pr[X] \) is the probability of the possible world \( X \), which is determined by the choice of live/blocked edges.

We now state the submodularity result for \( K \)-LT:

**Theorem 4.** Under the \( K \)-LT model, for any color \( C_i \), the expected influence spread \( \sigma_i(S_i, S_{-i}) \) is monotone and submodular in \( S_i \), for any fixed \( S_{-i} \) such that all seed sets are pairwise disjoint.

**Proof.** We prove this result by establishing the equivalence between the \( K \)-LT model and the competitive live-edge model (Definition 3). We shall prove this claim: Given \( K \) colors and their corresponding seed sets \( S_1, S_2, \ldots, S_K \) (all disjoint), for any color \( C_i \), the following two distributions over sets of nodes are equivalent:

(i) the distribution over \( C_i \)-active sets obtained by running the \( K \)-LT process to completion from \( S_1, S_2, \ldots, S_K \), and

(ii) the distribution over sets of \( C_i \)-reachable nodes according to the live-edge model.

The theorem follows from this claim.

We next prove the claim. If a node \( v \) has not become active after time step \( t \), then the probability that it becomes \( C_i \)-active at time step \( t+1 \) is

\[
\frac{\sum_{u \in A_t \setminus A_{t-1}} p_{u,v}}{1 - \sum_{u \in A_{t-1}} p_{u,v}} \cdot \frac{\sum_{u \in A_t' \setminus A_{t-1}} p_{u,v}}{\sum_{u \in A_t \setminus A_{t-1}} p_{u,v}} = \frac{\sum_{u \in A_t \setminus A_{t-1}} p_{u,v}}{1 - \sum_{u \in A_{t-1}} p_{u,v}},
\]

where the former quantity is the probability that \( v \) becomes active at \( t+1 \), and the latter is the probability that \( v \) adopts color \( C_i \), given that \( v \) gets activated.

For the competitive live-edge model, we start the “reach-out” process with seed sets \( S_1, S_2, \ldots, S_K \). In the first stage, if a node \( v \)'s selected live-edge is incident on \( S_i \), then \( v \) is \( C_i \)-reachable from a seed in \( S_i \). We denote the set of such nodes by \( A_t' \). In general, let \( A_t'' \) denote the set of nodes which are found to be \( C_i \)-reachable from a node in \( S_i \) in stage \( t \). In this way, we can obtain sets \( A_1'' \), \( A_2'' \), \ldots. Similarly, we can also obtain sets \( A_t', t = 1, 2, 3, \ldots \), which represent the set of nodes reachable from \( S_1 \cup S_2 \cup \ldots \cup S_K \) in stage \( t \). Now, if a node \( v \) has not yet been determined \( C_i \)-reachable by the end of stage \( t \), then the probability that \( v \) will be determined \( C_i \)-reachable at stage \( t+1 \) is the chance that its chosen edge is from \( A_t' \setminus A_{t-1} \), which is

\[
\frac{\sum_{u \in A_t' \setminus A_{t-1}} p_{u,v}}{1 - \sum_{u \in A_{t-1}} p_{u,v}}.
\]
Given that, the probability that \( v \) proceeds to become \( C_i \)-reachable is

\[
\sum_{u \in A'_t \setminus A'_{t-1}} \frac{P_{u,v}}{\sum_{u \in A'_t \setminus A'_{t-1}} P_{u,v}}
\]

By the product rule, the probability that \( v \) will be determined to be \( C_i \)-reachable at stage \( t+1 \), given that it is not already so determined, is

\[
\sum_{u \in A'_t \setminus A'_{t-1}} \frac{P_{u,v}}{1 - \sum_{u \in A'_{t-1}} P_{u,v}}.
\]

Applying induction on time steps (stages), it is easy to see that the distributions over \( A'_t \) and \( A'_{t-1} \) are identical, and the same holds for \( A_t \) and \( A'_{t-1} \), \( \forall t \).

3.4.2 Closed-form Expression for Influence Spread

We first introduce the needed notation. By virtue of the equivalence shown in Theorem 4, \( \sigma_i(S_i, S_{-i}) \) is equal to the expected number of \( C_i \)-reachable nodes under the competitive live-edge model. Let \( X \) be a possible world. For simplicity, we write \( V - S \) for \( V \setminus S \) and \( V - S + u \) for \( (V \setminus S) \cup \{u\} \) hereinafter. With node-sets as superscripts, we denote the corresponding induced subgraph: e.g., \( \sigma_{LT}^W(S) \), where \( W \subseteq V \), denotes the expected spread of the seed set \( S \) in the subgraph of \( G \) induced by the nodes \( W \). When there is no superscript, the entire graph \( G \) is meant by default.

We now derive the closed-form expression by establishing connections to the classical LT model. Let \( I_X^{V-S}(S_i, v) \) be the indicator function which takes 1 if there exists a node \( s \) in \( S_i \) and a path from \( s \) to \( v \), otherwise the function takes 0. Thus, by definition,

\[
\sigma_i(S_i, S_{-i}) = \sum_X \Pr[X] \cdot \sigma_{i,X}(S_i, S_{-i}),
\]

where \( \sigma_{i,X}(S_i, S_{-i}) \) is the number of \( C_i \)-reachable nodes in possible world \( X \). Then, because any live-edge path from any node \( u \in S_i \) to \( v \) must not go through any node \( w \in S_{-i} \), as all incoming edges to nodes in \( S_{-i} \) are blocked by definition of the live-edge model (in other words, it has the effect of removing nodes in \( S_{-i} \) from \( G \) and hence from the possible world \( X \)), we
Figure 3.2: An example graph for illustrating adjusted marginal gain

\[ \sigma_i(S_i, S_{-i}) = \sum_X \Pr[X] \cdot \sum_{v \in V} I_{X}^{V-S-i}(S_i, v). \]

Let \( W = V - S_{-i} \), the set of nodes after removing nodes in \( S_{-i} \). Then, by switching the summations, we have

\[ \sigma_i(S_i, S_{-i}) = \sum_{v \in V} \Pr[X] \cdot I_{X}^{W}(S_i, v) \]

\[ = \sum_{v \in V} \Upsilon_{S_i,v}^{W} \tag{3.5} \]

where \( \Upsilon_{S_i,v} \) is the probability that there exists a path from \( S_i \) to \( v \) in the subgraph induced by \( V - S_{-i} \). Since \( S_i \) is the seed set for company \( C_i \), it also denotes the probability that \( v \) becomes \( C_i \)-active on the corresponding subgraph. Note that the indicator function depends only the seed set \( S_i \) and the subgraph \( W \), and not on the seeds for other colors. Therefore, \( \Upsilon_{S_i,v}^{W} \) is equal to the probability that \( v \) is activated in the subgraph induced by \( V - S + u \), under classical LT model, with seed set \( S_i \).

### 3.4.3 Adjusted Marginal Gain

Next, we introduce the notion of adjusted marginal gain, which is key to solving Problem 3.

**Definition 4 (Adjusted Marginal Gain).** Given a set \( S \) of seeds, for any \( u \in S \), the adjusted marginal gain of \( u \), denoted \( \delta_u \), is the expected spread of influence of \( \{u\} \) on the graph induced by \( V - S + u \) under the classical LT model. That is, \( \delta_u = \sigma_{LT}^{V-S+u}(\{u\}) \).

Consider the example in Fig. 3.2. Suppose \( S = \{u_1, u_2\} \) is the seed set. Then, one can verify that \( \delta_{u_1} \) is the expected spread of \( u_1 \) on graph consisting of \( u_1 \) and \( u_3 \) only, which is \( 1 + 0.3 = 1.3 \).

Next, we show the following useful result for the \( K \)-LT model, which says that given a set of seeds \( S \) selected by the host, the expected spread for
company $C_i$ only depends on the seeds $S_i$ allocated it, and not on how the remaining seeds $S - S_i$ are distributed among the other companies.

**Theorem 5.** Consider an allocation of seed sets, where the seed set $S_i \subseteq S$ is assigned to company $C_i$ and the remaining seeds $S - S_i$ are allocated arbitrarily to other companies (denoted by $S_{-i}$). Then under the K-LT model,

$$
\sigma_i(S_i, S_{-i}) = \sum_{u \in S_i} \delta_u
$$

(3.6)

**Proof.** Consider the right hand side of the equation. Since $S$ is the set of all seeds, that is, $S = S_i + S_{-i}$, we have by Definition 4,

$$
\sum_{u \in S_i} \delta_u = \sum_{u \in S_i} \sigma_{LT}^{V - S_i - S_i + u}(\{u\})
= \sum_{u \in S_i} \sum_{v \in V} \Upsilon_{u,v}^{V - S_i - S_i + u},
$$

where $\Upsilon_{u,v}^{V - S_i - S_i + u}$ is the probability with which $v$ is activated given seed set $\{u\}$, on the subgraph induced by the nodes $V - S_i - S + u$, under LT model. We next make use of the proof of Theorem 1 of [72]. There, it is shown that, under LT model,

$$
\Upsilon_{S_i,v} = \sum_{u \in S_i} \Upsilon_{u,v}^{W - S_i + u},
$$

for any set $S_i \subseteq W \subseteq V$, where $\Upsilon_{S_i,v}$ is the probability that $v$ becomes active, given seed set $S_i$, on the subgraph induced by the nodes $W - S_i + u$. Let $W = V - S_i$, then by switching the summations and applying this result, we get

$$
\sum_{u \in S_i} \delta_u = \sum_{v \in V} \Upsilon_{S_i,v}^W.
$$

From the equivalence with the live-edge model, and Eq. 3.5, the theorem follows. \qed

Consider again the example shown in Figure 3.2. Suppose there are two companies with $S_1 = \{u_1\}$ and $S_2 = \{u_2\}$. Then, $\sigma_1(S_1, S_{-1}) = \delta_{u_1} = 1.3$. Similarly, $\sigma_2(S_1, S_{-2}) = \delta_{u_2} = 1.5$. Also, note that $\sigma_{all} = \sigma_{LT}(\{u_1, u_2\}) = 2.8$. 

53
3.5 Fair Seed Allocation Algorithms

In this section, we first show that Fair Seed Allocation is NP-hard. We then design several algorithms to tackle this problem, including two exact algorithms – Dynamic Programming and Integer Linear Programming – as well as a greedy heuristic called Needy-Greedy. All three algorithms leverage Theorem 5, which says that given the union seed set $S$, the expected spread of $C_i$ is solely determined by the seeds $S_i \subseteq S$ that are allocated to company $C_i$, and it can be calculated by taking the sum of the adjusted marginal gain of the seeds in $S_i$ in appropriate subgraphs. Therefore, the input to those algorithms will be the union seed set $S$ chosen by the greedy algorithm (Algorithm 1), the adjusted marginal gain $\delta_u$ for all $u \in S$, and the budget $b_i$ for each company $C_i$.

3.5.1 Hardness Results

Having established the notion of adjusted marginal gain, we first show that Fair Seed Allocation is NP-hard.

**Theorem 6.** Fair Seed Allocation is NP-hard under the K-LT model.

*Proof.* We prove the theorem by reduction from 3-PARTITION [59]. In 3-PARTITION, we are given a set $A$ of $3m$ elements, and a size $s(a) \in Z^+$ for each element. Let $Y$ be the sum of sizes of all elements, i.e., $Y = \sum_{a \in A} s(a)$, then the question is whether there exists a partition of $A$ into $m$ disjoint subsets $A_1, A_2, \ldots, A_m$, each with exactly 3 elements, such that the sum of sizes of elements in each subset is the same, i.e., $\sum_{a \in A_i} s(a) = Y/m$. This problem is known to be strongly NP-hard [59]. Recall that a problem is strongly NP-hard if it remains NP-hard even when the numerical parameters of the problem are bounded by a polynomial of the input size. In the context of 3-PARTITION, it implies that the problem remains NP-hard even when $Y$ is bounded by a polynomial in $m$.

Let $I$ be an arbitrary instance of 3-PARTITION. We reduce it to an instance $J$ of FSA as follows. First, create $m$ companies, and for each element $a \in A$ with size $s(a)$, create a seed $u_a$ in instance $J$, with its adjusted marginal gain set to $\delta_{u_a} := s(a)$. Then we set the budget of each company to $3$. Suppose there exists a polynomial time algorithm $A$ that provides an optimal solution to FSA. Then by running this algorithm on instance $J$ and checking whether the minimum amplification factor is equal to $Y/3m$, we can separate the YES-instances from the NO-instances of 3-PARTITION, which is not possible unless P = NP.
In the above, we performed the reduction entirely in terms of adjusted marginal gains, instead of creating a graph, which is a required input to FSA. It is easy to create an input graph whose seed nodes $u_a$ satisfy the adjusted marginal gains above. E.g., create $3 \cdot m$ disjoint trees, each rooted at a node $u_a$. The root $u_a$ has exactly $s(a) - 1$ children, with influence weights on all edges set to 1. Since the trees are disjoint, $\delta_{u_a} = s(a)$. Notice this reduction is polynomial time in $m$ since $Y$ is a polynomial in $m$.

**Inapproximability.** We note that the approximability of FSA is an open problem for the max-min objective as defined in Problem 3. However, if the problem is defined using the $L_p$-norm (Equation (3.4)), then FSA is trivially inapproximable within any factor because the optimal solution in this case may yield zero. The same result holds if the objective is to minimize $\alpha_{\text{max}} - \alpha_{\text{min}}$, which may also be zero.

**Special Case.** When $K = 2$, Fair Seed Allocation resembles the PARTITION problem, which is weakly NP-hard and admits an exact dynamic programming algorithm in pseudo-polynomial time. We can adapt it to solve Fair Seed Allocation, as described in the next subsection.

### 3.5.2 Dynamic Programming

In the special case where $K = 2$, the FSA problem can be solved by a dynamic programming (DP) algorithm in polynomial time. If all adjusted marginal gain are integers, then this approach yields an optimal solution. For instances with real-valued adjusted marginal gain, a standard technique is to multiple all input numbers with a factor of $10^d$ and round to the nearest integer. In such cases, the dynamic programming algorithm is optimal w.r.t. the resultant precision, but is not guaranteed to be such w.r.t. the original real-valued instance. Generally, the loss of accuracy should be insignificant, as we empirically verified in Section 3.6.1 by comparing the output of dynamic programming with that of integer linear programming (to be proposed in the next subsection, which by definition solves the problem optimally for any input instance). Note that there is a trade-off between accuracy and complexity: As $d$ increases, accuracy improves but the size of the dynamic programming table also grows, which in turn translates into higher time and space complexity.

The dynamic programming algorithm is set up as follows. Let the seed set be $S = \{u_1, u_2, \ldots, u_B\}$, where $B = b_1 + b_2$. Also let $S^j$ denote the “partial”
seed set \( \{u_1, \ldots, u_j\} \) for \( j \in \{1, 2, \ldots, B\} \). Then, we define

\[
P(j, \mu, \ell) = \begin{cases} 
1, & \text{if } \exists Q \subseteq S^j: |Q| = \ell \text{ and } \sigma_1(Q, S^j - Q) = \mu \\
0, & \text{otherwise}
\end{cases}
\]

Here, the variable \( j \) keeps track of which seeds in \( S \) have been explored, while \( \ell \) tracks the size of a seed set \( Q \), such that with \( Q \) allocated to \( C_1 \) and \( S^j - Q \) allocated to \( C_2 \), \( \sigma_1(Q, S^j - Q) \), is exactly \( \mu \). The size of \( Q \) is bounded by \( b_1 \), the budget of \( C_1 \).

To derive the dynamic programming formula, a key observation here is that in order to have a subset of \( S^j \) of size \( \ell \) yielding influence spread \( \mu \) for \( C_1 \), one or both of the following must be true:

1. There is a subset of \( Q \subseteq S^{j-1} \) of size \( \ell \), which when allocated to \( C_1 \), yields spread \( \mu \);
2. There is a subset of \( S^{j-1} \) of size \( \ell - 1 \), which does not give spread \( \mu \) for \( C_1 \) itself, but will if we add \( u_j \) to \( C_1 \)'s allocation.

More formally, \( P(j, \mu, \ell) = 1 \) if \( P(j - 1, \mu, \ell) = 1 \), or \( P(j - 1, \mu - \delta_{u_j}, \ell - 1) = 1 \). This gives the following dynamic programming equation:

\[
P(j, \mu, \ell) = \max\{P(j - 1, \mu, \ell), P(j - 1, \mu - \delta_{u_j}, \ell - 1)\}.
\]

And the base case is \( P(0, 0, 0) = P(1, 0, 0) = 1 \). Whenever the context is clear, we refer to \( P(\cdot, \cdot, \cdot) \) as the “DP table”.

The spread of \( C_1 \) would be equal to \( Z = \frac{b_1}{b_1 + b_2} \sigma_{all} \) in the theoretically best allocation. Thus, after all non-zero cells in the DP table is populated, to obtain the actual seed set partition we can set the target be the number \( t \) obtained by amplifying and rounding the number \( Z \). If \( P(B, t, b_1) = 1 \) then we have found this ideal partition. Otherwise, we search for \( t' \) such that \( P(B, t', b_1) = 1 \) and \( |t - t'| \) is minimized. It is easy to see that the subset that satisfies \( P(B, t', b_1) = 1 \) is then set to be \( S_1 \) and for the other company, its seed set \( S_2 \) is \( S \setminus S_1 \). By construction, \( |S_2| = B - b_1 = b_2 \).

**Time complexity.** From the ranges for \( j, \mu, \) and \( \ell \), the size of the DP table is \( O(b_1(b_1 + b_2)\tau) \). This in turn determines the running time. Note that typically, \( b_1 \) and \( b_2 \) are much smaller than \( \tau \). In our implementation, we apply a couple of optimizations. First, there is no need to populate cells with \( \ell > j \). Second, if \( \mu < \delta_{u_j} \), there is no need to examine the second argument in the RHS of the dynamic programming equation.

We also note that the DP table can be represented using sparse data structures instead of a full 3-dimensional array. This would avoid storing
any zeros in memory, and thus can reduce memory usage and make the implementation more scalable. For example, a hashing-based set can be used, where the keys are triples \((j, \mu, \ell)\) for which \(P(j, \mu, \ell) = 1\).

### 3.5.3 Integer Linear Programming

When \(K > 2\), the dynamic programming algorithm described above no longer works. In general, another exact approach for solving the FSA problem is Integer Linear Programming (ILP).

For convenience, we first introduce several useful notations. First let \([n]\) denote the integer set \(\{1, 2, \ldots, n\}\). We use \(i \in [K]\) to index companies and \(j \in [m]\) to index seeds in the union seed set \(S\), where \(m := |S|\). Let \(x_{ij}\) be a binary indicator variable such that: That is,

\[
x_{ij} = \begin{cases} 
1 & \text{if } s_j \in S_i, \\
0 & \text{otherwise.}
\end{cases}
\]

We can formulate an integer program by directly translating the objective and all constraints as defined in Problem 3.

\[
\text{maximize} \quad \min_{i \in [K]} \frac{\sum_{j=1}^{m} \delta_j x_{ij}}{b_i} \\
\text{subject to} \quad \sum_{i=1}^{K} x_{ij} = 1, \quad \forall j \in [m] \\
\sum_{j=1}^{m} x_{ij} = b_i, \quad \forall i \in [K] \\
x_{ij} \in \{0, 1\}, \quad \forall i \in [K], \forall j \in [m].
\]

However, the max-min objective is not linear and the formulation above needs to be transformed and standardized. To this end, we introduce a _slack variable_ \(z\) to represent the minimum amplification factor: Let

\[
z = \min_{i \in [K]} \frac{\sum_{j=1}^{m} \delta_j x_{ij}}{b_i}.
\]
Since \( z \) is the minimum amplification factor, then by definition \( z \) must be no greater than all amplification factors:

\[
z \leq \sum_{j=1}^{m} \frac{\delta_j x_{ij}}{b_i}, \quad \forall i \in [K].
\] (3.7)

Then, using \( z \) as the maximization objective and insert the above \( K \) constraints (Equation (3.7)) into the original formulation to get a standardized ILP formulation.

\[
\text{maximize} \quad z \quad \text{(3.8)}
\]

subject to \( z \leq \sum_{j=1}^{m} \frac{\delta_j x_{ij}}{b_i}, \quad \forall i \in [K] \)

\[
\sum_{i=1}^{K} x_{ij} = 1, \quad \forall j \in [m]
\]

\[
\sum_{j=1}^{m} x_{ij} = b_i, \quad \forall i \in [K]
\]

\[
x_{ij} \in \{0,1\}, \quad \forall i \in [K], \forall j \in [m]
\]

In any feasible solution to the above ILP, we have \( z \leq \min_{i \in [K]} \frac{\sum_{j=1}^{m} \delta_j x_{ij}}{b_i} \) as per (3.7). In case \( z \) is strictly smaller than the minimum amplification factor, we can simply increase \( z \) (3.8) without breaking any constraint. This can be done until \( z \) becomes equal to the minimum amplification factor.

### 3.5.4 An Efficient Greedy Heuristic

Thus far, we have described two exact algorithms for Fair Seed Allocation. However, since the problem is NP-hard under the \( K \)-LT model (Theorem 6), such exact algorithms may not scale to large problem instances. Therefore, we now propose an efficient greedy heuristic by drawing on the literature of machine scheduling and load balancing.

The heuristic is named Needy-Greedy, which comes from the intuition that in each iteration, the algorithm greedily chooses the “neediest” company to allocate a seed. The pseudo-code is illustrated in Algorithm 6. The algorithm takes as input all seeds in the union seed set \( S \) sorted in decreasing order of adjusted marginal gain and the budget for each company. It then begins by setting all individual seed sets \( S_i \) to the empty set (line 2). We process one seed in \( S \) per iteration. Let \( C \) be the set of of companies of
Algorithm 6: Needy-Greedy for Fair Seed Allocation with max-min objective function

Data: Union seed set $S$ (with $\delta_u$ for all $u \in S$) and budget $b_i$ for all $i \in [K]$.

Result: $K$-partition of $S$, with $|S_i| = b_i$ for all $i \in [K]$.

begin
$S_i \leftarrow \emptyset$ for all $i \in [K]$

sort all seeds $u \in S$ in decreasing order of $\delta_u$

foreach $u \in \text{ordered } S$ do

$C \leftarrow \{i \in [K] : |S_i| < b_i\}$

$J \leftarrow \arg\min_{i \in C} \frac{\sigma_i(S_i, S_{-i})}{b_i}$

if $|J| > 1$ then

$j^* \leftarrow \arg\max_{j \in J} (b_j - |S_j|)$ // breaking ties (if argmax returns multiple, return one randomly)

else

$j^* \leftarrow \text{the only element in } J$

end

$S_{j^*} \leftarrow S_{j^*} \cup \{u\}$

end

which the budget has not been exhausted, i.e., all $i \in [K]$ such that $|S_i| < b_i$ (line 5): note that we can allocate a new seed only to such companies. Next, it finds the subset $J$ of companies which have the smallest amplification factor (line 6). It is possible that $J$ has more than one element: for example, in the very first iteration, all companies have an amplification factor of zero and hence they are all in $J$. Tie-breaking is required in such a case. Since seeds are visited in decreasing order of adjusted marginal gain, as a reasonable heuristic we break ties by favoring the company with the largest deficiency in seed set size, namely the largest difference between budget and current seed set size (line 7 and line 8). On the other hand, if $J$ does contain only one company (line 10), then we proceed by assigning the current seed $u$ to this company.

Adapting to $L_p$-norm objectives. Needy-Greedy not only works with the max-min objective as defined in Problem 3, but can also be flexibly adapted to deal with alternative optimization objectives for Fair Seed Allocation. For instance, in order to minimize the $L_p$-norm:

$$\left( \sum_{i=1}^{K} \left| \sigma_i(S_i, S_{-i}) - \frac{b_i}{B} \sigma_{\text{all}} \right|^p \right)^{1/p},$$
the adaptation can be done by replacing line 6 of Algorithm 6 to

\[ J \leftarrow \arg \max_{i \in \mathcal{C}} \left| \sigma_i(S_i, S_{-i}) - \frac{b_i}{B} \sigma_{\text{all}} \right|. \]

That is, the “neediest” companies are those having the largest gap (absolute difference) between its current spread and the ideal “target” spread \( \frac{b_i}{B} \sigma_{\text{all}} \).

**Time complexity.** The optimal time complexity of Needy-Greedy as in Algorithm 6 is \( O(B(\log B + \log K)) \). The term \( B \log B \) comes from sorting the union seed set; \( B \log K \) comes from the fact that there are in total \( B \) iterations, each of which examines the state of all companies to determine \( j^* \). Using a priority queue, we can perform the necessary searching and updates in \( O(\log K) \) time. When a priority queue is not used, the complexity increases to \( O(B(\log B + K)) \).

### 3.5.5 Discussion on Strategic Behaviors

It is natural to think of companies participating in viral marketing campaigns as selfish agents. We now discuss interesting game-theoretical implications on fair seed allocation. In game theory and algorithmic mechanism design, a mechanism is said to be *truthful* if each agent’s dominant strategy is to bid its true valuation. That is, no agent can benefit from lying. Taking the classical single-item auctions as example, the second-price auction is a truthful mechanism while the first-price auction is not [132]. In the context of seed allocation in viral marketing, truthfulness means that for each company, regardless of the seed budgets of other companies, reporting its true seed budget always yields the best expected influence spread.

The following counter-example shows that Needy-Greedy is not truthful. For the sake of discussion and illustration, we follow a similar analysis in [53] to assume that companies have complete knowledge of the bids of their competitors. As common in game theory, we assume companies are rational agents, and their strategic behaviors are driven by the reasoning on how to “best react” to the other company’s bid so that their own expected influence spread can be maximized.

**Example 2** (Non-truthfulness of Needy-Greedy). Consider two companies \( C_1 \) and \( C_2 \) with true budgets \( b_1 = 30 \) and \( b_2 = 29 \). Suppose the adjusted marginal gain of the seeds are as follows: The first three seeds \( s_1, s_2, \) and \( s_3 \) have 100, 96, and 95 respectively. And then there is a significant drop
in terms of seed quality: for simplicity we assume the rest of the seeds have identical adjusted marginal gain $c$, where $c$ is constant and $c < 91/3^3$.

When both $C_1$ and $C_2$ bid truthfully (30 and 29 respectively), Needy-Greedy will assign $s_1$ to $C_1$, $s_2$ to $C_2$. At this point, one can verify that the amplification factor of $C_1$ is 3.33 while the amplification factor of $C_2$ is slightly less at 3.31. This means Needy-Greedy will assign $s_3$ to $C_2$. This implies that the spread of $C_1$ would be $100 + 29c$ and the spread of $C_2$ would be $191 + 27c$.

Suppose $C_1$ knows that $b_2$ is 29 and reacts by lowering its bid to 28 (underreporting). One can verify that Needy-Greedy would instead assign $s_1$ to $C_2$, and assign both $s_2$ and $s_3$ to $C_1$. The spread of $C_1$ in this case would become $191 + 26c$. As long as $c < 91/3$, implying that $100 + 29c < 191 + 26c$, company $C_1$ would be better off by lying and underreporting its budget to be 28. This violates the condition of truthfulness.

**Remarks.** Example 2 shows the myopic nature of a greedy algorithm: When deciding which company gets seed $s_3$, Needy-Greedy picks $C_2$ as the amplification factor of $C_2$ is smaller, although the difference is quite small ($96/29$ vs. $100/30$). Furthermore, in this setup, the adjusted marginal gain of the first three seeds are quite close, and much larger than the rest of the seeds, this enables $C_1$ to have incentives to underreport.

Interestingly, the example no longer applies if we just make a very small change to the input values. First, suppose the true budget of $C_2$ is 28 instead of 29, then since $96/28 > 100/30$, $C_1$ will get $s_3$ and as a result $C_1$ no longer has incentive to underreport the budget. Second, suppose the adjusted marginal gain of seed $s_2$ is 97 instead of 96, then again $C_1$ will get $s_3$ because $97/29 > 100/30$.

**Cyclic Trend in Strategic Behaviors**

We now conduct further analysis on the setup presented in Example 2, which indicates that the strategic behaviors of the two companies could lead to an interesting cyclic trend, if the host runs multiple campaigns in a row and the constant $c$ is small enough (less than $91/3$, to be exact). This, to some extent, bears a resemblance to the cyclic behaviors observed from real-world Generalized First-Price (GFP) auctions for display ads, on a search engine called Overture [53]. The authors of [53] estimated that in 2002 and

---

As we shall see shortly, the exact value of $c$ does not matter for the purpose of this counter-example and the following analysis.
Table 3.1: Cyclic behaviors of $C_1$ and $C_2$, assuming $c < 91/3$. The rightmost column represents the best response by a company. E.g., $(C_1, 28)$ means that company $C_1$ will change its bid to 28 in the next round. Note that the round 7 is identical to round 1, indicating a cyclic trend of the strategic behaviors.

<table>
<thead>
<tr>
<th>Round</th>
<th>Bid of $C_1$</th>
<th>Bid of $C_2$</th>
<th>Spread of $C_1$</th>
<th>Spread of $C_2$</th>
<th>Strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>29</td>
<td>$100 + 29c$</td>
<td>$191 + 27c$</td>
<td>$(C_1, 28)$</td>
</tr>
<tr>
<td>2</td>
<td>28</td>
<td>29</td>
<td>$191 + 26c$</td>
<td>$100 + 28c$</td>
<td>$(C_2, 27)$</td>
</tr>
<tr>
<td>3</td>
<td>28</td>
<td>27</td>
<td>$100 + 27c$</td>
<td>$191 + 25c$</td>
<td>$(C_1, 26)$</td>
</tr>
<tr>
<td>4</td>
<td>26</td>
<td>27</td>
<td>$191 + 24c$</td>
<td>$100 + 26c$</td>
<td>$(C_2, 25)$</td>
</tr>
<tr>
<td>5</td>
<td>26</td>
<td>25</td>
<td>$100 + 25c$</td>
<td>$191 + 23c$</td>
<td>$(C_1, 24)$</td>
</tr>
<tr>
<td>6</td>
<td>30</td>
<td>25</td>
<td>$195 + 28c$</td>
<td>$96 + 24c$</td>
<td>$(C_2, 29)$</td>
</tr>
<tr>
<td>7</td>
<td>30</td>
<td>29</td>
<td>$100 + 29c$</td>
<td>$191 + 27c$</td>
<td>$(C_1, 28)$</td>
</tr>
</tbody>
</table>

2003 Overture suffered about 7.8% loss of revenue in auctions on popular keywords.

To see how the cyclic trend may take place, let us recall Example 2. When $C_1$ drops its seed budget from 30 to 28 to “manipulate” Needy-Greedy to get seeds $s_2$ and $s_3$, the competitor $C_2$ can best respond by lowering the budget from 29 to 27, in which case $C_1$ ends up with $s_1$ and $C_2$ get $s_2$ and $s_3$ back. Then, to best respond again, $C_1$ can further lower from 28 to 26 to get $s_2$ and $s_3$ back. This could go on for a few more rounds, and we illustrate the complete scenario in Table 3.1. Also note that the companies are rational agents seeking to maximize its expected influence spread, so in this setup, when they decide to lower the bid, there is no incentive to declare an budget equal to the other company’s. We shall further remark on this point shortly.

As can be seen from Table 3.1, from round 1 to round 5, the two companies alternatively decrease their respective bid, until $C_1$ bids at 26 and $C_2$ bids at 25. Here, the best response of $C_1$ is actually to increase again to its valuation 30, in which case it gets both $s_1$ and $s_3$, and have a spread of $195 + 28c$. This action is better than lowering again to 24, in which case that the spread of $C_1$ is merely $96 + 23c$: Because Needy-Greedy would allocate $s_1$ to $C_2$ and $s_2$ to $C_1$, and at this point both companies have an amplification factor of 4. Due to Needy-Greedy’s tie-breaking rule that favors companies with larger bids, $C_2$ would get $s_3$. By a similar token, bidding at 30 is also better than 25. In that case, in the first iteration both companies tie at amplification factor (zero) and budgets (25), and thus Needy-Greedy will allocate...
the first seed uniformly at random: with probability 0.5, $C_1$ gets $s_1$ and end up with a spread of $100 + 24c < 195 + 28c$; with another 0.5 probability, $C_2$ gets $s_1$ and the spread of $C_1$ is $191 + 23c < 195 + 28c$.

**Further remarks on avoiding equal budgets.** Note that in our setup, when $C_1$ or $C_2$ has the incentive to lower its bid, they would avoid declaring equal budgets (cf. round 1 to round 5 in Table 3.1). The reason is that companies are clearly better off by getting seeds $s_2$ and $s_3$ together, as opposed to getting $s_1$. When the budgets are equal, there exists a non-zero probability for both companies to end up with $s_1$ due to Needy-Greedy’s tie-breaking rules. E.g., consider the scenario when $C_1$ and $C_2$ bid at 30 and 29 respectively. If $C_1$ then bids at 29, then the random tie-breaking rule is in effect: $C_1$ and $C_2$ have an equal probability of 0.5 to get $s_1$. In contrast, if $C_1$ declares slightly lower at 28, it is deterministically better off.

**Comparisons to GFP Auctions.** In [53], the authors discussed a similar, but much more “drastic” cyclic scenario that happened in real-world GFP auctions, on search engine Overture. To show key differences between the cyclic strategic behaviors in Needy-Greedy seed allocation and Overture’s GFP auctions, we first recap the analysis in [53]. There are $K$ agents (advertisers) bidding for $N$ positions on a search results page to display ads. Each agent declares a cost-per-click value to the search engine. In GFP, the agent with the $i$-th highest wins $i$-th position, for all $1 \leq i \leq N$, and the payment is equal to its own bid value.

GFP is not truthful, either. Edelman and Ostrovsky [53] analyzed the following scenario: Suppose there are two agents $A$ and $B$, whose true valuations are 0.6 and 0.55 respectively. $B$ can drop its bid all the way to 0.01 (which is assumed as the minimum required bid) as it would still win the second position. To respond, $A$ will also decrease its bid to 0.02, barely beating $B$ and still winning the first position. Now, since $B$ in fact has a valuation of 0.55 per click, it will bid 0.03 to beat $A$ and win the first position. This alternating behaviors will continue till $B$ hits the valuation 0.55 but can no longer win the first position. After that, $B$ will again drop to the minimum bid 0.01, resulting in a cyclic trend.

It is straightforward to see that in our setup, dropping all the way to the minimum bid (e.g., 1) is *not* a good strategy under the assumption that companies are rational. Consider a generic case where there are $K$ competing companies. If there exists a company $C_i$ that bids at 1, then assuming no other companies bid the same, it will only get one seed with the $K$-th highest adjusted marginal gain. If $C_i$ can afford more seeds, it will be decisively better off by bidding higher than 1: In the worst case where $C_i$ still has the lowest
budget amongst all companies, it will still get the $K$-th seed, plus another $b_i - 1$ seeds which result in a higher spread.

**Truthful Mechanisms by Borodin et al. [23].** In a recent work, Borodin et al. [23] studied truthful mechanisms for the same problem setting as in this chapter, though they did not look into fairness. When $K = 2$ (two companies), they showed that there exists a randomized allocation algorithm that is truthful. When $K > 3$, there exists an even simpler result: the mechanism that allocates seeds uniformly at random, coupled with the greedy seed selection algorithm, is truthful. More specifically, given $K \geq 3$ companies and their budgets $b_1, b_2, \ldots, b_K$, the host runs the classic greedy algorithm for $B := \sum_{i=1}^{K} b_i$ iterations. In each iteration, it selects the seed that maximizes the marginal gain w.r.t. the spread under the classical LT model, and randomly assigns it to a company whose budget has not been exhausted.

Our experiments in the next section shall show that the uniform random allocation performs substantially poorly w.r.t. fairness, compared to Needy-Greedy. Even though random allocation guarantees that no company can benefit from underreporting budgets, its pure random nature make it unsuitable as a building block for a solid and sustainable business model for social network hosts. This motivates a study of how to design a seed allocation algorithm that is truthful (like the random allocation in [23]), but at the same time achieves a higher level of fairness like Needy-Greedy.

**Remarks on Envy-freeness.** Last but not the least, Example 2 also shows that Needy-Greedy does not always guarantee envy-free outcomes. In the literature studying fair division (such as the classical cake-cutting problem) [41], envy-freeness means that no agent prefers another agent’s allocation than its own. Although the classic cake-cutting mainly focuses on divisible goods, while seeds in viral marketing are not, we can still “port” the definition of envy-freeness to our context. We say that a seed assignment is envy-free if there does not exist two companies $i$ and $j$ such that $b_i > b_j$ but $\sigma_i(S_i, S_{-i}) < \sigma_j(S_j, S_{-j})$. In Example 2, as long as $c < 91/2$, we have $100 + 29c < 191 + 27c$, in which case company $C_1$ would envy company $C_2$ because $b_1 > b_2$. Hence, designing an envy-free seed allocation algorithm and studying its possible connections to a truthful allocation algorithm is also an interesting direction for future research on viral marketing.
3.6 Experiments

3.6.1 Settings
To evaluate our proposed algorithms for FSA, we conducted simulations on four real-world networks—Epinions, Flixster, NetHEPT, and LiveJournal. Table 3.2 presents the statistics of the datasets.

<table>
<thead>
<tr>
<th></th>
<th>Epinions</th>
<th>Flixster</th>
<th>NetHEPT</th>
<th>LiveJournal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of nodes</td>
<td>76K</td>
<td>7.6K</td>
<td>15K</td>
<td>4.8M</td>
</tr>
<tr>
<td>Number of edges</td>
<td>509K</td>
<td>50K</td>
<td>62K</td>
<td>69M</td>
</tr>
<tr>
<td>Average out-degree</td>
<td>13.4</td>
<td>6.5</td>
<td>4.12</td>
<td>28.5</td>
</tr>
</tbody>
</table>

Table 3.2: Datasets statistics

<table>
<thead>
<tr>
<th></th>
<th>Equal budgets</th>
<th>Unequal budgets</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K = 2$</td>
<td>30 each</td>
<td>20, 40</td>
</tr>
<tr>
<td>$K = 3$</td>
<td>20 each</td>
<td>10, 20, 30</td>
</tr>
<tr>
<td>$K = 6$</td>
<td>10 each</td>
<td>5, 5, 10, 15, 15</td>
</tr>
</tbody>
</table>

Table 3.3: Test cases with varying budget distribution

Network Data and Influence Weights. Epinions is a who-trust-whom social network extracted from the consumer review website Epinions.com. If user $v$ trusts the reviews of user $u$, then we drew a directed edge $(u, v)$. We apply the Jaccard model to compute influence weights on edges [68]:

\[ p_{u,v} = \frac{A_{u|v}}{A_{u|v} + A_{v|u}} \]

where $A_{u|v}$ is the number of actions either $u$ or $v$ has performed. After computing these weights, we normalized them to ensure that the sum of incoming weights to each node $v \in V$ is 1.

NetHEPT is a collaboration network from the High Energy Physics Theory section on arXiv.org with nodes representing authors and edges representing co-author relationships. We calculated the weights as $p_{u,v} = \frac{A_{u,v}}{N_v}$ where $A_{u,v}$ is the number of papers $u$ and $v$ co-authored. Flixster is a friendship network from social movie site (www.flixster.com), for which $A_{u,v}$ is the number of movies rated by both $u$ and $v$. In both datasets, $N_v$ is the normalizing factor to ensure the sum of weights incoming to $v$ is 1.

LiveJournal is a directed social network (www.livejournal.com) where users write an online blog, journal, or diary. We calculated the influence weight $p_{u,v}$ as $1/\text{deg}^{\text{in}}(v)$ since there is no action log data available.
Algorithms and Baselines. In the experiments, we evaluated the following algorithms as proposed in Section 3.5: Dynamic Programming (DP), Integer Linear Programming (ILP), and Needy Greedy (NG). Two baselines were also tested: The first one is a simple Round-Robin (RR) allocation strategy: it first fixes a random permutation of the $K$ companies, and then allocates seeds to the companies in a round-robin fashion according to that order. The second one is the random allocation (RA) proposed in [23].

Implementations of all algorithms and baselines are in Python. For Integer Linear Programming we used the Coin-OR Cut-and-Branch (Cbc) solver [56]. For Dynamic Programming, we set the scaling factor to be 10. All experiments were conducted on a Linux server (Redhat Enterprise 6.6) with 12 Intel Xeon CPUs at 2.10GHz each and 64GB RAM.

Competition Settings. We varied $K$ – the number of competing companies – to be 2, 3 and 6. In each case, we further considered two scenarios: equal budgets and unequal budgets. The detailed set-up can be found in Table 3.3.
Later we also increased the problem size for the scalability test on Needy-Greedy – more details shortly.

The selection of the union seed set and the computation of adjusted marginal gains were done by invoking the martingale-based influence maximization algorithm [136] (IMM), which is the state-of-the-art influence maximization approximation algorithms for the LT model. The running time of IMM was within 10 seconds for all datasets, matching the results reported by [136]. Figure 3.3 illustrates the adjusted marginal gain of seeds in decreasing order, for all four datasets.

3.6.2 Results and Analysis

Minimum and maximum amplification factors. Figure 3.4 and Figure 3.5 give comparisons on quality: they depict the minimum and maximum amplification factors, respectively, achieved by various algorithms on all four datasets. Note that on the X-axis, “2eq” refers to the setting of $K = 2$ &
equal budgets, while “neq” refers to the case of unequal budgets. A missing bar means that either the corresponding algorithm failed to complete within a reasonable amount of running time (one week)\(^4\), or the algorithm does not apply (e.g., dynamic programming for \(K > 2\)). These two sets of plots offer a direct comparisons on the quality and effectiveness of algorithms for the FSA problem. For Figure 3.4, the higher the bar, the better; while for Figure 3.5, lower bar is better.

As expected, the Integer Linear Programming method yielded optimal solutions by definition. Dynamic Programming also achieved the same level of performance, which indicates that the loss of precision due to rounding the adjusted marginal gains to integers is negligible. On the other hand, the two baselines, Round-Robin and Random Allocation yielded poor results.

Importantly, Needy-Greedy performed well on these two metrics: In all instances where Needy-Greedy, Dynamic Programming, and Integer Linear

\(^4\)Note that, to put the quantity into perspective to select the union seed set for allocation, the current state-of-the-art algorithm [136] finishes within 10 seconds for all datasets.
Table 3.4: Comparing Needy-Greedy and Integer Linear Programming: For each dataset, we show the largest deviation between Needy-Greedy’s outcome and Integer Linear Programming’s outcome, in percentage, among all instances where Integer Linear Programming finished.

<table>
<thead>
<tr>
<th></th>
<th>Epinions</th>
<th>Flixster</th>
<th>NetHEPT</th>
<th>LiveJournal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deviation on min. AF</td>
<td>0.1%</td>
<td>0.8%</td>
<td>0.4%</td>
<td>0.8%</td>
</tr>
<tr>
<td>Deviation on max. AF</td>
<td>0.05%</td>
<td>0.5%</td>
<td>0.1%</td>
<td>0.4%</td>
</tr>
</tbody>
</table>

Note that both Equation (3.9) and Equation (3.10) are well-defined, as Integer Linear Programming is guaranteed to be optimal while Needy-Greedy is not, and thus the numerators as defined are always nonnegative. As can be seen from Table 3.4, for all cases presented, the deviation percentage is under 1%, and in some case it is as low as 0.05%. This further confirms the strong performance of Needy-Greedy.

**Running time.** Figure 3.7 illustrates the comparison on running time for all algorithms on all datasets. When a bar in the plot touches the top of Y-axis, it means the algorithm did not finish within a reasonable amount of
time, which is often the case of Integer Linear Programming. Needy-Greedy is consistently much more efficient than Dynamic Programming and Integer Linear Programming – it was several orders of magnitude faster. For instance, on NetHEPT when $K = 3$ (with unequal budgets), Needy-Greedy finished in 0.0002 seconds while Integer Linear Programming finished in 33.3 seconds (165,000 times slower). On instances when both Dynamic Programming and Integer Linear Programming finish, they have comparable performance with Integer Linear Programming being slightly faster. However, it is evident that significant scalability issue exists for Integer Linear Programming as it quickly becomes much slower when $K$ increases from 2 to 3 or 6. For instance, on NetHEPT with unequal budget cases, even though the running time of Integer Linear Programming was 0.42 seconds for $K = 2$, 33.3 seconds for $K = 3$, but it failed to finish when $K = 6$. Hence, we conclude that neither Dynamic Programming or Integer Linear Programming is
particularly scalable and practical, while Needy-Greedy is consistently more efficient.

**Scalability Tests on Needy-Greedy.** We ran Needy-Greedy with larger problem instances on the LiveJournal dataset to test its scalability. For simplicity, for all cases described below the budget is equal amongst all companies. In particular, we first varied the total number of seeds to be allocated from 1000 to 5000 and fixed the number of companies to be 10. The results of this test are shown in Figure 3.8(a). We then fixed the total number of seeds to be 5000 and varied the number of companies to be 10 to 50, and the results are depicted in Figure 3.8(b). In both cases, it is evident that Needy-Greedy scaled well as the size of the problem instance increased.

In conclusion, we have demonstrated the effectiveness of our proposed solutions to the Fair Seed Allocation problem: Needy-Greedy, Integer Linear Programming, and Dynamic Programming. In addition, we showed that
Needy-Greedy is also much more efficient and scalable than the other two algorithms. On the other hand, the baselines including the Random Allocation mechanism of [23] yielded much inferior results w.r.t. fairness.

3.7 Discussion and Future Work

Investigations of competitive influence propagation and fair seed allocation under other diffusion models (e.g., in which the virality of different products may vary) are worthy future work. Note that for our $K$-LT model, we are able to compute the adjusted marginal gains and use them as the input to the fair seed allocation problem. This result is specific to $K$-LT, and thus may not be easily replicated under other models such as the competitive IC models [35]. It is also interesting to further investigate whether other truthful mechanism exists besides the uniform random allocation [23] (which performs poorly in terms of fairness).

An important issue that arises here is data privacy and security in the viral marketing setting described in this chapter. Note that for the host to accurately select and allocate seeds, it needs access to user action logs to compute pairwise user influence strength in the social network [68, 129]. If the user actions are restricted to those on the social networking website itself, such as clicking a link, liking or commenting on a post, and sharing a video, the host will have direct knowledge from its own tracking data. However, when the actions are product adoptions, note that users’ purchase histories are more likely to be stored externally, e.g., in the participating companies’ own databases or the databases of third-party stores such as Amazon or eBay. For the host to estimate influence strength, one possible solution to use its own tracking data as proxies. An (arguably) better alternative is to
establish a protocol with the participating companies and use their ground-truth historical sales data. However, the challenge is that neither the host nor the companies want their proprietary data to be disclosed and leaked, so there is a need to protect privacy on both ends. To solve this problem, Tassa and Bonchi [138] proposed a privacy-preserving protocol for the host and the participating companies to jointly compute social influence strength. Their method is certainly appealing and useful, but it do not account for purchases in third-party stores, and hence the resultant estimations may be biased. Exploring possible improvements to eliminate such bias (if any) would be an interesting albeit quite challenging future work (e.g., how to collect data and how to ensure privacy guarantees when more parties are involved?).

More generally, privacy-preserving influence maximization is itself an important, interesting, yet often overlooked problem, in spite of a large body of work on privacy-preserving social network mining [9, 103, 153]. For example, supposes the host wishes to publishes (part, or all of) its social network data for purposes such as research or outsourcing\(^5\), the data must be published in a privacy-preserving manner such that (i) user privacy (identity, connections, interests, etc) can be safeguarded from adversaries and (ii) the published data still has sufficient utility for data mining applications. It is interesting to investigate whether well-established privacy models, such as \(k\)-anonymity by Zhou et al. [153] and \(k\)-degree-anonymity by Liu and Terzi [103] are suitable, and how influence maximization algorithms can be devised or adapted to achieve high-quality solutions.

\(^5\)For example, in the 2014 Economic Graph Challenge held by LinkedIn, the world’s largest professional social network company invited participations from US academic institutions to propose data mining and data analytics problems and solutions that would make good and novel use of LinkedIn’s data.
Chapter 4

Comparative Influence
Diffusion and Maximization

4.1 Introduction

Most existing work in computational social influence focuses on two types of diffusion models — *single-entity models* and *pure-competition models*. A single-entity model has only one propagating entity for social network users to adopt: the classic Independent Cascade (IC) and Linear Thresholds (LT) models [86] belong to this category. These models, however, ignore complex social interactions involving multiple propagating entities. Considerable work has been done to extend IC and LT models to study competitive influence maximization, but almost all models assume that the propagating entities are in pure competition and users adopt at most one of them [18, 23, 24, 29, 31, 34, 79, 106, 123].

In reality, the relationship between different propagating entities is certainly more general than pure competition. In fact, consumer theories in economics have two well-known notions: *substitute goods* and *complementary goods* [113, 133]. Substitute goods are those that can be used for the same purpose and purchased one in place of the other, e.g., smartphones of various brands. Complementary goods are those that tend to be purchased together, e.g., iPhone and its accessories, computer hardware and software, etc. There are also varying *degrees* of substitutability and complementarity: buying a product could lessen the probability of buying the other without necessarily eliminating it; similarly, buying a product could boost the probability
of buying another to any degree. Pure competition only corresponds to the special case of perfect substitute goods.

The limitation of pure-competition models can be exposed by the following example. Consider a viral marketing campaign featuring iPhone 6 and Apple Watch. It is vital to recognize the fact that Apple Watch generally needs an iPhone to be usable, and iPhone’s user experience can be greatly enhanced by a pairing Apple Watch (see, e.g., http://bit.ly/1GOqesc). Clearly none of the pure-competition models is suitable for this campaign because they do not even allow users to adopt both the phone and the watch!

This motivates us to design a more powerful, expressive, yet reasonably tractable model that captures not only competition, but also complementarity, and any possible degrees associated with these notions. To this end, we propose the \textit{Comparative Independent Cascade} model, or ComIC for short, which, unlike most existing diffusion models, consists of two critical components that work jointly to govern the dynamics of diffusions:

- Edge-level information propagation: This is similar to the social influence propagation dynamics captured by the classical IC and LT models, but only controls information awareness.

- Node-level decision-making: The model features a \textit{Node-Level Automaton} (NLA) that ultimately makes adoption decisions based on a set of model parameters known as the \textit{Global Adoption Probabilities} (GAPs).

The NLA is a novel feature and is unique in our model. Indeed, the term “comparative” comes from the fact that once a user is aware, via edge-level propagation, of multiple products, intuitively she makes a comparison between them by “running” her NLA. Notice that “comparative” subsumes “competitive” and “complementary” as special cases. In theory, the ComIC model is able to accommodate any number of propagating entities (items) and cover the entire spectrum from competition to complementarity between pairs of items, reflected by the values of GAPs.

In this work, as the first step toward comparative influence diffusion and viral marketing, we focus on the case of two items. At any time, w.r.t. any item \( A \), a user in the network is in one of the following four states: \( A \)-idle, \( A \)-suspended, \( A \)-rejected, or \( A \)-adopted. The NLA sets out probabilistic transition rules between states, and different GAPs are applied based on a given user’s state w.r.t. the other item \( B \) and the relationship between \( A \) and \( B \). Intuitively, competition (complementarity) is modeled as reduced probability (resp., increased probability) of adopting the second item after the first item is already adopted. After a user adopts an item, she propagates
this information to her neighbors in the network, making them aware of the item. The neighbor may adopt the item with a certain probability, as governed by her NLA.

We then study the influence maximization problem in the context of ComIC model for two complementary items $A$ and $B$. Specifically, here it asks for $k$ seeds for $A$ such that given a fixed set of $B$-seeds, the expected number of $A$-adopted nodes is maximized. To the best of our knowledge, we are the first to systematically study influence maximization for complementary items. We show that the problem is NP-hard under the ComIC model. Moreover, two important properties of set functions, submodularity and monotonicity, which would allow a greedy approximation algorithm frequently used in discrete optimization problems, do not hold in unrestricted ComIC model (where there are no constraints on the GAPs). Even when we restrict ComIC to mutual complementarity, submodularity still does not hold in general.

To circumvent the aforementioned difficulties, we first show that submodularity holds for a subset of the complementary parameter space. We then make a non-trivial extension to the Reverse-Reachable Set (RR-set) techniques [22,136,137], originally proposed for influence maximization with single-entity models, to obtain effective and efficient approximation solutions to Influence Maximization. Next, we propose a novel Sandwich Approximation (SA) strategy which, for a given non-submodular set function, provides an upper bound function and/or a lower bound function, and uses them to obtain data-dependent approximation solutions w.r.t. the original function.

We further note that both techniques are applicable to a larger context beyond the model and problems studied in this paper: for RR-sets, we provide a new definition and general sufficient conditions not covered by [22,136,137] that apply to a large family of influence diffusion models, while SA applies to the maximization of any non-submodular functions that are upper- and/or lower-bounded by submodular functions.

In the experiments, we first learn GAPs from user action logs from two social networking sites — Flixster.com and Douban.com. We demonstrate that our approximation algorithms based on RR-sets and SA techniques consistently outperform several intuitive baselines, typically by a significant margin on real-world networks.

To summarize, we make the following contributions:

- We propose the ComIC model to characterize influence diffusion dynamics of products with arbitrary degree of competition or complementarity (Section 4.3).
Table 4.1: Frequently used acronyms.

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ComIC</td>
<td>Comparative Independent Cascade</td>
</tr>
<tr>
<td>GAP</td>
<td>Global Adoption Probability</td>
</tr>
<tr>
<td>NLA</td>
<td>Node-Level Automation</td>
</tr>
<tr>
<td>RR-set</td>
<td>Reverse-Reachable Set</td>
</tr>
<tr>
<td>SA</td>
<td>Sandwich Approximation</td>
</tr>
<tr>
<td>TIM</td>
<td>Two-Phase Influence Maximization</td>
</tr>
</tbody>
</table>

- We identify a subset of the parameter space under which submodularity and monotonicity of influence spread hold, paving the way for designing approximation algorithms (Section 4.5).

- We study Influence Maximization for complementary products under the ComIC model (Section 4.4). The problem remains NP-hard, and thus we devise efficient and effective approximation solutions by non-trivial extensions to RR-set techniques and by proposing Sandwich Approximation, both having applicability beyond this work (Section 4.6).

- We conduct empirical evaluations on four real-world social networks and demonstrate the superiority of our algorithms over intuitive baselines (Section 4.8).

- We also propose a methodology for learning global adoption probabilities for the ComIC model from user action logs of social networking sites (Section 4.8).

Table 4.1 summarizes frequently used acronyms in this chapter.

4.2 Related Work

Datta et al. [43] studied influence maximization with items whose propagations are independent. Narayanam et al. [118] studied a setting with two sets of products, where a product can be adopted by a node only when it has already adopted a corresponding product in the other set. Their model extends LT. In Section 4, we depart by defining a significantly more powerful and expressive model in ComIC, compared to theirs which only covers the special case of perfect complementarity. Our technical contributions for addressing the unique challenges posed by ComIC are substantially different from [118].
Meyers and Leskovec [117] analyzed Twitter data to study the effect of different cascades on users and predicted the likelihood of a user adopting a piece of information (e.g., URLs in tweets) given cascades that the user was previously exposed to. McAuley et al. [114] used logistic regression to learn substitute or complementary relationships between products from user reviews. Both studies primarily focus on data analysis and behavior prediction, instead of providing diffusion modeling for competing and complementary items, nor do they study the influence maximization problem in this context.

Substitutability and complementarity have been studied in other computer science subfields, such as theory. A representative example is the allocation problem in combinatorial auctions: Given a set of items and each bidder’s valuation function, the task is to allocate items to bidders so that the total utility of all bidders (i.e., the social welfare) is maximized. The greedy algorithm can be naturally applied: Enumerate the items in an arbitrary order, and each item is allocated to the bidder with the largest marginal valuation under the current allocation. When all valuation functions are submodular and a value oracle is assumed\(^1\), Lehmann et al. showed that the greedy algorithm achieves 1/2-approximation [97]. Vondrak [142] proposed a continuous greedy algorithm with \((1 - 1/e)\) approximation factor. However, maximizing social welfare is much more difficult when there exists complementarity. In a recent work, Abraham et al. [2] proposed a hypergraph-based model to succinctly represent valuations with complements and designed approximation algorithms under that model.

Although the ComIC model also characterizes complementarity and substitutability between propagating entities in social influence diffusions, fundamental differences exist between social welfare maximization in combinatorial auctions and influence maximization under ComIC. More specifically, the substitute or complementary relationship is present amongst items to be allocated, which are not subject to any dynamic propagation. Neither the items, nor the agents that they are allocated to are involved in any recursive propagation. In contrast, in our setting (influence maximization), while seeds need to be allocated to different products, this allocation needs to factor in the effect of stochastic, recursive propagation of those products through a

\(^1\)The value oracle directly returns the value of the valuation function on any given subset of items. For example, suppose the function is denoted by \(v\) and the set is denoted by \(S\), then the value oracle returns \(v(S)\) when queried with \(S\). More powerful oracles such as the demand oracle and general oracle have also been studied in the combinatorial auction literature, see, e.g., [49].
network as guided by the stochastic propagation rules under the ComIC model.

4.3 The Comparative Independent Cascade Model

We start this section by highlight the essential ideas of our new model. In the Comparative IC (ComIC) model, there are at least two propagating entities (products, technologies, information pieces, opinions, etc). For ease of exposition, we focus on the case of two products, denoted by $\mathcal{A}$ and $\mathcal{B}$ respectively. The diffusion dynamics unfold in discrete time steps $0, 1, \ldots$ Nodes (users) in the social network can be in any of the following states: \{idle, suspended, adopted, rejected\} w.r.t. each of the products. Initially, all nodes in the joint state of ($\mathcal{A}$-idle, $\mathcal{B}$-idle).

One of the biggest differences between ComIC and IC is the separation of information diffusion (edge-level) and the actual adoption decisions (node-level). Edges only control the information that flows to a node: e.g., when $u$ adopts a product, its out-neighbor $v$ may be informed of this fact. Once that happens, $v$ uses its own Node-Level Automaton (NLA) to decide which state to transit to. This depends on $v$’s current state w.r.t. the two products as well as parameters corresponding to the state transition probabilities of the NLA, namely the Global Adoption Probabilities, defined below.

A concise representation of the NLA is shown in Figure 4.1. Each state is indicated by the label. E.g., with probability $q_{\mathcal{A}|\emptyset}$, a node transits from a state where it’s $\mathcal{A}$-idle to $\mathcal{A}$-adopted, regardless of whether it was $\mathcal{B}$-idle or $\mathcal{B}$-suspended. From the $\mathcal{A}$-suspended state, it transits to $\mathcal{A}$-adopted w.p. $\rho_{\mathcal{A}}$ and to $\mathcal{A}$-rejected w.p. $1 - \rho_{\mathcal{A}}$. The probability $\rho_{\mathcal{A}}$, called reconsideration probability, as well as the reconsideration process will be explained below.

Note that in a ComIC diffusion process defined in Section 4.3.3, not all joint state is reachable from the initial ($\mathcal{A}$-idle, $\mathcal{B}$-idle) state, e.g., ($\mathcal{A}$-idle, $\mathcal{B}$-id...
rejected). Since all unreachable states are irrelevant to adoptions, they can be safely ignored. We shall list and prove all unreachable states in Section 4.3.2.

4.3.1 Global Adoption Probability (GAP)

The Global Adoption Probabilities, consisting of four parameters $Q = (q_A|\emptyset, q_A|B, q_B|\emptyset, q_B|A) \in [0,1]^4$, are important parameters of the NLA which decide the likelihood of adoptions after a user is informed of an item. Formally,

- $q_A|\emptyset :=$ the probability that a user adopts $A$ given that she is $A$-informed but not $B$-adopted;
- $q_A|B :=$ the probability that a user adopts $A$ given that she is $B$-adopted;
- $q_B|\emptyset :=$ the probability that a user adopts $B$ given that she is $B$-informed but not $A$-adopted;
- $q_B|A :=$ the probability that a user adopts $A$ given that she is $B$-adopted.

Intuitively, GAPs reflect the overall popularity of products and how they are perceived by the entire market. They are considered aggregate estimates and hence are not user specific in our model. In other words, users are assumed to be homogeneous in terms of preferences on consuming the two items. We shall provide further justifications at the end of this section and describe a way to learn GAPs from user action log data in Section 4.8. In addition, we shall discuss in Section 4.9 how the ComIC model can be extended to accommodate multiple types of users.

GAPs enable ComIC to characterize competition and complementarity to arbitrary extent. We say that $A$ competes with $B$ iff

$$q_B|A \leq q_B|\emptyset.$$  

Similarly, $A$ complements $B$ iff

$$q_B|A \geq q_B|\emptyset.$$  

We include the special case of $q_B|A = q_B|\emptyset$ in both cases above for convenience of stating our technical results, and it actually means that the propagation of $B$ is completely independent of $A$ (cf. Lemma 5). Competition and complementarity in the other direction are similar.
The degree of competition and complementarity is determined by the difference between the two relevant GAPs, i.e., \(|q_{B|A} - q_{B|0}|\) and \(|q_{A|B} - q_{A|0}|\). For convenience, we use \(Q^+\) to refer to an arbitrary set of GAPs representing mutual complementarity:

\[(q_{A|0} \leq q_{A|B}) \land (q_{B|0} \leq q_{B|A}),\]

and similarly, \(Q^-\) for an arbitrary set of GAPs representing mutual competition:

\[(q_{A|0} \geq q_{A|B}) \land (q_{B|0} \geq q_{B|A}).\]

### 4.3.2 Unreachable States of ComIC Model

Before an influence diffusion starts under the ComIC model, all nodes are in the initial joint state (\(A\)-idle, \(B\)-idle). According to the diffusion dynamics defined in Figure 4.2, there exist five unreachable joint states, which are not material to our analysis and problem-solving, since none of these is relevant to actual adoptions, the objectives studied in Influence Maximization. For completeness, we list these states here.

1. \((A\text{-idle}, B\text{-rejected})\)
2. \((A\text{-suspended}, B\text{-rejected})\)
3. \((A\text{-rejected}, B\text{-idle})\)
4. \((A\text{-rejected}, B\text{-suspended})\)
5. \((A\text{-rejected}, B\text{-rejected})\)

**Lemma 1.** In any instance of the ComIC model (no restriction on GAPs), no node can reach the state of \((A\text{-idle}, B\text{-rejected})\), from its initial state of \((A\text{-idle}, B\text{-idle})\).

**Proof.** Let \(v\) be an arbitrary node from graph \(G = (V, E, p)\). Note that for \(v\) to reject \(B\), it must be first be informed of \(B\) (otherwise it remains \(B\)-idle, regardless of its state w.r.t. \(A\)), and then becomes \(B\)-suspended (otherwise it will be \(B\)-adopted, a contradiction). Now, note that \(v\) is never informed of \(A\), and hence it will not be triggered to reconsider \(B\), the only route to the state of \(B\)-rejected, according to the model definition. Thus, \((A\text{-idle}, B\text{-rejected})\) is unreachable. \(\square\)
The argument for (A-rejected, B-idle) being unreachable is symmetric, and hence omitted.

**Lemma 2.** In any instance of the ComIC model (no restriction on GAPs), no node can reach the state of (A-suspended, B-rejected), from its initial state of (A-idle, B-idle).

**Proof.** Let \( v \) be an arbitrary node from graph \( G = (V, E, p) \). Note that for \( v \) to reject \( B \), it must be first be informed of \( B \) (otherwise it remains \( B \)-idle, regardless of its state w.r.t. \( A \)), and then becomes \( B \)-suspended (otherwise it will be \( B \)-adopted, a contradiction). Now, \( v \) transits from \( A \)-idle to \( A \)-suspended, meaning that \( v \) does not adopt \( A \). This will not further trigger reconsideration, and hence \( v \) stays at \( B \)-suspended. \( \square \)

The argument for (A-rejected, A-suspended) being unreachable is symmetric, and hence omitted. Finally, it is evident from the proof of Lemma 2 that, the joint state of (A-suspended, A-suspended) is a sunken state, meaning the node will not get out it to adopt or reject any product. This implies that (A-rejected, B-rejected) is also unreachable.

### 4.3.3 Diffusion Dynamics in the ComIC Model

Let \( G = (V, E, p) \) be a directed social graph with pairwise influence probabilities. Let \( S_A, S_B \subset V \) be the seed sets for \( A \) and \( B \). Influence diffusion under ComIC proceeds in discrete time steps. Initially, every node is \( A \)-idle and \( B \)-idle.

At time step 0, every \( u \in S_A \) becomes \( A \)-adopted and every \( u \in S_B \) becomes \( B \)-adopted. No generality is lost in assuming seeds adopt an item without testing the NLA: for every \( v \in V \), we can create two dummy nodes \( v_A, v_B \) and edges \((v_A, v)\) and \((v_B, v)\) with \( p_{v_A,v} = p_{v_B,v} = 1 \). Requiring seeds to go through NLA is equivalent to constraining that \( A \)-seeds (/\( B \)-seeds) be selected from all \( v_A \)’s (resp. \( v_B \)’s). If \( u \in S_A \cap S_B \), we randomly decide the order of \( u \) adopting \( A \) and \( B \) with a fair coin. For ease of understanding, we describe the rest of the diffusion process in a modular way in Figure 4.2. We use \( N^+(v) \) and \( N^-(v) \) to denote the set of out-neighbors and in-neighbors of \( v \), respectively.

We draw special attention to tie-breaking and reconsideration. Tie-breaking is used when a node’s in-neighbors adopt different products and try to inform the node at the same step. Node reconsideration concerns the situation that a node \( v \) did not adopt \( A \) initially but later after adopting \( B \) it may reconsider adopting \( A \): when \( B \) competes with \( A \) (\( q_{A|\emptyset} \geq q_{A|B} \)),
Global iteration. At every time step $t \geq 1$, for all nodes that became $A$- or $B$-adopted at $t - 1$, their outgoing edges are tested for transition (1 below). After that, for each node $v$ that has at least one in-neighbor (with a live edge) becoming $A$- and/or $B$-adopted at $t - 1$, $v$ is tested for possible state transition (rules 2–4 below).

1. Edge transition. For an untested edge $(u, v)$, flip a biased coin independently: $(u, v)$ is live w.p. $p_{u,v}$ and blocked w.p. $1 - p_{u,v}$. Each edge is tested at most once in the entire diffusion process.

2. Node tie-breaking. Consider a node $v$ to be tested at time $t$. Generate a random permutation $\pi$ of $v$’s in-neighbors (with live edges) that adopted at least one product at $t - 1$. Then, test $v$ with each such in-neighbor $u$ and $u$’s adopted item ($A$ and/or $B$) following $\pi$. If there is a $w \in N^-(v)$ adopting both $A$ and $B$, then test both products, following their order of adoption by $w$.

3. Node adoption. Consider the case of testing an $A$-idle node $v$ for adopting $A$ (Figure 4.1). If $v$ is not $B$-adopted, then w.p. $q_{A|\emptyset}$, it becomes $A$-adopted and w.p. $1 - q_{A|\emptyset}$ it becomes $A$-suspended. If $v$ is $B$-adopted, then w.p. $q_{A|B}$, it becomes $A$-adopted and w.p. $1 - q_{A|B}$ it becomes $A$-rejected. The case of adopting $B$ is symmetric.

4. Node reconsideration. Consider an $A$-suspended node $v$ that just adopts $B$ at time $t$. Define

\[ \rho_A = \text{def} \frac{\max\{q_{A|B} - q_{A|\emptyset}, 0\}}{1 - q_{A|\emptyset}}. \]  

(4.1)

Then, $v$ reconsiders to become $A$-adopted w.p. $\rho_A$, or $A$-rejected w.p. $1 - \rho_A$. The case of reconsidering $B$ is symmetric, and the reconsideration probability of $B$ can be similarly defined:

\[ \rho_B = \text{def} \frac{\max\{q_{B|A} - q_{B|\emptyset}, 0\}}{1 - q_{B|\emptyset}}. \]  

Figure 4.2: ComIC model: Diffusion dynamics

83
$v$ will not reconsider adopting $A$, but when $B$ complements $A$ (specifically, $q_{A|B} < q_{A|B}$), $v$ will reconsider adopting $A$. In the latter case, the probability of adopting $A$, $\rho_A$, is defined in such a way that the overall probability of adopting $A$ is equal to $q_{A|B}$. That is,

$$q_{A|B} = q_{A|\emptyset} + (1 - q_{A|\emptyset}) \cdot \rho_A,$$

where $\rho_A$ is defined as in Equation (4.1).

### 4.3.4 Design Considerations

The design of ComIC not only draws on the essential elements from a classical diffusion model (IC) proposed in mathematical sociology, but also closes a gap between theory and practice, in which diffusions typically do not occur just for one product or with just one mode of pure competition.

With GAPs in the NLA, the model can characterize any possible relationship between two propagating entities: competition, complementarity, and any degree associated with them. GAPs are fully capable of handling asymmetric relationship between products. Furthermore, introducing NLA with GAPs and separating the propagation of product information from actual adoptions reflects Kalish’s famous characterization of new product adoption [85]: customers go through two stages – awareness followed by actual adoption. In Kalish’s theory, product awareness is propagated through word-of-mouth effects; after an individual becomes aware, she would decide whether to adopt the item based on other considerations. Edges in the network can be seen as information channels from one user to another. Once the channel is open (live), it remains so. This modeling choice is reasonable as competitive goods are typically of the same kind and complementary goods tend to be adopted together.

We remark that ComIC encompasses previously-studied single-entity and pure-competition models as special cases. When $q_{A|\emptyset} = q_{B|\emptyset} = 1$ and $q_{A|B} = q_{B|A} = 0$, ComIC reduces to the (purely) Competitive Independent Cascade model [35]. If, in addition, $q_{B|\emptyset}$ is 0, the model further reduces to the classic IC model.

### 4.3.5 An Equivalent Possible World Model

To facilitate a better understanding of the ComIC model and our submodularity analysis in (Section 4.5), we now describe a Possible World (PW)
model that provides an equivalent view of influence diffusion processes under the ComIC model.

Given a graph $G = (V, E, p)$ and a diffusion model, a possible world consists of a deterministic graph sampled from a probability distribution over all subgraphs of $G$. For ComIC, we also need some variables for each node to fix the outcomes of random events in relation to the NLA (i.e., adoption, tie-breaking, and reconsideration), so that influence cascade is fully deterministic in a single possible world.

4.3.5.1 Definition of the Possible World Model

**Generative rules.** Let $W$ be any possible world. To generate such $W$, we first process edges: Retain each edge $(u, v) \in E$ with probability $p_{u,v}$ (live edge) and drop it with probability $1 - p_{u,v}$ (blocked edge). This generates a deterministic graph $G_W = (V, E_W)$ where $E_W$ is the set of all live edges.

Next we process nodes. For all $v \in V$:

1. Choose “thresholds” $\alpha^{v,W}_A$ and $\alpha^{v,W}_B$ independently and uniformly at random from the interval $[0, 1]$. These two values are used for comparison with GAPs in adoption decisions. When the possible world $W$ is clear from context, we write $\alpha^v_A$ and $\alpha^v_B$ for simplicity;

2. Generate a random permutation $\pi_v$ of all in-neighbors $u \in N^-(v)$. This is for tie-breaking;

3. Sample a discrete value $\tau_v \in \{A, B\}$, where each value has a probability of 0.5. This is used for tie-breaking in case $v$ is a seed of both $A$ and $B$.

**Deterministic diffusions in a possible world.** At time step 0, the $A$-seeds in $S_A$ become $A$-adopted and the $B$-seeds in $S_B$ become $B$-adopted (ties, if any, are broken based on $\tau_v$).

Then, iteratively for each time step $t \geq 1$, we say that a node $v$ is reachable by $A$ at time step $t$ if $t$ is the length of a shortest path from any seed $u \in S_A$ to $v$ consisting entirely of live edges and $A$-adopted nodes. Node $v$ then becomes $A$-adopted at step $t$ if $\alpha^v_A \leq x$, where

$$x = \begin{cases} q_A(0), & \text{if } v \text{ is not } B\text{-adopted}; \\ q_A|B, & \text{if } v \text{ is already } B\text{-adopted}. \end{cases}$$

For re-consideration, suppose $v$ just becomes $B$-adopted at time step $t$ and it is $A$-suspended (i.e., $v$ became reachable by $A$ before $t$ time steps
but $\alpha^v_A > q_{A|\emptyset}$. Then, $v$ adopts $A$ if $\alpha^v_A \leq q_{A|B}$. The reachability and reconsideration tests for item $B$ are symmetric.

For tie-breaking, if $v$ is reached by both $A$ and $B$ at time step $t$, the permutation $\pi_v$ is used to determine the order in which $A$ and $B$ are considered. In addition, if $v$ is reached by $A$ and $B$ from the same in-neighbor, e.g., $u$, then the order in which $v$ is informed of $A$ and $B$ is the same as the order in which $u$ itself adopted $A$ and $B$.

4.3.5.2 Equivalence to ComIC

The following lemma establishes the equivalence between the possible world model defined above and the ComIC mode, from the standpoint of influence diffusion and the probability distribution of adopted nodes. This allows us to analyze model properties such as monotonicity and submodularity (Section 4.5) using the PW model, which tends to be more convenient technically.

Lemma 3. For any fixed $A$-seed set $S_A$ and fixed $B$-seed set $S_B$, the joint distributions of the sets of $A$-adopted nodes and $B$-adopted nodes obtained by (i) running a ComIC diffusion from $S_A$ and $S_B$ and (ii) randomly sampling a possible world $W$ and running a deterministic cascade from $S_A$ and $S_B$ in $W$, are the same.

Proof. The proof is based on establishing equivalence on all edge-level and node-level activities in these two models.

By the principle of deferred decisions and the fact that each edge is only tested once in one diffusion, edge transition processes are equivalent. To generate a possible world, the live/blocked status of an edge is pre-determined and revealed when needed, while in a Com-IC process, the status is determined on-the-fly.

Tie-breaking is also equivalent. Note that each node $v$ only needs to apply the random permutation $\pi_v$ for breaking ties at most once. For ComIC, we need to apply $\pi_v$ only when $v$ is transitioning out of state ($A$-idle, $B$-idle) after being informed of both $A$ and $B$. Clearly, this transition occurs at most once for each node. The same logic applies to the PW model. Thus, the equivalence is obvious due to the principle of deferred decisions.

The equivalence of decision-making for adoption is straightforward as $\alpha^v_A$ and $\alpha^v_B$ are chosen uniformly at random from $[0, 1]$. Hence, $\Pr[\alpha^v_A \leq q] = q$, where $q \in \{q_{A|\emptyset}, q_{A|B}, q_{B|\emptyset}, q_{B|A}\}$.

As to reconsideration, w.l.o.g. we consider $A$. In ComIC, the probability of reconsideration is $\rho_A = \max\{(q_{A|B} - q_{A|\emptyset}), 0\}/(1 - q_{A|\emptyset})$. In PW, when $q_{A|B} \geq q_{A|\emptyset}$, this amounts to the probability that $\alpha^v_A \leq q_{A|B}$ given $\alpha^v_A > q_{A|\emptyset}$.
which is \((q_{A|B} - q_{A|\emptyset})/(1 - q_{A|\emptyset})\). On the other hand, when \(q_{A|B} < q_{A|\emptyset}\), \(\alpha_A^v > q_{A|\emptyset}\) implies \(\alpha_A^v > q_{A|B}\), which means reconsideration is meaningless, and this corresponds to \(\rho_A = 0\) in ComIC. Thus, the equivalence is established.

Finally, the seeding protocol is trivially the same. Combining the equivalence for all edge-level and node-level activities, we can see that the two models are equivalent and yield the same distribution of \(A\)- and \(B\)-adopted nodes, for any given \(S_A\) and \(S_B\).

Note that since \(\alpha_A^v\)’s and \(\alpha_B^v\)’s are real values in the interval \([0, 1]\), theoretically the number of valid possible worlds can be infinite. However, from the perspective of influence diffusion (i.e., state transitions of all nodes), the number of all “effective” possible worlds is still finite. To see why, notice that instead of the exact values, it is the interval in which \(\alpha_A^v\) or \(\alpha_B^v\) falls into that ultimately decides the outcomes of the random events regarding adoption.

For \(\alpha_A^v\), there are three possibilities: \([0, q_{A|\emptyset})\), \([q_{A|\emptyset}, q_{A|B})\), and \([q_{A|B}, 1]\), and the same applies to \(\alpha_B\). Thus, consider two possible worlds \(W_1\) and \(W_2\) where everything else is the same but \(\alpha_{A,W_1}^v \neq \alpha_{A,W_2}^v\) and \(\alpha_{B,W_1}^v \neq \alpha_{B,W_2}^v\), for some \(v \in V\). If, for instance, \(\alpha_{A,W_1}^v, \alpha_{A,W_2}^v \in [0, q_{A|\emptyset})\) and \(\alpha_{B,W_1}^v, \alpha_{B,W_2}^v \in [q_{B|A}, 1]\), then the diffusion dynamics in \(W_1\) and \(W_2\) will still be the same for any fixed seed sets.

This observation would be particular useful for showing submodularity, since it suffices to show that submodularity holds in an arbitrary possible world (see, e.g., the proof of Theorem 8 in Section 4.5).

### 4.4 Influence Maximization with Complementary Goods

Many interesting optimization problems can be formulated thanks to the expressiveness of ComIC model. In this work, we focus on solving influence maximization in a novel context, where two propagating entities are complementary\(^2\).

Given the seed sets \(S_A\) and \(S_B\), we define \(\sigma_A(S_A, S_B)\) to be the expected number of \(A\)-adopted nodes. Similarly, let \(\sigma_B(S_A, S_B)\) denote the expected number of \(B\)-adopted nodes. We can see that both \(\sigma_A\) and \(\sigma_B\) are real-valued bi-set functions mapping \(2^V \times 2^V\) to \([0, |V|]\), for any fixed \(Q\).

Unless otherwise noted, GAPs are not considered as arguments to these two influence spread functions as \(Q\) is constant in a given instance of ComIC.

---

\(^2\)Recall that competitive viral marketing has been studied extensively in the literature (see Section 3.2 and Section 4.2).
Following conventions, these two functions are called influence spread functions. For simplicity, we will refer to $\sigma_\mathcal{A}(\cdot, \cdot)$ as “$\mathcal{A}$-spread” and $\sigma_\mathcal{B}(\cdot, \cdot)$ as “$\mathcal{B}$-spread”.

Without loss of generality we define the influence maximization problem in terms of $\mathcal{A}$-spread as follows.

**Problem 4** (Influence Maximization under ComIC). Given a directed graph $G = (V, E, p)$ with pairwise influence probabilities, $\mathcal{B}$-seed set $S_\mathcal{B} \subset V$, a cardinality constraint $k$, and a set of GAPs $\mathcal{Q}^+$ representing mutual complementarity, find an $\mathcal{A}$-seed set $S_\mathcal{A}^* \subset V$ of size $k$, such that the expected number of $\mathcal{A}$-adopted nodes is maximized under ComIC:

$$S_\mathcal{A}^* \in \arg\max_{T \subseteq V, |T| = k} \sigma_\mathcal{A}(T, S_\mathcal{B}).$$

Influence Maximization under ComIC is obviously NP-hard, as it subsumes the original problem under the classic IC model when $S_\mathcal{B} = \emptyset$ and $q_{\mathcal{A}|\emptyset} = q_{\mathcal{A}|\mathcal{B}} = 1$.

### 4.5 Properties of ComIC Model

Since Influence Maximization cannot be solved in polynomial time unless $P = NP$, we now study submodularity and monotonicity for the ComIC model which will pave the way for designing approximation algorithms. Without loss of generality, we focus on $\sigma_\mathcal{A}$ only. Note that this influence spread function is a bi-set function taking arguments $S_\mathcal{A}$ and $S_\mathcal{B}$, so submodularity and monotonicity can be defined w.r.t. each of the two arguments. For the purpose of Influence Maximization, it suffices to study these two properties w.r.t. $S_\mathcal{A}$ only.

#### 4.5.1 Monotonicity

It turns out that if $\mathcal{A}$ competes with $\mathcal{B}$, but $\mathcal{B}$ complements $\mathcal{A}$, monotonicity does not hold in general, as shown in the following counter-example.

![Figure 4.3: The graph for Example 3](image-url)
Example 3 (Non-Monotonicity). Consider the graph in Figure 4.3. All edges have probability 1. GAPs are \( q_{A|\emptyset} = q \in (0,1), q_{A|B} = q_{B|\emptyset} = 1, q_{B|A} = 0 \), which means that \( A \) competes with \( B \) but \( B \) complements \( A \). Let \( S_B = \{y\} \). If \( S_A \) is \( S = \{s_1\} \), the probability that \( v \) becomes \( A \)-adopted is 1, because \( v \) is informed of \( A \) from \( s_1 \), and even if it does not adopt \( A \) at the time, later it will surely adopt \( B \) propagated from \( y \), and then \( v \) will reconsider \( A \) and adopt \( A \). If it is \( T = \{s_1,s_2\} \), that probability is \( 1 - q + q^2 < 1 \): \( w \) gets \( A \)-adopted w.p. \( q \) blocking \( B \) and then \( v \) gets \( A \)-adopted w.p. \( q \); \( w \) gets \( B \)-adopted w.p. \( (1-q) \) and then \( v \) surely gets \( A \)-adopted. Replicating sufficiently many \( v \)'s, all connected to \( s_1 \) and \( w \), will lead to \( \sigma_A(T,S_B) < \sigma_A(S,S_B) \). The intuition is that the additional \( A \)-seed \( s_2 \) “blocks” \( B \)-propagation as \( A \) competes with \( B \) \((q_{B|A} < q_{B|\emptyset})\) but \( B \) complements \( A \) \((q_{A|B} > q_{A|\emptyset})\). Clearly \( \sigma_A \) is not monotonically decreasing in \( S_A \) either (e.g., in a graph when all nodes are isolated). Hence, \( \sigma_A \) is not monotone in \( S_A \). \( \square \)

The counter-example above has \( A \) competing with \( B \), but \( B \) complementing \( A \), which is unnatural. Hence, we now focus on mutual competition \((Q^-)\) and mutual complementary cases \((Q^+)\), and show that monotonicity is satisfied in these settings.

Theorem 7. For any fixed \( B \)-seed set \( S_B \), the influence spread function of \( A - \sigma_A(S_A,S_B) - \) is monotonically increasing in \( S_A \) for any set of GAPs in \( Q^+ \) and \( Q^- \). Also, \( \sigma_A(S_A,S_B) \) is monotonically increasing in \( S_B \) for any GAPs in \( Q^+ \), and monotonically decreasing in \( S_B \) for any \( Q^- \).

For ease of exposition, we also state a symmetric version of Theorem 7 w.r.t. \( \sigma_B \). That is, given any fixed \( A \)-seed set, \( \sigma_B(S_A,S_B) \) is monotonically increasing in \( S_B \) for any set of GAPs in \( Q^+ \) and \( Q^- \). Also, \( \sigma_B(S_A,S_B) \) is monotonically increasing in \( S_A \) for any GAPs in \( Q^+ \), and monotonically decreasing in \( S_A \) for any \( Q^- \). For technical reasons and notational convenience, in the proof of Theorem 7 presented below, we “concurrently” prove both Theorem 7 and this symmetric version, without loss of generality.

Proof of Theorem 7. We first fix a \( B \)-seed set \( S_B \). Since \( S_B \) is always fixed, in the remaining proof we ignore \( S_B \) from the notations whenever it is clear from context. It suffices to show that monotonicity holds in an arbitrary, fixed possible world, which implies monotonicity holds for the diffusion model. Let \( W \) be an arbitrary possible world generated according to \( \S 4.3.5 \).

Define \( \Phi_A^W(S_A) \) (resp. \( \Phi_B^W(S_A) \)) to be the set of \( A \)-adopted (resp. \( B \)-adopted) nodes in possible world \( W \) with \( S_A \) being the \( A \)-seed set (and \( S_B \) being the fixed \( B \)-seed set). Furthermore, for any time step \( t \geq 0 \), define
\( \Phi^W_A(S_A, t) \) (resp. \( \Phi^W_B(S_A, t) \)) to be the set of \( A \)-adopted (resp. \( B \)-adopted) nodes in \( W \) by the end of step \( t \), given \( A \)-seed set \( S_A \). Clearly, \( \Phi^W_A(S_A) = \bigcup_{t \geq 0} \Phi^W_A(S_A, t) \) and \( \Phi^W_B(S_A) = \bigcup_{t \geq 0} \Phi^W_B(S_A, t) \). Let \( S \) and \( T \) be two sets, with \( S \subseteq T \subseteq V \).

**Mutual Competition** \( Q^- \). Our goal is to prove that for any \( v \in V \), (a) if \( v \in \Phi^W_A(S) \), then \( v \in \Phi^W_A(T) \); and (b) if \( v \in \Phi^W_B(T) \), then \( v \in \Phi^W_B(S) \). Item (a) implies self-monotonic increasing property while item (b) implies cross-monotonic decreasing property. We use an inductive proof to combine the proof of above two results together, as follows. For every \( t \geq 0 \), we inductively show that (i) if \( v \in \Phi^W_A(S, t) \), then \( v \in \Phi^W_A(T, t) \); and (ii) if \( v \in \Phi^W_B(T, t) \), then \( v \in \Phi^W_B(S, t) \).

Consider the base case of \( t = 0 \). If \( v \in \Phi^W_A(S, 0) \), then it means \( v \in S \), and thus \( v \in T = \Phi^W_A(T, 0) \). If \( v \in \Phi^W_B(T, 0) \), it means \( v \in S_B \), and thus \( v \in \Phi^W_B(S, 0) = S_B \).

For the induction step, suppose that for all \( t < t' \), (i) and (ii) hold, and we show (i) and (ii) also hold for \( t = t' \). For (i), we only need to consider \( v \in \Phi^W_A(S, t') \setminus \Phi^W_A(S, t' - 1) \), i.e. \( v \) adopts \( A \) at step \( t' \) when \( S \) is the \( A \)-seed set. Since \( v \) adopts \( A \), we know that \( \alpha^v_A \leq q_{A|\emptyset} \). Let \( U \) be the set of in-neighbors of \( v \) in the possible world \( W \). Let \( U_A(S_A) = U \cap \Phi^W_A(S_A, t' - 1) \) and \( U_B(S_A) = U \cap \Phi^W_B(S_A, t' - 1) \), i.e. \( U_A(S_A) \) (resp. \( U_B(S_A) \)) is the set of in-neighbors of \( v \) in \( W \) that adopted \( A \) (resp. \( B \)) by time \( t' - 1 \), when \( S_A \) is the \( A \)-seed set. Since \( v \in \Phi^W_A(S, t') \), we know that \( U_A(S) \neq \emptyset \). By induction hypothesis, we have \( U_A(S) \subseteq U_A(T) \) and \( U_B(T) \subseteq U_B(S) \).

Thus, \( U_A(T) \neq \emptyset \), which implies that by step \( t' \), \( v \) must have been informed of \( A \) when \( T \) is the \( A \)-seed set. If \( \alpha^v_A \leq q_{A|B} \), then no matter if \( v \) adopted \( B \) or not, \( v \) would adopt \( A \) by step \( t' \) according to the possible world model. That is, \( v \in \Phi^W_A(T, t') \).

Now suppose \( q_{A|B} < \alpha^v_A \leq q_{A|\emptyset} \). For a contradiction suppose \( v \notin \Phi^W_B(T, t') \), i.e., \( v \) does not adopt \( A \) by step \( t' \) when \( T \) is the \( A \)-seed set. Since \( v \) has been informed of \( A \) by \( t' \), the only possibility that \( v \) does not adopt \( A \) is because \( v \) adopted \( B \) earlier than \( A \), which means \( v \in \Phi^W_B(T, t') \).

Two cases arise:

First, if \( v \in \Phi^W_B(T, t' - 1) \), then by the induction hypothesis \( v \in \Phi^W_B(S, t' - 1) \). Since \( v \in \Phi^W_A(S, t') \setminus \Phi^W_A(S, t' - 1) \), it means that when \( S \) is the \( A \)-seed set, \( v \) adopts \( B \) first before adopting \( A \), but this contradicts to the condition that \( q_{A|B} < \alpha^v_A \). Therefore, \( v \notin \Phi^W_B(T, t' - 1) \).

Second, \( v \in \Phi^W_B(T, t') \setminus \Phi^W_B(T, t' - 1) \). Since \( v \notin \Phi^W_A(T, t') \), it means that \( v \) is informed of \( A \) at step \( t' \) when \( T \) is the \( A \)-seed set, and thus the tie-breaking rule must have been applied at this step and \( B \) is ordered first.
implies that when induction hypothesis, we have the in-neighbors of \( v \) has been informed of means that due to condition is that set. if implies that by step hypothesis we have that show that of the mutual competition case. Our goal is to prove that for any Mutual Complementarity definitions of \( A \)-seed set, we keep this \( A \)-seed set, the same tie-breaking rule at \( A \) would still order \( B \) first before \( A \), but this would result in \( v \) not adopting \( A \) at step \( t' \), a contradiction. Therefore, we know that \( v \in \Phi^W_A(T, t') \).

The statement of (ii) is symmetric to (i): if we exchange \( A \) and \( B \) and exchange \( S \) and \( T \), (ii) becomes (i). In fact, one can check that we can literally translate the induction step proof for (i) into the proof for (ii) by exchanging pair \( A \) and \( B \) and pair \( S \) and \( T \) (except that (a) we keep the definitions of \( U_A(S_A) \) and \( U_B(S_A) \), and (b) whenever we say some set is the \( A \)-seed set, we keep this \( A \)). This concludes the proof of the mutual competition case.

**Mutual Complementarity** \( Q^+ \). The proof structure is very similar to that of the mutual competition case. Our goal is to prove that for any \( v \in V \), (a) if \( v \in \Phi^W_A(S) \), then \( v \in \Phi^W_A(T) \); and (b) if \( v \in \Phi^W_B(S) \), then \( v \in \Phi^W_B(T) \). To show this, we inductively prove the following: For every \( t \geq 0 \), (i) if \( v \in \Phi^W_A(S, t) \), then \( v \in \Phi^W_A(T, t) \); and (ii) if \( v \in \Phi^W_B(S, t) \), then \( v \in \Phi^W_B(T, t) \). The base case is trivially true.

For the induction step, suppose (i) and (ii) hold for all \( t < t' \), and we show that (i) and (ii) also hold for \( t = t' \).

For (i), we only need to consider \( v \in \Phi^W_A(S, t') \setminus \Phi^W_A(S, t' - 1) \), i.e. \( v \) adopts \( A \) at step \( t' \) when \( S \) is the \( A \)-seed set. Since \( v \) adopts \( A \), we know that \( \alpha^v_A \leq q_{A|B} \). Since \( v \in \Phi^W_A(S, t') \), we know that \( U_A(S) \neq \emptyset \). By induction hypothesis we have \( U_A(S) \subseteq U_A(T) \). Thus we know that \( U_A(T) \neq \emptyset \), which implies that by step \( t' \), \( v \) must have been informed of \( A \) when \( T \) is the \( A \)-seed set. if \( \alpha^v_A \leq q_{A|\emptyset} \), then no matter \( v \) adopted \( B \) or not, \( v \) would adopt \( A \) by step \( t' \) according to the possible world model. Thus, \( v \in \Phi^W_A(T, t') \).

Now suppose \( q_{A|\emptyset} < \alpha^v_A \leq q_{A|B} \). Since \( v \in \Phi^W_A(S, t') \), the only possibility is that \( v \) adopts \( B \) first by time \( t' \) so that after reconsideration, \( v \) adopts \( A \) due to condition \( \alpha^v_A \leq q_{A|B} \). Thus we have \( v \in \Phi^W_B(S, t') \), and \( \alpha^v_B \leq q_{B|\emptyset} \).

If \( v \in \Phi^W_B(S, t' - 1) \), by induction hypothesis \( v \in \Phi^W_B(T, t' - 1) \), which means that \( v \) adopts \( B \) by time \( t' - 1 \) when \( T \) is the \( A \)-seed set. Since \( v \) has been informed of \( A \) by time \( t' \) when \( T \) is the \( A \)-seed set, condition \( \alpha^v_A \leq q_{A|B} \) implies that \( v \) adopts \( A \) by time \( t' \) when \( T \) is the \( A \)-seed set, i.e. \( v \in \Phi^W_A(T, t') \).

Finally we consider the case of \( v \in \Phi^W_B(S, t') \setminus \Phi^W_B(S, t' - 1) \). Looking at the in-neighbors of \( v \) in \( W \), \( v \in \Phi^W_B(S, t') \), implies that \( U_B(S) \neq \emptyset \). By the induction hypothesis, we have \( U_B(S) \subseteq U_B(T) \), and thus \( U_B(T) \neq \emptyset \). This implies that when \( T \) is the \( A \)-seed set, node \( v \) must have been informed of
B by time \( t' \). Since \( \alpha_B^V \leq q_B\emptyset \), we have that \( v \) adopts \( B \) by time \( t' \) when \( T \) is the \( \mathcal{A} \)-seed set. Then the condition \( \alpha_A^V \leq q_{A|B} \) implies that \( v \) adopts \( A \) by time \( t' \) when \( T \) is the \( \mathcal{A} \)-seed set, i.e. \( v \in \Phi_A^W(T, t') \).

This concludes the inductive step for (i) in the mutual complementarity case. The inductive step for (ii) is completely symmetric, and hence omitted. Therefore, we have completed the proof for the mutual complementarity case. As a result, the whole theorem holds.

4.5.2 Submodularity in Complementary Setting

Next, we analyze the submodularity of \( \sigma_A(S_A, S_B) \) w.r.t. \( S_A \) for mutual complementary GAPs. This has direct impact on the approximability of Influence Maximization (Problem 4). We show that submodularity is satisfied in the case of “one-way complementarity”, i.e., \( B \) complements \( A \) (\( q_{A|B} \leq q_A\emptyset \)), but \( A \) does not affect \( B \) (\( q_{B|A} = q_{B|A} \)), or vice versa (Theorem 8). However, this property are not satisfied in general as we show below.

![Graph](image)

Figure 4.4: The graph for Example 4

**Example 4** (Non-Submodularity). Consider the possible world in Figure 4.4. All edges are live. The node thresholds are: for \( w \): \( \alpha_A^w \leq q_{A|w} \), \( q_{B|w} < \alpha_B^w \leq q_{B|w} \); for \( z \): \( \alpha_A^z > q_{A|B} \), \( \alpha_B^z < q_{B|z} \); for \( v \): \( q_{A|\emptyset} < \alpha_A^v \leq q_{A|B} \), \( \alpha_B^v \leq q_{B|\emptyset} \); Then fix \( S_B = \{y\} \). For \( S_A \), let \( S = \emptyset \), \( T = \{x\} \), and \( u \) is the additional seed. It can be verified that only when \( S_A = T \cup \{u\} \), \( v \) becomes \( \mathcal{A} \)-adopted, violating submodularity.

A concrete example of \( Q \) for which submodularity does not hold is as follows: \( q_{A|\emptyset} = 0.078432; q_{A|B} = 0.24392; q_{B|\emptyset} = 0.37556; q_{B|A} = 0.99545 \). Seed sets are the same as above. We denote by \( p_v(S_A) \) the probability that \( v \) becomes \( \mathcal{A} \)-adopted with \( A \)-seed set \( S_A \). It can be verified that: \( p_v(S) = 0 \), \( p_v(S \cup \{u\}) = 8.898 \times 10^{-5} \), \( p_v(T) = 0.027254 \), and \( p_v(T \cup \{u\}) = 0.027383 \). Clearly, \( p_v(T \cup \{u\}) - p_v(T) > p_v(S \cup \{u\}) - p_v(S) \). Hence, replicating \( v \) sufficiently many times will lead to \( \sigma_A(T \cup \{u\}, S_B) - \sigma_A(T, S_B) \geq \sigma_A(S \cup \{u\}, S_B) - \sigma_A(S, S_B) \), violating submodularity.
In what follows, we first give two useful lemmas. Thanks to Lemma 4 below, we may assume w.l.o.g. that tie-breaking always favours $A$ in complementary cases.

**Lemma 4.** Consider any ComIC instance with $Q^+$. Given fixed $A$- and $B$-seed sets, for all nodes $v \in V$, all permutations of $v$’s in-neighbors are equivalent in terms of determining if $v$ becomes $A$-adopted and $B$-adopted. This implies that tie-breaking is not needed for mutual complementary case.

**Proof.** Without loss of generality, we only need to consider a node $v$ and two of its in-neighbours $u_A$ and $u_B$ which become $A$-adopted and $B$-adopted at $t-1$ respectively. In a possible world, there are nine possible combinations of the values of $\alpha_A^v$ and $\alpha_B^v$. We show that in all such combinations, the ordering $\pi_1 = \langle u_A, u_B \rangle$ and $\pi_2 = \langle u_B, u_A \rangle$ produce the same outcome for $v$.

1. $\alpha_A^v \leq q_A |\emptyset \land \alpha_B^v \leq q_B |\emptyset$. Both $\pi_1$ and $\pi_2$ make $v$ $A$-adopted and $B$-adopted.

2. $\alpha_A^v \leq q_A |\emptyset \land q_B |\emptyset < \alpha_B^v \leq q_B |A$. Both $\pi_1$ and $\pi_2$ make $v$ $A$-adopted and $B$-adopted. With $\pi_2$, $v$ first becomes $B$-suspended, then $A$-adopted, and finally $B$-adopted due to re-consideration.

3. $\alpha_A^v \leq q_A |\emptyset \land \alpha_B^v > q_B |A$. Both $\pi_1$ and $\pi_2$ makes $v$ $A$-adopted only.

4. $q_A |\emptyset < \alpha_A^v \leq q_A |B \land \alpha_B^v \leq q_B |\emptyset$. Symmetric to (2).

5. $q_A |\emptyset < \alpha_A^v \leq q_A |B \land q_B |\emptyset < \alpha_B^v \leq q_B |A$. In this case, $v$ does not adopt any item.

6. $q_A |\emptyset < \alpha_A^v \leq q_A |B \land \alpha_B^v > q_B |A$. In this case, $v$ does not adopt any item.

7. $\alpha_A^v > q_A |B \land \alpha_B^v \leq q_B |\emptyset$. Symmetric to (3): $v$ is $B$-adopted only.

8. $\alpha_A^v > q_A |B \land q_B |\emptyset < \alpha_B^v \leq q_B |A$. Symmetric to (6).

9. $\alpha_A^v > q_A |B \land \alpha_B^v > q_B |A$. In this case, $v$ does not adopt any item.

Since the possible world model is equivalent to ComIC (Lemma 3), the lemma holds true. 

**Lemma 5.** In the ComIC model, if $B$ is indifferent to $A$ (i.e., $q_{B|A} = q_{B|\emptyset}$), then for any fixed $B$-seed set $S_B$, the probability distribution over sets of $B$-adopted nodes is independent of $A$-seed set. Symmetrically, the probability distribution over sets of $A$-adopted nodes is also independent of $B$-seed set if $A$ is indifferent to $B$. 

93
Claim holds by a simple induction along $P_B$ would still be a live-edge path from $S_B$ to $v$ such that for all nodes $w$ on $P_B$ (excluding seeds), $\alpha^w_B \leq q$. Since $q_{B|\emptyset} = q_{B|A}$, this condition under which $v$ becomes $B$-adopted in $W$ is completely independent of any node's state w.r.t. $A$. Thus, the propagation of $B$-adoption is completely independent of the actual $A$-seed set (even empty). Due to the equivalence of the possible world model and Com-IC, the lemma holds.

**Theorem 8.** For any instance of Com-IC model with $q_{A|\emptyset} \leq q_{A|B}$ and $q_{B|\emptyset} = q_{B|A}$, we have

1. $\sigma_A(S_A, S_B)$ is submodular w.r.t. $A$-seed set $S_A$, for any fixed $B$-seed set $S_B$.

2. $\sigma_B(S_A, S_B)$ is submodular w.r.t. $B$-seed set $S_B$, and is independent of $A$-seed set $S_A$.

**Proof.** First of all, the submodularity of $\sigma_B$ holds trivially. Lemma 5 implies that the diffusion of $A$ does not affect the diffusion of $B$ whatsoever. Thus, $\sigma_B(S_A, S_B) = \sigma_B(\emptyset, S_B)$. It can be shown that the function $\sigma_B(\emptyset, S_B)$ is both monotone and submodular w.r.t. $S_B$, for any $q_{B|\emptyset}$, through a straightforward extension to the proof of Theorem 2.2 in Kempe et al. [86].

Now we prove the submodularity of $\sigma_A$. First, fix any possible world $W$ and any $B$-seed set $S_B \neq \emptyset$. Let $\Phi_W^A(S_A)$ be the set of $A$-adopted nodes in possible world $W$ with $A$-seed set $S_A$ ($S_B$ omitted when it is clear from the context). Consider two sets $S \subseteq T \subseteq V$, some node $u \in V \setminus T$, and finally a node $v \in \Phi_W^A(T \cup \{u\}) \setminus \Phi_W^A(T)$. There must exist a live-edge path $P_A$ from $T \cup \{u\}$ consisting entirely of $A$-adopted nodes. We denote by $w_0 \in T \cup \{u\}$ the origin of $P_A$. We first prove the following claim.

**Claim 1.** All nodes on path $P_A$ remain $A$-adopted even when $S_A = \{w_0\}$.

**Proof of Claim 1.** Consider any node $w_i \in P_A$. In this possible world, if $\alpha^w_A \leq q_{A|\emptyset}$, then regardless of the diffusion of $B$, $w_i$ will adopt $A$ as long as its predecessor $w_{i-1}$ adopts $A$. If $q_{A|\emptyset} < \alpha^w_A \leq q_{A|B}$, then there must also be a live-edge path $P_B$ from $S_B$ to $w_i$ that consists entirely of $B$-adopted nodes, and it boosts $w_i$ to adopt $A$. Since $q_{B|\emptyset} = q_{B|A}$, $A$ has no effect on the diffusion of $B$ (Lemma 5), and $P_B$ always exists and all nodes on $P_B$ would still be $B$-adopted through $S_B$ (fixed) irrespective of $A$-seeds. Thus, $P_B$ always boosts $w_i$ to adopt $A$ as long as $w_{i-1}$ is $A$-adopted. Hence, the claim holds by a simple induction along $P_A$ starting from $w_0$. \qed
Then, it is easy to see $w_0 = u$. Suppose otherwise for a contradiction. Then, $w_0 \in T$ must be true. By Claim 1 and the monotonicity of $\sigma_A$ (Theorem 7), $v \in \Phi_A^W(\{w_0\})$ implies that $w \in \Phi_A^W(T)$, a contradiction. Therefore, we have $v \notin \Phi_A^W(S)$ and $v \in \Phi_A^W(S \cup \{u\})$. This by definition implies that $|\Phi_A^W(\cdot)|$ is submodular for any $W$ and $S_B$. This is sufficient to show that $\sigma_A(S_A, S_B)$ is indeed submodular in $S_A$, due to the fact that a nonnegative linear combination of submodular functions is also submodular, and that the number of effective possible worlds are finite.

4.6 Scalable Approximation Algorithms

In this section, we derive a general framework (§4.6.1) to obtain approximation algorithms for Influence Maximization (§4.6.2).

Recall that for Influence Maximization under the classic IC model, the TIM algorithm proposed by Tang et al. [137] is able to produce a $(1 - 1/e - \varepsilon)$-approximation with at least $1 - |V|^{-\ell}$ probability in $O((k + \ell)(|E| + |V|) \log |V|/\varepsilon^2)$ expected running time. TIM relies on the notion of Reverse-Reachable sets (RR-sets) [22] for computing influence spread accurately and efficiently.

**Reverse-Reachable Sets.** In a deterministic directed graph $G_d = (V_d, E_d)$, given any $v \in V_d$, we say that all nodes that can reach $v$ in $G_d$ form an RR-set rooted at $v$ [22]. Let $R(v)$ denote such a set:

$$R(v) = \{u \in V_d : u \text{ can reach } v \text{ via edges in } E_d\}.$$

A *random* RR-set encapsulates two levels of randomness: (i) the “root” node $v$ is chosen uniformly at random from the graph, and (ii) a deterministic graph is sampled according to a certain probabilistic rule that retains a subset of edges from the graph. E.g., for the IC model, each edge $(u, v) \in E$ is independently removed with probability $(1 - p_{u,v})$.

The TIM algorithm first computes a lower bound on the optimal influence spread (which itself is NP-hard to compute). Then it uses the lower bound to derive the number of random RR-sets to be sampled, denoted $\theta$. To guarantee approximation solutions, the following inequality must be satisfied:

$$\theta \geq \varepsilon^{-2} (8 + 2\varepsilon)|V| \cdot \frac{\ell \log |V| + \log \binom{|V|}{k} + \log 2}{OPT_k}, \quad (4.2)$$

where $OPT_k$ is the optimal influence spread achievable amongst all size-$k$ node-sets, and $\varepsilon$ represents the trade-off between efficiency and quality: a
smaller $\varepsilon$ implies more RR-sets (longer running time), but gives a better approximation factor. The approximation guarantee of TIM relies on a key result from [22], re-stated here:

**Proposition 2** (Lemma 9 in [137]). Fix a set $S \subseteq V$ and a node $v \in V$. Under the Triggering model, let $\rho_1$ be the probability that $S$ activates $v$ in a cascade, and $\rho_2$ be the probability that $S$ overlaps with a random RR-set $R(v)$ rooted at $v$. Then, $\rho_1 = \rho_2$.

### 4.6.1 A General Framework Extending TIM

The solution framework proposed in [137] is promising but does not work as is for Influence Maximization with the ComIC model. The primary challenge is how to correctly generate random RR-sets, so that the influence spread computed using those RR-sets are accurate w.r.t. the ComIC model and the same approximation guarantee can be obtained for Problem 4. In what follows, we use Possible World (PW) models to generalize the theory in [22, 137] and show that the extended framework is capable of delivering approximation algorithms for our Problem 4.

For a generic stochastic diffusion model $M$, an *equivalent* PW model $M'$ is a model that specifies a distribution over $W$, the set of all possible worlds, where influence diffusion in each possible world in $W$ is deterministic. Further, given a seed set (or two seed sets $S_A$ and $S_B$ as in ComIC), the distribution of the sets of active nodes (or $A$- and $B$-adopted nodes in ComIC) in $M$ is the same as the corresponding distribution in $M'$. Then, we define a generalized concept of RR-set through the PW model:

**Definition 5** (Generalized RR-Set). For any possible world $W \in W$ and any root node $v$, the reverse-reachable set (RR-set) of $v$ in $W$ — denoted by $R_W(v)$ — is defined as the set of all nodes $u$ such that the singleton set $\{u\}$ would activate $v$ in $W$. A random RR-set of $v$ is a set $R_W(v)$ where $W$ is randomly sampled from $W$ using the probability distribution specified by $M'$.

It is easy to see that Definition 5 encompasses the RR-set definition in [22, 137] for IC, LT, and Triggering models as special cases. For the entire solution framework to work, the key property that RR-sets need to satisfy is the following:

**Definition 6** (Activation Equivalence Property). Let $M$ be a stochastic diffusion model and $M'$ be its equivalent possible world model. Let $G = (V, E, p)$ be a graph. Then, RR-sets have the Activation Equivalence Property if for
any fixed $S \subseteq V$ and any fixed $v \in V$, the probability that $S$ activates $v$ according to $M$ is the same as the probability that $S$ overlaps with a random RR-set generated from $v$ in a possible world in $M'$.

As shown in [137], the entire correctness and complexity analysis is based on the above property, and in fact in their latest improvement [136], they directly use this property as the definition of general RR-sets. Proposition 2 shows that the activation equivalence property holds for the triggering model.

We now provide a more general sufficient condition for the activation equivalence property to hold (Lemma 7), which gives concrete conditions on when the RR-set based framework would work. More specifically, we show that for any diffusion model $M$, if there is an equivalent PW model $M'$ of which all possible worlds satisfy the following two properties, then the RR-sets as defined in Definition 5 will enjoy the activation equivalence property.

(P1). Given two seed sets $S \subseteq T$, if a node $v$ can be activated by $S$ in a possible world $W$, then $v$ shall also be activated by $T$ in $W$.

(P2). If a node $v$ can be activated by $S$ in a possible world $W$, then there exists $u \in S$ such that the singleton seed set $\{u\}$ can also activate $v$ in $W$.

In fact, (P1) and (P2) are equivalent to monotonicity and submodularity respectively, as we formally state below.

**Lemma 6.** Let $W$ be a fixed possible world. Let $f_{v,W}(S)$ be an indicator function that takes on 1 if $S$ can activate $v$ in $W$, and 0 otherwise. Then, $f_{v,W}(\cdot)$ is monotone and submodular for all $v \in V$ if and only if both (P1) and (P2) are satisfied in $W$.

**Proof.** First consider “if”. Suppose both properties hold in $W$. Monotonicity directly follows from Property (P1). For submodularity, suppose $v$ can be activated by set $T \cup \{x\}$ but not by $T$, where $x \notin T$. By Property (P2), there exists some $u \in T \cup \{x\}$ such that $\{u\}$ can activate $v$ in $W$. If $u \in T$, then $T$ can also activate $v$ by Property (P1), a contradiction. Hence we have $u = x$. Then, consider any subset $S \subseteq T$. Note that by Property (P1), $S$ cannot activate $v$ (otherwise so could $T$), while $S \cup \{x\}$ can. Thus, $f_{v,W}(\cdot)$ is submodular.

Next we consider “only if”. Suppose $f_{v,W}(\cdot)$ is monotone and submodular for every $v \in V$. Property (P1) directly follows from monotonicity. For Property (P2), suppose for a contradiction that there exists a seed set $S$ that can activate $v$ in $W$, but there is no $u \in S$ so that $\{u\}$ activates
alone. We repeatedly remove elements from $S$ until the remaining set is the minimal set that can still activate $v$. Let the remaining set be $S'$. Note that $S'$ contains at least two elements. Let $u \in S'$, and then we have $f_{v,W}(\emptyset) = f_{v,W}(\{u\}) = f_{v,W}(S' \setminus \{u\}) = 0$, but $f_{v,W}(S') = 1$, which violates submodularity, a contradiction. This completes the proof.

Lemma 7. Let $M$ be a stochastic diffusion model and $M'$ be its equivalent possible world model. If $M'$ satisfies Properties (P1) and (P2), then the RR-sets as defined in Definition 5 have the activation equivalence property as in Definition 6.

Proof. It is sufficient to prove that in every possible world $W \in W$, $S$ activates $v$ if and only if $S$ intersects with $v$’s RR set in $W$, denoted by $R_W(v)$.

Suppose $R_W(v) \cap S \neq \emptyset$. Without loss of generality, we assume a node $u$ is in the intersection. By the definition of RR set, set $\{u\}$ can activate $v$ in $W$. Per Property (P1), $S$ can also activate $v$ in $W$.

Now suppose $S$ activates $v$ in $W$. Per Property (P2), there exists $u \in S$ such that $\{u\}$ can also activate $v$ in $W$. Then by the RR-set definition, $u \in R_W(v)$. Therefore, $S \cap R_W(v) \neq \emptyset$.

Comparing with directly using the activation equivalence property as the RR-set definition in [136], our RR-set definition provides a more concrete way of constructing RR-sets, and our Lemma 6 and Lemma 7 provide general conditions under which such constructions can ensure algorithm correctness.

Algorithm 7, GeneralTIM, outlines a general solution framework based on RR-sets and TIM. It provides a probabilistic approximation guarantee for any diffusion models that satisfy (P1) and (P2). Note that the estimation of a lower bound $LB$ of $OPT_k$ (line 2) is orthogonal to our contributions and we refer the reader to [137] for details. Finally, we have:

Theorem 9. Suppose for a stochastic diffusion model $M$ with an equivalent PW model $M'$, that for every possible world $W$ and every $v \in V$, the indicator function $f_{v,W}$ is monotone and submodular. Then for the influence maximization problem under $M$ with graph $G = (V,E,p)$ and seed set size $k$, GeneralTIM (Algorithm 7) applied on the general RR-sets (Definition 5) returns a $(1-1/e-\varepsilon)$-approximate solution with at least $1-|V|^{-\varepsilon}$ probability.

Theorem 9 follows from Lemmas 6 and 7, and the fact that all theoretical analysis of TIM relies only on the Chernoff bound and the activation equivalence property, “without relying on any other results specific to the IC model” [137].
Algorithm 7: GeneralTIM—Generalized Two-phase Influence Maximization Algorithm

**Data:** graph $G = (V,E,p)$, $k$, $\varepsilon$, $\ell$

**Result:** seed set $S$

1. begin
2. 
3. $LB \leftarrow$ lower bound of $OPT_k$ estimated by method in [137]
4. Compute $\theta$ using Equation (4.2) with $LB$ replacing $OPT_k$
5. $\mathcal{R} \leftarrow$ generate $\theta$ random RR-sets according to Definition 5
6. $S \leftarrow \emptyset$
7. for $i = 1$ to $k$ do
8.     $v_i \leftarrow$ the node appearing in the most RR-sets in $\mathcal{R}$
9.     $S \leftarrow S \cup \{v_i\}$
10. Remove all RR-sets in which $v_i$ appears

Next, we describe how to generate RR-sets correctly for Influence Maximization under the ComIC model (line 4 of Algorithm 7), which is much more involved than the generation process for IC/LT models [137]. We will focus on submodular settings ($q_A|\emptyset \leq q_A|B$ and $q_B|\emptyset = q_B|A$, cf. Theorem 8) first, and then in §4.7, we propose Sandwich Approximation to handle any mutual complementary cases in which submodularity does not hold in general.

4.6.2 Generating RR-Sets for Influence Maximization with ComIC

We present two algorithms, RR-ComIC and RR-ComIC++, for generating random RR-sets per Definition 5. The overall algorithm for Influence Maximization can be obtained by plugging RR-ComIC or RR-ComIC++ into GeneralTIM (Algorithm 7).

According to Definition 5, for Influence Maximization, the RR-set of a root $v$ in a possible world $W$ is the set of nodes $u$ such that if $u$ is the only $A$-seed, then $v$ would be $A$-adopted in $W$ with any fixed $B$-seed set $S_B$. By Theorems 7 and 8 (whose proofs indeed show that the indicator function $f_{v,W}(S)$ is monotone and submodular), along with Lemmas 6 and 7, we know that RR-sets following Definition 5 have the activation equivalence property. We now focus on how to construct RR-sets that satisfy Definition 5. Recall that in ComIC, adoption decisions for $A$ are based on a number of factors
Algorithm 8: RR-ComIC—Generating RR-set for Problem 4

Data: Graph $G = (V, E, p)$, root node $v$, $B$-seed set $S_B$
Result: RR-set $R_W(v)$

1 begin
2 Create an empty FIFO queue $Q$ and empty set $R_W(v)$
3 Enqueue all nodes in $S_B$ into $Q$ /* start forward labeling */
4 while $Q$ is not empty do
5   $u \leftarrow Q$.dequeue()
6   Mark $u$ as $B$-adopted
7   foreach $v \in N^+(u)$ such that $(u, v)$ is live do
8     if $\alpha^u_W \leq q_{B|\emptyset} \land v$ is not visited then
9        $Q$.enqueue($v$) /* also mark $v$ as visited */
10    Clear $Q$, and then enqueue $v$ /* start backward BFS */
11 while $Q$ is not empty do
12   $u \leftarrow Q$.dequeue()
13   $R_W(v) \leftarrow R_W(v) \cup \{u\}$
14   if $(u$ is $B$-adopted $\land \alpha^u_W \leq q_{A|\emptyset} \land (u$ is not $B$-adopted $\land \alpha_A^u \leq q_{A|\emptyset})$ then
15      foreach $w \in N^-(u)$ such that $(w, u)$ is live do
16         if $w$ is not visited then
17            $Q$.enqueue($w$)
18            mark $w$ visited

such as whether $v$ is reachable via a live-edge path from $S_A$ and its state w.r.t. $B$ when reached by $A$. Note that $q_{B|\emptyset} = q_{B|A}$ implies that $B$-diffusion is independent of $A$ (Lemma 5). Our algorithms take advantage of this fact, by first revealing node states w.r.t. $B$, which gives a sound basis for generating RR-sets for $A$.

4.6.2.1 The RR-ComIC Algorithm

Conceptually, RR-ComIC (Algorithm 8) proceeds in three phases.

Phase I. Sample a possible world as described in §4.3.5 (omitted from the pseudo-code).

Phase II. A forward labeling process from the input $B$-seed set $S_B$ (lines 3 to 9): a node $v$ becomes $B$-adopted if $\alpha^v_W \leq q_{B|\emptyset}$ and $v$ is reachable
from \( S_B \) via a path consisting entirely of live edges and \( B \)-adopted nodes.

**Phase III.** Randomly select a node \( v \) and generate RR-set \( R_W(v) \) by running a Breadth-First Search (BFS) backwards by following incoming edges (lines 10 to 18).

Note that the RR-set generation for IC and LT models [137] is essentially a simpler version of the first and third phases.

**Backward BFS.** For possible world \( W \), an RR-set \( R_W(v) \) is formed by all nodes explored in the backward BFS as follows. Initially, we enqueue \( v \) into a FIFO queue \( Q \). We repeatedly dequeue a node \( u \) from \( Q \) for processing until \( Q \) is empty.

- **Case 1:** \( u \) is \( B \)-adopted. There are two sub-cases: (i) If \( \alpha^u_A \leq q_{A|B} \), then \( u \) is able to transit from \( A \)-informed to \( A \)-adopted. Thus, we continue to examine \( u \)'s in-neighbors. For all unexplored \( w \in N^+(u) \), if edge \((w,u)\) is live, then enqueue \( w \); (ii) If \( \alpha^u_A > q_{A|B} \), then \( u \) cannot transit from \( A \)-informed to \( A \)-adopted, and thus \( u \) has to be an \( A \) seed to become \( A \)-adopted. In this case, \( u \)'s in-neighbors will not be examined.

- **Case 2:** \( u \) is not \( B \)-adopted. Similarly, if \( \alpha^u_A \leq q_{A|B} \), perform actions as in 1(i); otherwise perform actions as in 1(ii).

**Theorem 10.** Under one-way complementarity (\( q_{A|\emptyset} \leq q_{A|B} \) and \( q_{B|\emptyset} = q_{B|A} \)), the RR-sets generated by the RR-ComIC algorithm satisfy Definition 5 for Problem 4. As a result, Theorem 9 applies to GeneralTIM with RR-ComIC in this case.

**Proof.** It suffices to show that, given a fixed possible world \( W \), a fixed \( B \)-seed set \( S_B \), and a certain node \( u \in V \), for any node \( v \notin \Phi^W_A(\emptyset, S_B) \) with \( \alpha^v_A \leq q_{A|B} \), we have: \( v \in \Phi^W_A(\{u\}, S_B) \) if and only if there exists a live-edge path \( P \) from \( u \) to \( v \) such that for all nodes \( w \in P \), excluding \( u \), \( w \) satisfies \( \alpha^w_A \leq q_{A|B} \), and in case \( \alpha^w_A > q_{A|B} \), then \( w \) must be \( B \)-adopted.

The “if” direction is straightforward as \( P \) will propagate the adoption information of \( A \) all the way to \( v \). If \( \alpha^v_A \leq q_{A|B} \), it adopts \( A \) without question. If \( \alpha^v_A \in (q_{A|\emptyset}, q_{A|B}] \), then \( v \) must be \( B \)-adopted by the definition of \( P \), which makes it \( A \)-adopted. For the “only if” part, suppose no such \( P \) exists for \( u \). This leads to a direct contradiction since \( u \) is the only \( A \)-seed, and \( u \) lacks a live-edge path to \( v \), it is impossible for \( v \) to get informed of \( A \), let alone adopting \( A \). Next,
suppose there is a live-edge path $P$ from $u$ to $v$, but there is a certain node $w \in P$ which violates the conditions set out in the lemma. First, $w$ could be have a “bad” threshold: $\alpha^w_A > q_A|B|$. In this case, $w$ will not adopt $A$ regardless of its status w.r.t. $B$, and hence the propagation of $A$ will not reach $v$. Second, $w$ could have a threshold such that $\alpha^w_A \in (q_A|\emptyset|, q_A|B|]$ but it does not adopt $B$ under the influence of the given $S_B$. Similar to the previous case, $w$ will not adopt $A$ and the propagation of $A$ will not reach $v$. This completes the “only if” part.

Then by Definition 5, the theorem follows. ⪜

**Lazy sampling.** For RR-ComlC to work, it is *not* necessary to sample all edge- and node-level variables (i.e., the entire possible world) up front, as the forward labeling and backward BFS are unlikely to reach the whole graph. Hence, we can simply reveal edge and node states on demand (“lazy sampling”), based on the principle of deferred decisions. In light of this observation, the following improvements are made to RR-ComlC.

First, the first phase is simply skipped. Second, in Phase II, edge states and $\alpha$-values are sampled as the forward labeling from $S_B$ goes on. We record the outcomes, as it is possible to encounter certain edges and nodes again in the third phase. Next, for Phase III, consider any node $u$ dequeued from $Q$. We need to perform an additional check on every incoming edge $(w, u)$. If $(w, u)$ has already been tested live in Phase II, then we just enqueue $w$. Otherwise, we first sample its live/block status, and enqueue $w$ if it is live. Algorithm 8 provides the pseudo-code for RR-ComlC, where sampling is assumed to be done whenever we need to check the status of an edge or the $\alpha$-values of a node.

**Expected time complexity.** For the entire seed selection (Algorithm 7 with RR-ComlC) to guarantee approximate solutions, we must estimate a lower bound $LB$ of $OPT_k$ and use it to derive the minimum number of RR-sets required, defined as $\theta$ in Equation (4.2). In expectation, the algorithm runs in $O(\theta \cdot EPT)$ time, where $EPT$ is the expected number of edges explored in generating one RR-set. In our case, $EPT = EPT_F + EPT_B$, where $EPT_F$ ($EPT_B$) is the expected number of edges examined in forward labeling (resp., backward BFS). Thus, we have the following result.

**Lemma 8.** The expected running time complexity of GeneralTIM with RR-ComlC is

$$O \left( (k + \ell)(|V| + |E|) \log |V| \left( 1 + \frac{EPT_F}{EPT_B} \right) \right).$$
Proof. Given a fixed RR-set $R \subseteq V$, let $\omega(R)$ be the number of edges in $G$ that point to nodes in $R$. Since in RR-ComIC, it is possible that we do not examine incoming edges to a node added to the RR-set (cf. Cases 1(ii) and 2(ii) in the backward BFS), we have:

$$EPT_B \leq \mathbb{E}[\omega(R)],$$

where the expectation is taken over the random choices of $R$. By Lemma 4 in [137] (note that this lemma only relies on the activation equivalence property of RR-sets, which holds true in our current one-way complementarity setting),

$$\frac{|V|}{|E|} \cdot \mathbb{E}[\omega(R)] \leq OPT_k.$$

This gives

$$\frac{|V|}{|E|} \cdot EPT_B \leq OPT_k.$$

Following the same analysis as in [137] one can check that the lower bound $LB$ of $OPT_k$ obtained by the estimation method in [137] guarantees that $LB \geq EPT_B \cdot |V|/|E|$. Since in our algorithm we set $\theta = \lambda/LB$, where (following Eq.(4.2))

$$\lambda = \varepsilon^{-2} \left( (8 + 2\varepsilon)|V| \left( \ell \log |V| + \log \left( \frac{|V|}{k} \right) + \log 2 \right) \right),$$

then the expected running time of generating all RR-sets is:

$$O(\theta \cdot EPT) = O \left( \frac{\lambda}{LB} \cdot (EPT_F + EPT_B) \right)
= O \left( \frac{\lambda |E|}{|V| EPT_B} (EPT_F + EPT_B) \right)
= O \left( \frac{\lambda |E|}{|V|} \left( 1 + \frac{EPT_F}{EPT_B} \right) \right)
= O \left( (k + \ell)(|V| + |E|) \log |V| \left( 1 + \frac{EPT_F}{EPT_B} \right) \right).$$

103
The time complexity for estimating {$LB$} and for calculating the final seed set given RR-sets are the same as in [137], and thus the final complexity is

$$O \left( (k + \ell) (|V| + |E|) \log |V| \left( 1 + \frac{EPT_F}{EPT_B} \right) \right).$$

This completes our analysis.

Observe that {$EPT_F$} increases when the input {$B$}-seed set grows. Intuitively, it is reasonable that a larger {$B$}-seed set may have more complementary effect and thus it may take longer time to find the best {$A$}-seed set. However, it is possible to reduce {$EPT_F$} as described in the next algorithm.

### 4.6.2.2 The RR-ComIC++ Algorithm

The RR-ComIC algorithm may incur loss of efficiency because some of the work done in forward labeling (Phase II) may not be used in backward BFS (Phase III). For instance, consider an extreme situation where all nodes explored in forward labeling are in a different connected component of the graph than the root {$v$} of the RR-set. In this case, forward labeling can be skipped safely and entirely! To leverage this observation, we propose RR-ComIC++ (pseudo-code in Algorithm 9), of which the key idea is to run two rounds of backward BFS from the random root {$v$}. The first round determines the necessary scope of forward labeling, while the second one generates the RR-set.

**First backward BFS.** As usual, we create a FIFO queue {$Q$} and enqueue the randomly chosen root {$v$}. We also sample {$\alpha_B^W$} uniformly at random from $[0, 1]$. Then we repeatedly dequeue a node {$u$} until {$Q$} is empty: for each incoming edge {(w, u)}, we test its live/blocked status based on probability {$p_{w,u}$}, independently. If (w, u) is live and w has not been visited before, enqueue w and sample its {$\alpha_B^W$}.

Let {$T_1$} be the set of all nodes explored. If {$T_1 \cap S_B = \emptyset$}, then none of the {$B$}-seeds can reach the explored nodes, so that forward labeling can be completely skipped. The above extreme example falls into this case. Otherwise, we run a residual forward labeling only from {$T_1 \cap S_B$} along the explored nodes in {$T_1$}; if a node {$u \in T_1 \setminus S_B$} is reachable by some {$s \in T_1 \cap S_B$} via a live-edge path with all {$B$}-adopted nodes, and {$\alpha_B^{u,W} \leq q_{B \emptyset}$}, {$u$} becomes {$B$}-adopted. Note that it is not guaranteed in theory that this always saves time compared to RR-ComIC, since the worst case of RR-ComIC++ is that {$T_1 \cap S_B = S_B$}, which means that the first round is wasted. However, our experimental re-
Algorithm 9: RR-ComIC++ for Generating RR-set for Problem 4

Data: Graph $G = (V, E, p)$, root node $v$, $B$-seed set $S_B$

Result: RR-set $R_W(v)$

1 begin
2 Create an FIFO queue $Q$ and empty sets $R_W(v)$, $T_1$
3 $Q$.enqueue($v$) /* 1st backward BFS */
4 while $Q$ is not empty do
5 $u \leftarrow Q$.dequeue()
6 $T_1 \leftarrow T_1 \cup \{u\}$
7 foreach unvisited $w \in \mathcal{N}^-(u)$ such that $(w, u)$ is live do
8 $Q$.enqueue($w$)
9 Mark $w$ visited
10 if $T_1 \cap S_B \neq \emptyset$ then /* auxiliary forward pass to determine $B$ adoption */
11 Clear $Q$
12 Enqueue all nodes of $T_1 \cap S_B$ into $Q$
13 Execute line 4 to line 9 in Algorithm 8 /* 2nd backward BFS */
15 execute line 10 to line 18 in Algorithm 8

Results (§4.8) indeed show that RR-ComIC++ is often one magnitude faster than RR-ComIC.

Second backward BFS. This round is largely the same as Phase III in RR-ComIC, but there is a subtle difference. Suppose we just dequeued a node $u$. It is possible that there exists an incoming edge $(w, u)$ whose status is not determined. This is because we do not enqueue previously visited nodes in BFS. Hence, if in the previous round, $w$ is already visited via an out-neighbor other than $u$, $(w, u)$ would not be tested. Thus, in the current round we shall test $(w, u)$, and decide if $w$ belongs to $R_W(v)$ accordingly. To see RR-ComIC++ is equivalent to RR-ComIC, it suffices to show that for each node explored in the second backward BFS, its adoption status w.r.t. $B$ is the same in both algorithms.

We now show the RR-ComIC and RR-ComIC++ are indeed equivalent.

Lemma 9. Consider any possible world $W$ of the ComIC model. Let $v$ denote the root node. For any $u \in V$ that can reach $v$ via live-edges in $W$, $u$ is determined as $B$-adopted in RR-ComIC if and only if $u$ is determined as $B$-adopted in RR-ComIC++.
Proof. We first prove the "if" part. Suppose \( u \) is determined as \( \mathcal{B} \)-adopted in \( \text{RR-ComIC}^{++} \). This means that there exists a node \( s \in T_1 \cap S_\mathcal{B} \), such that there is a path from \( s \) to \( u \) consisting entirely of live-edges and \( \mathcal{B} \)-adopted nodes (every node \( w \) on this path satisfies that \( \alpha^w \leq q_{\mathcal{B}|\emptyset} \)). Therefore, in \( \text{RR-ComIC} \), where \( W \) is generated upfront (or revealed on-the-fly using lazy sampling), this live-edge path must still exist. Thus, \( u \) must be also \( \mathcal{B} \)-adopted in \( \text{RR-ComIC} \) as well.

Next we prove the "only if" part. By definition, if \( u \) is determined as \( \mathcal{B} \)-adopted in possible world \( W \), then there exists a path \( P \) from some \( s \in S_\mathcal{B} \) to \( u \) such that the path consists entirely of live edges and all nodes \( w \) on the path satisfy that \( \alpha^w \leq q_{\mathcal{B}|\emptyset} \). It suffices to show that if \( u \) is reachable by \( v \) backwards in \( W \), then \( P \) will be explored entirely by \( \text{RR-ComIC}^{++} \).

Suppose otherwise. That is, there exists a node \( z \in P \) that is not explored by the first round backward BFS from \( v \). We have established that in the completely revealed possible world \( W \), there is a live-edge path from \( z \) to \( u \) and from \( u \) to \( v \) respectively. Thus, connecting the two paths at node \( u \) gives a single live-edge path \( P_z \) from \( z \) to \( v \). Now recall that the continuation of the first backward BFS phase in \( \text{RR-ComIC}^{++} \) relies solely on edge status (as long as an edge \( (w,u) \) is determined live, \( w \) will be visited by the BFS). This means that \( z \) must have been explored in the first backward BFS, a contradiction. This completes the proof.

Lemma 9, together with the fact that all relevant edges will be tested for liveness (Phase III in \( \text{RR-ComIC} \) and the second backward BFS in \( \text{RR-ComIC}^{++} \)), it then follows that \( \text{RR-ComIC} \) and \( \text{RR-ComIC}^{++} \) are equivalent in terms of generating RR-sets for the ComIC model.

**Expected time complexity.** The analysis is similar: We can show that the expected running time of \( \text{RR-ComIC}^{++} \) is

\[
O \left( (k + \ell)(|V| + |E|) \log |V| \left( 1 + \frac{\text{EPT}_{B1}}{\text{EPT}_{B2}} \right) \right),
\]

where \( \text{EPT}_{B1} \) (\( \text{EPT}_{B2} \)) is the expected number of edges explored in the first (resp., second) backward BFS.

Compared to \( \text{RR-ComIC} \), \( \text{EPT}_{B2} \) is the same as \( \text{EPT}_B \) in \( \text{RR-ComIC} \), so \( \text{RR-ComIC}^{++} \) will be faster than \( \text{RR-ComIC} \) if \( \text{EPT}_{B1} < \text{EPT}_F \), i.e., if the first backward BFS plus the residual forward labeling explores fewer edges, compared to the full forward labeling in \( \text{RR-ComIC} \).
4.7 The Sandwich Approximation Strategy

In this section, we present the *Sandwich Approximation* strategy that leads to algorithms with data-dependent approximation factors for Influence Maximization in the general mutual complement case of ComIC \((q_{A|\emptyset} \leq q_{A|B} \text{ and } q_{B|\emptyset} \leq q_{B|A})\). In fact, Sandwich Approximation can be seen as a general strategy, applicable to any non-submodular maximization problems for which we can find submodular upper or lower bound functions. Thus, our presentation below is generic and independent of ComIC and Problem 4.

Let \(\sigma : 2^V \to \mathbb{R}_{\geq 0}\) be a non-submodular set function. Let \(\mu\) and \(\nu\) be submodular and defined on the same ground set \(V\) such that \(\mu(S) \leq \sigma(S) \leq \nu(S)\) for all \(S \subseteq V\). That is, \(\mu\) (\(\nu\)) is a lower (resp., upper) bound on \(\sigma\) everywhere. Consider the problem of maximizing \(\sigma\) subject to a cardinality constraint \(k\). Notice that if the objective function were \(\mu\) or \(\nu\), the problem would be approximable within \(1 - 1/e\) (e.g., max-\(k\)-cover) or \(1 - 1/e - \varepsilon\) (e.g., influence maximization) by the greedy algorithm \([86, 119]\). A natural question is: Can we leverage the fact that \(\mu\) and \(\nu\) “sandwich” \(\sigma\) to derive an approximation algorithm for maximizing \(\sigma\)? The answer is yes.

4.7.1 Methodology

First, run the greedy algorithm on all three functions. It produces an approximate solution for \(\mu\) and \(\nu\). Let \(S_\mu, S_\sigma, S_\nu\) be the solution obtained for \(\mu, \sigma,\) and \(\nu\) respectively. Then, select the final solution to \(\sigma\) to be

\[
S_{\text{sand}} = \arg \max_{S \in \{S_\mu, S_\sigma, S_\nu\}} \sigma(S).
\]  

(4.3)

**Theorem 11.** Sandwich Approximation solution gives:

\[
\sigma(S_{\text{sand}}) \geq \max \left\{ \frac{\sigma(S_\nu)}{\nu(S_\nu)}, \frac{\mu(S_\sigma)}{\sigma(S_\sigma)} \right\} \cdot (1 - 1/e) \cdot \sigma(S_\sigma^*),
\]  

(4.4)

where \(S_\sigma^*\) is the optimal solution maximizing \(\sigma\) (subject to cardinality constraint \(k\)).
Proof. Let $S^*_{\mu}$ and $S^*_{\nu}$ be the optimal solution to maximizing $\mu$ and $\nu$ respectively. We have

$$
\sigma(S_{\nu}) = \frac{\sigma(S_{\nu})}{\nu(S_{\nu})} \cdot \nu(S_{\nu}) \\
\geq \frac{\sigma(S_{\nu})}{\nu(S_{\nu})} \cdot (1 - 1/e) \cdot \nu(S^*_{\nu}) \\
\geq \frac{\sigma(S_{\nu})}{\nu(S_{\nu})} \cdot (1 - 1/e) \cdot \nu(S^*_{\sigma}) \\
\geq \frac{\sigma(S_{\nu})}{\nu(S_{\nu})} \cdot (1 - 1/e) \cdot \sigma(S^*_{\sigma}),
$$

(4.5)

and

$$
\sigma(S_{\mu}) \geq \mu(S_{\mu}) \geq (1 - 1/e) \cdot \mu(S^*_{\mu}) \\
\geq (1 - 1/e) \cdot \mu(S^*_{\sigma}) \\
\geq \frac{\mu(S^*_{\sigma})}{\sigma(S^*_{\sigma})} \cdot (1 - 1/e) \cdot \sigma(S^*_{\sigma}).
$$

(4.6)

The theorem follows by applying Equation (4.3). $\square$

Without loss of generality, in the theorem statement we use approximation factor $1 - 1/e$. In cases where the function value must be estimated using MC simulations, the factor drops to $1 - 1/e - \epsilon$, for any $\epsilon > 0$. However, this does not affect our analysis.

Further Remarks. While the factor in Equation (4.4) involves $S^*_{\sigma}$ that generally is not computable in polynomial time for problems such as Influence Maximization, the first term inside $\max \frac{\sigma(S_{\nu})}{\nu(S_{\nu})} \cdot \frac{\mu(S^*_{\sigma})}{\sigma(S^*_{\sigma})}$ involves $S_{\mu}$ can be computed efficiently and can be of practical value (see Table 4.9 in §4.8). We emphasize that Sandwich Approximation is much more general and is not restricted to cardinality constraints. E.g., for a general matroid constraint, simply replace $1 - 1/e$ with $1/2$ in (4.4), as the greedy algorithm is a 1/2-approximation in this case [119].

Furthermore, monotonicity is not important, as maximizing general submodular functions can be approximated within a factor of $1/2$ [28], and thus Sandwich Approximation applies regardless of monotonicity. On the other hand, the true effectiveness of Sandwich Approximation depends on how close $\nu$ and $\mu$ are to $\sigma$: e.g., a constant function can be a trivial submodular upper bound function but would only yield trivial data-dependent approxi-
mation factors. Thus, an interesting question is how to derive $\nu$ and $\mu$ that are as close to $\sigma$ as possible, while maintaining submodularity.

### 4.7.2 Applying Sandwich Approximation to Influence Maximization

For Problem 4, GeneralTIM (Algorithm 7) with RR-ComIC (Algorithm 8) or RR-ComIC++ (Algorithm 9) provides a $(1 - 1/e - \epsilon)$-approximate solution with high probability, when $q_{A|\emptyset} \leq q_{A|B}$ and $q_{B|\emptyset} = q_{B|A}$. When $q_{B|\emptyset} < q_{B|A}$, the upper bound function $\nu$ can be obtained by increasing $q_{B|\emptyset}$ to $q_{B|A}$, while the lower bound function $\mu$ can be obtained by decreasing $q_{B|A}$ to $q_{B|\emptyset}$. The correctness of this approach is ensured by the following theorem.

**Theorem 12.** Suppose $q_{A|\emptyset} \leq q_{A|B}$ and $q_{B|\emptyset} \leq q_{B|A}$. Then, under the ComIC model, for any fixed $A$ and $B$ seed sets $S_A$ and $S_B$, $\sigma(S_A, S_B)$ is monotonically increasing w.r.t. any one of $\{q_{A|\emptyset}, q_{A|B}, q_{B|\emptyset}, q_{B|A}\}$ with other three GAPs fixed, as long as after the increase the parameters are still in $Q^+$.

**Proof (Sketch).** The detailed proof would follow the similar induction proof structure for each possible world as in the proof of Theorem 7. Intuitively, we would prove inductively that at every step increasing $q_{A|\emptyset}$ or $q_{A|B}$ would increase both $A$-adopted and $B$-adopted nodes. \qed

Putting it all together, the final algorithm for Problem 4 is GeneralTIM with (i) RR-ComIC/RR-ComIC++ and (ii) Sandwich Approximation. It is important to see how useful and effective Sandwich Approximation is in practice. We address this question head on in §4.8, where we “stress test” the idea behind Sandwich Approximation. Intuitively, if $q_{B|\emptyset}$ and $q_{B|A}$ are close to each other, the upper and lower bounds ($\nu$ and $\mu$) obtained for Influence Maximization can be expected to be quite close to $\sigma$ in terms of function values. More details are presented in the next section.

### 4.8 Experiments

We performed extensive experiments on three real-world social networks from which four datasets are derived and tested on. We first present results with synthetically generated GAPs (Section 4.8.1). Then we shall propose a method for learning GAPs using action log data (Section 4.8.2), and conduct experiments using learned GAPs (Section 4.8.3).

**Datasets.** Flixster was collected from a movie website with social networking features, and we extracted a strongly connected component from the original
network [82]. Douban\(^3\) is collected from a Chinese social network [147], where users rate and review numerous books, movies, music, etc. We crawled all movie & book ratings of the users in the graph, and derive two datasets from book and movie ratings: Douban-Book and Douban-Movie (details later). Last.fm is taken from the popular music website\(^4\) with social networking features. For all graphs, we learned influence probabilities on edges using the method in [68], which has been widely adopted in the field [35]. Table 4.2 presents the basic stats of the datasets.

### 4.8.1 Experiments with Synthetic Adoption Probabilities

We first evaluated our proposed RR-set-based algorithms described in Section 4.6 under various combinations of synthetic GAPs, and compared with two baselines:

1. **VanillaIC**: It selects \(k\) seeds using TIM algorithm [137] under the classic IC model, essentially ignoring the other product and the NLA in ComIC model; Intuitively, VanillaIC ought to be effective for Influence Maximization when \(q_A|B - q_A|\emptyset\) is small, in which case complementary effects are less strong and hence it is “safer” to ignore \(B\)-seeds and all the GAPs. Recall that the TIM algorithm is a highly scalable approximation algorithm for influence maximization and was shown to dominate many advanced heuristics [37, 39, 72] in terms of both seed set quality and running time in [137].

2. **Copying**: For Influence Maximization, it simply selects the top-\(k\) \(B\)-seeds to be \(A\)-seeds (assuming \(|S_B| \geq |S_A|\)).

We set \(q_A|B = q_B|A = 0.75\), \(q_B|\emptyset = 0.5\). And \(q_A|\emptyset\) was set to 0.1, 0.3, 0.5, which represent strong, moderate, and low complementarity respectively.

![Table 4.2: Statistics of graph data (all directed)](attachment:table4.2)

<table>
<thead>
<tr>
<th></th>
<th>Douban-Book</th>
<th>Douban-Movie</th>
<th>Flixster</th>
<th>Last.fm</th>
</tr>
</thead>
<tbody>
<tr>
<td># nodes</td>
<td>23.3K</td>
<td>34.9K</td>
<td>12.9K</td>
<td>61K</td>
</tr>
<tr>
<td># edges</td>
<td>141K</td>
<td>274K</td>
<td>192K</td>
<td>584K</td>
</tr>
<tr>
<td>avg. out-degree</td>
<td>6.5</td>
<td>7.9</td>
<td>14.8</td>
<td>9.6</td>
</tr>
<tr>
<td>max. out-degree</td>
<td>1690</td>
<td>545</td>
<td>189</td>
<td>1073</td>
</tr>
</tbody>
</table>

\(^3\)http://www.douban.com/, last accessed October 6, 2015.

\(^4\)http://www.last.fm/, last accessed October 6, 2015.
Lots of possibilities exist for fixing the input $B$-seeds. We tested the following three representative cases:

1. Run VanillalC and select the 101st to 200th nodes – this models a situation where we assume those seeds are moderately influential.

2. Randomly select 100 nodes – this models our complete lack of knowledge;

3. Run VanillalC and select the top-100 nodes – this models a situation where we assume the advertiser might use an advanced algorithm such as TIM to target highly influential users;

<table>
<thead>
<tr>
<th></th>
<th>VanillaC</th>
<th>Copying</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_{A/B} = 0.1$</td>
<td>5.89%</td>
<td>85.7%</td>
</tr>
<tr>
<td></td>
<td>0.93%</td>
<td>207%</td>
</tr>
<tr>
<td></td>
<td>0.50%</td>
<td>301%</td>
</tr>
<tr>
<td>Douban-Book</td>
<td>24.7%</td>
<td>13.3%</td>
</tr>
<tr>
<td></td>
<td>3.30%</td>
<td>68.8%</td>
</tr>
<tr>
<td></td>
<td>1.72%</td>
<td>122%</td>
</tr>
<tr>
<td>Douban-Movie</td>
<td>35.5%</td>
<td>16.7%</td>
</tr>
<tr>
<td></td>
<td>11.3%</td>
<td>48.0%</td>
</tr>
<tr>
<td></td>
<td>5.15%</td>
<td>84.8%</td>
</tr>
<tr>
<td>Flixster</td>
<td>31.5%</td>
<td>22.6%</td>
</tr>
<tr>
<td></td>
<td>2.75%</td>
<td>88.5%</td>
</tr>
<tr>
<td></td>
<td>0.70%</td>
<td>168%</td>
</tr>
<tr>
<td>Last.fm</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.3: Percentage improvement of GeneralTIM with RR-ComIC over VanillalC & Copying, where the fixed $B$-seed set is chosen to be the 101st to 200th ones from the VanillalC order

<table>
<thead>
<tr>
<th></th>
<th>VanillaC</th>
<th>Copying</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_{A/B} = 0.1$</td>
<td>2.16%</td>
<td>133%</td>
</tr>
<tr>
<td></td>
<td>1.12%</td>
<td>419%</td>
</tr>
<tr>
<td></td>
<td>0.71%</td>
<td>676%</td>
</tr>
<tr>
<td>Douban-Book</td>
<td>4.38%</td>
<td>236%</td>
</tr>
<tr>
<td></td>
<td>1.49%</td>
<td>737%</td>
</tr>
<tr>
<td></td>
<td>0.87%</td>
<td>1283%</td>
</tr>
<tr>
<td>Douban-Movie</td>
<td>10.6%</td>
<td>134%</td>
</tr>
<tr>
<td></td>
<td>0%</td>
<td>352%</td>
</tr>
<tr>
<td></td>
<td>0%</td>
<td>641%</td>
</tr>
<tr>
<td>Flixster</td>
<td>3.76%</td>
<td>398%</td>
</tr>
<tr>
<td></td>
<td>2.65%</td>
<td>1355%</td>
</tr>
<tr>
<td></td>
<td>1.65%</td>
<td>2525%</td>
</tr>
<tr>
<td>Last.fm</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.4: Percentage improvement of GeneralTIM with RR-ComIC over VanillalC & Copying, where the fixed $B$-seed set is randomly chosen

Table 4.3 shows the percentage improvement of our algorithms over the two baselines, for the case of selecting the 101st to 200th nodes output by VanillalC as the fixed $B$-seed set. As can be seen, GeneralTIM performed consistently better than both baselines, and in many cases by a large margin.
Table 4.5: Percentage improvement of GeneralTIM with RR-ComIC over VanillaIC & Copying, where the fixed $B$-seed set is chosen to be the top-100 nodes by VanillaIC.

<table>
<thead>
<tr>
<th></th>
<th>VanillaIC</th>
<th>Copying</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$q_{A\emptyset} = 0.1$</td>
<td>$0.1$</td>
</tr>
<tr>
<td>Douban-Book</td>
<td>0.34%</td>
<td>0.34%</td>
</tr>
<tr>
<td></td>
<td>0.3%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Douban-Movie</td>
<td>0.64%</td>
<td>0.64%</td>
</tr>
<tr>
<td></td>
<td>0.54%</td>
<td>0.54%</td>
</tr>
<tr>
<td></td>
<td>0.36%</td>
<td>0.36%</td>
</tr>
<tr>
<td>Flixter</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Last.fm</td>
<td>1.73%</td>
<td>1.73%</td>
</tr>
<tr>
<td></td>
<td>1.38%</td>
<td>1.38%</td>
</tr>
<tr>
<td></td>
<td>0.80%</td>
<td>0.80%</td>
</tr>
</tbody>
</table>

Table 4.4 shows the percentage improvement of GeneralTIM over VanillaIC and Copying baselines when the fixed $B$-seeds were chosen uniformly at random from $V$. As can be seen, GeneralTIM was significantly better except when comparing to VanillaIC. This is not surprising, as when $B$-seeds were chosen randomly, they were unlikely to be influential, so it is rather safe to ignore them when selecting $A$-seeds, matching what VanillaIC essentially does.

Table 4.5 shows the results when the fixed $B$-seed set was chosen to be the top-100 nodes from VanillaIC. These nodes represent the most influential ones under the IC model that can be found efficiently in polynomial time. We remark that VanillaIC and Copying are equivalent in this case. Here the advantage of GeneralTIM is less significant. On Flixter, the three algorithms achieve the same influence spread.

Overall, considering Tables 4.3, 4.4, and 4.5 all together, we can see that in the vast majority of all test cases, GeneralTIM outperformed the two baselines, often by a large margin. This demonstrates that GeneralTIM is robust with respect to different $B$-seeds. Furthermore, in real-world scenarios, the $B$-seed set may simply consist of “organic” early adopters, i.e., users who adopt the product spontaneously. The robustness of GeneralTIM is thus highly desirable as it is often difficult to foresee which users would actually become organic early adopters in real life.

Also, in our model the influence probabilities on edges were assumed to be independent of the product; without this assumption it is expected that Copying and VanillaIC would perform even more poorly. If we additionally assume that the GAPs are user-dependent, VanillaIC would deteriorate further.

---

5Recall that influence maximization is NP-hard under the IC model, and thus the optimal seed set of cardinality 100 is difficult to obtain. The seeds found by VanillaIC can be regarded as a good proxy.
In contrast, our GeneralTIM and RR-set generation algorithm (RR-ComIC) can be easily adapted to both these scenarios.

4.8.2 Learning Global Adoption Probabilities

Extracting Signals from Data

For Flixster and Douban, we learned GAPs from the available timestamped rating data. These datasets can be viewed as action logs in the following sense. Each entry is a quadruple \((u, i, a, t_{u,i,a})\), indicating \(u\) performed action \(a\) on item \(i\) at time \(t_{u,i,a}\). We counted a rating quadruple as one adoption action and one informing action: If someone rated an item, she must have been informed of it first; we assume only adopters rate items.

A key challenge is how to find actions that can be mapped to informing events that do not lead to adoptions. Fortunately, there are special ratings providing such signals in Flixster and Douban. Flixster allows users to indicate if they “want to see” a movie, or are “not interested” in one. We mapped both signals to the actions of a user being informed of a movie. Douban allows users to put items into a wish list. Thus, if a book/movie was in a user’s wish list, we treated it as an informing action.

Learning Methodology and Results

Consider two items \(A\) and \(B\) in an action log. Let \(R_A\) and \(I_A\) be the set of users who rated \(A\) and who were informed of \(A\), respectively. Clearly, \(R_A \subseteq I_A\). Thus,

\[
q_{A|\emptyset} = \frac{|R_A \setminus R_{B <_{rate} A}|}{|I_A \setminus R_{B <_{inform} A}|};
\]

where \(R_{B <_{rate} A}\) is the set of users who rated both items with \(B\) rated first, and \(R_{B <_{inform} A}\) is the set of users who rated \(B\) before being informed of \(A\).

Next, \(q_{A|B}\) is computed as follows:

\[
q_{A|B} = \frac{|R_{B <_{rate} A}|}{|R_{B <_{inform} A}|};
\]

Similarly, \(q_{B|\emptyset}\) and \(q_{B|A}\) can be computed in a symmetric way.

GAPs learned from Flixster and Douban. Tables 4.6 – 4.8 demonstrate selected GAPs learned from Flixster, Douban-Book, and Douban-Movie datasets. Here we not only show the estimated probabilities, but also give 95% confidence intervals (the standard Wald interval) [27]. By the defi-
Table 4.6: Selected GAPs learned for pairs of movies in Flixster

| A                                  | B                   | $q_{A|\emptyset}$ | $q_{A|B}$ | $q_{B|\emptyset}$ | $q_{B|A}$ |
|------------------------------------|---------------------|--------------------|-----------|--------------------|-----------|
| Monster Inc.                       | Shrek               | .88 ± .01          | .92 ± .01 | .92 ± .01          | .96 ± .01 |
| Gone in 60 Seconds                 | Armageddon          | .63 ± .02          | .77 ± .02 | .67 ± .02          | .82 ± .02 |
| Harry Potter: Prisoner of Azkaban  | What a Girl Wants   | .85 ± .01          | .84 ± .02 | .66 ± .02          | .67 ± .02 |
| Shrek                              | The Fast and the Furious | .92 ± .02        | .94 ± .01 | .80 ± .02          | .79 ± .02 |

Table 4.7: Selected GAPs learned for pairs of books in Douban-Book

| A                                  | B                   | $q_{A|\emptyset}$ | $q_{A|B}$ | $q_{B|\emptyset}$ | $q_{B|A}$ |
|------------------------------------|---------------------|--------------------|-----------|--------------------|-----------|
| The Unbearable Lightness of Being  | Norwegian Wood      | .75 ± .01          | .85 ± .02 | .92 ± .01          | .97 ± .01 |
| Harry Potter: Philosopher’s Stone  | Harry Potter: Half-Blood Prince | .99 ± .01        | 1.0 ± 0   | .97 ± .01          | .98 ± .01 |
| Stories of Ming Dynasty III       | Stories of Ming Dynasty VI | .94 ± .01        | 1.0 ± 0   | .88 ± .03          | .98 ± .01 |
| Fortress Besieged                 | Love Letter         | .89 ± .01          | .91 ± .03 | .82 ± .02          | .83 ± .03 |

Table 4.8: Selected GAPs learned for pairs of movies in Douban-Movie

![Histograms](image.png)

Figure 4.5: Complementary effects learned from data: The histogram of all $(q_{A|B} - q_{A|\emptyset})$ and $(q_{B|A} - q_{B|\emptyset})$ values on Flixster, Douban-Book, and Douban-Movie (10000 pairs of items each).

nition of GAPs (Section 4.3), we can treat each GAP as the parameter of a Bernoulli distribution. Consider any GAP, denoted $q$, and let $\bar{q}$ be its estimated value from action log data. The confidence interval of $\bar{q}$ is computed as follows:

$$
\left[ \bar{q} - 1.96\sqrt{\bar{q}(1-\bar{q})/n_q}, \quad \bar{q} + 1.96\sqrt{\bar{q}(1-\bar{q})/n_q} \right],
$$

114
where $n_q$ is the number of samples used for estimating $q$.

**Complementary effects.** Figure 4.5 plots the histogram of all $(q_{A|B} - q_{A|\emptyset})$ and $(q_{B|A} - q_{B|\emptyset})$ values on Flixster, Douban-Book, and Douban-Movie. For each dataset, we selected the top 10K pairs of items ranked by the number of common ratings received. The plots clearly demonstrate that complementarity and substitutability exist in the data.

### 4.8.3 Experimental Settings with Learned Adoption Probabilities

**Baselines.** In the following experiments, we compare our approximation algorithm, namely GeneralTIM with RR-ComIC / RR-ComIC++ and Sandwich Approximation, to several baselines commonly used in the influence maximization literature.

1. **HighDegree**: choose the $k$ highest out-degree nodes as seeds;
2. **PageRank**: choose the $k$ nodes with highest PageRank score as seeds;
3. **Random**: choose $k$ seeds uniformly at random.

**Items and Global Adoption Probabilities.** The following pairs of items are tested:

- **Flixster**: We chose movies *Monster Inc* as $\mathcal{A}$ and *Shrek* as $\mathcal{B}$, where $Q = \{.88, .92, .92, .96\}$.
- **Douban-Book**: We chose books *The Unbearable Lightness of Being* as $\mathcal{A}$ and *Norwegian Wood* as $\mathcal{B}$, and $Q = \{.75, .85, .92, .97\}$.
- **Douban-Movie**: We chose movies *Fight Club* as $\mathcal{A}$ and *Se7en* as $\mathcal{B}$, and $Q = \{.84, .89, .89, .95\}$.
- **Last.fm**: There is no signal in the data to indicate informing events, thus our learning method in Section 4.8.2 is not applicable. As a result, we used synthetic $Q = \{.5, .75, .5, .75\}$.

**Other Parameters.** Unless otherwise stated, following the convention we set $k = 50$ as the number of $\mathcal{A}$-seeds to mine. For GeneralTIM, we set $\ell$ to 1 so that a success probability (of obtaining approximate solutions) of $1 - 1/|V|$ was ensured [137]. We set parameter $\varepsilon$ to be 0.5 for RR-ComIC and RR-ComIC++, as this choice strikes the balance between efficient running time
and high quality of seeds, as we shall show in Figure 4.6. For the input $B$-seed set, we ran VanillaIC to extract 200 seeds and took the bottom 100 to be $B$-seeds. This is to simulate the situation where the $B$-seeds are moderately influential.

All algorithms were implemented in C++ and compiled using g++ 03 optimization. All experiments were conducted on an openSUSE server with 2.93GHz CPUs and 128GB RAM.
4.8.4 Results and Analysis

Effects of $\varepsilon$. We first evaluated the effects of $\varepsilon$ on GeneralTIM with RR-ComIC and RR-ComIC++. As mentioned in Section 4.6, the parameter $\varepsilon$ controls the trade-off between approximation ratio and efficiency. The larger $\varepsilon$ is, the more RR-sets will be generated. Figure 4.6 depicts influence spread and running time (log-scale) side-by-side, as a function of $\varepsilon$. We can see that as $\varepsilon$ goes up from 0.1 to 0.5, the running time of both versions of GeneralTIM (RR-ComIC & RR-ComIC++) decreases dramatically, often by orders of magnitude, while the quality of seed sets in terms of influence spread (Influence Maximization) are almost completely unaffected (the largest difference among all test cases is only 0.45%). This allows us to set $\varepsilon = 0.5$ to achieve best efficiency without sacrificing seed set quality much.

Figure 4.6 also illustrates that RR-ComIC++ was consistently faster than RR-ComIC, up to one order of magnitude. For instance, on Douban-Book, RR-ComIC++ was 10.4, 11.4, 12.8, 11.9, and 11.7 times as fast as RR-ComIC when $\varepsilon$ was set to 0.1, 0.2, 0.3, 0.4, and 0.5 respectively.

Quality of Seed Sets. The quality of seeds (output by an algorithm) is measured by the influence spread it achieves. We evaluated the spread of seed sets computed by all algorithms by MC simulations with 10000 iterations to ensure a fair comparison. As can be seen from Figure 4.7, our approximation algorithm GeneralTIM was consistently the best in all test cases, often by a significant margin. More specifically, GeneralTIM with RR-ComIC++ was 99.9%, 12.5%, 2.7%, and 12.6% better than the next best algorithm on Flixster, Douban-Book, Douban-Movie, and Last.fm respectively. HighDegree typically has good performance, especially in graphs with many nodes having very large out-degrees (Douban-Book and Douban-Movie), while PageRank has good quality seeds only on Last.fm. Random is consistently the worst.

Running Time. We plot the running time of GeneralTIM with RR-ComIC and RR-ComIC++ ($\varepsilon = 0.5$) and compare to Greedy with MC simulations, as depicted in Figure 4.8. Note that the maximum value of the Y-axis is set to be two weeks of time, and touching it means the algorithm did not finish within this limit. On all datasets, we can see that GeneralTIM with RR-ComIC and RR-ComIC++ was about two to three orders of magnitude faster than Greedy. In addition, RR-ComIC++ is 7.1, 11.7, 8.0, and 2.2 times as fast as than RR-ComIC on Flixster, Douban-Book, Douban-Movie, and Last.fm respectively. We omit the running time of HighDegree, PageRank, and Random since they are typically very efficient [35,37,39].

Scalability Tests on Synthetic Graphs. We then used larger synthetic graphs to conduct scalability tests. We generated power-law random graphs.
of 0.2, 0.4, ..., up to 1 million nodes with a power-law degree exponent of 2.16 [37]. These graphs have an average degree of about 5. The GAPs were set to be the same as Flixster. As can be seen from Figure 4.9, the running time of both algorithms grew linearly w.r.t. graph size, demonstrating good scalability. Consistent with Figure 4.6 and Figure 4.8, RR-ComIC++ was consistently faster than RR-ComIC in all graphs we tested.
Approximation Factors by Sandwich Approximation. Recall from Section 4.7 that the approximation factor yielded by Sandwich Approximation is data-dependent:

$$\sigma(S_{sand}) \geq \max \left\{ \frac{\sigma(S_\nu)}{\nu(S_\nu)}, \frac{\mu(S^*_\sigma)}{\sigma(S^*_\sigma)} \right\} \cdot (1 - 1/e - \varepsilon) \cdot \sigma(S^*_\sigma).$$
To see evaluate the effectiveness of Sandwich Approximation in real-world graphs, we computed $\sigma(S_\nu)/\nu(S_\nu)$, as although $S_\sigma^*$ is unknown due to NP-hardness of Influence Maximization, Sandwich Approximation is guaranteed to have an approximation factor of at least $(1 - 1/e - \varepsilon)\sigma(S_\nu)/\nu(S_\nu)$. Note that in the GAPs learned from data, both $q_{B|A} - q_{B|\emptyset}$ and $q_{A|B} - q_{A|\emptyset}$ are small, hence “friendly” to Sandwich Approximation as mentioned in Section 4.7. Thus, we further “stress tested” the algorithm with more adversarial settings: We set $q_{A|\emptyset} = .3$, $q_{A|B} = .8$, $q_{B|A} = 1$ and vary $q_{B|\emptyset}$ from \{.1, .5, .9\}; This ensures mutual complementarity.

Table 4.9 presents the results on all datasets with both learned GAPs and artificial GAPs. As can be seen, with real GAPs, the ratio was quite close to 1, matching our intuition. For artificial GAPs, the ratio was not as high. E.g., in the case of $q_{B|\emptyset} = 0.5$, $\sigma(S_\nu)/\nu(S_\nu)$ ranges from 0.628 (Last.fm) to 0.969 (Douban-Movie), which correspond to an approximation factor of 0.40 and 0.61 (omitting $-\varepsilon$). Even the smallest ratio 0.492 would yield a decent factor at about 0.3. This has shown that Sandwich Approximation is fairly effective and robust for solving non-submodular cases of Problem 4.

### 4.9 Discussion and Future Work

Our work in this chapter opens up a number of interesting avenues for future research. One direction is to design more efficient approximation algorithms or heuristics for Influence Maximization: e.g., whether near-linear time algorithm is still available for these problems is open. Another direction is to fully characterize the entire GAP space $Q$ in terms of monotonicity and submodularity properties. In addition, an important direction is to extend the model to multiple items. Given the current framework, ComIC can be extended to accommodate $k$ items, if we allow $k \cdot 2^{k-1}$ GAP parameters — for each item, we specify the probability of adoption for every combination of other items that have been adopted. However, how to simplify the model and make it tractable, how to reason about the complicated two-way or multi-way competition and complementarity, how to analyze monotonicity and submodularity, and how to learn GAP parameters from real-world data, etc. remain interesting challenges.

Last but not the least, as mentioned in Section 4.3.1 the current ComIC model assumes user homogeneity and the set of GAPs – $q_{A|\emptyset}$, $q_{A|B}$, $q_{B|\emptyset}$, and $q_{B|A}$ – are the same for all users. It is worth considering an extension of the ComIC model to characterize heterogeneous users. Let $T$ be the number of user types and let $[T] := \{1, 2, \ldots, T\}$ denote the set of all types. For
each type $i \in [T]$, there is a set of type-specific adoption probabilities: $q^j_{A|\emptyset}$, $q^j_{A|B}$, $q^j_{B|\emptyset}$, and $q^j_{B|A}$. For any user $u$ of type $i$ in the propagation process, she decides to adopt $A$ or $B$ using the aforementioned type-specific probabilities. The rest of the model remains the same.

Under the extended ComIC model, one may consider investigate if the submodularity and monotonicity results still hold: Establishing submodularity for a subset of the parameter space is important as it will enable the application of the generalized TIM algorithm and the Sandwich Approximation technique proposed in this chapter. In addition, how to leverage available datasets to learn the larger set of parameters ($4T$ adoption probabilities in total) is challenging. In principle, the methods described in Section 4.8.2 can still be used, but one may face data sparsity issues since the new model is in finer granularity and has much more parameters.
Chapter 5

Recommendations with Attraction, Aversion, and Social Influence Diffusion

5.1 Introduction

In this chapter, we describe how social influence makes big impact in a widely-used data mining application: recommender systems. The primary goal of a recommender system is to infer user interest from data (e.g., numeric ratings, text reviews, or even binary adoption data) and suggest a list of personalized relevant items to each user. Relevant and accurate suggestions boost user engagement, deliver better user experiences, which are highly beneficial to the growth of the service. In this digital era, Recommender systems are now ubiquitous, and can be found in almost every major Internet services such as Amazon (product recommendation), Netflix (movie recommendation), Spotify (music recommendation), Facebook and LinkedIn (friend recommendation). We refer interested readers to surveys [3, 127] for comprehensive treatments on a large variety of computational models and techniques employed by modern recommender systems.

To come up with high-quality and highly relevant recommendations, one of the most crucial tasks is to accurately model and infer user interests. Importantly and naturally, users’ content consumption patterns evolve over time. For example, a user may be attracted towards content that is popular, content recommended to her by a service, or content being enjoyed by her friends. Alternatively, users may get tired of certain types of content, e.g.,
romantic comedy movies, and desire to consume something different and new.

A key challenge for recommender systems is accurately modeling such user preferences as they evolve over time. Although traditional matrix factorization approaches can be extended to incorporate temporal dynamics of user behavior [92,93], such extensions do not identify or explicitly analyze the factors that influence the drift in interests.

A “classic” factor influencing user interests is attraction: users may be attracted to content they are exposed to repeatedly and often (such as, e.g., a song played often on the radio). This phenomenon, known in psychology as the “mere-exposure effect” [148], is natural and intuitive, and is the main premise behind advertising [54,75]. Nonetheless, repetition and/or overexposure can also have the opposite effect, leading to aversion: recent research argues that users often desire serendipitous, novel, previously unseen content [1,5,115,130]. This notion is also quite natural and intuitive, but is usually not taken into account by recommender systems, yielding overspecialized, predictable recommendations [5,115].

Importantly, a third factor affecting a user’s interests is social influence: users may feel attracted to content consumed and liked by their friends. Trend adoption through “word-of-mouth” or “viral” marketing is a well documented phenomenon [26,50], and has been extensively studied since the seminal paper by Kempe et al. [86]. Nonetheless, to the best of our knowledge, the effect of social influence on interests, and its implications for recommender systems, has received attention only recently [70,82,88,112,130,152]. Furthermore, those existing work in social recommendation rarely consider the dynamics of influence cascades and how modelling and computational issues in recommender systems would be affected by those complex stochastic processes. Instead, they merely consider the effects of social influence within each user’s ego-network (neighbourhood).

Incorporating these influence factors in a recommender system raises several challenges. To begin with, under attraction and aversion, a recommender can no longer be treated as a passive entity: recommendations it makes may alter user interests, pushing them either towards or away from certain topics. Hence, traditional methods that merely profile a user and then cater to this specific profile may fall short of keeping up with these dynamics. Second, social influence implies that recommendation decisions to different users cannot be made in isolation anymore: as recommendations alter a user’s interests through attraction and aversion, social influence can spread these changes, resulting in an interest cascade. Therefore, optimal recommenda-
tion decisions across users need to be computed globally, taking into account the joint effect they have over the user’s social network.

In this chapter, we make the following contributions:

• We formulate a global recommendation problem in the presence of attraction, aversion, and social influence. In particular, we propose a mathematical model that incorporates these phenomena, and study the steady state behavior of user interests as a function of the recommender’s strategy in selecting which items to show to users. Under this model, we seek the optimal recommendation strategy, i.e., one that maximizes the users’ social welfare in steady state (Section 5.3).

• We show that, for a large recommender item catalog, obtaining the optimal recommendation strategy amounts to solving a quadratically-constrained quadratic optimization problem (QCQP). Though this problem may not be convex, we present a semi-definite program (SDP) relaxation that can be solved in polynomial time. In many cases, this solution is also guaranteed to be an optimal solution; when the solution is not optimal, we show how a solution with a provable approximation guarantee can be constructed through randomization. We discuss how to determine whether the solution is optimal, and identify special cases for which an optimal solution is always reached, and randomization is unnecessary (Section 5.4).

• We provide evidence for the existence of attraction and aversion in three real-life rating datasets. We do so by developing and applying a method for learning the weight (i.e., importance) of these factors from rating data (Section 5.5). Applied to three real life datasets, our method indicates that between 14.0% to 18.6% of users show strong aversive or attractive behavior (Section 5.6).

• We conduct extensive experiments on real world datasets, and show that our recommendation algorithm is 15.4% to 107.4% better than a baseline algorithm in terms of social welfare achieved (Section 5.6).

As we shall see, our work assumes that users are generally aware of items consumed by their friends (so that social influence is effective). There are plenty of real-life examples: For instance, friends often discuss popular movies and TV series with each other. In addition, content streaming services such as Netflix and Spotify allow users to import their social network data (e.g., from Facebook) and post activities as feed. We also assume that the service
host has access to social network connections amongst its users, and that the host is capable of computing pairwise influence strength amongst its users by, e.g., using timestamped action logs stored in its databases [68].

Our data analysis indicates that (i) the phenomena of attraction and aversion are present in real-life datasets and (ii) accounting for them can lead to significant gains in the improvement of recommendations. Figure 5.1 provides a quick illustration of these two facts (see Section 5.6 for a detailed account on the derivation these two figures). Figure 5.1(a) presents the distribution of a score measuring aversion and attraction among different users in MovieLens (with $-1$ indicating users with the strongest aversive behavior, and $+1$ indicating users with the strongest attractive behavior). About 7.0% of users are strongly aversive (score $\leq -0.5$) while 9.0% are strongly attractive (score $\geq 0.5$). Accounting for such users can lead to a significant impact on recommendations: as shown in Figure 5.1(b), the user social welfare more than doubles when incorporating this knowledge in recommendation decisions. Though there are clearly many factors of user behavior that are not accounted for in our analysis, we believe that these two facts, along with the SDP relaxation yielding optimal recommendations, indicate that investigating and accommodating for such phenomena is both important and tractable.
5.2 Related Work

There has been a significant interest in modeling the temporal dynamics of user interests for various settings close to ours [92,125,130,143]. Early work on matrix factorization (MF) by Koren [92] incorporated time-variant user profiles, an approach that we also adopt. We depart from this line of work by modeling, and also including in the MF process (see Section 5.5), factors that impact these drifts, including attraction, aversion, and social influence.

Several studies have highlighted the need for serendipity and diversity in the context of recommender systems, both of which relate to the notion of aversion we describe here. The need for serendipity was first identified by McNee et al. [115]. To address this, Yu et al. [146] and Abbassi et al. [1] proposed algorithms for recommending items that maximize a score that combines both relevance to a user as well as diversity. Ge et al. [60] focused on evaluating the lack of serendipity and diversity, and how it hurts the quality of recommendations. We depart from these works by modeling how recommendations themselves may instigate aversion or attraction among users, through a dynamic evolution of user interests.

Our approach to incorporating the effects of aversion is closer to Das Sarma et al. [130], which considered users that iteratively consume items in one out of several categories. They incorporate “boredom” and social influence in a manner similar to us: inherent item values decrease as a function of a weighted frequency of past consumption, and a user’s utility is averaged among her friend’s utilities. The authors provide bounds of the steady state performance of different consumption strategies under such dynamics. We depart by modelling user interests as multi-dimensional vectors, and using a factor-based model for user utilities, whose dynamics and steady state behavior cannot be captured by the (one-dimensional) model in [130].

The literature on social influence is vast, motivated by the viral marketing applications introduced by Domingos and Richardson [50] and further studied by Kempe et al. [86]. Our influence model is closer to gossiping [131], in that the interest/state of each user results from averaging the interests of her neighbors. Though we depart from classic gossiping protocols in that we incorporate additional dynamics (through attraction and aversion), similar techniques as in [131] could potentially be used to study our system in scenarios where interest evolution is asynchronous across users. In the context of matrix factorization, Jamali et al. [82] proposed incorporating the distance of a user’s profile to the average profile of users in their social circle as a regularization factor in MF. This is consistent with the social influence behavior we outline in Section 5.3.3. We depart from this work by modeling
dynamic profiles, and studying the additional effect of recommendations on
user profiles through attraction and aversion.

Semi-definite programming (SDP) relaxation for quadratically con-
strained quadratic programs (QCQP) lies at the core of our algorithmic con-
tributions. Building on the seminal work by Goemans and Williamson [62],
several papers have demonstrated classes of QCQPs for which an SDP re-
 laxation gives a constant approximation guarantee [120, 121, 145]. Moreover,
exact solutions of rank 1 were known to be attainable for several classes
of QCQP, including when the problem has one [25] or two quadratic con-
straints [14]. Of special interest is the case where the quadratic objective
involves non-negative off-diagonal elements, and constraints involve only
quadratic terms of one variable [150], as the attraction-dominant case of our
problem falls into this class (see Section 5.4.3). We refer the interested reader
to [140] for SDP in general, and to [111, 121] for applications to quadratic
programming.

5.3 Problem Formulation
In what follows, we present our mathematical model that describes how users
interact with a recommender system. We use bold script (e.g., $\mathbf{x}, \mathbf{y}, \mathbf{u}, \mathbf{v}$) to
denote vectors, and capital script (e.g., $A, B, H$) to denote matrices. For ei-
ther matrices or vectors, we use $\geq$ to indicate element-wise inequalities. For
symmetric matrices, we use $\succeq$ to indicate dominance in the positive semide-
finite sense; in particular, $A \succeq 0$ implies that $A$ is positive semidefinite. For
square matrices $A$, we denote by $\text{tr}(A), \text{diag}(A), \text{rank}(A)$ the trace, diagonal
and rank of $A$, respectively. Finally, given an $n \times m$ matrix $A$, we denote by
$\text{col}: \mathbb{R}^{n \times m} \rightarrow \mathbb{R}^{nm}$ the column-major order representation of $A$; i.e., $\text{col}(M)$
maps the elements of $A$ to a vector, by stacking the $m$ columns of $A$ on top of
each other.

5.3.1 Overview
Our model assumes that user interests are dynamic: they are affected both
by recommendations users receive, as well as by how other users’ interests
evolve. In particular, our model of user behavior takes into account the fol-
lowing factors:

1. Inherent interests. Our model accounts for an inherent predisposi-
tion users may have, e.g., towards particular topics or genres. This is
static and does not change through time.
2. **Social influence.** A user’s behavior can be affected by what people in her social circle (e.g., her friends or family) are interested in.

3. **Attraction.** As per the mere-exposure effect, users may exhibit attractive behavior: if a type of content is shown very often by the recommendation service, this might reinforce the desire of a user to consume it.

4. **Aversion.** Users may also exhibit aversive behavior: a user can grow tired of a topic that she sees very often, and may want to see something new or rare.

Under the joint effect of the factors above, suggestions made by the recommender instigate an *interest cascade* over the users. Suggestions alter user interests through attraction or aversion; in turn, these changes affect neighboring users as well, on account of their social behavior. These effects propagate dynamically over the users’ social network.

### 5.3.2 Recommender System and User Utilities

We now formally describe how each of the four factors mentioned above is incorporated in our model. We consider $n$ users receiving recommendations from an entity we call the recommender. We denote by $[n] \equiv \{1, 2, \ldots, n\}$ the set of all users. At each time step $t \in \mathbb{N}$, the recommender suggests an item to each user in $[n]$, selected from a catalog $C$ of available items. The user accrues a utility from the item recommended. As discussed below, the recommender’s goal is to suggest items that maximize the aggregate user utility, i.e., the social welfare.

Following the standard convention in recommender systems, we assume factor-based user utilities. At each $t \in \mathbb{N}$, each user $i \in [n]$ has an *interest profile* represented by a $d$-dimensional vector $\mathbf{u}_i(t) \in \mathbb{R}^d$. Moreover, the item recommended to user $i$ at time $t$ is represented by a $d$-dimensional *feature profile* $\mathbf{v}_i(t) \in \mathbb{R}^d$. Then, the expected rating a user $i$ would give to the item suggested to her at time $t$ is given by $F(\mathbf{u}_i(t), \mathbf{v}_i(t))$, where

$$F(\mathbf{u}, \mathbf{v}) = \text{def} \; \langle \mathbf{u}, \mathbf{v} \rangle = \sum_{k=1}^{d} u_k v_k,$$

i.e., the inner product between the interest and feature profiles [91,93]. Intuitively, each coordinate of a feature profile can be perceived as an item-specific feature, e.g., a movie’s genre or an article’s topic. The corresponding

---

1 In practice, Equation (5.1) best approximates *centered* ratings, i.e., ratings offset by a global average across users.
coordinate in an interest profile captures the propensity of the user to react positively or negatively to this feature.

We call \( F(u_i(t), v_i(t)) \) the utility of user \( i \) from the suggested item at time \( t \). Without loss of generality\(^2\), we assume that the item profiles \( v_i \in \mathbb{R}^d \) are normalized, i.e.: \( \|v_i(t)\|_2 = 1 \) for all \( i \in [n] \), \( t \in \mathbb{N} \). Under this assumption, given that a user’s profile is \( u_i \), the best item to recommend to user \( i \) is the one that yields the highest expected rating; indeed, this is

\[
\arg \max_{v \in \mathbb{R}^d: \|v\|_2 = 1} F(u, v) = u/\|u\|_2,
\]

i.e., the item that maximizes the utility of a user \( i \). Note that identifying items that maximize the aggregate utility across users (i.e., the sum of expected ratings to suggested items), is a natural goal for the recommender.

### 5.3.3 Interest Evolution

The evolution of user interests captures the four factors outlined in §5.3.1. At each time step \( t \in \mathbb{N} \), the interest profile of a user \( i \in [n] \) is chosen alternately between either a personalized or a social behavior. If personalized, the behavior of a user is again selected among three possible outcomes, each corresponding to inherent interests, attraction, and aversion, respectively.

The selection of which of these four behaviors takes place at a given time step is random, and occurs independently of selections at other users as well as selections at previous time slots. We denote with \( \beta \in [0, 1] \) the probability that the user selects a social behavior at time slot \( t \). The probability of selecting a personalized behavior is thus \( 1 - \beta \). Interests at these two distinct events are as follows:

**Personalized Behavior.** If a user’s interest is selected through a personalized behavior, the user selects her profile through one of the three personalized factors outlined in §5.3.1. In particular, for every \( i \in [n] \), there exist probabilities \( \alpha_i, \gamma_i, \delta_i \in [0, 1] \) such that \( \alpha_i + \gamma_i + \delta_i = 1 \), and:

- **Inherent interests.** With probability \( \alpha_i \), user \( i \) follows her inherent interests. That is, \( u_i(t) \) is sampled from a probability distribution \( \mu_i^0 \) over \( \mathbb{R}^d \). This distribution does not vary with \( t \) and captures the user’s inherent predisposition.

\(^2\)Note that \( F(u, v) = F(su, \frac{1}{s}v) \), for any scalar \( s \in \mathbb{R} \), so we can assume that either user or feature profiles have a bounded norm.
• **Attraction.** With probability $\gamma_i$, user $i$ selects a profile that is attracted to the items suggested to her in the past. To capture this notion denote by $V_i(t) = \{v_i(\tau)\}_{\tau \leq t}$ the history of (profiles of) items suggested to a user $i$. Then, the interest of a user under attraction is given by $g(V_i(t-1))$, a weighted average of the items suggested to it in the past. That is:

$$u_i(t) = g(V_i(t-1)) = \frac{\sum_{\tau=0}^{t-1} w_{t-\tau} v(\tau)}{\sum_{\tau=0}^{t-1} w_{t-\tau}}. \quad (5.2)$$

• **Aversion.** With probability $\delta_i$, user $i$ selects a profile that is repulsed by the items suggested to her in the past; that is, her interest profile is given by

$$u_i(t) = -g(V_i(t-1)) = -\frac{\sum_{\tau=0}^{t-1} w_{t-\tau} v(\tau)}{\sum_{\tau=0}^{t-1} w_{t-\tau}}. \quad (5.3)$$

To gain some intuition on Equation (5.2) and Equation (5.3), recall that a user’s utility at time $t$ is given by Equation (5.1). Therefore, a profile generated under Equation (5.2) implies that the suggestion that maximizes her utility at time $t$ would be one that aligns perfectly (i.e., points in the same direction as) the weighted average $g$ up to time $t-1$. In contrast, under the aversive behavior Equation (5.3), the same suggestion minimizes the user’s utility.

Note that the weighted average $g$ is fully determined by the sequence weights $\{w_\tau\}_{\tau \in \mathbb{N}}$. By selecting decaying weights, a higher importance can be placed on more recent suggestions.

**Social Behavior.** User $i$’s profile is selected through social behavior with probability $\beta$. Conditioned on this event:

• **Social Influence.** A user aligns her interests with a user $j$ selected from her social circle with probability $P_{ij}$. That is:

$$u_i(t) = u_j(t-1), \quad \text{with probability } P_{ij}, \quad (5.4)$$

where $\sum_j P_{ij} = 1$.

The probability $P_{ij} \in [0, 1]$ captures the influence that user $j$ has on user $i$. Note that users $j$ for which $P_{ij} = 0$ (i.e., outside $i$’s social circle) have no influence on $i$. Moreover, the set of pairs $(i, j)$ s.t. $P_{ij} \neq 0$, defines the social network among users. We denote by $P \in [0, 1]^{n \times n}$ the stochastic matrix
with elements $P_{ij}$, $i, j \in [n]$; we assume that $P$ is ergodic (i.e., irreducible and aperiodic) [58].

Under these dynamics, interests evolve in the form of a dynamic cascade: suggestions made by the recommender act as a forcing function, altering interests either through attraction or aversion. Such changes propagate across users through the social network.

### 5.3.4 Recommended Item Distribution

In practice, the recommender has access to a finite “catalog” of items. Recalling that feature profiles have norm 1, the recommender’s catalog can be represented as a set $C \subseteq \mathbb{B}$, where $\mathbb{B} = \{ \mathbf{v} \in \mathbb{R}^d : \|\mathbf{v}\|_2 = 1 \}$ is the set of items of norm 1 (i.e., the $d$-dimensional unit ball).

We assume that the recommender selects the items $\mathbf{v}_j(t) \in \mathbb{B}$ suggested to user $i \in [n]$ by sampling them from a discrete distribution $\nu_i$ over $\mathbb{B}$, whose support is $C$. Note that the expected feature profile of a suggested item is a weighted average among the vectors in $C$. As such, it belongs to the convex hull of catalog $C$; formally:

$$\bar{\mathbf{v}}_i = \int_{\mathbf{v} \in \mathbb{B}} \mathbf{v} d\nu_i \in \text{conv}(C), \quad (5.5)$$

Note that conv($C$) is a convex polytope included in $\mathbb{B}$.

As we will see later in our analysis (cf. Theorem 13), the steady state user utilities depend only on the expectations $\bar{\mathbf{v}}_i$, $i \in [n]$, rather than the entire distributions $\nu_i$. We will thus refer to $\{\bar{\mathbf{v}}_i\}_{i \in [n]}$ as the recommender strategy; it is worth keeping in mind that, given a $\bar{\mathbf{v}}_i \in \text{conv}(C)$, finding a $\nu_i$ such that (5.5) holds can be computed in polynomial time in $|C|$ (see also §5.4.4).

We further assume that the catalog $C$ is large; in particular, for large catalog size $|C|$, we have:

$$\text{conv}(C) \simeq \mathbb{B}. \quad (5.6)$$

This would be true if, for example, each item in the catalog are generated in an i.i.d. fashion from a distribution that covers the entire ball $\mathbb{B}$; this distribution need not be uniform. Formally, $\lim_{|C| \to \infty} \text{conv}(C) = \mathbb{B}$ w.p. 1 if, e.g., items in catalog $C$ are sampled independently from a probability distribution absolutely continuous to the uniform distribution on $\mathbb{B}$. We revisit the
issue of how to pick a distribution \( \nu_i \) given \( \bar{v}_i \), as well as how to interpret our results in the case of a finite catalog, in §5.4.4.

5.3.5 Recommendation Objective

Observe that, under the above dynamics, the evolution of the system is an Markov chain, whose state comprises the interest and feature profiles. We define the objective of the recommender as maximizing the social welfare, i.e., the sum of expected user utilities, in steady state. Formally, we wish to determine a strategy \( \{ \bar{v}_i \}_{i \in [n]} \) (and, hence, distributions \( \nu_i \)) that maximizes:

\[
\lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T} \sum_{i \in [n]} \langle u_i(t), v_i(t) \rangle = \lim_{t \to \infty} \sum_{i \in [n]} \mathbb{E}[\langle u_i(t), v_i(t) \rangle],
\]

where the equality above holds w.p. 1 by the renewal theorem [58]. It is important to note that, under the interest dynamics described in §5.3.3, optimal recommendations to a user \( i \) cannot be obtained independently of recommendations to other users: user \( i \)’s profile depends on recommendations made not only directly to this user, but also to any user reachable through \( i \)’s social network.

5.4 Algorithm Design

In this section, we discuss how the recommender selects which items to present to users to maximize the system’s social welfare. We begin by obtaining a closed-form formula for the social welfare in steady state, and then discuss algorithms for its optimization.

5.4.1 Steady State Social Welfare

Recall that \( \mu_i^0 \) is the inherent profile distribution of user \( i \in [n] \), and let \( \mu_i \) be the steady state distribution of the profile of user \( i \). We denote by \( \bar{u}_i = \int_{u \in \mathbb{R}^d} u d\mu_i \) and \( \bar{u}_i^0 = \int_{u \in \mathbb{R}^d} u d\mu_i^0 \) the expected profile of \( i \in [n] \) under the steady state and inherent profile distributions, respectively. Moreover, denote by \( \bar{U}, \bar{U}^0, \bar{V} \in \mathbb{R}^{n \times d} \) the matrices of dimensions \( n \times d \) whose rows comprise the expected profiles \( \bar{u}_i, \bar{u}_i^0, \bar{v}_i, i \in [n] \), respectively. Let also \( \Lambda, \Gamma, \Delta \in \mathbb{R}^{n \times n} \) be the \( n \times n \) diagonal matrices whose diagonal elements are the coefficients \( (1 - \beta) \alpha_i, (1 - \beta) \gamma_i, \) and \( (1 - \beta) \delta_i \), respectively.

Then, the steady state social welfare can be expressed in closed form according to the following theorem.
Theorem 13. The expected social welfare in steady state is:

\[ G(\bar{V}) \equiv \text{tr} \left[ (I - \beta P)^{-1} \left( A\bar{U}^{0}\bar{V}^{T} + (\Gamma - \Delta)\bar{V}\bar{V}^{T} \right) \right], \tag{5.7} \]

where \( \text{tr}(\cdot) \) denotes the matrix trace.

Proof. Observe that at any time step \( t \), the profiles \( u_i(t) \) and \( v_i(t) \) are independent random variables. Hence,

\[
\lim_{t \to \infty} \sum_{i \in [n]} \mathbb{E}[\langle u_i(t), v_i(t) \rangle] = \lim_{t \to \infty} \sum_{i \in [n]} (\mathbb{E}[u_i(t)], \mathbb{E}[v_i(t)])
= \sum_{i \in [n]} \langle \bar{u}_i, \bar{v}_i \rangle = \text{tr} \left( \bar{U}\bar{V}^{T} \right). \tag{5.8}
\]

Observe that by the linearity of expectation \( \mathbb{E}[g(V_i(t))] = \bar{v}_i \) for all \( t \in \mathbb{N} \) and \( i \in [n] \). Thus, for \( U(t) = [u_i(t)]_{i \in [n]} \in \mathbb{R}^{n \times d} \) the matrix of all user profiles at time \( t \), we get that

\[ \mathbb{E}[U(t)] = A\bar{U}^{0} + \beta P\mathbb{E}[U(t - 1)] + \Gamma\bar{V} - \Delta\bar{V}. \]

As \( \beta P \) is sub-stochastic and ergodic, the Perron-Frobenius theorem [58] implies that \( \bar{U} = \lim_{t \to \infty} \mathbb{E}[U(t)] \) exists and

\[ \bar{U} = A\bar{U}^{0} + \beta P\bar{U} + (\Gamma - \Delta)\bar{V}. \]

Solving the above linear system and substituting the solution for \( \bar{U} \) in (5.8) completes the proof. \( \square \)

An important consequence of Theorem 13 is that the steady state social welfare depends only on the expected profiles \( \bar{V} \), rather than the entire distributions \( \nu_i, i \in [n] \). Hence, determining the optimal recommender strategy amounts to solving the following quadratically-constrained quadratic optimization problem (QCQP):

**Global Recommendation**

Max.: \[ \text{tr} \left[ (I - \beta P)^{-1} \left( A\bar{U}^{0}\bar{V}^{T} + (\Gamma - \Delta)\bar{V}\bar{V}^{T} \right) \right] \]

subj. to: \[ \|\bar{v}_i\|^2 \leq 1, \text{ for all } i \in [n]. \] \tag{5.9}

where the norm constraint comes from Equation (5.6). Note that this is indeed a global optimization: to solve it, recommendations across different
users need to be taken into account jointly. This manifests in (5.9) through the quadratic term in the social welfare objective.

5.4.2 SDP Relaxation

The QCQP (5.9) is not necessarily convex. It is thus not a priori clear whether it can be solved in polynomial time. However, there is a way to reduce to a semi-definite program (SDP) relaxation, which can be solved in polynomial time. Interestingly, in many cases, the solution obtained for the SDP relaxation turns out to be an optimal solution to our original problem (5.9), and there is a simple and efficient test that can verify whether the obtained solution is optimal. Finally, when the solution is not optimal, it can be transformed to yield a constant-factor approximation. We are thus able to obtain a strong and elegant theoretical result for solving the GLOBALRECOMMENDATION problem.

It is important to note that the large-catalog assumption (5.6) is crucial to tractability: replacing the quadratic constraints with the linear constraints (5.5) does not lead to a problem that is amenable to an SDP relaxation. In fact, generic quadratic problems with linear constraints are known to be inapproximable, unless $P = NP$ [121].

Deriving an SDP Relaxation. We begin by describing first how to express (5.9) as “almost” an SDP, except for a rank constraint:

Theorem 14. There exists a symmetric matrix $H \in \mathbb{R}^{(nd+1) \times (nd+1)}$ and a convex polyhedral set $D \in \mathbb{R}^{nd+1}$ such that GLOBAL RECOMMENDATION (5.9) is equivalent to:

$$
\begin{align*}
\text{Max.:} & \quad \text{tr}(HY) \\
\text{subj. to:} & \quad Y \succeq 0, \; \text{diag}(Y) \in D, \; \text{rank}(Y) = 1,
\end{align*}
$$

(5.10)

where $Y \in \mathbb{R}^{(nd+1) \times (nd+1)}$.

Proof. Let $x = \text{col}(\vec{V}) \in \mathbb{R}^{nd}$ be the column-major order vector representation of the recommender’s strategy, and $b = \text{col}((I - \beta P)^{-1}A\bar{U}) \in \mathbb{R}^{nd}$ the vector representation of the linear term in (5.7). Moreover, for $Q = (I - \beta P)^{-1}(\Gamma - \Delta) \in \mathbb{R}^{n \times n}$, consider the following block-diagonal symmetric
matrix, where $\frac{Q+Q^T}{2}$ is repeated $d$ times:

$$H_0 = \begin{bmatrix}
\frac{Q+Q^T}{2} & 0 & \ldots & 0 \\
0 & \frac{Q+Q^T}{2} & \ldots & 0 \\
\ldots & \ldots & \ldots & \ldots \\
0 & 0 & \ldots & \frac{Q+Q^T}{2}
\end{bmatrix} \in \mathbb{R}^{nd \times nd}.$$ 

Under this notation, (5.9) can be written as

$$\begin{align*}
\text{Max.} & \quad b^T x + x^T H_0 x \\
\text{subj. to} & \quad x^2 \in \mathcal{D}_0,
\end{align*}$$

(5.11)

where $x^2 = [x_k^2]_{k \in [nd]} \in \mathbb{R}_+^{nd}$ results from squaring the elements of $x$, and $\mathcal{D}_0$ the set implied by the norm constraints:

$$\mathcal{D}_0 = \{ x' \in \mathbb{R}^n \mid \forall i \in [n], \sum_{j=1}^{nd} |x_{j \mod n = i \mod n} | x_{j} | \leq 1 \}.$$ 

Note that $\mathcal{D}_0$ is a convex polyhedral set defined by linear equality constraints. Moreover, (5.11) can be homogenized to a quadratic program without linear terms using the following standard trick (see also [111, 121, 140]). Introducing an auxiliary variable $t$, the objective can be replaced by $tb^T x + x^T H_0 x$, where $t$ satisfies the constraint $t^2 \leq 1$. Setting $y = (x, t) \in \mathbb{R}^{nd+1}$, this yields:

$$\begin{align*}
\text{Max.:} & \quad y^T H y \\
\text{subj. to:} & \quad y^2 \in \mathcal{D},
\end{align*}$$

(5.12)

where $H$ is the following symmetric matrix:

$$H = \begin{bmatrix}
H_0 & b/2 \\
b^T/2 & 0
\end{bmatrix} \in \mathbb{R}^{(nd+1) \times (nd+1)},$$

and

$$\mathcal{D} = \{ y' = (x', t') \in \mathbb{R}^{nd+1} \mid x' \in \mathcal{D}_0, t' \leq 1 \}.$$ 

To see that (5.12) and (5.11) are equivalent, observe that an optimal solution $(x, t)$ to (5.12) must be such that $t = -1$ or $t = +1$. If $t = +1$, then $x$ is an optimal solution to (5.11); if $t = -1$, then $-x$ is an optimal solution to (5.11). Finally, (5.12) is equivalent to (5.10), by setting $Y = yy^T$ and using the fact that $y^T H y = \text{tr}(Hyy^T)$. 

135
In particular, given an optimal solution $Y$ to (5.10), an optimal solution to $\text{GLOBALRECOMMENDATION}$ can be constructed as follows. Since $Y \succeq 0$ and $\text{rank}(Y) = 1$, there exists $y \in \mathbb{R}^{nd+1}$ such that $Y = yy^\top$. More specifically, $Y$ has a unique positive eigenvalue $\lambda$. If $e$ is the corresponding eigenvector, $y = (x, t) = \sqrt{\lambda} \cdot e$. An optimal solution to (5.9) is thus the matrix $\bar{V} \in \mathbb{R}^{n \times d}$ with column-major order representation $\text{col}(\bar{V}) = t \cdot x$.

Problem (5.10) is still not convex, on account of the rank constraint on $Y$. However, in light of Theorem 14, a natural relaxation for $\text{GLOBALRECOMMENDATION}$ is the following semi-definite program, resulting from dropping this rank constraint:

$$\text{SDP RELAXATION}$$

$$\begin{align*}
\text{Max.:} & \quad \text{tr}(HY) \\
\text{subj. to:} & \quad Y \succeq 0, \ \text{diag}(Y) \in \mathcal{D}.
\end{align*}$$

(5.13)

This is a relaxation, in the sense that it increases the feasible set: any solution to (5.10) will also be a solution to (5.13). Crucially, (5.13) is a convex SDP problem, and can be solved in polynomial time. Moreover, if it happens that the optimal solution $Y$ of (5.13) has rank 1, this solution is also guaranteed to be an optimal solution to (5.10), and can thus be used to construct an optimal solution to $\text{GLOBALRECOMMENDATION}$, by Theorem 14. If, on the other hand, $\text{rank}(Y) > 1$, we are not guaranteed to retrieve an optimal solution to (5.10). However, a solution with a provable approximation guarantee can still be constructed through a randomization technique, originally proposed by Goemans and Williamson [62].

**Approximation Algorithm.** Algorithm 10 summarizes the steps in the approach outlined above to solving $\text{GLOBALRECOMMENDATION}$. First, the algorithm obtains an optimal solution $Y$ to the SDP (5.13). It then tests if $\text{rank}(Y) = 1$, i.e., if this solution happens to have rank 1. If it does, then it is also a solution to (5.10), and an optimal solution to (5.9) can be constructed as outlined in the proof of Theorem 14. In particular, $Y$ can be written as $Y = yy^\top$, where $y = (x, t) \in \mathbb{R}^{nd+1}$ can be computed from the unique positive eigenvalue of $Y$ and its corresponding eigenvector. The optimal solution to (5.9) can subsequently be obtained as the matrix $\bar{V} \in \mathbb{R}^{n \times d}$ that has a column-major order representation $\text{col}(V) = t \cdot x$.

If, on the other hand, $\text{rank}(Y) > 1$, the algorithm returns a vector $(x, t)$ constructed in a randomized fashion. In particular, the algorithm returns
Algorithm 10: Global Recommendation Algorithm

Data: Model parameters $A, \Gamma, \Delta, \beta, P, \bar{U}^0$
Result: Expected item profiles $\bar{V}$

begin
1 Solve SDP Relaxation (5.13) and let $Y$ be its solution
2 if rank($Y$) = 1 then
3 Let $\lambda > 0$ be the unique positive eigenvalue of $Y$, and $e$ the corresponding eigenvector
4 $(x,t) \leftarrow \sqrt{\lambda} \cdot e$
5 else
6 Construct a factorization $Y = Z^T Z$, where $Z \in \mathbb{R}^{(nd+1) \times (nd+1)}$
7 Sample a random $u \in \mathbb{R}^{nd+1}$ from $\mathcal{N}(0, I_{nd+1})$
8 $\sigma \leftarrow \text{sgn}(Z^T u)$
9 $D \leftarrow \text{a diagonal matrix containing } \sqrt{\text{diag}(Y)}$
10 $(x,t) \leftarrow D \sigma$
11 else
12 Let $\bar{V} \in \mathbb{R}^{n \times d}$ be such that col($\bar{V}$) = $t \cdot x$

The vector $\sqrt{\text{diag}(Y)}$, namely the square root of $Y$’s diagonal elements, with each coordinate multiplied by a random sign (+1 or −1). The random sign vector $\sigma \in \{-1, +1\}^{nd+1}$ used in this multiplication is constructed as follows. Given that $Y \succeq 0$, there exists a matrix $Z \in \mathbb{R}^{n \times n}$ that factorizes $Y$, i.e., $Y = Z^T Z$. Such a matrix can be obtained in polynomial time from the eigendecomposition of $Y$. Having $Z$, the algorithm proceeds by sampling a random vector $u \in \mathbb{R}^{nd+1}$ from a standard Gaussian distribution. Then, $\sigma$ is the binary vector computed by applying the sign operator on the coordinates of vector $Z^T u$.

The resulting random $y = (x,t) \in \mathbb{R}^{nd+1}$ is guaranteed to be a feasible solution to (5.10). Most importantly, the following approximation guarantee for the quality of the corresponding solution to Global Recommendation can be provided:

Theorem 15 (Ye [121]). Let $G^*, G_s$ be the maximal and minimal values of the social welfare $G$ given by (5.7), evaluated over the feasible domain of (5.9). Let also $\bar{V}$ the solution generated by Algorithm 10 when rank($Y$) > 1. Then

\[
\frac{G^* - E_u[\bar{V}]}{G^* - G_s} \leq \frac{\pi}{2} - 1 < \frac{4}{7},
\]

where the expectation $E_u[\cdot]$ is over the Gaussian vector $u$. 

137
The existence of a simple test (namely \( \text{rank}(Y) = 1 \)) verifying that the solution produced by Algorithm (10) is optimal is important. In fact, in Section 5.6, we study an extensive set of instances, involving several social network topologies and combinations of aversive and attractive behavior. In each and every instance studied, Algorithm 10 yielded an optimal solution. Hence, although the quadratic program (5.9) is not known to be within the class of problems that can be solved exactly through an SDP relaxation, the experiments in Section 5.6 suggest that a stronger guarantee than the one provided by Theorem 15 is attained in practice.

### 5.4.3 Solvable Special Cases

Though for generic instances of (5.9) we cannot obtain a better theoretical guarantee than Theorem 15, there are specific instances of (5.9) for which optimality is always attained, and the approximation through randomization is not necessary. As these cases are also of practical interest, we briefly review them below.

**Attraction Dominance.** Consider a scenario where (a) \( \gamma_i > \delta_i \) for all \( i \in [n] \) and (b) \( \bar{U} \geq 0 \). Intuitively, (a) implies that attraction to proposed content is more dominant than aversion to content, while (b) implies that user profile features take only positive values. Hence, the matrix \( H \) in Theorem 14 has nonnegative off-diagonal elements. Although the QCQP (5.9) in this case is not convex, it is known that in this specific case Algorithm 10 provides an optimal, rank-1 solution [150].

**Uniform Aversion Dominance.** Consider a scenario where (a) all parameters are uniform across users, i.e., \( \gamma_i = \gamma \) and \( \delta_i = \delta \), for all \( i \in [n] \), and (b) \( \gamma < \delta \), i.e., aversion dominates user behavior. In this case, the QCQP (5.9) is convex and can thus be solved optimally by standard interior point methods in polynomial time [25].

**No Personalization.** Consider the scenario where the same item is recommended to all users, i.e., \( \bar{v}_i(t) = \bar{v}(t), \forall i \in [n] \). In this case, Global Recommendation reduces to a quadratic objective with a single quadratic constraint, in which case even though the problem may not be convex, Algorithm 10 is guaranteed to find a rank-1, optimal solution [25].

**No Social Network.** In the case where \( \beta = 0 \), and there is no social component to the optimization, the social welfare (5.7) becomes separable in \( \bar{v}_i \), i.e., \( G(\bar{V}) = \sum_{i \in [n]} G_i(\bar{v}_i) \), where \( G_i \) is a quadratic function. Then, the optimization is separable, and a solution to (5.9) can be obtained by solving \( \max_{\bar{v}_i \in \mathbb{R}^d : \|\bar{v}_i\| \leq 1} G(\bar{v}_i) \) for each \( i \in [n] \); these are again quadratic problems
with a single quadratic constraint, and can be solved exactly by Algorithm 10 [25].

5.4.4 Finite Catalog

Recall that our analysis assumes (5.6), which becomes applicable for a large catalog \( C \) covering the unit ball \( \mathbb{B} \). We describe below how a computed profile \( \bar{v}_i, i \in [n] \) can be used to construct a distribution \( \nu_i \) over catalog \( C \).

If \( \bar{v}_i \in \text{conv}(C) \), the recommender can select probabilities \( \nu_i(v) \), for \( v \in C \), that satisfy (5.5); this equality, along with the positivity constraints, and the constraint \( \sum_{v \in C} \nu_i(v) = 1 \) (as \( \nu_i \) is a distribution), are linear, and define a feasible set. Thus, finding a probability distribution satisfying (5.5) (i.e., that lies in the feasible set) is a linear program, which can be solved in polynomial time.

If, on the other hand, \( \bar{v}_i / \in \text{conv}(C) \), the same procedure can be applied to the projection of \( \bar{v}_i \) to \( \text{conv}(C) \). Given that \( \text{conv}(C) \) is a convex polytope, this can again be computed in polynomial time. Moreover, under (5.6), if the catalog \( C \) is large this projection will be close to the optimal value \( \bar{v}_i \).

5.5 Parameter Learning

In this section, we provide an algorithm for validating the existence of attraction and aversion phenomena in real datasets. In short, our approach involves incorporating aversion and attraction parameters into matrix factorization [91,93]; we treat parameters \( \alpha_i, \gamma_i \) and \( \delta_i \) as regularization terms, which are learned through cross validation.

Extending MF. We focus on datasets that comprise ratings generated by users, at known times (such as the datasets to be used in §5.6). More specifically, we assume access to a dataset represented by tuples of the form \( (i, j, r_{ij}, t) \) where \( i \in [n] \equiv \{1, \ldots, n\} \) is the ID of a user, \( j \in [m] \equiv \{1, \ldots, m\} \) the ID of an item, \( r \in \mathbb{R} \) the feedback (rating) provided by user \( i \) to item \( j \) and \( t \in [T] \) the time at which the rating took place. Denoting by \( E \subset [n] \times [m] \) the pairs appearing in tuples in this dataset, recall that matrix factorization (MF) amounts to constructing profiles \( u_i \in \mathbb{R}^d, v_j \in \mathbb{R}^d \) that are solutions to:

\[
\min_{u_i, v_j} \sum_{i \in [n], j \in [m], (i, j) \in E} (r_{ij} - \langle u_i, v_j \rangle)^2 + \lambda \sum_{i \in [n]} \|u_i\|^2_2 + \mu \sum_{j \in [m]} \|v_j\|^2_2
\]

(5.14)
where $\lambda, \mu > 0$ are regularization parameters to be learned through cross validation. Though this is not a convex problem, it is typically solved either through gradient descent or alternating least squares techniques, both of which perform well in practice [91, 93].

We incorporate attraction and aversion in this formulation as follows. First, at any time step $t \in [T]$, the profile of a user $i$ is given by $u_i(t) \in \mathbb{R}^d$. Let $E_i(t) \subseteq [m]$ be the set of items rated by user $i$ at time $t$ and

$$\mathcal{V}_i(t) = \{v_j : j \in E_i(\tau), 1 \leq \tau \leq t\}$$

be the set of items the user has interacted with up to time $t$ (inclusive). As in §5.3, we denote by $g(\mathcal{V}_i(t))$ the weighted average of items in $\mathcal{V}_i(t)$. Then, we propose obtaining $u_i(t)$ as solutions to:

$$\min_{u_i(t), i \in [n], t \in [T]} \sum_{i \in [n], t \in [T]} \sum_{(i,j) \in E_i(t)} (r_{ij} - \langle u_i(t), v_j \rangle)^2 + \sum_{i \in [n], t \in [T]} \left(\|u_i(t) - \alpha_i u_i^0 - (\gamma_i - \delta_i) g(\mathcal{V}_i(t))\|_2^2 + \kappa \|u_i(t)\|_2^2\right), \tag{5.15}$$

where $u_i^0, v_j$ are computed through standard MF (5.14), and $\alpha_i, \gamma_i, \delta_i, i \in [n]$ and $\kappa$ are also treated as regularization parameters, to be learned through cross validation. Note that, in contrast to (5.14), (5.15) is a simple linear regression problem, and the profiles $u_i(t)$, where $i \in [n], t \in [T]$, can be computed in closed form.

**Learning Procedure.** Based on this approach, our algorithm for learning the vectors $\alpha, \gamma, \delta$ is outlined in Algorithm 11. First, we learn the inherent profiles $u_i^0$ and the item feature profiles $v_j$ by solving (5.14), through stochastic gradient descent. Then, we use these profiles to learn $\alpha_i, \gamma_i, \delta_i$ through cross validation. In particular, we split the ratings dataset in $k$ folds, and use $k - 1$ folds as a training set, and one fold as a test set. In our evaluation, we set $k = 5$. We learn $u_i(t)$ by solving (5.15) on this restricted dataset. Using these, we compute the square error on the test set as:

$$\text{SE}_{\text{test}} = \sum_{(i,j,t) \in \text{test}} (r_{ij} - \langle u_i(t), v_j \rangle)^2.$$  

We repeat this process across $k$ folds and obtain an average $\overline{\text{SE}_{\text{test}}}$. We compute vectors $\alpha, \gamma, \delta$ that minimize the average $\overline{\text{SE}_{\text{test}}}$. Note that this is a function of the regularization parameters of (5.15), i.e.,
Algorithm 11: Attraction-Aversion Learning Algorithm

**Data:** Rating data

**Result:** \((\alpha_i, \gamma_i, \delta_i), \text{ for all } i \in [n]\)

1. Obtain \(u_i^0, \forall i \in [n]\) and \(v_j, \forall j \in [m]\) through standard MF (5.14)
2. Compute \(g(V_i(t)), \forall i \in [n], \forall t \in [T]\)
3. Split the dataset into \(k\) folds
4. Initialize values in \(\alpha, \gamma, \delta\) uniformly at random from \([0, 1]\)
5. Project \((\alpha_i, \gamma_i, \delta_i)\) to the set \(\{(x, y, z) \geq 0 : x + y + z = 1\}, \forall i \in [n]\)
6. repeat
7. \((\alpha, \gamma, \delta, \kappa) \leftarrow (\alpha, \gamma, \delta, \kappa) - \rho \nabla \mathcal{SE}_{\text{test}}(\alpha, \gamma, \delta, \kappa)\)
8. Project \((\alpha_i, \gamma_i, \delta_i)\) to \(\{(x, y, z) \geq 0 : x + y + z = 1\}, \forall i \in [n]\)
9. until change of \(\mathcal{SE}_{\text{test}}\) in two consecutive iterations is small enough

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Flixster</th>
<th>FilmTipSet</th>
<th>MovieLens</th>
</tr>
</thead>
<tbody>
<tr>
<td># users</td>
<td>4.6K</td>
<td>443</td>
<td>8.9K</td>
</tr>
<tr>
<td># items</td>
<td>25K</td>
<td>4.3K</td>
<td>3.8K</td>
</tr>
<tr>
<td># ratings</td>
<td>1.8M</td>
<td>118K</td>
<td>1.3M</td>
</tr>
<tr>
<td># SN edges</td>
<td>44K</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Table 5.1: Datasets statistics

\(\mathcal{SE}_{\text{test}} = \mathcal{SE}_{\text{test}}(\alpha, \gamma, \delta, \kappa)\). As (5.15) admits a closed form solution, so does \(\mathcal{SE}_{\text{test}}(\alpha, \gamma, \delta, \kappa)\). Using this, we find \(\alpha, \gamma, \delta\) through projected gradient descent, requiring that they sum to 1.

### 5.6 Experiments

We performed experiments to evaluate our parameter learning and social welfare-maximizing algorithms on three real-world rating datasets: Flixster, FilmTipSet, and MovieLens, as well as several synthetically generated traces. The implementations were done in Matlab and we made use of the CVX library [74] to solve the SDP in Algorithm 10. All experiments were run on a server with AMD Opteron 6272 CPUs (eight cores at 2.1GHz) and 128GB memory.

**Dataset Preparations.** We first describe the three real-world rating datasets, whose basic statistics are summarized in Table 5.1.

Flixster is a social movie rating site. The original dataset, collected by Jamali et al. [82], comprises 1M users, 14M undirected friendship edges, and
8.2M timestamped ratings (ranging from 0.5 to 5 stars). We used Graclus [48] to extract a dense subgraph of the social network. Further, we filtered out users and movies with less than 100 ratings so that there is enough data to learn temporal profile vectors. This left us with a core of 4.6K users, 44K edges, and 25K movies.

FilmTipSet\(^3\) is Swedish movie fans community. The data was originally published for a research competition in the CAMRa workshop\(^4\). It has 16K users, 67K movies, 2.8M timestamped ratings (on the scale of 1 to 5). We selected users rating no less than 100 movies in both 2004 and 2005. This gives a core of 443 users, 4.3K movies, and 118K ratings.

The third dataset is MovieLens\(^5\) (the largest version with 10 million users). We focused on users that have rated at least 20 movies in the year of 2000. Note that there is no social network in MovieLens. FilmTipSet contains some social networking information, which we however did not use in our analysis due to its extreme sparsity (85 edges for the 443 core users).

![Figure 5.2: The decreasing trend of Test RMSE, RMSE\(_{\alpha}\), RMSE\(_{\gamma}\), and RMSE\(_{\delta}\)](image)

5.6.1 Evaluation of Parameter Learning

**Learning on Synthetic Data.** We first ran Algorithm 11 on a synthetically generated dataset to examine its accuracy. We set \(n = 100\), \(T = 100\), and

\(^3\)http://www.filmtipset.se/, last accessed on September 24, 2015

\(^4\)http://www.dai-labor.de/camra2010/, last accessed on September 24, 2015

\(^5\)http://grouplens.org/datasets/movielens/, last accessed on September 24, 2015
Figure 5.3: Interest evolution probabilities learned on synthetic data, compared against the generated ground-truth values for accuracy

Each user $i \in [n]$ consumes one random item at every time step $t \in [T]$. For all users $i$, we generated “ground-truth” $\alpha_i$, $\gamma_i$, and $\delta_i$ uniformly at random from $[0,1]$ and normalize them so that $\alpha_i + \gamma_i + \delta_i = 1$. The expected inherent interest profiles and item profiles were generated uniformly at random from $[0,1]^d$. Interest profiles evolved according to the dynamics in Section 5.3.3, with $\beta = 0$, and weighted average $g$ with equal weights. At each step, users “generate” ratings which were computed by taking the inner product of appropriate profile vectors.

Both the learning rate $\rho$ and regularization parameter $\kappa$ in Algorithm 11 were set to 0.001 (determined by cross validation). The convergence condition of Algorithm 11 was set to be the change in $\text{SE}_{\text{test}}$ being smaller than $10^{-6}$. We repeated the process ten times with different random starting points and report the results obtained in the repetition that gives the smallest $\text{SE}_{\text{test}}$.

Let $\alpha^\ell_i$, $\gamma^\ell_i$, and $\delta^\ell_i$ be the learned evolution probabilities. We used RMSE to define the learning error w.r.t. $\alpha_i$’s:

$$\text{RMSE}_\alpha = \sqrt{\frac{\sum_{i=1}^n |\alpha_i - \alpha^\ell_i|^2}{n}}.$$  

RMSE$\gamma$ and RMSE$\delta$ can thus be defined in the same way. Figure 5.2 shows the decrease of these RMSEs as the number of iterations goes up. At convergence, they were 0.08, 0.58 and 0.56 respectively. In addition, this figure
shows that the test RMSE (computed as $\sqrt{SE_{\text{test}}/|\text{test}|}$) dropped steadily as the learning proceeds, which finally converged to 0.02 in a total of 5300 iterations.

In Figure 5.3, we show a scatter plot of the ground-truth probabilities and the learned probabilities (at convergence). Each data point has one ground-truth probability value in x-coordinate and the corresponding learned value in y-coordinate. The $y = x$ line indicates points for which the learned and ground truth probabilities are equal. As can be seen, the algorithm recovered $\alpha_i$’s almost perfectly, while the results for $\gamma_i$’s and $\delta_i$’s were also reasonably good. This gave us confidence in deploying the algorithm on real-world rating data, in which ground-truth parameters are not known.

**Learning on Real Data.** For each dataset, we sorted the ratings in chronological order and split them into $T = 10$ time steps. A single time step corresponded to 3, 1.2, and 2.5 calendar months in Flixster, MovieLens, and FilmTipSet, respectively. We then ran Algorithm 11 with learning rate $\rho = 0.001$, regularization parameter $\kappa = 0.001$, and number of latent features $d = 10$. 

![Graphs of Figure 5.4](image1.png)

**Figure 5.4:** Learned values of $\alpha_i$ on three real-world datasets

![Graphs of Figure 5.5](image2.png)

**Figure 5.5:** Values of $\gamma_i - \delta_i$ on three real-world datasets
Figure 5.4 shows the distributions of values learned for $\alpha_i$’s. Furthermore, to compare the number of attraction-dominant users and the number of aversion dominant users, in Figure 5.5 we display the distribution of $\gamma_i - \delta_i$, along with a Gaussian distribution fitted by data within the interval $[\mu - 1.8\sigma, \mu + 1.8\sigma]$, where $\mu$ and $\sigma^2$ is the mean and variance of all $\gamma_i - \delta_i$’s. As can be seen, the empirical distribution has tails that are heavier than the Gaussian (at about -0.5 and 0.5), indicating the existence of strongly aversive and strongly attracted users.

For reference, we also compared the average test RMSE on five-fold cross-validation achieved by our model and by standard MF. For standard MF, we implement the stochastic gradient descent method as in [93] with $d = 10$, learning rate 0.002, and regularization parameters determined by cross-validation. As shown in Figure 5.6, profiles learned by Algorithm 11 outperform standard MF in rating prediction, lowering the test RMSE by 11.8%, 11.9%, 6.18% on Flixster, FilmTipSet, and MovieLens respectively.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{test_rmse.png}
\caption{Test RMSE comparisons between our extended MF model and the standard MF model}
\end{figure}

### 5.6.2 Social Welfare Performance

Next, we evaluated Algorithm 10, hereafter referred to as GRA, and compare the social welfare it yielded with a baseline that ignores interest evolution. This baseline would recommend to each user the item profile maximizing the user’s utility under the inherent profile computed by standard MF. For each
user $i \in [n]$, this is
\[ v_i = \frac{u_i^0}{||u_i^0||} \text{, for all } i \in [n]. \]

It is thus easy to see that $||v_i||_2 = 1$ and it is in fact co-linear w.r.t. $u_i^0$. We hereafter refer to this baseline as $MF-Local$. In all experiments, following the literature of social influence propagation and maximization [35, 86], we set the influence probability of user $j$ on user $i$ to be $1/deg^{in}(i)$, where $deg^{in}(i)$ is the in-degree of node $i$ in the network graph.

5.6.2.1 Experiments on Synthetic Networks

We started by evaluating the social welfare achieved by $GRA$ and $MF-Local$ on three different random networks that mimic the structure of a social network: Forest-Fire [12], Kronecker [99], and Power-Law [4]. For each type, we considered the following settings: Forest-Fire with forward and backward burning probability being 0.38 and 0.32 respectively, Kronecker with initiator matrix being $[0.9, 0.5; 0.5, 0.3]$, and Power-Law with exponent 2.1.

We varied the size of network graphs (i.e., number of users, $n$), the value of $\beta$ (i.e., users’ tendency of getting influenced by friends), and the difference between $\gamma_i$ and $\delta_i$, to evaluate their effects on the performance of our $GRA$ algorithm in comparison to $MF-Local$. Unless otherwise noted, $\alpha_i$’s, $\gamma_i$’s, $\delta_i$’s, and inherent user profiles are sampled randomly and the process was repeated ten times, of which we took the average social welfare. Also, $d$ was fixed to be 10. In all cases, we plot the relative gap in social welfare, i.e.,
\[ \frac{SocialWelfare_{GRA} - SocialWelfare_{MF-Local}}{|SocialWelfare_{MF-Local}|}. \]

Effect of Network Size. We tested five different values for $n$: 10, 50, 100, 150, and 200 for Forest-Fire and Power-Law, and 16, 32, 64, 128, and 256 for Kronecker (by definition a random Kronecker graph has $2^w$ nodes where $w \in \mathbb{N}_+$ is the number of iterations of Kronecker product taken in the generation process [99]). As can be seen from Figure 5.7, the gap between $GRA$ and $MF-Local$ was close to 10% for small graphs, but increases on all three networks for larger values of $n$: $GRA$ achieved twice as much social welfare as $MF-Local$, for $n = 200$.

Effect of $\beta$. In this test, we varied the value of $\beta$ from 0 up to 0.5. Network size is fixed at 100 for Forest-Fire and Power-Law, and 128 for Kronecker. Figure 5.8 shows that $GRA$ significantly outperformed $MF-Local$, and more interestingly, the relative gap increased as $\beta$ increased. This intuitively sug-
suggests that when the influence among users is higher, ignoring the joint effect of recommendations becomes more detrimental to maximizing the social welfare.

**Effects of** $\gamma_i - \delta_i$. Next, we tested different values of $\gamma_i - \delta_i$, representing cases from extreme aversion dominance to attraction dominance. Network size was $n = 100$ and $\beta = 0.25$, while $\alpha_i = \alpha$, $\gamma_i = \gamma$ and $\delta_i = \delta$ for all $i \in [n]$. We set $\alpha$ s.t. $\alpha(1 - \beta) = 0.25$, and vary $\gamma - \delta$, where $\alpha + \gamma + \delta = 1$. Relative gaps are shown in Figure 5.9. All in all, we see that gaps are far
more pronounced in the strongly aversive regime, as targeting to the existing profiles of users leads to suboptimal recommendations. For values less than -0.3, the social welfare under \( MF-Local \) was actually negative; it went up as \( \gamma - \delta \) increases, i.e., users tend towards attractive behavior. In contrast, the social welfare of \( GRA \) was always positive, and always greater than the one under \( MF-Local \). As a result, there is a large gap for values at less than -0.3; the relative gap become small (but still positive) near -0.1, and then steadily increased.

It is important to note that in all evaluations of \( GRA \) over synthetic datasets, as well as the ones listed below on real datasets, \( GRA \) returned an optimal solution. That is, for all inputs tested, the matrix \( Y \) computed had rank 1. Hence, although the QCQP problem (5.9) is not known to be solvable in polynomial time, in practice, \( GRA \) outperformed the guarantee of Theorem 15.

5.6.2.2 Experiments on Real Data

We next compare the social welfare attained on Flixster, FilmTipSet, and MovieLens by \( GRA \) and \( MF-Local \). For FilmTipSet and MovieLens where there is no social network considered, \( GLOBAL \ RECOMMENDATION \) is separable and can thus be \textit{parallelized} (see Section 5.4.3): we can divide users into arbitrary subsets, run \( GRA \) on each of them, and then combine the total social welfare over all subsets as the final solution without any loss.
Figure 5.10: Social welfare, where FX(0.1) and FX(0.5) denote Flixster with $\beta = 0.1$ and 0.5 respectively; FT denotes FilmTipSet and ML denotes MovieLens.

To improve the scalability of GPA over Flixster, and parallelize its execution, we adopted the following heuristic. First, we split the social graph into 50 subgraphs using Graclus. Then, we solved SDP on each subgraph separately. Note that, in effect, this optimization ignored the edges between subgraphs, and thus only yielded an approximation to the social welfare.

Figure 5.10 illustrates the performance of GRA and MF-Local on those datasets, where the values of $\alpha_i$, $\gamma_i$, and $\delta_i$ were all from the learning results in Section 5.6.1 and we set the dimensionality $d$ to 10. We can see that GRA is significantly superior to MF-Local: on FilmTipSet (1461 vs. 757) and MovieLens (11092 vs. 4926), it achieved approximately twice the social welfare.

On Flixster, tested two cases for $\beta$: 0.1 and 0.5, representing weak and strong social behavior respectively. For GRA, we adopted the aforementioned clustering-based heuristic to compute $\bar{v}_i$’s, and evaluated the welfare achieved by GRA in two ways: (i) simply calculating the welfare on the subgraph and taking the sum over all subgraphs (termed GRA-heuristic); (ii) taking the $\bar{v}_i$’s to calculate the social welfare on the entire graph (termed GRA). The values computed by method (ii) were only slightly different from (i), indicating that our clustering heuristic closely followed the true social welfare, while enabling parallelization. The relative gain of GRA-heuristic over MF-Local was 39.0% when $\beta = 0.1$ and 13.4% when $\beta = 0.5$. The run-
ning time of GRA was reasonably good, e.g., on a subgraph of Flixster with 94 nodes and 276 edges, GRA finished in 90 seconds.

In summary, through extensive empirical evaluation on both real and synthetic data, we have demonstrated that first, the phenomenon of interest evolution, especially attraction and aversion, can indeed be observed from real-world rating data, and second, both of our learning algorithm and global recommendation algorithm are highly effective in their respective tasks.

5.7 Discussion and Future Work

Extensions to accommodate user segmentation. Notice that the framework presented in this chapter assumes equal treatment of all users, as the objective function (5.7) is an unweighted sum of the expected steady-state utility of all users. We can extend it to capture scenarios where the service provider of the recommender system wishes to provide differential treatments on different segments of users. For instance, the system may want to further optimize for VIP users who have premium subscriptions. Alternatively, we may want to weight active users more than inactive ones.

To this end, we extend the GLOBAL RECOMMENDATION problem formulation to use the weighted sum of the expected steady-state utility of all users as the maximization objective. More specifically, each user $i \in [n]$ is associated with a weight $w_i \in (0, 1]$. It indicates the importance of a particular user in the optimization.

By linearity of expectation and the linearity of inner product operations, the expected weighted social welfare in steady state is thus:

\[
\lim_{t \to \infty} \sum_{i \in [n]} w_i \cdot \mathbb{E}[(u_i(t), v_i(t))] = \lim_{t \to \infty} \sum_{i \in [n]} \langle w_i \cdot \mathbb{E}[u_i(t)], \mathbb{E}[v_i(t)] \rangle
\]

\[
= \sum_{i \in [n]} \langle w_i \tilde{u}_i, \tilde{v}_i \rangle = \text{tr} \left( (W\tilde{U}) \tilde{V}^T \right),
\]

where the matrix $W$ is a diagonal matrix with $W_{ii} = w_i$ and $W_{ij} = 0$ whenever $i \neq j$.

Then, by Theorem 13 we can obtain the formulation of the WEIGHTED GLOBAL RECOMMENDATION problem as follows.

\[
\text{Max.: } \text{tr} \left[ W(I - \beta P)^{-1} \left( AU^0 \tilde{V}^T + (\Gamma - \Delta)\tilde{V}\tilde{V}^T \right) \right]
\]

\[
\text{subj. to: } \| \tilde{v}_i \|^2 \leq 1, \text{ for all } i \in [n].
\]
Note that the property of the problem does not change, as (5.16) is still a QCQP, and hence our solution framework based on SDP relaxation (see Section 5.4.2) remains applicable. More precisely, this is because Theorem 14 can be easily extended to hold for (5.16): in its proof where we constructed an equivalent problem formulation consisting of a SDP and a rank-one constraint, we can simply modify vector $b$ and matrix $Q$ by multiplying $W$ to the left of $(I - \beta P)^{-1}$.

**Future Work.** In the experiments, we have exploited parallelizing execution over weakly connected partitions of the social graph. This highlights an approach for scalable, parallelizable solutions to the SDP relaxation. Further opportunities for improving efficiency exist: the sparse, block structure of the matrices in our SDP was not exploited by the generic solvers we employed. Investigating solutions that exploit this structure for higher efficiency is an interesting future direction.

Moreover, although the QCQP that expresses our problem is not known to be exactly solvable through an SDP relaxation, all solutions we obtained through our experiments were actually optimal. Understanding if optimality holds for a wider class than the ones presented in Section 5.4.3 is also an important open problem. Besides, there are clearly many phenomena beyond attraction, aversion and social influence that may affect a user’s interests. The quadratic nature of our problem arises from the standard factor-based model for utilities: understanding if other phenomena inducing drift on profiles can also be cast in this framework is also an interesting open question.
Chapter 6

Summary and Future Research

6.1 Summary

The rapid growth of online social networks and social media has opened up many opportunities in computational social influence research. Motivated by (i) the connection between social influence and two prevalent data mining applications – viral marketing and recommender systems, and (ii) the gaps between theories of computational social influence and practical applications, in this dissertation we have made the following contributions to the field of computational social influence:

- We proposed three novel influence diffusion models: Linear Thresholds with Valuation (LT-V), Linear Thresholds with $K$ advertisers ($K$-LT), and Comparable Independent Cascade (ComIC). All three models have more expressive power compared to the classical diffusion models such as IC and LT [86].

- We studied various optimization problems under these models, such as profit maximization over social networks (PROMAX), fair seed allocation for competitive viral marketing from the host perspective, and influence maximization for two complementary propagating entities.

- In the context of recommender systems, we modeled the dynamics of user interest evolution using social influence as well as users’ attraction and aversion behaviors, and solve the optimal recommendation problem using semi-definite programming techniques.

In Chapter 2, we extended the classical LT model by incorporating prices and valuations to capture monetary aspects in product adoption. We studied
an NP-hard profit maximization problem (ProMAX) under the LT-V model. In this problem, the objective function is submodular but non-monotone w.r.t. the seed set, for any fixed price vector. We proposed the PAGE algorithm which dynamically determines the optimal personalized discounts for targeted seeds. Our experimental results showed that PAGE is both efficient and effective, outperforming several intuitive baseline algorithms in all aspects evaluated, e.g., the expected total profit achieved and running time.

Our work in Chapter 3 also took a step towards closing this gap between influence maximization and real-world viral marketing. More specifically, we considered the fact that social networks are owned by a service provider (host) in real life, and competing advertisers cannot simply autonomously set up their campaigns. We considered a setting where hosts sells viral marketing as a service to multiple competing companies. We posed the novel problem of Fair Seed Allocation in which the host must allocate influential users to competing companies to guarantee “the bang for the buck” for the advertisers is as balanced as possible. We showed that the problem is NP-hard, and developed an efficient greedy heuristic called Needy-Greedy, as well as two exact algorithms based on dynamic programing and integer linear programming. We perform simulations on three real-world networks and show that our algorithms are both effective and efficient, significantly outperforming baseline allocation schemes such as random and round-robin.

In Chapter 4, we proposed the Comparative Independent Cascade (ComIC) model that characterizes both competition and complementarity relationships between two different propagating items, to any degree possible, and tackled the influence maximization problem when there are two complementary products. We identified parameter subspaces of ComIC under which submodularity and monotonicity are satisfied by influence spread functions, and develop non-trivial extensions to the RR-set techniques to obtain approximation algorithms. For non-submodular settings, we also devised a Sandwich Approximation scheme to achieve data-dependent approximation bounds. Our experiments demonstrated the effectiveness and efficiency of the novel approximation algorithms we proposed.

In Chapter 5, we applied social influence propagation, together with users’ attraction and aversion behaviors, to recommender systems to model user interests. Taking these factors into account, we first proposed an interest evolution model in which the probability distributions of user interest profiles form a Markov chain. Devising recommendation strategies in this setting is challenging, as the items suggested in previous time steps have direct impact in all users’ future interests due to the interest cascade triggered by influence propagation. Therefore, effective recommendations should be made in a
holistic fashion, in contrast to conventional methods that computes relevant items for each user separately. Our objective was to maximize the total expected utility of all users, which we formulated as a quadratically constrained quadratic program (QCQP). We showed that the optimal recommendation problem is NP-hard, and devised an approximation algorithm using SDP relaxation. We learned interest evolution parameters by adapting low-rank matrix factorization, and our experiments showed that the SDP-based algorithms clearly outperform baseline recommendation methods which do not take interest evolution into account.

6.2 Discussions and Future Work

The field of Computational Social Influence is still booming and there are abundant opportunities for future research.

Influence Modeling. Most stochastic propagation models (including those presented in the dissertation) do not account for non-social channels through which people obtain information, such as TV, newspapers, web browsing, etc. This calls for a unified model which takes both social influence and non-social information channels into consideration. One possible way is to augment the social network graph $G = (V, E)$ by creating dummy nodes and edges that represent those channels and their influence on users. For each non-social channel $c$, we create a dummy node $x_c$ that has a directed edge to every user-node $u \in V$ in the social network. The edge weight on $(x_c, u)$ represents the propensity to which a user $u$ obtains information from or gets influenced by channel $x_c$. As a future work, one could study the stability of influence optimization algorithms when such non-social information channels are accounted for.

Another direction is to factor in the heterogenous nature of activeness of social network users. For example, teenagers and college students are typically keen on modern social networking and social media apps and hence have more exposure to viral information, while people spent less time on those technologies (e.g., busy professionals) do not have the same level of exposure. Thus, one may consider segmenting social network users into multiple types, each corresponding to a different level of activeness, and study influence propagation models in the presence of such heterogeneity.

One may also consider multiple types of social relationships: family, co-worker, classmates, research collaborators, people sharing the same political belief, or same sports interest, etc. Similarly, the information being propagated over the social network can also be of multiple types, and depending on
the type of the relationship, the strength of influence may vary. For example, suppose Alice and Bob became friends for they are both enthusiastic fans of the Vancouver Canucks. The mutual influence between Alice and Bob may be stronger when the information or product being propagated is about ice hockey, as the same topic and relationship type (ice hockey) may create high synergy. Studying relevant influence optimization problems in such a setting is also worth considering.

**Influence Optimization.** The reliance of submodularity to obtain approximation guarantees is ubiquitous in the literature. Prior to the proposal of the Sandwich Approximation method in Chapter 4, resorting to heuristics is the only way of solving maximization problems in the absence of submodularity. However, for Sandwich Approximation to be effective, one should derive an upper and/or lower bound submodular function that is reasonably tight w.r.t. the original objective function. A general principle is to analyze if there is any aspect of a non-submodular model that can be changed, tuned, or relaxed, so that submodularity is satisfied. The difficulty of such an analysis varies from model to model. Thus, further studies on how to apply Sandwich Approximation effectively on influence maximization under well-known non-submodular models – e.g., The LT model with hard-wired node thresholds [86], the WPCLT-model [24] – is both interesting and worthwhile.

One may also be interested in designing optimization algorithms when additional constraints are imposed. For example, in a viral marketing campaign, many users could very well be inundated with signals, or impressions of its neighbors having performed a promoted action (e.g., purchased a product). To avoid this and ensure pleasant user experience, a social network host may want to place an upper bound on how many viral-marketing-related impressions each user will receive in a specific time frame. It is not difficult to extend simulation-based algorithms to choose seeds under such constraints (e.g., greedy algorithm with MC simulations), but how to extend the much more advanced and efficient TIM [137] or IMM [136] algorithms remains a challenge: First, the RR-set sampling component in General TIM assumes order-independence\(^1\), which may not hold due to the constraint on the number of impressions. Second, there is no guarantee that submodularity will still be satisfied. We remark that Sandwich Approximation may offer a viable solution in this case, as it is obvious that the influence spread computed without the impression limit constraint is naturally an upper bound on the

\(^1\)As a specific example, in the IC model, the order in which active in-neighbors attempt to influence an inactive node does not matter for spread estimation and seed selection.
influence spread computed with the constraint imposed\(^2\). Though this discussion sheds some lights on the power of Sandwich Approximation, a full-scope study in the future is certainly worthwhile.

**Influence Learning.** Another common assumption often made in computational social influence is that optimization algorithms have explicit knowledges on the influence strength between social network users. Getting ground-truth is extremely challenging, and there has been considerable work on learning influence strength in social networks \([68, 122, 129, 141]\). Although the thesis does not focus on learning, we stress that the accuracy of learnt influence strength w.r.t. the ground-truth is crucial in the sense that if the learnt values deviate substantially from the ground-truth, then even an optimal influence maximization algorithm is likely to return suboptimal solutions. He and Kempe \([78]\) found that for the IC model, if each of the estimated influence probability (one per edge) has more than 20\% relative error, the noise of the data would dominate the objective function and lead to a significant risk of suboptimal solutions. This indicates the need of searching for (\(i\)) capable influence learning methods that have proven bound on relative errors (e.g., well under 20\%) and (\(ii\)) influence maximization algorithms that are less susceptible to inaccurate and noisy input data.

**Distributed Influence Maximization.** The vast majority of the computational social influence literature assumes that for problems like influence maximization, the entire social network graph can fit into the main memory of a single machine. However, real-world social networks are getting larger and larger by the second\(^3\), and thus there is a pressing need to devise efficient and effective distributed algorithms.

Recently, Lucier et al. \([110]\) proposed a distributed MapReduce-based \([45]\) algorithm – INFEst – for estimating the expected influence spread of any given seed set under the IC model. They assumed the *link-server model* for querying and inter-machine communications, which is commonly used in distributed computing \([32, 46, 134]\). Under the link-server model, the social network graph \(G = (V, E)\) is stored on the disk of a centralized server, and each query to the server on any node \(v \in V\) returns the set of in- and out-neighbors of \(v\), together with the relevant influence probabilities. For any given seed set \(S \subset V\) and any \(\epsilon \in (0, 1/4)\), the estimation \(\hat{\sigma}(S)\) by INFEst is an \((1 + 8\epsilon)\)-approximation to the true value \(\sigma(S)\), with high probability.

\(^2\)For simplicity, we restrict this discussion to single-item propagation models only, as competitive models may introduce additional complications.

\(^3\)For example, according to the official quarterly earnings report of Facebook in the first quarter of 2016, it adds 63 million monthly active users. This roughly translates into a growth rate of 8 new monthly active users per second. Source: [http://investor.fb.com/](http://investor.fb.com/).
They showed that for estimating the spread of a singleton set \( \{u\} \) within a factor of \( 1 + \epsilon \), the InfEst algorithm requires \( \Omega(\sqrt{|V|}) \) queries to the link server.

This work offers a significant first step toward devising distributed algorithms for influence maximization. A natural extension is to treat InfEst as a value oracle in the classic greedy algorithm (Algorithm 1), which results in a MapReduce-based algorithm for influence maximization. More specifically, in line 4 of Algorithm 1, we invoke InfEst to estimate \( f(S \cup \{u\}) \). Note that the greedy algorithm requires \( O(k|V|) \) calls to the spread estimation oracle, and in the case of InfEst, each of which requires \( \Omega(\sqrt{|V|}) \) queries to the link server as mentioned above.

For future work, it is certainly worthwhile to search for better distributed influence maximization algorithms (in terms of I/O-complexity and/or solution quality). One intuitive direction is to consider making the RR-set based algorithms (e.g., TIM in [137], GeneralTIM in Chapter 4, and IMM [136]) to work in a distributed setting, as the single-machine version of these algorithms are proven to have much better time complexity than the classic greedy algorithm.
Bibliography


158


[34] Wei Chen, Alex Collins, Rachel Cummings, Te Ke, Zhenming Liu, David Rincón, Xiaorui Sun, Yajun Wang, Wei Wei, and Yifei Yuan. Influence maximization in social networks when negative opinions may emerge and propagate. In *SDM*, pages 379–390, 2011. → pages 2, 43, 74


163


165


[86] David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In KDD, pages 137–146, 2003. → pages 1, 2, 3, 4, 5, 6, 8, 15, 19, 22, 23, 30, 31, 41, 47, 49, 74, 94, 107, 123, 126, 146, 152, 155


169


