From Viral Marketing to Social Advertising: Ad Allocation under Social Influence

Çiğdem Aslay

TESI DOCTORAL UPF / 2016

Directores de la tesi:

Prof. Dr. Ricardo Baeza-Yates
Departament of Information and Communication Technologies

Dr. Francesco Bonchi
ISI Foundation

Her şeyimi borçlu olduğum anneme.

Acknowledgements

I want to express my sincerest gratitute to my supervisors, Prof. Dr. Ricardo Baeza-Yates and Dr. Francesco Bonchi, who accompanied me these last years with their knowledge and advice, and made it possible for me to work on a research field that has always fascinated me. I am very grateful to Ricardo for giving me the opportunity to do a PhD in the great research environment of Yahoo Labs, and for always having time and being helpful whenever I needed the most. I am still amazed with his ability to give immediate and very constructive feedback during his insanely busy schedule. I am deeply thankful to Francesco for supporting me any time and for all his technical and non-technical advice regarding the life of a researcher. I am also grateful to him for guiding me along the way when I was getting lost, and for giving me the opportunity to collaborate with other amazing researchers. I am very grateful to Prof. Dr. Laks Lakshmanan for giving me the opportunity to visit the University of British Columbia for a long internship under his supervision. Prof. Dr. Laks Lakshmanan thought me, among many other things, how to attack complex problems by starting from the simplest solutions as I had a tendency to try to solve everything at the same time. I want to thank him for all his teachings, for our ongoing collaboration, and our technical conversations which I truely enjoy.

I want to thank to my initial collaborators, Dr. Luca Aiello and Dr. Neil O'Hare, for introducing me to the world of research during my Master thesis internship at Yahoo Barcelona Lab. I had a quite amazing introduction to research under their supervision, which directly motivated me to continue with my PhD studies. During the first year of my PhD, I had the chance to collaborate with Dr. Nicola Barbieri, who is not only an amazing researcher, but also an amazing friend that I can still consult for advice without hesitation. I also want to thank to Dr. Amit Goyal and Dr. Wei Lu, with whom I had a chance to collaborate thanks to Prof. Dr. Laks Lakshmanan.

I am grateful to Yahoo Labs for accommodating me as an intern during the most of my PhD. It was great to be surrounded by amazing high-profile researchers. I am also very grateful to Dr. Carlos Castillo Ocaranza for fighting against the bureaucracy to be able to fund the last days of my PhD. Looking back, I also want to thank to the consortium and the organizers of the Erasmus Mundus Master Programme in Data Mining and Knowledge Management. This PhD wouldn't have been possible in the first place if they had not given me the opportunity, the formal training, and the scholarship to complete my Master's degree.

I also want to thank to my ex-manager Çağla Bakış at the Nielsen Company, where I was working as a statistician before I started to pursue my Master's degree. Çağla was always very fond and supportive of my interest in technical developments. Back then, around 2009, when I had first told her that I was planning to take some programming courses during the weekends just for learning and fun, she told me without hesitation that she will have the company fund it. As an old school statistician whose programming skills were not beyond SAS macro programming, I learned to code in several languages thanks to her support.

I want to thank to all the members of our legendary Yahoo Barcelona Lab family, who are now spread all over the world. I will never forget the fun times and parties. Special thanks to Bora Edizel, Ioanna Tsalouchidou, Diego Saez Trumper, David Laniado, Michele Trevisol, and Janette Lehmann for all the interesting conversations we had during the lunch or coffee breaks, and for all the fun we had inside and outside the lab. A special thanks also goes to Balkumbia and Bruce Vera for all the peace, refresh, and dance sessions.

Finally, this thesis wouldn't have been possible without the love and support of my mother, my brother, and my other half Tuğçe Akkuş. I am very grateful to them for always standing by my side, always supporting me, and enduring the long periods of being apart. I owe everything to my mother, who is the strongest woman I have ever seen, and who has always motivated me to follow my own path and create my own destiny. I am also very grateful to have an amazing roommate and friend like Marc Mora Radigales. After the coup attempt in Turkey last July, life took an unexpected turn for me. During this stressful period of exile combined with the stress of PhD thesis, Marc has given me incredible support when I needed the most. He calmed me down when things were falling apart, ignored my messiness, and even cooked for me. This thesis wouldn't have been possible without his compassionate support.

ABSTRACT

This thesis constitutes one of the first investigations that lie at the intersection of social influence propagation, viral marketing, and social advertising. The objective of this thesis is to take the algorithmic aspects of viral marketing out of the lab, and further enhance these aspects to account for the real world social advertisement models, by drawing on the viral marketing literature to study social influence aware ad allocation for social advertising. To this end, we take a first step towards enabling social influence online analytics in support of viral marketing decision making, and propose efficient influence indexing framework that can accurately answer topic-aware viral marketing queries with milliseconds response time. We then initiate investigation in the area of social advertising through the viral marketing lens, aligned with real world social advertisement models, and introduce two fundamental optimization problems, regarding the allocation of ads to social network users under social influence. We devise greedy approximation algorithms with provable approximation guarantees for the novel problems introduced. We also develop scalable versions of our approximation algorithms by leveraging the notion of reverse reachability sampling on social graphs, and experimentally confirm that our algorithms are scalable and deliver high quality solutions.

RESUM

Aquesta tesi constitueix una de les primeres investigacions en la intersecció entre propagació d'influència social, màrqueting viral i publicitat social. L'objectiu d'aquesta tesi és treure els aspectes algorítmics de màrqueting viral fora del laboratori, i millorar-los per tenir en compte els models de publicitat del món real en xarxes socials, fent ús de la literatura del màrqueting viral per estudiar l'assignació d'anuncis basada en la influència social per a la publicitat en xarxes socials. Amb aquesta finalitat, hem pres un primer pas cap al desenvolupament de anàlisi d'influència social en línia que ajudin en la presa de decisions en el màrqueting viral, i proposem un marc per a la indexació eficient d'influència que pugui respondre amb precisió a les consultes de màrqueting viral orientades a temes específics amb temps de resposta de mil·lisegons. A continuació, comencem una investigació en l'àrea de la publicitat social a través de la lent del màrqueting viral, en línia amb models de publicitat del món real, i introduïm dos nous problemes d'optimització pel que fa a l'assignació d'anuncis als usuaris de la xarxa social sota la influència social, amb garanties d'aproximació demostrables. També desenvolupem una versió escalable dels nostres algoritmes d'aproximació aprofitant la noció de presa de mostres d'accessibilitat inversa en grafs socials, i confirmem experimentalment que els nostres algoritmes són escalables i ofereixen solucions d'alta qualitat.

RESUMEN

Esta tesis constituye una de las primeras investigaciones en la intersección entre propagación de influencia social, marketing viral y publicidad social. El objetivo es sacar los aspectos algorítmicos de marketing viral fuera del laboratorio, y mejorarlos para tener en cuenta los modelos de publicidad del mundo real en redes sociales, haciendo uso de la literatura de marketing viral para estudiar asignación de anuncios basada en la influencia social. Con este fin, tomamos un primer paso hacia el desarrollo de análisis de influencia social en línea que ayuden en la toma de decisiones en el marketing viral, y proponemos un marco para la indexación eficiente de influencia que pueda responder con precisión a las consultas de marketing viral orientadas a temas específicos con tiempo de respuesta de milisegundos. A continuación, iniciamos una investigación en el área de la publicidad social a través de la lente del marketing viral, en línea con modelos de publicidad del mundo real, e introducimos dos nuevos problemas de optimización respecto a la asignación de anuncios a los usuarios de la red social bajo la influencia social, con garantías de aproximación demostrables. También desarrollamos una versión escalable de nuestros algoritmos de aproximación aprovechando la noción de toma de muestras de accesibilidad inversa en grafos sociales, y confirmamos experimentalmente que nuestros algoritmos son escalables y ofrecen soluciones de alta calidad.

ÖZET

Bu doktora tezi, sosyal etki, viral pazarlama, ve sosyal ağ pazarlaması kesişimindeki ilk akademik araştırmalardan biridir. Bu tezin amacı, viral pazarlama algoritmalarını labdan çıkartmak, ve viral pazarlama literatüründen yola çıkarak bu algoritmaları endüstrideki sosyal ağ pazarlama modellerini de kapsayacak şekilde geliştirmektir. Bu amaçla, milisaniyelik cevap süresi ile viral pazarlama ve sosyal etki analitik çözümlerini mümkün kılan indeks uygulamaları geliştirdik. Ayrıca, viral pazarlama literatüründen yola çıkarak, endüstrideki sosyal ağ pazarlaması uygulamalarına yönelik, reklamları sosyal ağ kullanıcılarına paylaştıran yeni optimizasyon problemleri tanımladık. Bu optimizasyon problemlerimiz icin teorik garantisi olan algoritmalar geliştirdik. Ayrıca ters ulaşılabilirlik örnekleme kavramından yararlanarak bu algoritmalarin hızlı ve kaliteli bir şekilde çözümler üreten ölçeklenebilir versiyonlarını geliştirdik.

Contents

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Motivation

The study of complex networks to describe the topologies of a wide variety of systems has received a great deal of attention from the scientific community over the past decades. One of the main problems studied extensively by many social scientists is the phenomena of how new trends, information, and innovations spread, analogous to the spread of a virus, through a social network. Earliest studies tackling this problem focused on the adoptions of medical [43] and agricultural [122] innovations, showing that the decisions about the adoption of an innovation were made in the context of a social structure, in which people were influenced by the decisions of their neighbors, friends, and colleagues. Since then, the research on social influence theory took off, providing remarkable evidence that social influence induces viral phenomena, such as the spread of obesity, back pain, suicide, and political beliefs through a social network of individuals [27, 39, 40, 65, 78, 121, 137].

With the emergence and wide spread use of online social networking (*e.g.*, Facebook), social microblogging (*e.g.*, Twitter), and social media (*e.g.*, Tumblr) platforms, the dynamics of social influence on these platforms started to attract the interests of computational scientists, technologists, and advertisers. From the advertisers' perspective, such processes of virality enabling to reach a wide audience is extremely appealing, and has lead to a popular business concept known as *viral marketing* [3, 4, 70, 82, 87, 99].

The term viral marketing was first coined in 1997 by Steve Jurvetsen, an investor of Hotmail, to illustrate the viral phenomenon "network enhanced word-of-mouth", inspired from the adoption pattern of the free e-mail service Hotmail [87]. After its launch in 1996, Hotmail included a promotional message with a clickable

URL of the service in every message sent by a Hotmail user. This way, every Hotmail user became an involuntary advertiser by simply using the Hotmail service. Hotmail's subscriber base grew from zero to 12 million users in just 18 months, more rapidly than any company in the history of the world, and became the largest e-mail provider in several countries, including the ones where it had performed no marketing activity [87]. The successful viral growth of Hotmail, using only a small advertising budget of $50,000, inspired several companies to adopt viral marketing strategies up to day, including the launch of Google's Gmail. Gmail achieved wide spread usage with "no" advertising cost, by initially sending limited invitations to a set of *carefully* chosen users, each of whom could thereafter invite more users [90].

Viral marketing is more powerful than traditional third-party advertising as it conveys an implied endorsement from a friend. It takes advantage of the networks of social influence among the individuals, by "targeting" the most *influential* individuals. By convincing them to adopt a product with free samples or promotions, marketers can exploit the power of the network effect, fueled by word-of-mouth. In this way, they can deliver their marketing message to a large portion of the social network through a self-replicating viral process. One of the most challenging components of creating a successful viral marketing campaign is the *seeding* strategy, *i.e.*, selecting the initial individuals to target, such that the spread of a marketing message in the social network is maximized. Kempe *et al.* [88] formulated this problem in their seminal work as a discrete optimization problem under the name *influence maximization*.

Influence maximization is the key algorithmic problem behind viral marketing. The problem, as originally defined by Kempe *et al.* [88] is as follows: given (*i*) a social network, represented by a directed graph with individuals as nodes, edges corresponding to social ties, and edge weights denoting the strength of social influence a node can exert on his neighbor in the graph; (*ii*) a stochastic propagation model that governs how a certain behavior would diffuse from a node to his neighbors; and (*iii*) a cardinality budget k; the goal is to identify a set of k nodes, called the"seed set", that should be targeted by the viral marketing campaign, such that the expected number of influenced nodes in the network is maximized. Kempe *et al.* show that influence maximization is NP-hard, and obtain provable approximation guarantees under several stochastic propagation models.

Following this seminal work, research on the dynamics of social influence propagation and influence maximization took off in several dimensions, from learning and modeling the real-world social influence propagation, to improving the efficiency and scalability of the influence maximization algorithms, as we will review in Chapter 2. Although the research on influence maximization and its application to viral marketing is advancing with promising theoretical and experimental results, its applicability to the real-world viral marketing scenarios is

still limited due to the computational challenges incurred by the hardness of the problem. Indeed, even the recently proposed state-of-the-art scalable influence maximization algorithms [22, 42, 112, 131, 132] can take several hours on reasonably large real-world online social networks, limiting their efficiency for applications that require milliseconds response time. Thus, in Chapter 3, we take a first step towards enabling social-influence online analytics in support of viral marketing decision making, and propose an efficient influence indexing framework for a very general type of viral marketing queries: topic-aware influence maximization queries.

In addition to viral marketing, the rise of online advertisement models, implemented by search engines, online social networking, or microblogging platforms, have generated even more opportunities for advertisers in terms of personalizing and targeting their marketing messages. When users access a platform, they leave a trail of information that can be correlated with their consumption tastes, enabling better targeting options for advertisers. In particular, social networking platforms can gather larger amount of users' own shared information that stretches beyond general demographic and geographic data. Hence, these platforms offer more advanced interest, behavioral, and connection-based targeting options, enabling a level of personalization that is not achievable by other online advertising channels. Consequently, advertising on social networking platforms has been one of the fastest growing sectors in the online advertising landscape, further fueled by the explosion of investments in mobile ads. For example, social advertising, a market that did not exist until Facebook launched its first advertising service in May 2005, is projected to generate 11 billion revenue by 2017, almost doubling the revenue obtained in 2013.[1]

Driven by the multi-billion dollar industry, the area of computational advertising has attracted a lot of interest during the last decade [102]. The central problem in the area is to find the "best match" between a given user in a given context, such as a query submitted to a search engine or a webpage visited, and a suitable set of advertisements. Considerable work has been done in sponsored search and display advertising [52, 62, 63, 68, 105]. However, when online advertising is performed on social networking and microblogging platforms, the context of the user includes not just her interests or queries, but also her social context: the users she follows and is influenced by, and the users that follow her and are influenced by her. Hence, with the advent of social advertising, the standard interest-driven allocation of ads to users has become inadequate as it fails to leverage the potential of social influence. Thus, in Chapter 4 and 5, we initiate the research in the area of social advertising through the viral marketing lens, and introduce novel optimization problems regarding the allocation of ads to social network users, aligned with

[1] http://www.unified.com/historyofsocialadvertising/

real-world social advertising models, and address the problems that viral marketing or computational advertising literature fail to address in isolation.

1.2 Contributions

This thesis constitutes one of the first investigations that lie at the intersection of social influence propagation, viral marketing, and social advertising. Our main work is divided in three parts, in Chapter 3, 4, and 5 respectively. Below we provide a brief summary of the chapters and the key contributions.

- **Online Topic-aware Influence Maximization Queries** (Chapter 3)
  In this part of the thesis, we take a first step towards enabling social-influence online analytics in support of viral marketing decision making, and propose efficient influence indexing framework for a very general type of viral marketing queries: topic-aware influence maximization queries. Given a directed social graph, where the arcs are associated with a topic-dependent user-to-user social influence strength, and given a budget k, the problem requires to find a set of k users that we shall target in a viral marketing campaign for a given new item, described as a distribution over topics. The main challenge is given by the enormous number of queries: any possible distribution over the topic space (*i.e.*, any possible item) induces a different probabilistic graph, and thus a different instance of the influence maximization problem.

  Regardless the substantial research effort devoted to improve the efficiency and scalability of the influence maximization algorithms, their efficiency is still limited for applications that require milliseconds response time. Thus, our goal was to build an index over pre-computed solution seed sets that allows to answer such queries in milliseconds, enabling online social influence analytics, what-if simulation, and marketing decision making.

  Exploiting a tree-based index for similarity search in non-metric spaces, a clever approximate nearest neighbors search over the tree, and a weighted rank aggregation mechanism, our index can provide, in few milliseconds, a solution very similar to the one produced by the standard offline influence maximization computation, while achieving a similar expected influence spread.

  Our work initiated the investigation of topic-aware influence indexing techniques in the influence maximization literature, and was published in *International Conference on Extending Database Technology* (EDBT), 2014, under the title "**Online Topic-aware Influence Maximization Queries**" [5].

- **Social Advertising: Regret Minimization** (Chapter 4)
  In this part of the thesis, we initiate the investigation in the area of social advertising through the viral marketing lens. We propose a novel problem domain of allocating users to advertisers for promoting advertisement posts, taking advantage of the network affect, while at the same time paying attention to important factors such as relevance of the ad, effect of social influence, users' limited attention span, and limited advertisers' budgets. We assume a real-world business model in which the advertisers approach the host (*i.e.*, social network owner) with a monetary budget, to pay for ad-engagements in return for the social advertising service provided by the host. We show that the allocation that takes into account the propensity of ads for viral propagation can achieve significantly better engagement rates. However, uncontrolled virality could be undesirable for the host as it creates room for exploitation by the advertisers: hoping to tap uncontrolled virality, an advertiser might declare a lower budget for its marketing campaign, aiming at the same large outcome with a smaller cost.

  This creates a challenging trade-off: on the one hand, the host aims at leveraging virality and the network effect to improve advertising efficacy, while on the other hand the host wants to avoid giving away free service due to uncontrolled virality. We formalize this as the problem of *minimizing regret* in allocating users to ads, which we show is NP-hard and inapproximable w.r.t. any factor. However, we devise an algorithm that provides approximation guarantees w.r.t. the total budget of all advertisers. We also develop a scalable version of our approximation algorithm, which we extensively test on four real-world data sets, confirming that our algorithm delivers high quality solutions, is scalable, and significantly outperforms several natural baselines.

  Our work initiated the investigation in the area of social advertising through the viral marketing lens, and was published in *International Conference on Very Large Databases* (VLDB), 2015, under the title "**Viral Marketing Meets Social Advertising: Ad Allocation with Minimum Regret**" [6].

- **Social Advertising: Revenue Maximization** (Chapter 5)
  In this part of the thesis, we study the novel advertisement model of *incentivised* social advertising. Under this model, those users who are selected by the host to be the *seeds* for the campaign on a specific ad, can take a "cut" on the social advertising revenue. These users are typically selected because they are influential or authoritative on the specific topic, brand, or market of the ad. In this context, we study the fundamental problem of revenue maximization from the host perspective: an advertiser enters into a commercial

agreement with the host to pay, following the cost-per-engagement model, a fixed price per each engagement to his ad. The agreement also specifies the finite budget of the advertiser for the incentivised social advertising campaign for his ad. The host has to carefully select the seed users for the campaign: given that the budget that it can receive from the advertiser is fixed, the host must try to achieve as many engagements on the ad as possible, while spending little on the incentives for "seed" users to increase his revenue. The host's task gets even more challenging by simultaneously accommodating many campaigns by different advertisers.

We show that, keeping all important factors such as topical relevance of ads, their propensity for social propagation, the topical influence of users, users incentives, and advertisers budgets in consideration, the problem of revenue maximization in incentivised social advertising is NP-hard and it corresponds to the problem of monotone submodular function maximization subject to a partition matroid constraint on the ads-to-seeds allocation and submodular knapsack constraints on the advertisers' budgets. For this problem we devise two natural variants of the greedy approximation algorithm for which we provide formal approximation guarantees.

Our work initiates the investigation in the area of *incentivised* social advertising through the viral marketing lens, and is being prepared for submission.

The thesis is organized as follows. In Chapter 2 we provide necessary background and review related work. In Chapter 3, we build an influence index that can efficiently and accurately process topic-aware influence maximization queries with milliseconds response time. In Chapter 4, we build a bridge between viral marketing and social advertising for the allocation of ads under social influence, and formally define and study the regret minimization problem from the perspective of the social network owner. In Chapter 5, we introduce the novel advertisement model of *incentivised* social advertising, and formally study the revenue maximization problem under this model from the perspective of the social network owner. In Chapter 6, we conclude the thesis by providing a summary of the main results, open problems, and various directions for future research.

CHAPTER 2

BACKGROUND

In this chapter, we provide necessary formal background that is at the core of our research. We start by formally introducing the influence maximization problem as originally defined by Kempe *et al.* [88].

2.1 Influence Maximization in Social Networks

Domingos and Richardson [54, 118] are the first to consider the propagation of influence, and the identification of influential users as a data mining problem. They used Markov Random Field (MRF) techniques to model and study the problem of finding an optimal set of individuals on which a company should perform marketing actions, so that the expected increase in the profit is maximized. This problem didn't receive much attention from the data mining community, until Kempe *et al.* [88] formulated the same problem in a discrete optimization setting, under the name *influence maximization*.

Kempe *et al.* [88] formalized the influence maximization problem based on the concept of a propagation model, *i.e.*, a stochastic diffusion model which governs how individuals influence each other and how propagations happen. Given a directed social graph $G = (V, E)$, a propagation model, and a cardinality budget k, the task of influence maximization is to find a set $S \subseteq V$ of k nodes, such that by targeting them initially for early activation, the expected number of activated nodes, denoted by $\sigma(S)$, is maximized. The initially activated set S of nodes is commonly referred as the *seed set*, and $\sigma(S)$ is commonly referred as the expected *influence spread* of S. Kempe *et al.* mainly focused on two propagation models from mathematical sociology, namely, the Independent Cascade [70] and the Linear Threshold [77] models.

Independent Cascade (IC) In an instance of the IC model, given a directed social graph $G = (V, E)$, each edge $(u, v) \in E$ is labeled with an influence probability $p_{u,v}$, representing the strength of influence that node u exerts over node v. At any time step t, each node is either active or inactive: an active node never becomes inactive. Initially all nodes are inactive: at time step 0, a set $S \subseteq V$ of seed nodes are activated, and the propagation process starts to proceed in discrete times steps. When a node u becomes active at time t, it has one chance at influencing each inactive out-neighbor $v \in N^{out}(u)$, succeeding with probability $p_{u,v}$, independent of the diffusion history so far. If the attempt succeeds, v becomes active at time $t + 1$. The diffusion process terminates when no more nodes can be activated.

Linear Threshold (LT) In an instance of the LT model, given a directed social graph $G = (V, E)$, each node $v \in V$ has an activation threshold θ_v uniformly distributed in the interval $[0, 1]$, which represents the minimum weighted fraction of active in-neighbors that are needed to activate v. Each edge $(u, v) \in E$ is associated with an influence weight $p_{u,v}$ such that the sum of incoming weights to v from the set of in-neighbors of v, denoted by $N^{in}(v)$, does not exceed 1:

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

In the LT model, time proceeds in discrete time steps: at time step 0, a set $S \subseteq V$ of seed nodes are activated. At any time step $t \geq 1$, any inactive node v becomes active if the total influence weight from its active in-neighbors reaches or exceeds θ_v:

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

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

Hardness and Approximation. Kempe *et al.* show that influence maximization is NP-hard under both IC and LT propagation models. However, they show that the objective function (expected influence spread $\sigma : 2^V \mapsto \mathbb{R}_{\geq 0}$) under both IC and LT models is *monotone* and *submodular*. Monotonicity of $\sigma(\cdot)$ implies $\sigma(S) \leq \sigma(T)$ whenever $S \subseteq T \subseteq V$. Submodularity of $\sigma(\cdot)$ implies $S \subseteq T$

$$\sigma(S \cup \{w\}) - \sigma(S) \geq \sigma(T \cup \{w\}) - \sigma(T),$$

Algorithm 1: Greedy Algorithm for Influence Maximization [88]

Input : G, k, σ
Output: seed set S

```
S ← ∅
while |S| < k do
    u ← arg max_{w∈V\S} (σ(S ∪ {w}) − σ(S))
    S ← S ∪ {u}
```

for all $S \subseteq T$ and $w \in V \setminus T$. Intuitively, monotonicity indicates that the expected influence spread cannot decrease as the seed set size increases. Similarly, submodularity indicates that the marginal gain $\sigma(S \cup \{w\}) - \sigma(S)$ from adding a new node w shrinks as the seed set grows. This property is also known as the *law of diminishing returns*. Based on the seminal result of Nemhauser *et al.* [109] for the maximization of a monotone submodular subject to a cardinality constraint, Kempe *et al.* show that the simple greedy algorithm, depicted in Algorithm 1, that at each iteration greedily extends the set of seeds with the node providing the largest marginal gain, produces a solution with provable approximation guarantee $(1 - 1/e)$.

To explore the boundaries of approximability under different discrete propagation models, Kempe *et al.* further unified their results under the General Threshold (GT) model, of which IC and LT models are special cases. GT model specifies that each node $v \in V$ is associated with a threshold function $f_v : 2^V \mapsto [0, 1]$ that is monotone w.r.t. the set of in-neighbors of v. Mossel and Roch [106] later show that whenever the threshold function at every node is monotone and submodular, the expected influence spread function $\sigma(\cdot)$ is also monotone and submodular, which was a conjecture posed in [88].

Kempe *et al.*, through extensive experimentation, verify that the greedy approximation algorithm for influence maximization (Algorithm 1) achieves significantly higher expected influence spread than the node selection heuristics based on the well studied notions of degree centrality and distance centrality.

2.2 Modeling and Learning Influence Propagation

Learning Influence Probabilities. Most of the literature devoted to improving the efficiency and scalability of algorithms for influence maximization assume that the weighted social graph is given and do not address how the influence probabilities $p_{u,v}$ on the edges can be obtained. This problem instead is addressed in [73, 125, 126]: Saito *et al.* [125] study how to learn the probabilities for the IC model from a log of past propagation data for a given social graph. They formalize

this as a likelihood maximization problem and apply the Expectation Maximization (EM) algorithm to learn the influence probabilities. However, their approach suffers from overfitting due to sparsity of the past propagations data, hence, the same set of authors propose to consider also the node attributes while learning influence probabilities in [126]. Goyal *et al.* [73] devise algorithms that can learn the influence probabilities for GT model in no more than 2 scan of the input log of past propagations: their probabilistic inference techniques also can predict when a user will perform an action accurately. These existing work assume that the social graph is given and focus only on learning the influence probabilities on the edges. A line of research instead focus on inferring both the unknown network structure and the influence probabilities in the case when the social graph is also not available [50, 71, 110].

Alternative Propagation Models. In addition to IC, LT, and GT propagation models, various other propagation models have been proposed to model social influence spread. One of earliest well known models was proposed by Bass [15], showing that the product diffusion follows a S-shaped curve, where product adoption starts slowly, takes off exponentially, and flattens at the end. He modeled the rate of adoption as a function of the individuals who have already adopted the product. However this model does not account for the structure of the social networks.

Another well known propagation model is Susceptible-Infected-Recovered (SIR) model, which has been extensively studied in the context of epidemics and disease propagation: in this model, a person who is susceptible to a disease becomes infected with a certain probability if there is an infected neighbor in the social network. The person then recovers at a certain rate and becomes immune to the disease. Within the context of social influence, a few papers study the SIR model and its variation Susceptible-Infected-Susceptible (SIS), which assumes that the recovered nodes can become infected again [89, 123, 124, 143].

Considerable work has been done in order to better capture the real-world social influence propagation dynamics: majority of the influence propagation models assume discrete time diffusion, instead a few papers study the modeling the propagation of influence in continuous time [56, 119, 120]. A different line of research focused on competitions of multiple propagation processes [18, 23, 24, 26, 30, 81, 96, 117], as well as, cooperations or complementarity between different propagation processes [97, 107, 108].

Topic-aware Influence Propagation. Regardless the fact that users authoritativeness, expertise, trust and influence are evidently topic-dependent, only few papers have looked at social influence from the topics perspective. Tang *et al.* [130] study the problem of learning user-to-user topic-wise influence strength. The input to their problem is the social network and a prior topic distribution for each node,

which is given as input and inferred separately. Liu *et al.* [95] propose a probabilistic model for the joint inference of the topic distribution and topic-wise influence strength: here the input is an heterogenous social network with nodes that are users and documents. The goal is to learn users' interest (topic distribution) and user-to-user influence. Lin *et al.* [93] study the joint modeling of influence and topics, by adopting textual models. However, none of these three papers define an influence propagation model, instead Barbieri et al. [13] extend the classic IC and LT models to be topic-aware: the resulting models are named Topic-aware Independent Cascade (TIC), and Topic-aware Linear Threshold (TLT). Barbieri et al. also devise methods to learn, from a log of past propagations, the model parameters, i.e., topic-aware influence strength for each link and topic-distribution for each item. Their experiments show that: (*i*) topic-aware influence propagation models are more accurate in describing real-world influence driven propagations than the state-of-the-art topic-blind models, and (*ii*) by considering the characteristics of the item a larger number of adoptions can be obtained in the influence maximization problem.

2.3 Alternative Optimization Objectives

Leskovec *et al.* [91] study the influence maximization problem from a different perspective, namely *outbreak detection*, which aims to find the nodes in a social network such that the spread of a virus is detected as fast as possible. Nguyen *et al.* [111] study *budgeted* influence maximization problem, in which the cardinality budget of the standard influence maximization problem is replaced with monetary budget, and seed users are paid non-uniform incentives in exchange for initial activation. The alternative problem definition of Leskovec *et al.* [91] also model the budgeted influence maximization problem. Mathioudakis *et al.* [100] proposed an algorithm for finding the k most important links in a social network that maximizes the likelihood of the observed propagations.

Motivated by the resource and time constraints on viral marketing campaigns, Goyal *et al.* [72] study two different optimization problems: (*i*) given a threshold η on the expected influence spread, their *Minimum Target Set Selection* problem asks to find the seed set of minimum size, whose activation can provide an expected influence spread that exceeds the threshold η; (*ii*) given a threshold on expected influence spread η, and a threshold k on the seed set size, their *Minimum Time* problem asks to find a seed set of at most k nodes whose activation can provide at least η expected influence spread in the minimum possible time.

Barbieri *et al.* [12] study the interplay between viral product design and social influence with the goal to design the features of a novel product such that its adoption, fueled by peer influence and word-of-mouth effect, is maximized. Lu *et*

al. [98] studied the distinction between social influence and actual product adoption by modeling the states of being influenced and of adopting a product with the goal to maximize the revenue of a viral marketing campaign. Budak *et al.* [26] study the problem of influence limitation for a bad campaign that starts propagating from certain nodes in the network by identifying the individuals to target with the competing good campaign. He *et al.* [81] study a similar problem in which one entity tries to block the influence propagation of its competing entity as much as possible by strategically selecting a number of seed nodes that could initiate its own influence propagation.

2.4 Improving Efficiency and Scalability

Though simple, the greedy approximation algorithm for influence maximization (Algorithm 1) is computationally prohibitive, since the step of selecting the node providing the largest marginal gain, depicted on Line 3, is #P-hard [37, 38]. Kempe *et al.* [88] run Monte Carlo simulations for sufficiently many times[1] to obtain an accurate estimate of the expected spread. In particular, they claim that for any $\phi > 0$, there is a $\delta > 0$ such that by using $(1 + \delta)$-approximate values of the expected spread, we can obtain a $(1-1/e-\phi)$-approximation for the influence maximization problem.

Accurate estimation of influence spread requires a large number of Monte Carlo simulations, thus, the greedy approximation algorithm for influence maximization (Algorithm 1) is computationally exhaustive due to its high time complexity of $O(knmr)$, where n is the number of nodes, m is the number of edges, and r is the number of Monte Carlo simulations.

Leskovec *et al.* [91], by exploiting submodularity of the influence spread function, devised a cost-effective lazy forward (CELF) technique that improves the run-time of Algorithm 1 up to 700 times. Goyal *et al.* [75] further optimized CELF, by look ahead optimization of marginal gain computations, and proposed CELF++ that empirically improves the run-time of CELF by $35\% - 55\%$.

Despite the big improvements of CELF [91] and CELF++ [75] over Algorithm 1, their efficiency and scalability are still limited: CELF++ [75], takes from few days to more than a week in order to extract a seed set of 50 nodes on a graph with only a few thousand of nodes [5]. Therefore, a number of heuristics have been proposed to improve the efficiency and scalability of the influence maximization computation [37, 38, 76, 86]: Chen *et al.* propose the Maximum Influence Arborescence (MIA) algorithm, a degree-discount heuristic, for the IC model [37], and the Local Directed Acyclic Graph (LDAG) algorithm for the LT

[1]The authors report 10,000 simulations.

model [38], all of which restrict the computation of influence spread to local trees and DAGs surrounding seed nodes. Goyal *et al.* [76] proposed SimPath algorithm for the LT model which operates based on enumerating paths originating from seeds. Jung *et al.* [86] propose the IRIE algorithm for the IC model that operates based on a belief propagation approach for their global influence ranking procedure. Although these heuristics approaches provide significant run-time improvement over the greedy approximation algorithm [88], and its optimizations [75,91], the scalability of these approaches and the quality of the solutions are limited.

Orthogonal to these efforts for devising scalable and efficient influence maximization algorithms, Mathioudakis *et al.* [100] proposed an algorithm for finding the k most important links in a social network that maximizes the likelihood of the observed propagations, which can also be used as a preprocessing step to aid with the scalability of influence maximization algorithms. Similarly, we introduced topic-aware influence indexing techniques to the influence maximization literature [5], and our work has several follow-ups [35, 36, 92], all of which similarly exploit pre-computed information, and are orthogonal to the efforts devoted to improving the efficiency and the scalability of influence maximization algorithms.

Recently, Borgs et al. [22] make a theoretical breakthrough by introducing the idea of sampling *"reverse-reachable"* (RR) sets in the graph for the efficient estimation of influence spread, referred as *Reverse-Influence Sampling* (RIS), and present a quasi-linear time randomized algorithm that runs in $O(k\ell^2(m+n)\log^2 n/\epsilon^3)$ time, returning $(1-1/e-\epsilon)$-approximate solution with at least $1-1/n^\ell$ probability. Based on RIS, Tang et al. [132] propose TIM, a more run-time efficient algorithm that runs in $O\left((k+l)(m+n)\log n/\epsilon^2\right)$, while providing the same approximation guarantee. Tang et al. [131] later proposed IMM, that improves over TIM by addressing its deficiencies that arises during the computation of a lower bound on the statistically required sample size for accurate estimation. Cohen et al. [42] proposed a sketch-based design for fast computation of influence spread, achieving efficiency and effectiveness comparable to TIM. Nguyen *et al.* [112], adapting ideas from TIM [132], and the sequential sampling design proposed by Dagum *et al.* [49], propose SSA that provides significant run-time improvement over TIM and IMM, while providing an influence spread estimate that keeps up with the $(1-1/e-\epsilon)$-approximation guarantee.

Next we provide a more detailed background on the RIS framework which was introduced by Borgs et al. [22].

Random RR sets. Interpreting G as a distribution over unweighted directed graphs, where each edge $(v,w)\in E$ is realized with probability $p_{v,w}$, let $g\sim G$ be a graph drawn from the random graph distribution G. For a given set S, Borgs

et al. [22] show that:

$$\sigma(S) = n \cdot \Pr_{u \sim V, g \sim G} [S \cap R_{u,g}(u) \neq \emptyset] \tag{2.1}$$

where $R_{u,g}(u)$ is a random RR set, with the subscripts denoting the 2 level of randomness in its creation: (*i*) selection of a root node $u \in V$ uniformly at random, (*ii*), sampling of a possible world rooted at node u from the transposed graph G^T, by removing each encountered edge (w, v) in G^T with probability $1 - p_{w,v}$. For notational convenience, we will simply use R to denote a random RR set $R_{u,g}(u)$, with the randomness over $u \sim V$ and $g \sim G$ already implied by definition.

Influence spread estimation. For a given set S, let $X^S \sim$ Bernoulli(μ^S) denote the indicator random variable for the event $[S \cap R \neq \emptyset]$, succeeding with probability μ^S, and failing with probability $1 - \mu^S$, where μ^S is the probability that a random RR set R has non-empty intersection with S:

$$\mu^S = \frac{\sigma(S)}{n} = \textbf{Pr}\,[S \cap R \neq \emptyset].$$

For a fixed set S, the problem of estimating $\sigma(S)$ reduces to the classic problem of estimating the unknown mean μ^S of a Bernoulli random variable X^S. A typical statistical approach to estimate an unknown mean is to design an experiment that produces independent copies of the random variable X^S, and use the average of the experiment outcomes as the estimate $\hat{\mu}^S$: let $X_1^S, X_2^S, \cdots, X_\theta^S$ be independently and identically distributed according to X^S, where each X_i^S denotes the outcome of the experiment $[S \cap R_i \neq \emptyset]$ for given a sequence $R_1, \cdots, R_\theta$ of θ randomly sampled RR sets (*with replacement*). Here, $\frac{\sum_{i=1}^{\theta} X_i^S}{\theta}$ is an unbiased estimator of μ^S, *i.e.*, given $\mathbb{E}[X_i^S] = \mu^S, \forall i \in [1, \theta]$, we have:

$$\mathbb{E}\left[\frac{\sum_{i=1}^{\theta} X_i^S}{\theta}\right] = \frac{\sum_{i=1}^{\theta} \mathbb{E}[X_i^S]}{\theta} = \mu^S.$$

While $\frac{\sum_{i=1}^{\theta} X_i^S}{\theta}$ is an unbiased estimator of μ^S, the accuracy ϵ, and the confidence $(1-\delta)$ of estimation depends on the choice of the sample size θ, s.t. the estimation

$\hat{\mu}^S$ satisfies:

$$\Pr[(1-\epsilon)\mu^S \leq \hat{\mu}^S \leq (1+\epsilon)\mu^S] \geq 1-\delta. \tag{2.2}$$

Following Dagum *et al.* [49], we can refer to $\hat{\mu}^S$ as (ϵ, δ)-approximation of μ^S if it satisfies Eq. 2.2.

Influence maximization with RIS. For a given input k, TIM [132] generates a sample $\mathcal{R}$ of θ random RR sets, such that, for any set $S \subseteq V$ of size k, the estimation $\hat{\sigma}(S) = n \cdot \frac{\sum_{i=1}^{\theta} X_i^S}{\theta}$ satisfies:

$$|\sigma(S) - \hat{\sigma}(S)| \leq \epsilon \cdot OPT_k, \tag{2.3}$$

with at least $1 - \delta/\binom{n}{k}$ probability. Note that, Eq. 2.3 corresponds to $(\epsilon', \delta/\binom{n}{k})$-approximation of $\sigma(S)$, for each S of size k, where $\epsilon_1 = \frac{\epsilon \cdot OPT_k}{2 \cdot \mu^S \cdot n}$. To obtain $(\epsilon', \delta/\binom{n}{k})$-approximation of $\sigma(S)$, for all S of size k, TIM uses Chernoff Bounds to find the following lower bound on θ:

$$\theta \geq \frac{(8 + 2 \cdot \epsilon) \cdot n \cdot (\log \delta + \log \binom{n}{k} + \log 2)}{\epsilon^2 \cdot OPT_k}. \tag{2.4}$$

such that Eq. 2.3 holds for all size k sets. TIM then returns the set that *greedily* covers the most number of random RR sets in $\mathcal{R}$ as the approximate greedy solution with at least $1 - \delta$ probability. The computation of θ using Eq. 2.4 requires the knowledge of the unknown OPT_k, hence, TIM also computes a lower bound on OPT_k: IMM [131] addresses the deficiencies of TIM that arises during the computation of a lower bound on OPT_k, by providing a tighter lower bound on OPT_k, which translates to tighter lower bound on θ, hence improved efficiency. IMM also optimizes the computation of a lower bound on OPT_k, by allowing dependencies during the generation of random RR sets, which allows to reuse the random RR sets produced for the determination of θ.

SSA [112], adaping ideas from SRA [49] and TIM, defines a stopping condition Λ_1: initially starting with an empty $\mathcal{R}$, SSA keeps adding Λ_1 random RR sets to $\mathcal{R}$ each time checking the value of the greedy-cover solution S_c on the evolving $\mathcal{R}$, until the first time the value of the greedy-cover computed on the evolving $\mathcal{R}$ is greater than Λ_1. Then, SSA this time starts with an empty $\mathcal{R}'$ and employs SRA, with a pre-defined coverage Λ_2, and estimates the influence spread

of S_c: if the two estimates for S_c, computed from $\mathcal{R}$ and $\mathcal{R}'$, are close to each other, SSA returns S_c, otherwise it repeats the same process by generating Λ_1 more random RR sets into the current $\mathcal{R}$. In the worst case, SSA terminates with $(8+2\cdot\epsilon)\cdot n\cdot(\log\delta+\log\binom{n}{k}+\log 2)/\epsilon^2$ random RR sets in $\mathcal{R}$, which is computed from Eq.2.4 by setting $OPT_k = 1$.

CHAPTER 3

ONLINE TOPIC-AWARE INFLUENCE MAXIMIZATION QUERIES

3.1 Introduction

Viral marketing, a popular concept in business literature, has recently attracted a lot of attention also in computer science, thanks to the fascinating computational challenges that it entails. In this area, the most studied computational problem, known as *influence maximization* [88], requires the identification of a set of k influential users (called the "seed set"), that should be targeted by the viral marketing campaign. Here, targeting might mean to give a free sample of a product, a special promotion or a big discount. In order to enjoy the special promotion, the targeted user has to accept to automatically re-post it on her timeline over the social networking platform, so that her followers are exposed to the same marketing message.

As we previously discussed in Chapter 2, the bulk of the literature on influence maximization problem just focuses on a generic item, thus implicitly assuming that the influence among users of the social network remains the same, regardless of the characteristics of the item being propagated. In this chapter, we drop such assumptions, and study how to address Topic-aware Influence Maximization (TIM) queries in an online fashion: given a directed social graph, in which the arcs are associated with a topic-dependent user-to-user social influence strength, and given a cardinality budget k, a TIM query requires to find a seed set of k users that we shall target in a viral marketing campaign for a given new product (described as a distribution over topics) in order to maximize the product adoption.

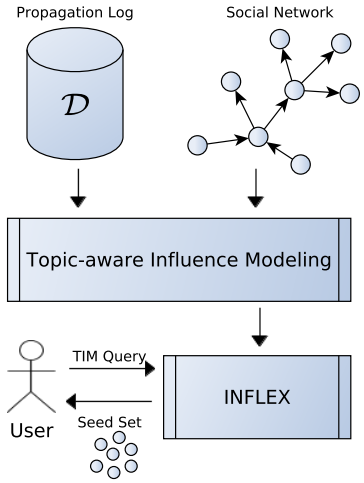


Figure 3.1: INFLEX high-level overview.

Regardless the substantial research effort devoted to improving the efficiency and scalability of the influence maximization algorithms [22, 37, 42, 86, 112, 131, 132], their efficiency is still limited for real world applications, such as interactive decision support systems, that require milliseconds response time on large-scale social networks.

As an application example, consider a social networking platform that allows to implement viral marketing campaigns: the users initially targeted by the campaign accept to spread the marketing message over the social network to their friends. Advertisers come to the platform with a description of the ad (e.g., a set of keywords) to be promoted and they compete for the attention of the users which are considered influential w.r.t. the given description. In this kind of setting, not only it is important to consider the given description for selecting the seed set appropriately, but such a decision must also be taken in an online fashion. Thus, in this chapter, our goal is to build an index over pre-computed solution seed sets that allows to answer such queries in milliseconds, enabling online social influence analytics, what-if simulation, and marketing decision making.

Our contribution is INFLEX, an index to answer TIM queries in milliseconds with excellent accuracy. INFLEX employs a tree-based index for similarity search with Bregman divergences, to efficiently retrieve a good-enough set of topic-wise neighbor points for the query item. Then, it performs rank aggregation on their seed sets to produce the final answer to the query. Experimental results on real data show that INFLEX can provide, in few milliseconds, a solution very similar (Kendall-τ distance < 0.1) to the one produced by the *offline* ground-truth computation [75], which usually takes several days. A high level depiction of our setting is provided in Figure 3.1.

3.1.1 Problem Definition

Given a directed social graph $G = (V, A)$ and a space of Z topics, we assume the TIC propagation model [13] that we previously presented in Chapter 2. In the TIC model, for each arc $(u, v) \in A$ and for each topic $z \in [1, Z]$, we have a probability $p^z_{u,v}$ that represents the strength of influence that user u exerts over user v for topic z. Similarly, an item i is described by a probability distribution $\vec{\gamma}_i$ over the topics: that is for each topic $z \in [1, Z]$, we are given γ^z_i, with $\sum_{z=1}^{Z} \gamma^z_i = 1$.

In the TIC model, a propagation happens as in the IC model: when a node u first becomes active on item i, it has one chance of influencing each inactive neighbor v, independently of the history thus far. The tentative succeeds with a probability that is the weighted average of the link probability w.r.t. the topic distribution of the item i:

$$p^i_{u,v} = \sum_{z=1}^{Z} \gamma^z_i p^z_{u,v}. \tag{3.1}$$

A TIM query $Q(\vec{\gamma}_q, k)$ takes as input an item description $\vec{\gamma}_q$ and an integer k, and it requires to find the seed set $S \subseteq V$, $|S| = k$, such that the expected number of nodes adopting item q, denoted by $\sigma(S, \vec{\gamma}_q)$, is maximized:

$$Q(\vec{\gamma}_q, k) = \underset{S \subseteq V, |S|=k}{\operatorname{argmax}} \ \sigma(S, \vec{\gamma}_q). \tag{3.2}$$

It is important to observe that a TIM query can always be processed by a standard influence maximization computation in the IC model: given the query item description, we can derive a directed probabilistic graph $G = (V, A, p)$, where the probability $p_{u,v}$ for each arc is defined as in Equation 3.1. This means that, TIM queries maintain the same properties of standard influence maximization, thus, they can exploit the standard algorithms and enjoy the usual approximation guarantees. However, our goal is to build a topic-aware influence index to efficiently process TIM queries with milliseconds response time, opening the door to online influence maximization analytics.

3.1.2 Contributions and Roadmap

The main challenge of using pre-computed information in our setting is the enormous number of potential queries: essentially any possible item description $\vec{\gamma}_i$ lying on the probability simplex that contains all possible probability distributions with state space $[Z]$. It is also very hard to build smart indexes exploiting the graph structure, as any potential query corresponds to a different probabilistic graph.

Motivated from the fact that similar items are likely to interest similar people, thus, are likely to have similar influence patterns, we propose INFLEX, an indexing framework to efficiently answer TIM queries. INFLEX is based on the idea of appropriately selecting a set of items, and by extracting their seed sets using a standard influence maximization process. Then, at query time, given a query item, we select a "large enough" set of neighbor index points and combine their pre-computed seed sets, by means of *rank aggregation*, into a final seed set that we return as the result to TIM query.

Next, for each step, we briefly describe the associated challenge and the intuition behind the proposed solution.

Selecting the items to build the index. The number of items used to build the index governs the trade-off between accuracy and space-time efficiency. In fact, for each index point, we have to run a standard influence maximization algorithm, which can be extremely time consuming, and store its seed. Another challenge is given by the space from which we have to select the index points: on one hand we want to follow the distribution observed in the catalog of items that we have available, because also new items are expected to come from the same distribution; on the other hand selecting index points directly from the catalog can be risky in the case of sparsely distributed catalog items - we might end up finding nearest neighbors which are not very similar to the query item. Our approach here is to select, for a given preprocessing budget, a reasonable number of points that can provide a good coverage of the space. This is obtained as follows: we use the catalog of available items to define, by means of the maximum likely Dirichlet distribution, the space from which we sample a large enough number of points. Then we apply Bregman K-means++ [10] to these points, and select the resulting centroids as our index points (details are provided in Section 3.4.1).

Fast, approximate, and unbounded nearest neighbors. At query time, given a TIM query, we want to efficiently retrieve the index points which are topic-wise similar to the query item. Given that index points and query items are probability distributions over the space of topics, we adopt Kullback-Leibler divergence as a measure of their distance. Our task is then a *similarity search* with Kullback-Leibler divergence.

Our task differs from other types of similarity search in the literature as it is not based on a pre-specified radius (*"range search"*), nor on a number of neighbors to return (*"k-NN search"*). Instead, how many points to retrieve depends on how close the points we retrieve are to the query item. The intuition is that, if we find index points extremely similar to the query item, then we can just use few of them (at an extreme, if we find exactly the query item in the index, then we can simply retrieve the associated seed set without looking for any other point). Instead, when there are no index points very close to the query point, we aggregate

a larger number of them.

Another requirement for our similarity search is to be fast, for which we drop exactness: our solutions are *approximate* nearest neighbors, in the sense that, if we return k index points, these are not necessarily the k nearest neighbors of the query item. For this task we adopt *Bregman ball tree* (Section 3.4.2) with a novel approximate nearest neighbors search procedure (Section 3.5.1).

Seed set aggregation. In the final step, we perform rank aggregation of the seed sets[1] of the retrieved index points. The goal is to provide a final list of nodes that has the minimum Kendall-τ distance to all the seed set lists. As this problem is NP-hard and we aim for quick computation, we look at approximate solutions. In particular, we adopt and compare Borda [21] and Copeland aggregation [47], both followed by the Local Kemenization procedure that have been shown to be fast and good in practise [127]. We enrich these two methods with a novel importance weighting scheme based on the KL-divergence of the index points from the query item: intuitively, the closer a point is to the query item, the more predominant its role will be in the aggregation (details are provided in Section 3.5.2).

Evaluation. The evaluation of our framework is straightforward. For a given query item, we assess the performance of the INFLEX framework in terms of accuracy and query evaluation time. For both we can compare against performing an influence maximization computation, for the given query, from scratch, as well as other smarter baselines. Our experiments on a real dataset (Section 3.6) show that INFLEX produces seed sets that are very close to the "best offline" ones (Kendall-τ distance generally < 0.1), while achieving an expected spread very close (NRMSE $< 2\%$) to the spread achieved by standard offline influence maximization computation, but it does so in few milliseconds instead of several hours or days of computation, thus opening the door to online influence maximization analytics.

3.2 Related Work

3.2.1 Influence Maximization

Kempe *et al.* [88] formalized the NP-hard influence maximization problem, and proposed a simple greedy approximation algorithm, which we reviewed in Chapter 2. Though simple, the greedy algorithm is computationally prohibitive, since the step of selecting the node providing the largest marginal gain is #P-hard [37, 38]. In their paper, Kempe *et al.* run Monte Carlo simulations for sufficiently many times to obtain an accurate estimate of the expected spread.

[1] It is important to note that, although usually called seed "sets", these are ranked lists of nodes.

However, running many propagation simulations is extremely costly on real-world social networks. Therefore, following [88], considerable effort has been devoted to developing methods for improving the efficiency and scalability of influence maximization [22, 37, 42, 74, 75, 86, 91, 112, 131, 132].

Alternatively, with the published contents of this chapter, we initiated the investigation of topic-aware influence indexing techniques in the influence maximization literature [5], and our work has several direct follow-ups [35, 36, 92], all of which similarly exploit pre-computed information and are orthogonal to the efforts devoted to improving the efficiency and the scalability of influence maximization algorithms.

3.2.2 Similarity Search

Similarity search (*a.k.a.* proximity search) studies the problem of searching the items of the database that are similar to a given query item: given a database X of items, a dissimilarity measure d, and a query item q, two typical similarity queries can be defined using d: (*i*) range query, that reports all the objects in X that are within a distance r to q; (*ii*) K-Nearest Neighbors query (K-NN), that reports the k closest objects to q in X.

If similarity is modeled with a dissimilarity measure d that satisfies the following metric axioms:

- Reflexivity: $d(x, y) = 0 \iff x = y$
- Non-negativity: $d(x, y) \geq 0$
- Symmetry: $d(x, y) = d(y, x)$
- Triangle Inequality: $d(x, y) \leq d(x, z) + d(z, y)$

then the dissimilarity measure is metric, and the set of objects in the database X is called a metric space. If the dissimilarity is not a metric, then the similarity search is referred as *non-metric similarity search*.

Similarity Search in metric spaces have important applications in many commercial databases and web search. For efficient processing of similarity queries in terms of I/O and CPU time, a common approach is to use data structures to filter out irrelevant items during the search to avoid the costly sequential scan of the database. To this end, many data structures have been proposed, which mainly operate on the decomposition of the metric space into smaller cells, with branch and bound exploration methods defined *w.r.t.* metric space axioms, particularly *triangle inequality* [16, 17, 66, 79, 141, 144]. Interested reader may refer to Chávez *et al.* [34] for an in-depth survey on similarity search in metric spaces.

Non-metric similarity search has recently attracted a lot of attention from researchers due to the increasing need to perform *content-based retrieval* of multimedia data [20, 46, 69, 101]. When the dissimilarity measure of interest fails to

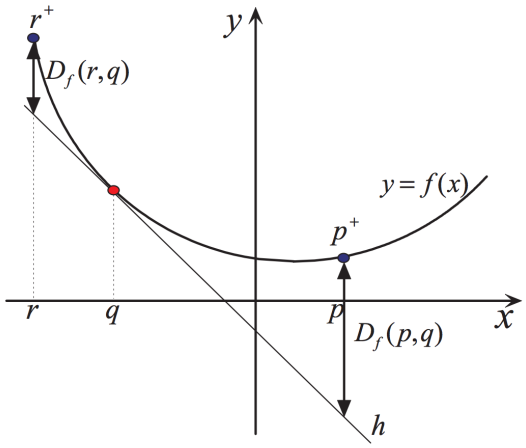


Figure 3.2: Bregman Divergence between 2 points p and q.

satisfy metrix axioms, data structures relying on metric space axioms cannot be directly used. Common approaches to perform non-metric similarity search include mapping of the problem to metric space, or designing index structures that rely on particular properties of the non-metric measure [28, 129].

Our dissimilarity measure of interest in this study, *Kullback-Leibler (KL) divergence*, is a non-metric information theoretic measure, and belongs to a broad of family of dissimilarity measures called *Bregman divergences*. Bregman Divergences form a family of distortion measures that are defined by a strictly convex and differentiable generator function $f : \mathcal{X} \mapsto \mathbb{R}^+$ on a multi-dimensional convex domain $\mathcal{X}$. The Bregman Divergence based on f is defined as:

$$d_f(p, q) = f(p) - f(q) - \langle \nabla f(q), p - q \rangle \tag{3.3}$$

where $\nabla f(x)$ is the gradient of the function $f(x)$ at point q and $\langle ., . \rangle$ denotes the dot product between the two vectors.

The Bregman Divergence $d_f(p, q)$ is geometrically measured as the vertical distance between the point $(p, f(p))$ and the hyperplane H_q that is tangent to f at the point $(q, f(q))$. Bregman divergence $d_f(p, q)$ can be interpreted as the distance between a function $f(p)$ and its linear approximation centered at q, in other terms, the distance between a function and its first-order Taylor series approximation. Some examples of Bregman Divergences include Squared Euclidean Distance, Kullback-Leibler Divergence, Mahalanobis Distance, and Itakura-Saito Distance.

Bregman divergences are not metrics, since none of them satisfies the triangle inequality, and some of them fail to satisfy the symmetry property as in the case

for KL-divergence. Cayton [31, 32] proposed *Bregman Ball Trees (bb-tree)* for nearest neighbor and range search with Bregman divergences, analogous to metric ball trees [135, 136, 144], which operates on the hierarchical decomposition of space by simple convex bodies. He designed optimization procedures for the pruning of nodes, based on convexity and projective duality properties of the Bregman divergences. Nielsen *et al.* [115] proposed *bb-tree++* as an improvement to Cayton's *bb-tree* [31, 32] in terms of construction time and solution quality. Nielsen *et al.* [114] later proposed *Bregman Vantage Point Tree*, coupled with a pruning strategy based on the intersection of Bregman balls, with an adaptation from its metric counterpart *vp-tree* [144] that uses triangle inequality of metric spaces.

3.2.3 Rank Aggregation

Rank aggregation addresses the problem of combining rankings of a set of candidates from different sources to one consensus ranking. It is a classical problem, dating back to the eighteenth century [21,44], from Social Choice and Voting Theory, in which each voter gives a preference on a set of alternatives, and the system outputs a single preference order on the set of alternatives, based on the voters' preferences. A rank aggregation function computes an aggregated ranking order which minimizes the distance to the set of orderings given as input. The ordering received as input can specify either a *full* or *partial* ranking on the objects of the domain and the techniques for rank aggregation differ slightly if they are applied to the first case or the latter.

All commonly used rank aggregation methods satisfy one or more of the desirable *goodness* properties: unanimity, neutrality, monotonicity, and Concordet Criterion. *Concordet Criterion* suggests that if a candidate is ranked ahead of all other candidates by an absolute majority of voters, it should be declared as the winner, thus, it should be ranked first [44]. Truchon *et al.* [133] made an extension to the Concordet criterion named as *Extended Concordet Criterion* which suggests that if there is any partition $\{C, R\}$ of a set S of candidates, such that for any $i \in C$ and $j \in R$, if a majority of rankers prefer i to j, then the aggregate ranking should prefer i to j. Aggregation mechanisms that satisfy the Concordet Criterion and its natural extensions such as the Extended Concordet Criterion are considered to yield *robust* results that cannot be "spammed" by a few bad voters.

The *Kemeny* optimal rank aggregation problem requires to identify the ranking that has the minimum number of pairwise disagreements, *i.e.*, minimum Kendall Tau distance, with all rankers. Kemeny optimal aggregation satisfies neutrality, monotonicity, Concordet Criterion and Extended Concordet Criterion. Kemeny optimal aggregated list corresponds to the true geometric median of the input lists. Bartholdi *et al.* [14] showed that computing Kemeny optimal aggregation is NP-hard.

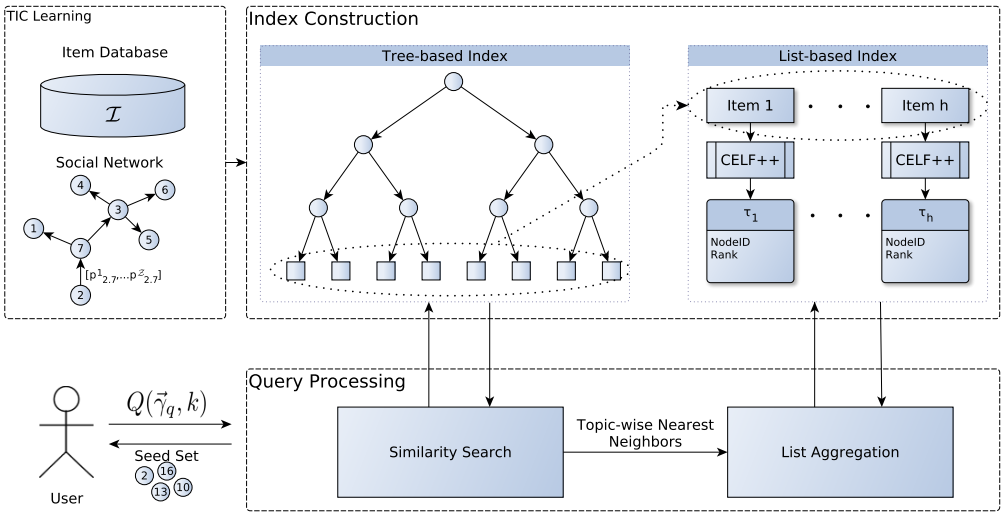


Figure 3.3: INFLEX detailed overview.

Recently, rank aggregation problem is studied widely in computer science in the context of meta-search engines [57, 58], cache-based query processing in search engines [29], similarity search in databases [61] and collaborative filtering [9]. Dwork *et al.* [57,58] applied rank aggregation to the context of metasearch in IR, showing that computing the Kemeny optimal aggregation is NP-hard even when there are at least four lists to aggregate, and proposed solutions based on Markov chains or minimum-cost matching in bipartite graphs. They also introduced the notion of *Local Kemenization*, as a relaxation of the Kemeny optimality, that ensures satisfaction of the extended concordet criterion but remains computationally tractable $O(kn \log n)$. ()A detailed comparison of the rank aggregation algorithms having constant factor approximation or PTAS is addressed by Schalekamp and van Zuylen [127].

3.3 Overview of the Framework

In Figure 3.1 we already provided a very high-level view of the framework. In this section we start giving more details, opening the INFLEX box and describing its components as depicted in Figure 3.3. The starting point is a social graph $G = (V, A)$ where each arc $(u, v) \in A$ has associated a probability $p^z_{u,v}$ for each topic $z \in [1, Z]$ representing the strength of influence that user u exerts over user v for topic z. Moreover we have a database $\mathcal{I}$ of items, where each item i is represented by a distribution $\vec{\gamma}_i$ over the topics. Both these two pieces of input (depicted in the upper-left corner of Figure 3.3) are jointly learnt in a pre-processing phase from a log of past propagation traces [13] (depicted in the top half of Figure 3.1).

The database of items $\mathcal{I}$ is used to define, by means of maximum likely Dirichlet distribution, the space from which we select h index points (details are given in Section 3.4.1).

Let $\mathcal{H} = \{\vec{\gamma}_1, \cdots, \vec{\gamma}_h\}$ be our index points. For each $\vec{\gamma}_i \in \mathcal{H}$, and for a fixed $\ell \in \mathbb{N}$ we extract a seed set of size ℓ, or equivalently, we solve the TIM query $Q(\vec{\gamma}_i, \ell)$, by transforming it to a standard influence maximization computation over the IC model (as discussed in Section 3.1.1) and running the standard greedy algorithm: in particular, we use its optimization CELF++ [75]. This phase is depicted in the top-right corner of Figure 3.3. It is important to note that, at the time of the publication of INFLEX [5], CELF++ [75], was the state-of-the-art influence maximization algorithm, hence, we used CELF++ for pre-computing the seed sets required by the index. Our methods are orthogonal to the latest advances on devising scalable influence maximization algorithms [22, 42, 112, 131, 132], which can replace CELF++ for the pre-computation of the seed sets more efficiently.

Let $\tau_1, \cdots, \tau_h$ denote the index lists containing pre-computed seed sets returned for the h index points. For a given query $Q(\vec{\gamma}_q, k)$ our goal is to *(i)* retrieve the points whose topic distributions are *similar* to the topic distribution of the query item q, *(ii)* combine their pre-computed seed sets, by means of *rank aggregation* into a final seed set ${\tau_q}^*$ and return as the result to TIM query. As already discussed in the previous section, the search of index points similar to the query items is developed on top of a *Bregman ball tree* index structure (depicted in the center of Figure 3.3 and described in full details in Section 3.4.2).

Note that the size of the seed set k requirement, can be satisfied this way even when $k > \ell$. In fact, the rank aggregation can return up to m seeds, where m is the cardinality of the union of the seed lists of all index points retrieved in the similarity search phase. By retrieving more index points, we can satisfy larger k requirements.

In the next section we present the offline phase of the index construction. Then in Section 3.5 we will present the online TIM query evaluation mechanism over INFLEX.

3.4 Index Construction

In the TIC propagation model, each item $i \in \mathcal{I}$ is represented by a distribution over topics, $\vec{\gamma}_i$, that lies on the probability simplex $\triangle^{Z-1}$. Each topic z encodes an abstract influence pattern. The assumption is that pairwise influence probabilities between users depend on the topic. More specifically, $p^z_{u,v} \in [0, 1]$ denotes the likelihood that user u will trigger the activation of user v, on topic z. Given an item i, the item-specific influence probability on each arc $(u, v) \in A$ is the dot

product of the user-to-user topic dependent influence probabilities and the item's topic distribution (Equation 3.1). Under these assumptions, two items that exhibit a similar distribution over topics will also exhibit a similar propagation pattern, as they will enable close pairwise influence probabilities.

This observation is the core of the overall approach, as it allows us to cast the efficient processing of TIM query as a *similarity search* problem. Intuitively, given a query $Q(\vec{\gamma}_q, k)$, we can retrieve the closest items for which the list of users to target is available, and exploit this information to provide a list of k seed nodes that can boost the adoption of q on the considered network.

The first step towards the design of the index is the formalization of the notion of similarity between two items. In this context, it is natural to instantiate the dissimilarity measure between two items as the KL-divergence between their respective topic distributions. Given two discrete distributions P and Q, the KL-divergence

$$D_{KL}(P\|Q) = \sum_i P(i)\ \log \frac{P(i)}{Q(i)}$$

quantifies the average information lost when we use Q to approximate P. Since KL is asymmetric, one must choose between the *right-sided* (the query item is the second argument) and *left-sided* formulation, or opt for a symmetrized version that can be computed by considering the average of the sided definitions. Since our task is to retrieve the nearest-neighbors for a given query item $\vec{\gamma}_q$, the definition of dissimilarity should penalize the difference between the topic distribution of each item i and the query item, proportionally to each component γ_i^z. The dissimilarity that best suits to this setting is the right-sided KL: $D_{KL}(\vec{\gamma}_i\|\vec{\gamma}_q)$, which prefers to stretch over all components γ_i^z, rather then focusing only on the highest mode of $\vec{\gamma}_q$ [113].

3.4.1 Selection of the Index Points

The topic distributions of items recorded in $\mathcal{I}$, form a *data-space* on $\triangle^{Z-1}$. This is the overall *search space* for similarity queries on topic distributions.

The first step to build INFLEX is to select a set of h index points $\mathcal{H} = \{\vec{\gamma}_1, \cdots, \vec{\gamma}_h\}$ where each $\vec{\gamma} \in \mathcal{H}$ lies on $\triangle^{Z-1}$. On one hand, we want the h points to provide a good coverage of $\triangle^{Z-1}$. On the other hand, the actual choice of the budget h depends on, both, limitations in terms of memory[2] and index construction time, due to the need of running a full influence maximization computation for each index point.

One way of selecting the index points would be to use a *space-based* approach, by selecting h items whose topic distributions are positioned *equi-distantly* on

[2]The cost of keeping one preprocessed index item in memory is $(Z-1)$ * sizeof(double) + ℓ * sizeof(int).

$\triangle^{Z-1}$. This would provide a fair coverage of the space. The drawback is that it disregards the available workload: in fact, the topic distribution of the items learnt from past data, might be clustered in some area of the simplex.

At the opposite extreme, we have the fully *data-driven* approach: assuming that future items will follow the same distribution of the items learnt from past data, we might select as index points, items from the catalog $\mathcal{I}$. However, this way we might end up finding nearest neighbors which are actually not very close to the query item if there are some items in the catalog whose topic distributions are sparse.

To realize a good compromise between these two indexing approaches, we resort on a sampling strategy on the simplex. By applying the *Maximum-Likelihood Dirichlet Estimation* procedure described in [104], given the topic distributions learnt from data $\Theta_{\mathcal{I}} = \{\vec{\gamma}_1, \cdots, \vec{\gamma}_{|\mathcal{I}|}\}$, we estimate the hyper-parameters $\boldsymbol{\alpha} = \{\alpha_1, \cdots, \alpha_Z\}$ which define the Dirichlet distribution that maximizes:

$$\prod_{i \in \mathcal{I}} P(\vec{\gamma}_i | \boldsymbol{\alpha}) = \prod_{i \in \mathcal{I}} \frac{\Gamma(\sum_z \alpha_z)}{\prod_z \Gamma(\alpha_z)} \prod_z (\gamma_i^z)^{\alpha_z - 1}.$$

Then, the next step is to generate a large number of samples from $Dirichlet(\boldsymbol{\alpha})$, identify h clusters by applying K-means++ [10] and finally use their centroids as the topic distributions of the index points.

After the selection of index items, we start building the *list-based index* which will store the seed sets returned to the preprocessing of TIM queries for the selected index items. For each item $i \in \mathcal{H}$, let G_i denote an instance of $G = (V, A, p)$ which is obtained by assigning item-specific influence probability, according to Eq. 3.1, to each arc $(u, v) \in A$. We then compute the seed set τ_i for each index item $i \in [1 : h]$ using a standard influence maximization computation.

3.4.2 Bregman-ball Tree Index

As already anticipated in the previous sections, the problem of TIM query processing can be addressed by retrieving index points which are similar to the query item. To make the similarity search phase efficient, we turn our attention to index structures to organize the points in $\mathcal{H}$.

The choice of indexing strategy for *similarity search* is naturally tied to the choice of similarity/dissimilarity function. As discussed above, INFLEX employs an information theoretic measure, the KL-divergence, which belongs to the family of Bregman divergences. This family comprises distortion measures (Squared Euclidean distance, Mahalanobis distance, Itakura-Saito distance, and

KL-divergence, just to cite a few) that are defined by a strictly convex and differentiable generator function $f : \mathcal{X} \mapsto \mathbb{R}^+$ on a $d - dimensional$ convex domain $\mathcal{X}$. The Bregman divergence based on f is defined as:

$$d_f(p, q) = f(p) - f(q) - \langle \nabla f(q), p - q \rangle \tag{3.4}$$

where $\nabla f(x)$ is the gradient of the function $f(x)$ at point q and $\langle ., . \rangle$ denotes the dot product between two vectors. Bregman divergences are not metrics since none of them satisfies the triangle inequality and some of them fail to satisfy the symmetry property as in the case for KL-divergence. When the dissimilarity measure of interest fails to satisfy metric axioms, data structures relying on metric space axioms cannot be directly used.

For efficient similarity search with KL-divergence, we adopt the *Bregman ball tree* (bb-tree) [31, 115], a *tree-based index* structure designed to work with the family of Bregman divergences, to avoid the costly sequential scan of the database of index points with $O(Zh)$ time.

Similar to its metric counterparts [34], bb-tree is built in a *top-down* fashion, by recursive partitioning of the database of items to be indexed ($\mathcal{H}$), and thus defining a hierarchical space partition based on convex bodies called *Bregman balls*. A Bregman ball with a center μ and a radius R is defined as:

$$B_f(\mu, R) = \{i \in \mathcal{H} \mid d_f(i, \mu) \leq R\}. \tag{3.5}$$

Each node of the bb-tree corresponds to a set of database items $H_i \subseteq \mathcal{H}$ and is associated with a Bregman ball $B_f(\mu, R)$ such that $H_i \subset B_f(\mu, R)$ which covers all data points indexed in the subtree rooting at the node. Following Nielsen *et al.* [115], the tree is built from the root to leaves, by recursively applying Bregman K-means++ at each node to generate child nodes from the parent node. The tree-branching factor is computed by applying Gaussian clustering, which allows to find the optimal number of children that avoids the overlapping of the Bregman balls of the child nodes. The tree is built in $O(h \log h)$ time for h index points.

We equip bb-tree with a novel approximate nearest neighbors search procedure, that we introduce in the next section.

3.5 Query Processing

In this section, we present the query evaluation mechanism of the INFLEX framework. As anticipated in the previous sections it consists of two phases:

1. *similarity search* aimed at quickly retrieving a good set of index points for the given query item (Section 3.5.1);

2. *rank aggregation* of the seed list associated to the retrieved index points (Section 3.5.2).

3.5.1 Searching for Topic-wise Similar Items

The kind of search needed in the INFLEX framework has several peculiar requirements, that make the standard approaches, such as *range search* or *K-NN search*, unsuitable. Let us first discuss these requirements, and then present our solution.

- The search is neither based on a pre-specified radius as in *range search*, nor on a number of neighbors as in *"k-NN search"*. Instead, how many points to retrieve is decided dynamically as the points are retrieved. The intuition is that if we find, in the currently visited leaf, index points similar to the query item, then we can stop the search, otherwise we might need to visit more leaves.

- An extreme case is when there is an index point whose topic-distribution is identical (or extremely similar) to the query item: in that case we want to directly return the seed set of the index point, without further looking for similar points and performing rank aggregation.

- The search must be fast and can be approximate, in the sense that if it returns k results, those do not necessarily have to be the k nearest neighbors. In any case, the selected points will have a weight in the rank aggregation, proportional to their distance from the query item: so unimportant points will be treated accordingly.

Given these requirements, the similarity search in INFLEX is implemented as follows. We visit the bb-tree in *depth-first search* order, from the root to the leaf nodes, heuristically moving towards the branch whose associated Bregman ball has the center closer to the query item $\vec{\gamma}_q$, and adding the other children to a *priority queue* to ensure the early successive exploration of sub-trees that most likely contain the nearest neighbors. When we reach a leaf node, we compute the divergence of the query item from all the index points stored in the node. At this point we have three options:

1. there exists a point i in the leaf such that $D_{KL}(\vec{\gamma}_i \parallel \vec{\gamma}_q) \leq \epsilon$ for $\epsilon \approx 0$. In this case we say we have an ϵ-exact match: we stop the search and return the top-k elements in τ_i;

2. the population of points in the leaf is considered "similar enough" to the query item: we stop the search and move to the rank aggregation phase with this group of points;

3. the population of points in the leaf is not considered "similar enough" to the query item: in this case we consider the next leaf.

We have to formally define what does it mean for a group of nodes to be "similar enough". As anticipated abstractly before, this concept depends on the number of index points retrieved and on their distance from the query item. We instantiate this concept by resorting to the application of the *Anderson-Darling* test [2], which assesses whether a sample of data comes from a population following a specific distribution. Following [115], this test has been previously applied in the phase of building the tree index, to learn the branching factor of the bb-tree by applying the G-means procedure [80]. Given a population of points which currently define a node in the bb-tree, we apply the Anderson-Darling normality test to check if, given a confidence level α, the hypothesis of normality is rejected. If this happens, the node should be split. In a similar fashion, here we check if the query item and the population of items contained in the current leaf are compatible with a normal distribution.[3] If we accept the null hypothesis that the underlying distribution is Normal, then it is likely that the population of indexed items in the current leaf can already provide good neighbors for the query item, and hence we stop the search. The early stopping criterion based on this test achieves good performance in our framework, as we will show in Section 3.6.

Let B_n denote the Bregman ball $B(\mu_n, R_n)$ associated with node n, where μ_n is its center and R_n the radius, and let X_n denote set of data points contained in node n. The overall search procedure on bb-tree is specified in Algorithm 2.

Some implementation details are hidden in the pseudocode in Algorithm 2. The function $similar\_enough(\cdot, \cdot)$ implements the Anderson-Darling test discussed earlier. For practical reasons the function has been implemented with maximum number of leaves to consider. In all our experiments we keep this value equal to 5.

When we need to select the next node to explore, we can use the current solution set to produce a bound that helps us to avoid the exploration of unpromising subtrees while traversing back the tree. In particular, we use the maximum divergence of the query point from the current solution set NN. Let δ be such divergence:

$$\delta = \max_{i \in NN} D_{KL}(\vec{\gamma}_i \parallel \vec{\gamma}_q).$$

[3]The test is performed by projecting all the considered points in one dimension, where the application of the test is straightforward, and assuming unknown mean and unknown variance.

Algorithm 2: INFLEX similarity search

Input : bb-tree T, query item $\vec{\gamma}_q$
Output: approximate nearest neighbors of $\vec{\gamma}_q$

$PQ \leftarrow T.root$ // init. priority queue
$NN \leftarrow \emptyset$ // init. solution set
while $PQ \neq \emptyset$ **do**
  $n \leftarrow$ top element in PQ
  while n *is not leaf* **do**
    $c \leftarrow \text{argmin}_{c \in n.Children} D_{KL}(\mu_c \parallel \vec{\gamma}_q)$
    PQ.insert($n.Children \setminus \{c\}$)
    $n \leftarrow c$
  if n *is leaf* **then**
    if $\exists \vec{\gamma}_i \in X_n$ *s.t.* $D_{KL}(\vec{\gamma}_i \parallel \vec{\gamma}_q) \leq \epsilon$ **then**
      return $\vec{\gamma}_i$
    $NN \leftarrow NN \cup X_n$
    if $similar\_enough(X_n, q)$ **then**
      return NN
return NN

Analogous to triangle inequality of metric spaces, we apply the following pruning strategy. A yet unexplored node n should be visited only if the divergence of $\vec{\gamma}_q$ from the closest point in B_n (*i.e.* Bregman projection of q onto B_n), is less than δ, otherwise the subtree rooted at node n can be pruned:

$$\min_{\vec{\gamma}_x \in B(\mu_n, R_n)} D_{KL}(\vec{\gamma}_x \parallel \vec{\gamma}_q) < \delta. \tag{3.6}$$

To test whether a node should be explored based on this strategy, we use the bisection search algorithm proposed by Cayton [31] that calculates the Bregman projection onto a Bregman ball efficiently, by using primal and dual function evaluations as the stopping criterion.

3.5.2 Aggregation of Seed Sets

We have retrieved a good-enough set of index points for the given query $Q(\vec{\gamma}_q, k)$, let us denote this set $NN(\vec{\gamma}_q)$. The next step is to aggregate their seed sets to produce the final answer to the given TIM query. Let $\mathcal{L}_q$ denote the set of the pre-computed seed lists for the index points in $NN(\vec{\gamma}_q)$, i.e., $\mathcal{L}_q = \{\tau_i \mid \vec{\gamma}_i \in NN(\vec{\gamma}_q)\}$. Our task can be nicely formalized as an optimization problem named

rank aggregation. Intuitively, the goal is to combine the rankings provided by the pre-computed seed sets for topic-wise nearest neighbors to one *consensus* ranking which minimizes the overall disagreement. A rank aggregation function computes an aggregated ranking order which minimizes the distance to the set of orderings given as input. The ordering received as input can specify either a *full* or *partial* ranking on the objects of the domain and the techniques for rank aggregation differ slightly if they are applied to the first case or the latter.

Assume that two full ranking lists τ_1 and τ_2, defined on the same domain of n objects, are available. We can compute their distance by measuring the number of pairwise disagreements among their rankings. The Kendall-τ ($\mathcal{K}$) distance between two full lists is defined as:

$$\mathcal{K}(\tau_1, \tau_2) = \sum_{i=1}^{n} \sum_{j=1}^{n} 1\{\tau_1(i) \prec \tau_1(j) \ \wedge \ \tau_2(j) \prec \tau_2(i)\} \tag{3.7}$$

where $\mathbb{1}\{.\}$ is the indicator function and $i \prec j$ is the comparison operator to denote if i is ranked ahead of j.

As we are dealing with the aggregation of lists that contain top-ℓ ranked nodes instead of complete rankings on the set of users in the network, we can employ the extension of Kendall-τ to the top-ℓ case [60]:

$$\mathcal{K}(\tau_1, \tau_2) = \sum_{\{i,j\} \in \tau_1 \cup \tau_2} \bar{K}_{i,j}^{(p)}(\tau_i, \tau_j) \tag{3.8}$$

where $0 \leq p \leq 1$ is a fix parameter[4] and $\bar{K}_{i,j}^{(p)}(\tau_i, \tau_j)$ is the *penalty* defined accordingly to the following four cases:

- when i and j appear in both lists: if they appear in the same order (i is ranked ahead of j in both lists or vice versa), then $\bar{K}_{i,j}^{(p)}(\tau_i, \tau_j) = 0$;
- when i and j both appear in one list and only i appears in the other: if i is ahead of j in the list where they both appear, then $\bar{K}_{i,j}^{(p)}(\tau_i, \tau_j) = 0$, otherwise $\bar{K}_{i,j}^{(p)}(\tau_i, \tau_j) = 1$;
- when i appears in one list and j appears in the other list, then $\bar{K}_{i,j}^{(p)}(\tau_i, \tau_j) = 1$;
- when i and j both appear in only one of the lists, then $\bar{K}_{i,j}^{(p)}(\tau_i, \tau_j) = p$.

[4] In our calculations, we use the *neutral* approach by setting $p = 0.5$.

We normalize the Kendall-τ distance to lie in the $[0, 1]$ interval, where a distance of 0 corresponds to *identical* lists, by dividing Eq.3.7 and Eq.3.8 with the maximum number of possible disagreements among two full lists and among two top-ℓ lists, which is equal to $\frac{\ell(\ell-1)}{2}$ and $\ell^2 + \ell(\ell-1)p$ respectively.

The *Kemeny* optimal rank aggregation problem requires to identify the ranked list of nodes, τ_q^*, that has the minimum Kendall-τ distance to all the ranked lists $\mathcal{L}_q$:

$$\tau_q^* = \underset{\tau_q}{\operatorname{argmin}} \ \frac{1}{|\mathcal{L}_q|} \sum_{\tau_i \in \mathcal{L}_q} K(\tau_i, \tau_q). \tag{3.9}$$

This optimization problem has shown to be NP-hard when there are at least four lists to aggregate [57]. Solutions based on Markov chains or by casting the problem as minimum-cost matching in bipartite graphs have been proposed.

As efficiency is one of our main design requirements, we turn our attention to fast rank aggregation techniques, such as Borda [21] and Copeland aggregation [47], whose result can be improved by implementing a Local Kemenization procedure.

Motivated by Social Choice Theory, rank aggregation methods treat all available rankings with equal importance. However, in our context, the aggregation should favor index lists of items who are more similar to the query item. This idea is implemented in INFLEX by incorporating importance weights into the rank aggregation.

The weighting can be further exploited to prune, for efficiency sake, lists that contribute only marginally to the final aggregation, gaining considerably in query execution time, while losing very little in terms of accuracy.

Weighted ranking aggregation techniques and a procedure for weight-based list pruning are discussed next.

Importance weights for rank aggregation. The rank aggregation module receives as input a set $\mathcal{L}_q$ of seed sets. During the aggregation, the idea is to favor the rankings entailed by index lists which correspond to the closest neighbors with respect to query item. For each $i \in NN(\vec{\gamma}_q)$ we compute a rank aggregation weight $0 \leq w_i \leq 1$, which is inversely proportional to the KL-divergence from the query item. Recall that minimum value for KL-divergence is 0, while this measure is not bounded above. The weighting function $W : [0, \infty) \mapsto [0, 1]$ can

be specified by applying a non-linear transformation of the KL-divergence values:

$$W(\vec{\gamma}_i, \vec{\gamma}_q) = \frac{e^{KL_{max}} - e^{D_{KL}(\vec{\gamma}_i \| \vec{\gamma}_q)}}{1 - e^{-KL_{max}}}, \tag{3.10}$$

where KL_{max} is an empirical upper bound of the KL-divergence, computed as the distance between two corners of the considered simplex and employing a smoothing factor of machine-ε value to handle zero probabilities during the computation of KL-divergence.

Selection of nearest neighbors. Since the task of rank aggregation introduces a heavy processing burden, a careful selection of which (and how many) seed lists to consider in the aggregation is a key component towards speeding up the query evaluation phase. Therefore, we propose a procedure for the empirical selection of a subset of $\mathcal{L}_q$, based on weighting scheme introduced above.

The idea is that by iteratively inspecting the retrieved index points, from the largest to the smallest weight, we can automatically distinguish neighbors which will contribute to the weighted rank aggregation, from neighbors whose contribute is marginal. The goal is to determine the minimum number $t \leq |\mathcal{L}_q|$, such that the top-t nearest neighbors hold the highest impact in the procedure of weighted rank aggregation. We implement this test by comparing the weight assigned to the t-th index lists with the ones assigned to the previous. If the top-t nearest neighbors are equally close to the query item, then their normalized weights should tend to $\frac{1}{t}$. Let $\tilde{w}_t$ be the normalized weight assigned to the t closest point, where the normalization is over all the weights up to t. Using a threshold $\omega \in [0, 1]$, we scan iteratively the set of points and stop as soon as we find a t such that:

$$\tilde{w}_t - \frac{1}{t} \geq \omega.$$

Borda aggregation. Borda aggregation [21] is a *positional* method that corresponds to the descending order arrangement of the average Borda score for each element averaged across all ranker preferences, where Borda score for an element is the number of candidates below it in each ranker's preferences. Ordering an element by its Borda score is equivalent to ranking the vertices by increasing in-degree in the corresponding weighted feedback arc set problem in tournaments and has a factor 5 approximation of the optimal Kemeny ranking [48].

Let $\mathcal{U} \subseteq V$ denote the union of users belonging to the nearest neighbors' precomputed seed lists, i.e. $\mathcal{U} = \bigcup_{i \in [1,t]} \tau_i$. Moreover, let $\tau_i(v)$ denote the rank of the node v in the index list τ_i, and let w_i be the importance weight assigned to the

i-th index list. In the case of top-ℓ lists aggregation, the weighted Borda score for each $v \in V$ can be defined as follows:

$$Borda^{w}(v) = \begin{cases} \sum_{i=1}^{t} w_i \left(\ell - \tau_i(v) + 1\right) & \text{if } v \in \mathcal{U} \\ \ell + 1 & \text{otherwise} \end{cases}$$

When $w_i = 1,\ \forall i \in [1, t]$, weighted Borda score calculation is equal to the normal Borda score calculation. For a given query $Q(\vec{\gamma}_q, k)$, the top-k nodes having highest score are returned as output.

Copeland aggregation. Copeland aggregation [47] is a form of majority voting where the pairwise comparison among the elements in the ranked lists are taken into account. Copeland score of an element v corresponds to the number of elements v' such that v was ranked ahead, $v \prec v'$, in the majority of the lists. Copeland aggregation corresponds to the sorting of elements by non-increasing indegree on the majority tournament. The Markov Chain method (MC4) [57] is a generalization of the Copeland aggregation. For a given query $Q(\vec{\gamma}_q, k)$, we formulate the Copeland score calculation for each $v \in \mathcal{U}$ as:

$$\begin{aligned} Copeland(v) = \sum_{v' \in \mathcal{U}} 1\{(\sum_{j=1}^{t} 1\{\tau_j(v) \prec \tau_j(v')\} > \\ \sum_{j=1}^{t} 1\{\tau_j(v') \prec \tau_j(v)\}\} \end{aligned} \tag{3.11}$$

This can be implemented by introducing a pairwise comparison matrix $P_{v,v'}$ that stores the number of times that v preceeds v' among given lists. We propose to incorporate the importance weights by promoting, in the calculation of the pairwise matrix P, those comparisons which come from index lists having greater importance weight. This weighting schema for Copeland aggregation is described in Algorithm 3. Again, for a given query $Q(\vec{\gamma}_q, k)$, the top-k nodes having the highest Copeland scores are returned as output.

Local Kemenization. Local Kemenization [57] is a greedy post-processing step which takes an initial aggregation result τ_q and computes a locally Kemeny optimal aggregation of $\{\tau_1, ..., \tau_t\}$, that is *maximally consistent* with τ_q^*. This means that no better list, in terms of lower Kendall-τ distance to all the ranked lists in input, can be achieved by just flipping an adjacent pair of elements.

We implement the procedure by an insertion sort algorithm applied on the

Algorithm 3: Weighted Copeland

Input : Seed set τ_i, importance weight $w_i \; \forall \; \vec{\gamma}_i \in NN(\vec{\gamma}_q)$
Output: Weighted Copeland scores $Copeland^w$

$\mathcal{U} \leftarrow \cup_{i \in [1,t]} \tau_i$
$P_{v,v'} \leftarrow 0 \; \forall \; \{v, v'\} \in \mathcal{U}$
$Copeland^w \leftarrow 0 \; \forall \; v \in \mathcal{U}$
for *each* $\{v, v'\} \in \mathcal{U}$ **do**
    for $i \leftarrow 1$ **to** t **do**
        if $\tau_i(v) \prec \tau_i(v')$ **then**
            $P_{v,v'} \leftarrow P_{v,v'} + w_i$
        else if $\tau_i(v') \prec \tau_i(v)$ **then**
            $P_{v',v} \leftarrow P_{v',v} + w_i$
for *each* $v \in \mathcal{U}$ **do**
    for *each* $v' \in \mathcal{U}$ **do**
        $Copeland^w(v) \leftarrow Copeland^w(v) + P_{v,v'}$

aggregated final list. The sorting starts from the lowest ranked element in the list which is "bubbled up" as long as it is preferred by the majority of the input rankings. To apply this procedure for the weighted counterparts of Borda and Copeland aggregation, we incorporate weights into this procedure by *(i)* using weighted ranks for applying this on top of weighted Borda aggregation results, and *(ii)* using weighted pairwise comparisons on top of weighted Copeland aggregation results.

3.6 Experiments

In this section we describe the experimental setup for evaluating the effectiveness and efficiency of INFLEX [5]. The overall evaluation aims at:

- Understanding and quantifying the relationship between distance of items in the simplex and the distance between their respective ranked list of seed nodes. In other words, to what extent the ranked seed list for an item can be used to approximate the one of its neighbors in the simplex?
- Evaluating the overall retrieval accuracy of the approximate nearest neighbors search on the bb-tree index, and the performances of the early stopping criterion based on the Anderson-Darling test.
- Comparing the performance of different rank aggregation methods and assessing the gain of the weighted versions.

[5]The software is available from `https://github.com/aslayci/INFLEX`

Parameter	Value
Nr. of topics (Z)	10
Nr. index items (h)	1000
Max. nr of children of a node	4
Max. nr of items in a leaf	50
Max. leaf radius	0.01
Significance level for AD Test (α)	0.05
K-determination threshold ω	0.005
Seed set budget for pre-computed lists (ℓ)	50

Table 3.1: Parameters used in experimentation.

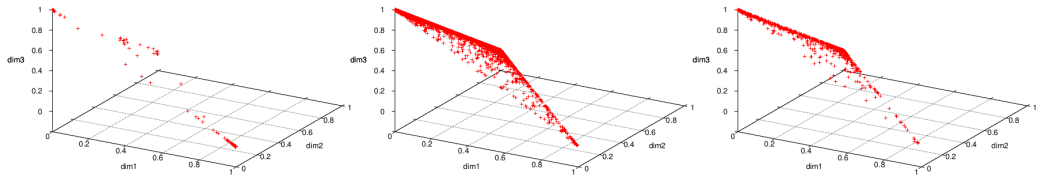
- Finally, and more importantly, evaluating the accuracy of the answers provided by the overall framework and its effectiveness.

Experimental setting and dataset. Experiments were performed on a real-world dataset from Flixster:[6] a social movie web site, where users can discover new movies and share reviews and ratings with their friends.

The network is defined by roughly 30k users and 425k unidirectional social links between them, while the propagation log records the timestamp at which a user provided a rating on a particular movie, out of a catalog of 12k items. This dataset comes with the social graph and a log of past propagations (ratings on movies), and it has been widely used to test the effectiveness of social influence propagation models and influence maximization problems [13, 74]. We focus on the influence episode defined by a user v rating a movie which is later on rated by one of his friend u: in this case we see it as a potential influence of v over u. In the movie context, it is natural to assume that each item can exhibit several topics (i.e genres) and each user may exhibit different degree of influence on different topics. We learn the topic-aware influence probabilities and the item specific topic distributions, by applying the TIC learning procedure provided in [13] with employing $Z = 10$ topics.

To evaluate the framework with respect to the aforementioned dimensions of analysis, we generated TIM queries according to, both, a data-driven and a random perspective. This differentiation allows us to study the performances of INFLEX under the assumption that query items will follow the same distributions of already indexed items, but also to assess its robustness to very diverse data distributions. To this aim, out of a total of 200 query items, half were generated by sampling from the Dirichlet distribution learnt from the item-specific distributions over topics provided by TIC learning, and the remaining were randomly generated by sampling from a uniform distribution on the simplex. We assess the

[6] http://www.cs.sfu.ca/~sja25/personal/datasets/



(a) Distribution of TIC items (b) Distribution of sampled items from *Dirichlet*($\boldsymbol{\alpha}$) (c) Distribution of index items

Figure 3.4: Selection of index items: from the catalog items (a), we learn a Dirichlet distribution that we use for sampling a large number of points (b). Index items are identified as the centroids provided by K-Means++ (c).

performance of the INFLEX framework in terms of accuracy and query evaluation time.

A summary of the setting of all the parameters considered in this evaluation is given in Table 3.1. A detailed analysis of the experimental evaluation is provided next.

Index construction. As discussed in Sec. 3.4.1, the procedure for selection of items to include in the index starts with estimating the Dirichlet distribution that maximizes the likelihood of generating the item-specific distributions over topics learnt from data. To this end, we apply the *generalized* Newton iteration procedure,[7] described by Minka [104]. Then, we run Bregman K-means++ [10] over 100k samples from the Dirichlet distribution with a number of clusters equal to the number of items that we are willing to index. We use $h = 1000$. The h centroids are identified by the clustering procedure form our set of index items $\mathcal{H}$. The output of this 3-phase process is given in Fig. 3.4: by applying dimensionality reduction on the mapping of the $\triangle^{Z-1}$ simplex to Euclidean $\mathcal{R}^{Z-1}$ with isometric log-ratio [59], we show (a) the distribution over the topics for items in the Flixster dataset, (b) 100k samples from the Dirichlet distribution, and (c) index items.

Finally, for each item in the index we run the CELF++ algorithm for selecting their seed set: on average, the computation required 60 hours for an item, when employing 5k Monte Carlo trials with $\ell = 50$. Due to extremely heavy computational burden of standard influence maximization computation, we limit the seed budget for influence maximization to $\ell = 50$.

To confirm the soundness of the main assumption that motivated INFLEX, we investigate in Figure 3.5 the relationship between the KL-divergence among randomly selected pair of items in the index, and the Kendall-τ distance among their corresponding ranked lists. The high correlation coefficient clearly shows that the items that are close in the simplex will tend to agree on the ranking of seed nodes, while their agreement in identifying the influential nodes consistently decreases with their distance.

Retrieval accuracy of similarity search. To assess the accuracy of the similarity search on bb-tree (Algorithm 2), we measure the recall of the search procedure in

[7] `http://research.microsoft.com/en-us/um/people/minka/software/fastfit/`

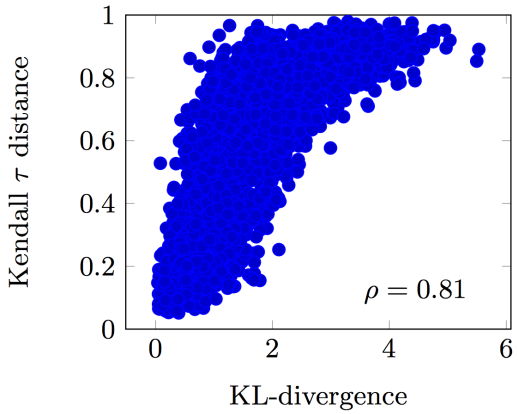


Figure 3.5: High correlation between the KL-divergence of items' topic distributions and the Kendall-τ distance among their seed sets.

identifying the top-K true nearest neighbors, with $K \in [5, 10, 15, 20]$ on 40 randomly chosen query items. Figure 3.6(a) reports such recall for varying number of visited leaves, without using the Anderson-Darling test as the early-stopping criterion. The plot shows that when $K \leq 20$, the 80% of the top-K true nearest neighbors can be found in the first 5 visited leaves.

In Figure 3.6(b), we compare the accuracy of the search procedure equipped with the Anderson-Darling test, with the one achieved by visiting the first 5 leaves. We found the average number of leaves visited when applying early-stopping criterion to be 3.65. We further checked the statistical performance of leaf-by-leaf retrieval against Anderson-Darling test based early-stopping criterion by paired t-tests, $(p < 0.05)$ on (*i*) the maximum KL-divergence observed, (*ii*) the retrieval recall, and (*iii*) the number of KL-divergence calculations, and found that our early-stopping criterion statistically works better than visiting upto 3 leaves, *i.e.*, smaller KL-divergences and higher retrieval recall, while, visiting 3 or more leaves statistically have smaller KL-divergences and higher retrieval recall (although with lower confidence level $(p < 0.10)$). Our findings are expected as the number of visited leaves increases in bb-tree, the probability of finding true nearest neighbors increases. While on the other hand, by using the early-stopping criterion, we significantly have lower number of divergence calculations $(p < 0.01)$, in average, half of the KL-divergence calculations (101 vs 200). Thus, the choice of using an early-stopping criterion is a trade-off between retrieval recall and runtime, since visiting each internal node during the traversal of the tree has a computational overload of solving a convex optimization problem via Newton iterations, that is more costly than a linear-time Anderson Darling test. Thus, while exhibiting a limited loss in recall, the early stopping criterion results to be more effective.

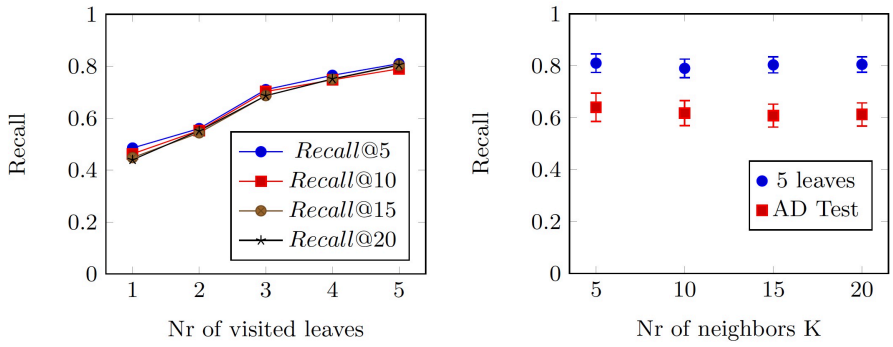


Figure 3.6: (a) Retrieval accuracy in case of leaf-based search.; (b) Comparison in retrieval accuracy between the nearest neighbors search with early stop criterion based on the Anderson-Darling test and the visit of the first 5 leaves of the bb-tree.

Accuracy of rank aggregation. In order to assess which rank aggregation technique is better suited for INFLEX, we conduct an analysis on the accuracy of the aggregations provided by Borda and Copeland. In Table 3.2, we report the average Kendall-τ distance for both unweighted and weighted aggregations, obtained by processing TIM queries with different seed set sizes, while employing top-10 exact nearest neighbors search to retrieve the similar items. In general, weighted versions outperform the unweighted standard ones. Copeland^w achieves the highest accuracy (lowest Kendall-τ) and outperforms all the other techniques (paired t-tests, $(p < 0.05)$). We also obtained similar results with varying values for K.

TIM query evaluation. As highlighted by the previous analysis, given a list of ranked lists, their best aggregation can be achieved by employing the weighted Copeland aggregation technique. However, how to effectively select the ranked lists to pass to the aggregation module is still an open question.

INFLEX implements a fast approximate nearest neighbors search based on an early stopping criterion and a procedure for the automatic selection of ranking lists to aggregate. To evaluate the effect of the combination of these two components in retrieving good seed lists for the aggregation phase, we compare the final performance of INFLEX with the following alternatives:

- **exactKNN**: K-NN exact nearest neighbors search. This is implemented by a complete visit of the bb-tree, which provides true K nearest neighbors. The main drawback of this approach is the costly traversal time.
- **approxKNN**: K-NN approximate nearest neighbors search, realized by setting a maximum number of leaves to explore during the traversal of the bb-tree. The K nearest neighbors among the index items in the visited leaves

k	$Borda$	$Borda^w$	$Copeland$	$Copeland^w$
5	0.100	0.096	0.104	**0.087**
10	0.073	0.066	0.068	**0.062**
15	0.071	0.065	0.068	**0.061**
20	0.068	0.063	0.068	**0.061**
25	0.068	0.066	0.069	**0.064**
30	0.068	0.067	0.071	**0.066**
35	0.071	**0.069**	0.072	**0.069**
40	0.074	0.073	0.075	**0.072**
45	0.079	0.076	0.077	**0.075**
50	0.081	0.080	0.079	**0.077**

Table 3.2: Kendall-τ distance between the seed sets produced by aggregation algorithms and the ground truth computed by standard offline influence maximization computation.

are returned as output. This procedure provides approximate nearest neighbors as output, while it exhibits a speed up over exact search.

- **approxKNN + Sel**: K-NN approximate nearest neighbors search with automatic seed lists selection. Neighbors retrieved by the approximate nearest neighbors search are further refined by applying our procedure of automatic nearest neighbors selection. This is expected to speed up the phase of rank aggregation.

- **approxAD**: fast approximate nearest neighbors search based on the Anderson Darling test. In this case, at each leaf visited, we apply Anderson Darling test, to decide whether or not to continue the search. This heuristic stopping criterion is expected to speed up the search in bb-tree. Its difference from INFLEX is that we do not apply the procedure of nearest neighbors selection.

Preliminary experiments shows that the best accuracy for K-NN based methods is achieved by employing 10 neighbors, which we assume as K in the following analysis. Figure 3.7(a) summarizes the accuracy performance of the considered methods. INFLEX outperforms in accuracy both the approximate K-NN search with automatic selection of index points and the fast search procedure based on the Anderson Darling test. The effectiveness of the procedure for the automatic selection of index points is witnessed by consistent gain of INFLEX over **approxAD**. The paired t-test between Kendall-τ values for INFLEX and **approxKNN** shows that there is no statistical difference between their performance in accuracy $(p < 0.01)$. As expected, we see that the top performing method in

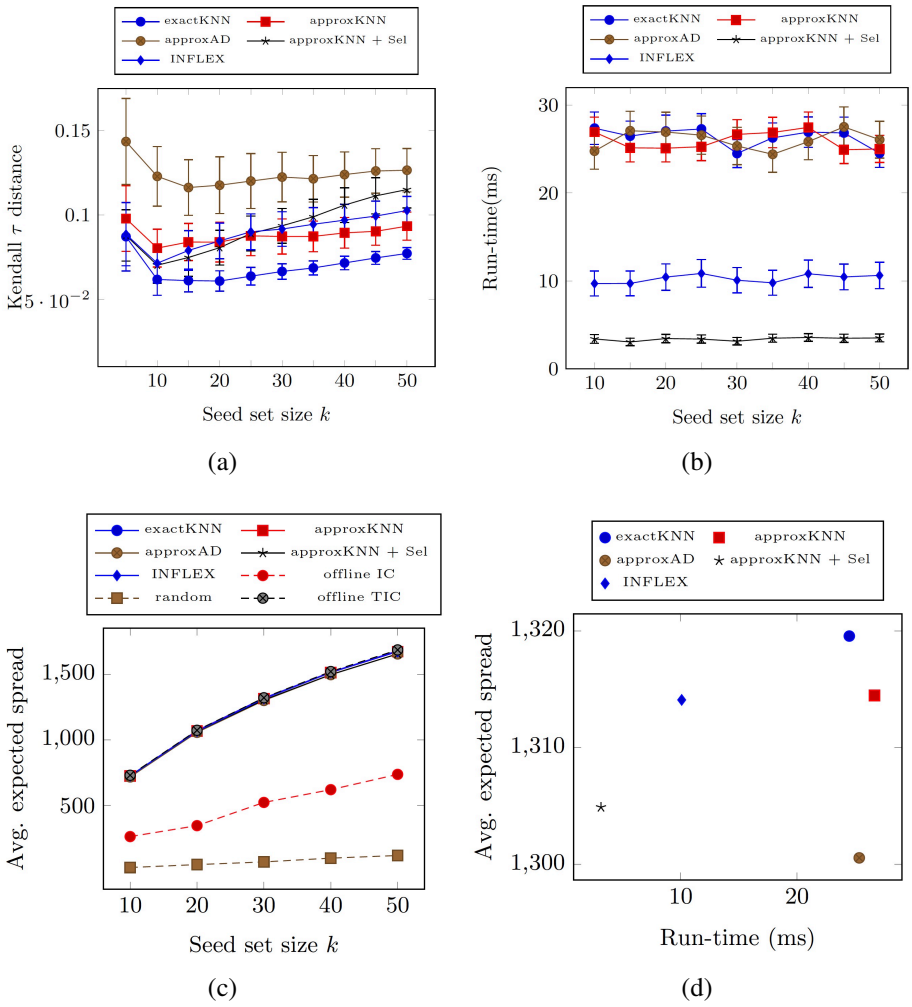


Figure 3.7: (a)Accuracy comparison; (b) Run-time comparison; (c) Expected spread comparison; (d) Run-time vs. expected spread trade-off.

terms of running time, given in Figure 3.7(b), is **approxKNN + Sel** since it applies the procedure of automatic selection of neighbors to already pre-determined number of points and reduce the computational cost of the rank aggregation procedure. INFLEX exhibits a consistent gain in running time over K-NN based methods that do not implement the automatic selection of index lists. Overall, the general framework is able to provide highly accurate estimate of seed sets in less than 30 milliseconds.

An alternative way of assessing the effectiveness of seed sets produced by INFLEX is to consider their resulting expected spread, which can be computed by running Monte Carlo simulations employing the TIC propagation model. More

Method	Exp.Spread	RMSE	NRMSE
offline TIC	1686.31 ± 60.06	-	-
exactKNN	1679.47 ± 60.12	33.23	**0.020**
INFLEX	1673.26 ± 60.80	40.05	**0.023**
approxKNN	1673.24 ± 60.84	39.24	**0.023**
approxAD	1655.30 ± 61.45	55.05	0.033
approxKNN + Sel	1655.79 ± 61.31	98.83	0.059
offline IC	737.15 ± 0.00	1020.61	1.384
random	118.47 ± 5.70	1609.30	13.583

Table 3.3: Avg. Expected Spread of the seed sets for k = 50.

specifically, we compare the spread achieved by seed sets provided by INFLEX with the ones achieved by: (*i*) standard offline TIC influence maximization computation (offline TIC), (*ii*) topic-blind version of standard offline influence maximization computation that is achieved by running the TIC model with a uniform topic distribution (offline IC), and (*iii*) randomly selected seed sets for each item (random).

As we can see from Figure 3.7(c), the seed sets produced by INFLEX and similar aggregation-based alternatives proposed in this chapter can achieve an expected spread that is very close to the one achieved by considering seed sets produced by running compute-intensive TIC influence maximization. On the other hand, seed sets identified by running topic-blind influence maximization computation perform badly, achieving less than half of the spread of the seed nodes provided by TIC influence maximization. We also provide in Table 3.7(c) the Root Mean Square Error (RMSE) and its normalized version (NRMSE) between spread values achieved by different approaches and the ones achieved by offline TIC, which is assumed as ground truth. We see that, although **approxKNN + Sel** performs better than INFLEX in terms of running time, the expected spreads achieved by the seed sets produced by INFLEX are much closer to the ones produced by offline TIC, when compared to the expected spreads achieved by the seed sets of **approxKNN + Sel** (on average 40.05 vs 98.83). The low deviation in terms of expected spread achieved by INFLEX with respect to the ground truth, which is stable also for different choices of k as shown in Table 3.4, statistically confirms the accuracy as well as the robustness of the framework.

3.7 Discussion and Future Work

As a first step towards enabling social-influence online analytics in support of viral marketing decision making, in this chapter we propose an efficient index for

k	INFLEX	offline TIC	RMSE	NRMSE
10	725.33 ± 32.24	730.39 ± 32.11	9.95	0.014
20	1066.20 ± 44.17	1073.28 ± 43.82	14.35	0.013
30	1314.08 ± 51.39	1321.32 ± 50.96	18.50	0.014
40	1513.73 ± 56.91	1521.21 ± 55.96	24.73	0.016
50	1673.26 ± 60.80	1686.31 ± 60.07	40.85	0.024

Table 3.4: Accuracy of the expected spread of seed sets produced by INFLEX.

a very general type of viral marketing queries: influence maximization queries where each item is described by a distribution over a space of topics. The challenge is given by the enormous number of possible queries: essentially any point on the simplex of the topics space. Exploiting a tree-based index for similarity search in non-metric spaces, a clever approximate nearest neighbors search over the tree, and a weighted rank aggregation mechanism, our index can provide, in few milliseconds, a solution very similar (Kendall-$\tau < 0.1$) to the one produced by the standard ground-truth computation, achieving also similar expected spread (NRMSE $< 3\%$).

In our future work, we plan to investigate the automatic determination of the number of index items that is required for maintaining the accuracy of the framework. At the time of the publication of INFLEX [5], CELF++ [75], was the state-of-the-art influence maximization algorithm: on the problem instances that we consider in this chapter, the pre-processing step with CELF++ [75] took from few days to more than a week in order to extract 50 seed nodes for a single item. With the latest advances on devising scalable influence maximization algorithms [22, 42, 112, 131, 132], CELF++ can be directly replaced for the precomputation of the seed sets, which would provide greater flexibility in choosing the number of index and query items, and testing influence indexing techniques.

Another interesting future direction is studying efficient evaluation of other types of viral marketing queries, *e.g.*, when specific market segments are targeted by the viral marketing campaign, combined with *what-if* analysis and visualization paradigms for social-influence online analytics.

CHAPTER 4

SOCIAL ADVERTISING: REGRET MINIMIZATION

4.1 Introduction

Advertising on social networking and microblogging platforms is one of the fastest growing sectors in digital advertising, further fueled by the explosion of investments in mobile ads. Social ads are typically implemented by platforms such as Twitter, Tumblr, and Facebook through the mechanism of *promoted posts* shown in the "timeline" (or feed) of their users. A promoted post can be a video, an image, or simply a textual post containing an advertising message. Similar to organic (non-promoted) posts, promoted posts can propagate from user to user in the network by means of social actions such as *"likes"*, *"shares"*, or *"reposts"*.[1] Below, we blur the distinction between these different types of action, and generically refer to them all as *clicks*. These actions have two important aspects in common: (1) they can be seen as an explicit form of acceptance or endorsement of the advertising message; (2) they allow the promoted posts to propagate, so that they might be visible to the *"friends"* or *"followers"* of the endorsing (i.e., clicking) users. In particular, the platform may supplement the ads with *social proofs* such as *"X, Y, and 3 other friends clicked on it"*, which may further increase the chance that a user will click [7, 134].

This type of advertisement usually follows a *cost per engagement* (CPE) model. The *advertiser* enters into an agreement with the platform owner, called the *host*: the advertiser agrees to pay the host an amount $cpe(i)$ for each click

[1] `Tumblr`'s CEO David Karp reported (CES 2014) that a normal post is reposted on average 14 times, while promoted posts are on average reposted more than 10 000 times: `http://yhoo.it/1vFfIAc`.

received by its ad i. The clicks may come not only from the users who saw i as a promoted ad post, but also their (transitive) followers, who saw it because of viral propagation. The agreement also specifies a budget B_i, that is, the advertiser a_i will pay the host the total cost of all the clicks received by i, *up to a maximum of* B_i. Naturally, posts from different advertisers may be promoted by the host concurrently.

Given that promoted posts are inserted in the timeline of the users, they compete with organic social posts and with one another for a user's attention. A large number of promoted posts (ads) pushed to a user by the system would disrupt user experience, leading to disengagement and eventually abandonment of the platform. To mitigate this, the host limits the number of promoted posts that it shows to a user within a fixed time window, e.g., a maximum of 5 ads per day per user: we call this bound the *user-attention bound*, κ_u, which may be user specific [94].

A subtle point here is that, ads directly promoted by the host count against user attention bound. On the contrary, an ad i that flows from a user u to her follower v should not count toward v's attention bound. In fact, v is receiving ad i from user u, whom she is voluntarily following: as such, it cannot be considered "promoted".

A naïve ad allocation[2] would match each ad with the users that are most likely to click on the ad. However, the above strategy fails to leverage the possibility of ads propagating virally from endorsing users to their followers. We next illustrate the gains achieved by an allocation that takes viral ad propagation into account.

Viral ad propagation: why it matters. For our example we use the toy social network in Figure 4.1. We assume that each time a user clicks on a promoted post, the system produces a social proof for such engagement action, thanks to which her followers might be influenced to click as well. In order to model the propagation of (promoted) posts in the network, we can borrow from the rich body of work done in diffusion of information and innovations in social networks. In particular, the *Independent Cascade* (IC) model [88], adapted to our setting, says that once a user u clicks on an ad, she has one independent attempt to try to influence each of her neighbors v. Each attempt succeeds with a probability $p^i_{u,v}$ which depends on the topics of the specific ad i and the influence exerted by u on her neighbor v. The propagation stops when no new users get influenced. Similarly, we model the *intrinsic relevance* of a promoted post i to a user u, as the probability $\delta(u, i)$ that u will click on ad i, based on the content of the ad and her own interest profile, i.e., the prior probability that the user will click on a promoted post in the absence of any social proof. Since the model is probabilistic,

[2] In the rest of the chapter we use the form "allocating ads to users" as well as "allocating users to ads" interchangeably.

we focus on the number of clicks that an ad receives in *expectation*. Formal details of the propagation model, the topic model, and the definition of expected revenue are deferred to Section 4.1.1.

Consider the example in Figure 4.1, where we assume peer influence probabilities (on edges) are equal for all the four ads $\{a, b, c, d\}$. The figure also reports $\delta(u, i)$ and advertiser budgets. For each advertiser, CPE is 1 and the attention bound for every user is 1, i.e., no user wants more than one ad promoted to her by the host. The expected revenue for an allocation is the same as the resulting expected number of clicks, as the CPE is 1. Below, for simplicity, we round all numbers to the second decimal *after* calculating them all.

Let us consider two ways of allocating users to ads by the host. In allocation $\mathcal{A}$, the host matches each user to her top preference(s) based on $\delta(u, i)$, subject to not violating the attention bound. This results in ad a being assigned to all six users, since it has the highest engagement probability for every user. No further ads may be promoted without violating the attention bound. In allocation $\mathcal{B}$, the host recognizes viral propagation of ads and thus assigns a to v_1 and v_2, b to v_3, c to v_4 and v_5, and d to v_6.

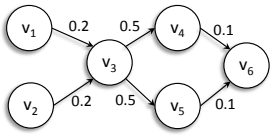
Under allocation $\mathcal{A}$, clicks on a may come from all six users: v_1, v_2 click on a with probability 0.9. However, v_3 clicks on a w.p. $(1-(1-0.9\cdot 0.2)^2(1-0.9)) = 0.93$. This is obtained by combining three factors: v_3's engagement probability of 0.9 with ad a, and probability $0.9 \cdot 0.2$ with which each of v_1, v_2 clicks on a and influences v_3 to click on a. In a similar way one can derive the probability of clicking on a for v_4, v_5, and v_6 (reported in the figure). The overall expected revenue for allocation $\mathcal{A}$ is the sum of all clicking probabilities: $2 \times 0.9 + 0.93 + 2 \times 0.95 + 0.92 = \mathbf{5.55}$.

Under allocation $\mathcal{B}$, the ad a is promoted to only v_1 and v_2 (which click on it w.p. 0.9). Every other user that clicks on a does so solely based on social influence. Thus, v_3 clicks on a w.p. $1 - (1 - 0.9 \cdot 0.2)^2 = 0.33$. Similarly one can derive the probability of clicking on a for v_4, v_5, and v_6 (reported in the figure). Contributions to the clicks on b can only come from nodes v_3, v_4, v_5, v_6. They click on b, respectively, w.p. 0.8, $0.8 \cdot 0.5 = 0.4$, $0.8 \cdot 0.5 = 0.4$, and $1 - (1 - 0.8 \cdot 0.5 \cdot 0.1)^2 = 0.08$.

Finally, it can be verified that the expected number of clicks on ad c is $0.7 + 0.7 + (1 - (1 - 0.7 \cdot 0.1)^2)$, while on d is just 0.6. The overall number of expected clicks under allocation $\mathcal{B}$ is **6.3**.

Observations: (1) Careful allocation of users to ads that takes viral ad propagation into account can outperform an allocation that merely focuses on immediate clicking likelihood based on the content relevance of the ad to a user's interest profile. It is easy to construct instances where the gap between the two can be arbitrarily high by just replicating the gadget in Figure 4.1.

- Ads = $\{a, b, c, d\}$
- p^i_{uv} (on the edges) are the same $\forall i \in \{a, b, c, d\}$
- $\forall u \in \{v_1, \ldots, v_6\}: \quad \delta(u, a) = 0.9, \quad \delta(u, b) = 0.8,$ $\delta(u, c) = 0.7, \quad \delta(u, d) = 0.6$
- $B_a = 4, B_b = 2, B_c = 2, B_d = 1$
- $\kappa_u = 1 \quad \forall u \in \{v_1, \ldots, v_6\}$



Allocation $\mathcal{A}$: *maximizing* $\delta(u, i)$
$\langle v_1, a\rangle, \langle v_2, a\rangle, \langle v_3, a\rangle, \langle v_4, a\rangle, \langle v_5, a\rangle, \langle v_6, a\rangle$

$Pr^{\mathcal{A}}(click(v_1, a)) = Pr^{\mathcal{A}}(click(v_2, a)) = 0.9$
$Pr^{\mathcal{A}}(click(v_3, a)) = 1 - (1 - 0.9 \cdot 0.2)^2(1 - 0.9) = 0.93$
$Pr^{\mathcal{A}}(click(v_4, a)) = Pr^{\mathcal{A}}(click(v_5, a)) = 1 - (1 - 0.93 \cdot 0.5)(1 - 0.9) = 0.95$
$Pr^{\mathcal{A}}(click(v_6, a)) = 1 - (1 - 0.95 \cdot 0.1)^2(1 - 0.9) = 0.92$

Expected number of clicks = $2 \times 0.9 + 0.93 + 2 \times 0.95 + 0.92 = \mathbf{5.55}$

Allocation $\mathcal{B}$: *leveraging virality*
$\langle v_1, a\rangle, \langle v_2, a\rangle, \langle v_3, b\rangle, \langle v_4, c\rangle, \langle v_5, c\rangle, \langle v_6, d\rangle$

$Pr^{\mathcal{B}}(click(v_1, a)) = Pr^{\mathcal{B}}(click(v_2, a)) = 0.9$
$Pr^{\mathcal{B}}(click(v_3, a)) = 1 - (1 - 0.9 \cdot 0.2)^2 = 0.33$
$Pr^{\mathcal{B}}(click(v_4, a)) = Pr^{\mathcal{B}}(click(v_5, a)) = 0.33 \cdot 0.5 = 0.16$
$Pr^{\mathcal{B}}(click(v_6, a)) = 1 - (1 - 0.16 \cdot 0.1)^2 = 0.03$
$Pr^{\mathcal{B}}(click(v_3, b)) = 0.8$
$Pr^{\mathcal{B}}(click(v_4, b)) = Pr^{\mathcal{B}}(click(v_5, b)) = 0.8 \cdot 0.5 = 0.4$
$Pr^{\mathcal{B}}(click(v_6, b)) = 1 - (1 - 0.8 \cdot 0.5 \cdot 0.1)^2 = 0.08$
$Pr^{\mathcal{B}}(click(v_4, c)) = Pr^{\mathcal{B}}(click(v_5, c)) = 0.7$
$Pr^{\mathcal{B}}(click(v_6, c)) = 1 - (1 - 0.7 \cdot 0.1)^2 = 0.14$
$Pr^{\mathcal{B}}(click(v_6, d)) = 0.6$

Expected number of clicks = $2 \cdot 0.9 + 0.33 + 2 \cdot 0.16 + 0.03 + 0.8 + 2 \cdot 0.4 + 0.08 + 2 \cdot 0.7 + 0.14 + 0.6 = \mathbf{6.3}$.

Figure 4.1: Illustrating viral ad propagation. For simplicity, we round all numbers to the second decimal.

(2) Even though allocation $\mathcal{A}$ ignores the effect of viral ad propagation, it still benefits from the latter, as shown in the calculations. This naturally motivates finding allocations that expressly exploit such propagation in order to maximize the expected revenue.

In this context, we study the problem of *how to strategically allocate users to the advertisers, leveraging social influence and the propensity of ads to propagate.* The major challenges in solving this problem are as follows. Firstly, the host needs to strike a balance between assigning ads to users who are likely to click and assigning them to "influential" users who are likely to boost further propagation of the ads. Moreover, influence may well depend on the "topic" of the ad. E.g., u may influence its neighbor v to different extents on cameras versus health-related products. Therefore, ads which are close in a topic space *will naturally compete* for users that are influential in the same area of the topic space. Summarizing, a good allocation strategy needs to take into account the different CPEs and budgets for different advertisers, users' attention bound and interests, and ads' topical distributions.

An even more complex challenge is brought in by the fact that uncontrolled virality could be undesirable for the host, as it creates room for exploitation by the advertisers: hoping to tap uncontrolled virality, an advertiser might declare a lower budget for its marketing campaign, aiming at the same large outcome with a smaller cost. Thus, from the host perspective, it is important to make sure that the expected revenue from an advertiser is as close to the budget as possible: both undershooting and overshooting the budget results in a *regret* for the host, as illustrated in the following example.

Example 4.1. *Consider again our example in Figure 4.1. Rounding to the first decimal, allocation $\mathcal{A}$ leads to an overall regret of* $|4 - 5.6| + |2 - 0| + |2 - 0| + |1 - 0| = \mathbf{6.6}$*: the expected revenue exceeds the budget for advertiser a by* 1.6 *and falls short of other advertiser budgets by* $2, 2, 1$ *respectively. Similarly, for allocation $\mathcal{B}$, the regret is* $|4 - 2.5| + |2 - 1.7| + |2 - 1.5| + |1 - 0.6| = \mathbf{2.7}$.

The host knows it will not be paid beyond the budget of each advertiser, so that any excess above the budget is essentially "free service" given away by the host, which causes regret, and any shortfall w.r.t. the budget is a lost revenue opportunity which causes regret as well. This creates a challenging trade-off: on the one hand, the host aims at leveraging virality and the network effect to improve advertising efficacy, while on the other hand, the host wants to avoid giving away free service due to uncontrolled virality.

4.1.1 Problem Definition

The Ingredients. The computational problem studied in this chapter is from the host perspective. The host owns: (i) a *directed social graph* $G = (V, E)$, where an

arc (u, v) means that v follows u, thus v can see u's posts and can be influenced by u; (ii) a *topic model* for ads and users' interest, defined on a space of K topics; (iii) a *topic-aware influence propagation model* defined on the social graph G and the topic model.

The key idea behind the topic modeling is to introduce a hidden variable Z that can range among K states. Each topic (i.e., state of the latent variable) represents an abstract interest/pattern and intuitively models the underlying cause for each data observation (a user clicking on an ad). In our setting the host owns a pre-computed probabilistic topic model. The actual method used for producing the model is not important at this stage: it could be, e.g., the popular *Latent Dirichlet Allocation* (LDA) [19], or any other method. What is relevant is that the topic model maps each ad i to a topic distribution $\vec{\gamma_i}$ over the latent topic space, formally: $\gamma_i^z = Pr(Z = z|i)$ with $\Sigma_{z=1}^{K} \gamma_i^z = 1$.

Propagation Model. The propagation model governs the way that ads propagate in the social network driven by social influence. In this work, we extend a simple topic-aware propagation model introduced by Barbieri et al. [13], with Click-Through Probabilities (CTPs) for seeds: we refer to the set of users S_i that receive ad i directly as a promoted post from the host as the *seed set* for ad i. In the *Topic-aware Independent Cascade* model (TIC) of [13], the propagation proceeds as follows: when a node u first clicks an ad i, it has one chance of influencing each inactive neighbor v, independently of the history thus far. This succeeds with a probability that is the weighted average of the arc probability w.r.t. the topic distribution of the ad i:

$$p_{u,v}^i = \sum\nolimits_{z=1}^{K} \gamma_i^z \cdot p_{u,v}^z. \tag{4.1}$$

For each topic z and for a seed node u, the probability $p_{H,u}^z$ represents the likelihood of u clicking on a promoted post for topic z. Thus the CTP $\delta(u, i)$ that u clicks on the promoted post i in absence of any social proof, is the weighted average (as in Eq. (5.1)) of the probabilities $p_{H,u}^z$ w.r.t. the topic distribution of i. In our extended TIC-CTP model, each $u \in S_i$ accepts to be a seed, i.e., clicks on ad i, with probability $\delta(u, i)$ when targeted. The rest of the propagation process remains the same as in TIC.

Following the literature on influence maximization we denote with $\sigma_i(S_i)$ the *expected number of clicks* (according to the TIC-CTP model) for ad i when the seed set is S_i. The corresponding expected revenue is $\Pi_i(S_i) = \sigma_i(S_i) \cdot cpe(i)$, where $cpe(i)$ is the cost-per-engagement that a_i and the host have agreed on.

We observe that for a fixed ad i, with topic distribution $\vec{\gamma_i}$, the TIC-CTP model boils down to the standard *Independent Cascade* (IC) model [88] with CTPs,

where again, a seed may activate with a probability. We next expose the relationship between the expected spread a $\sigma^{ic}(S)$ for the classical IC model without CTPs, and the expected spread under the TIC-CTP model for a given ad i.

Lemma 4.1. *Given an instance of the TIC-CTP model, and a fixed ad i, with topic distribution $\vec{\gamma}_i$, build an instance of IC by setting the probability over each edge (u, v) as in Eq. 5.1. Now, consider any node u, and any set S of nodes. Let $\delta(u, i)$ be the CTP for u clicking on the promoted post i. Then we have*

$$\delta(u,i)[\sigma^{ic}(S \cup \{u\}) - \sigma^{ic}(S)] = \sigma_i(S \cup \{u\}) - \sigma_i(S). \tag{4.2}$$

Proof. The proof relies on the possible world semantics. For the IC model [88], consider a graph $G = (V, E)$ with influence probability $p_{u,v}$ on each edge $(u, v) \in E$. A possible world, denoted X, is a deterministic graph generated as follows. For each edge $(u, v) \in E$, we flip a biased coin: with probability $p_{u,v}$, the edge is declared "live", and with probability $1 - p_{u,v}$, it is declared "blocked".

Define an indicator function $\mathbb{I}_X(S, v)$, which takes on 1 if v is reachable by S via a path consisting entirely of live edges in X, and 0 otherwise. In the IC model,

$$\begin{aligned}
&\sigma^{ic}(S \cup \{u\}) - \sigma^{ic}(S) \\
&= \sum_X \Pr[X] \cdot (|\{w : \mathbb{I}_X(S \cup \{u\}, w) = 1\}| - |\{w : \mathbb{I}_X(S, w) = 1\}|) \\
&= \sum_X \Pr[X] \cdot |\{w : \mathbb{I}_X(S \cup \{u\}, w) = 1 \wedge \mathbb{I}_X(S, w) = 0\}| \\
&= \sum_X \Pr[X] \cdot |\{w : \mathbb{I}_X(\{u\}, w) = 1 \wedge \mathbb{I}_X(S, w) = 0\}|.
\end{aligned}$$

Notice that for a node to be active in a possible world, it must be reachable from a seed. In each of the possible worlds, node u has probability $\delta(u, i)$ to accept to become a seed. Thus, in the TIC-CTP model, we have:

$$\begin{aligned}
&\sigma_i(S \cup \{u\}) - \sigma_i(S) \\
&= \delta(u,i) \cdot \sum_X \Pr[X] \cdot |\{w : \mathbb{I}_X(\{u\}, w) = 1 \wedge \mathbb{I}_X(S, w) = 0\}|.
\end{aligned}$$

This directly leads to

$$\delta(u,i)(\sigma^{ic}(S \cup \{u\}) - \sigma^{ic}(S)) = \sigma_i(S \cup \{u\}) - \sigma_i(S),$$

which was to be shown. □

A corollary of the above lemma is that, for a fixed $\vec{\gamma_i}$, the expected spread $\sigma_i(\cdot)$ function under the TIC-CTP model, inherits the properties of monotonicity and submodularity from the IC model (see Section 4.2 and [13, 88]). In turn, $\Pi_i(S_i) = cpe(i) \cdot \sigma_i(S_i)$ is also monotone and submodular, being a non-negative linear combination of monotone submodular functions.

Budget and Regret. As in any other advertisement model, we assume that each advertiser a_i has a finite budget B_i for a campaign on ad i, which limits the maximum amount that a_i will pay the host. The host needs to allocate seeds to each of the ads that it has agreed to promote, resulting in an allocation $\mathcal{S} = (S_1, ..., S_h)$. The expected revenue from the campaign may fall short of the budget (i.e., $\Pi_i(S_i) < B_i$) or overshoot it (i.e., $\Pi_i(S_i) > B_i$). An advertiser's natural goal is to make its expected revenue as close to B_i as possible: the former situation is lost opportunity to make money whereas the latter amounts to "free service" by the host to the advertiser. Both are undesirable. Thus, one option to define the host's regret for seed set allocation S_i for advertiser a_i is as $|B_i - \Pi_i(S_i)|$.

Note that this definition of regret has the drawback that it does not discriminate between small and large seed sets: given two seed sets S_1 and S_2 with the same regret as defined above, and with $|S_1| \ll |S_2|$, this definition does not prefer one over the other. In practice, it is desirable to achieve a low regret with a small number of seeds. By drawing on the inspiration from the optimization literature [25], where an additional penalty corresponding to the complexity of the solution is added to the error function to discourage overfitting, we propose to add a similar penalty term to discourage the use of large seed sets. Hence we define the *overall regret* as

$$\mathcal{R}_i(S_i) = |B_i - \Pi_i(S_i)| + \lambda \cdot |S_i|. \tag{4.3}$$

Here, $\lambda \cdot |S_i|$ can be seen as a penalty for the use of a seed set: the larger its size, the greater the penalty. This discourages the choice of a large number of poor quality seeds to exhaust the budget. When $\lambda = 0$, no penalty is levied and the "raw" regret corresponding to the budget alone is measured. We assume w.l.o.g. that the scalar λ encapsulates CPE such that the term $\lambda|S_i|$ is in the same

monetary unit as B_i. How small/large should λ be? We will address this question in the next section. The overall regret from an allocation $\mathcal{S} = (S_1, ..., S_h)$ to all advertisers is

$$\mathcal{R}(\mathcal{S}) = \sum\nolimits_{i=1}^{h} \mathcal{R}_i(S_i). \tag{4.4}$$

Example 4.2. *In Example 4.1, the regrets reported for allocations $\mathcal{A}$ (6.6) and $\mathcal{B}$ (2.7) correspond to $\lambda = 0$. When $\lambda = 0.1$, the regrets change to $6.6+0.1\times6 = 7.2$ for $\mathcal{A}$ and to $2.7 + 0.1 \times 6 = 3.3$ for $\mathcal{B}$.*

As noted earlier in Section 4.1, in practice, the number of ads that can be promoted to a user may be limited. The host can even personalize this number depending on users' activity. We model this using a user-specific attention bound κ_u for each user $u \in V$. An allocation $\mathcal{S} = (S_1, ..., S_h)$ is called *valid* provided for every user $u \in V$, $|\{S_i \in \mathcal{S} \mid u \in S_i\}| \leq \kappa_u$, i.e., no more than κ_u ads are promoted to u by the allocation. We are now ready to formally state the problem we study.

Problem 4.1 (REGRET-MINIMIZATION)**.** *We are given h advertisers $a_1, \ldots, a_h$, where each a_i has an ad i described by topic-distribution $\vec{\gamma}_i$, a budget B_i, and a cost-per-engagement $cpe(i)$. Also given is a social graph $G = (V, E)$ with a probability $p^z_{u,v}$ for each edge $(u, v) \in E$ and each topic $z \in [1, K]$, an attention bound κ_u, $\forall u \in V$, and a penalty parameter $\lambda \geq 0$. The task is to compute a valid allocation $\mathcal{S} = (S_1, \ldots, S_h)$ that minimizes the overall regret:*

$$\mathcal{S} = \underset{\substack{\mathcal{T}=(T_1,\ldots,T_h):T_i\subseteq V \\ \mathcal{T} \text{ is valid}}}{\operatorname{argmin}} \mathcal{R}(\mathcal{T}).$$

Discussion. Note that $\Pi_i(S_i)$ denotes the expected revenue from advertiser a_i. In reality, the actual revenue depends on the number of engagements the ad *actually* receives. Thus, the uncertainty in $\Pi_i(S_i)$ may result in a loss of revenue. Another concern could be that regret on the positive side ($\Pi_i(S_i) > B_i$) is more acceptable than on the negative side ($\Pi_i(S_i) < B_i$), as one can argue that maximizing revenue is a more critical goal even if it comes at the expense of a small and reasonable amount of free service. Our framework can accommodate such concerns and can easily address them. For instance, instead of defining raw regret as $|B_i - \Pi_i(S_i)|$, we can define it as $|B'_i - \Pi_i(S_i)|$, where $B'_i = (1+\beta)\cdot B_i$. The idea is to artificially boost the budget B_i with parameter β allowing maximization of revenue while keeping the free service within a modest limit. This small change has no impact

on the validity of our results and algorithms. Theorem 4.2 provides an upper bound on the regret achieved by our allocation algorithm (Section 4.3.1). The bound remains intact except that in place of the original budget B_i, we should use the boosted budget B'_i. This remark applies to all our results. We henceforth study the problem as defined in Problem 4.1.

4.1.2 Contributions and Roadmap

In this chapter we make the following major contributions:

- We propose a novel problem domain of allocating users to advertisers for promoting advertisement posts, taking advantage of the network effect, while paying attention to important practical factors like relevance of ad, effect of social proof, user's attention bound, and limited advertiser budgets (Section 4.1.1).
- We formally define the problem of *minimizing regret* in allocating users to ads (Section 4.1.1), and show that it is NP-hard and is NP-hard to approximate within any factor (Section 4.3).
- We develop a simple greedy algorithm and establish an upper bound on the regret it achieves as a function of advertisers' total budgets (Section 4.3.1).
- We then devise a scalable instantiation of the greedy algorithm by leveraging the notion of *random reverse-reachable sets* [22, 132] (Section 4.4).
- Our extensive experimentation on four real datasets confirms that our algorithm is scalable and it delivers high quality solutions, significantly outperforming natural baselines (Section 4.5).

To the best of our knowledge, regret minimization in the context of promoting multiple ads in a social network, subject to budget and attention bounds has not been studied before. Related work is discussed in Section 4.2, while Section 4.6 concludes the chapter discussing future work.

4.2 Related Work

Datta et al. [51] study influence maximization with multiple items, under a user attention constraint. However, as in classical influence maximization, their objective is to maximize the overall influence spread, and the budget is w.r.t. the size of the seed set, so without any CPE model. Their diffusion model is the (topic-blind) IC model, which also doesn't model the competition among similar items. Du et al. [55] study influence maximization over multiple non-competing products subject to user attention constraints and budget constraints, and develop approximation algorithms in a continuous time setting. Lin et al. [94] study the problem

of maximizing influence spread from a website's perspective: how to dynamically push items to users based on user preference and social influence. The push mechanism is also subject to user attention bounds. Their framework is based on Markov Decision Processes (MDPs).

Our work departs from the body of work in this field by looking at the possibility of integrating viral marketing into existing social advertising models and by studying a fundamentally different objective: *minimize host's regret*. A noteworthy feature of our work is that, as will be shown in §4.5, the budgets we use are such that thousands of seeds are required to minimize regret. Scalability of algorithms for selecting thousands of seeds over large networks has not been demonstrated before.

While social advertising is still in its infancy, it fits in the more general (and mature) area of computational advertising that has attracted a lot of interest during the last decade. The central problem of computational advertising is to find the "best match" between a given user in a given context and a suitable advertisement. The context could be a user entering a query in a search engine ("sponsored search"), reading a web page ("content match" and "display ads"), or watching a movie on a portable device, etc. The most typical example is sponsored search: search engines show ads deemed relevant to user-issued queries, in the hope of maximizing click-through rates and in turn, revenue. Revenue maximization in this context is formalized as the well-known *Adwords* problem [103]. We are given a set Q of keywords and N bidders with their daily budgets and bids for each keyword in Q. During a day, a sequence of words (all from Q) would arrive online and the task is to assign each word to one bidder *upon its arrival*, with the objective of maximizing revenue for the given day while respecting the budgets of all bidders. This can be seen as a generalized online bipartite matching problem, and by using linear programming techniques, a $(1 - 1/e)$ competitive ratio is achieved [103]. Considerable work has been done in sponsored search and display ads [52, 62, 63, 68, 105]. For a comprehensive treatment, see a recent survey [102]. Our work fundamentally differs from this as we are concerned with the *virality* of ads when making allocations: this concept is still largely unexplored in computational advertising.

Recently, Tucker [134] and Bakshy et al. [7] conducted field experiments on Facebook and demonstrated that adding social proofs to sponsored posts in Facebook's News Feed significantly increased the click-through rate. Their findings empirically confirm the benefits of social influence, paving the way for the application of viral marketing in social advertising, as we do in our work.

4.3 Theoretical Analysis

We first show that REGRET-MINIMIZATION is not only NP-hard to solve optimally, but is also NP-hard to approximate within any factor (Theorem 4.1). On the positive side, we propose a greedy algorithm and conduct a careful analysis to establish a bound on the regret it can achieve as a function of the budget (Theorems 4.2-4.4).

Theorem 4.1. REGRET-MINIMIZATION *is NP-hard and is NP-hard to approximate within any factor.*

Proof. We prove hardness for the special case where $\lambda = 0$, using a reduction from 3-PARTITION [67].

Given a set $X = \{x_1, ..., x_{3m}\}$ of positive integers whose sum is C, with $x_i \in (C/4m, C/2m)$, $\forall i$, 3-PARTITION asks whether X can be partitioned into m disjoint 3-element subsets, such that the sum of elements in each partition is the same (= C/m). This problem is known to be strongly NP-hard, i.e., it remains NP-hard even if the integers x_i are bounded above by a polynomial in m [67]. Thus, we may assume that C is bounded by a polynomial in m.

Given an instance $\mathcal{I}$ of 3-PARTITION, we construct an instance $\mathcal{J}$ of REGRET-MINIMIZATION as follows. First, we set the number of advertisers $h = m$ and let the cost-per-engagement (CPE) be 1 for all advertisers. Then, we construct a directed bipartite graph $G = (U \cup V, E)$: for each number x_i, G has one node $u_i \in U$ with $x_i - 1$ outneighbors in V, with all influence probabilities set to 1. We refer to members of U (resp., V) as "U" nodes (resp., "V" nodes) below. Set all advertiser budgets to $B_i = C/m$, $1 \leq i \leq m$ and the attention bound of every user to 1. This will result in a total of C nodes in the instance of REGRET-MINIMIZATION. Since C is bounded by a polynomial in m, the reduction is achieved in polynomial time.

We next show that if REGRET-MINIMIZATION can be solved in polynomial time, so can 3-PARTITION, implying hardness. To that end, assume there exists an algorithm $\mathbf{A}$ that solves REGRET-MINIMIZATION optimally. We can use $\mathbf{A}$ to distinguish between YES- and NO-instances of 3-PARTITION as follows. Run $\mathbf{A}$ on $\mathcal{J}$ to yield a seed set allocation $\mathcal{S} = (S_1, ..., S_m)$. We claim that $\mathcal{I}$ is a YES-instance of 3-PARTITION iff $\mathcal{R}(\mathcal{S}) = 0$, i.e., the total regret of the allocation $\mathcal{S}$ is zero.

($\Longrightarrow$): Suppose $\mathcal{R}(\mathcal{S}) = 0$. This implies the regret of every advertiser must be zero, i.e., $\Pi_i(S_i) = B_i = C/m$. We shall show that in this case, each S_i must consist of 3 "U" nodes whose spread sums to C/m. From this, it follows that the 3-element subsets $X_i := \{x_j \in X \mid u_j \in S_i\}$ witness the fact that $\mathcal{I}$ is a YES-instance. Suppose $|S_i| \neq 3$ for some i. It is trivial to see that each seed set S_i can contain only the "U" nodes, for the spread of any "V" node is just 1. If

Algorithm 4: Greedy Algorithm

Input : $G = (V, E)$; λ; attention bounds $\kappa_u, \forall u \in V$; items $\vec{\gamma}_i$ with $cpe(i)$ & budget B_i, $i = 1, \ldots, h$; $\delta(u, i), \forall u \forall i$

Output: $S_1, \ldots, S_h$

1 $S_i \leftarrow \emptyset, \forall i = 1, \ldots, h$

2 **while** *true* **do**

3 $(u, a_i) \leftarrow \text{argmax}_{v,a_j} \mathcal{R}_j(S_j) - \mathcal{R}_j(S_j \cup \{v\})$, subject to: $|\{S_\ell | v \in S_\ell\}| < \kappa_v \ \wedge \ \mathcal{R}_j(S_j \cup \{v\}) \leq \mathcal{R}_j(S_j))$

4 **if** (u, a_i) is null **then return** ;

5 **else** $S_i \leftarrow S_i \cup \{u\}$;

$|S_i| \neq 3$, then $\Pi_i(S_i) = \sum_{u_j \in S_i} x_j \neq C/m$, since all numbers are in the open interval $(C/4m, C/2m)$. This shows that every seed set S_i in the above allocation must have size 3, which was to be shown.
($\Longleftarrow$): Suppose $X_1, ..., X_m$ are disjoint 3-element subsets of X that each sum to C/m. By choosing the corresponding "U"-nodes we get a seed set allocation whose total regret is zero.

We just proved that REGRET-MINIMIZATION is NP-hard. To see hardness of approximation, suppose $\mathbf{B}$ is an algorithm that approximates REGRET-MINIMIZATION within a factor of α. That is, the regret achieved by algorithm $\mathbf{B}$ on any instance of REGRET-MINIMIZATION is $\leq \alpha \cdot OPT$, where OPT is the optimal (least) regret. Using the same reduction as above, we can see that the optimal regret on the reduced instance $\mathcal{J}$ above is 0. On this instance, the regret achieved by algorithm $\mathbf{B}$ is $\leq \alpha \cdot 0 = 0$, i.e., algorithm $\mathbf{B}$ can solve REGRET-MINIMIZATION optimally in polynomial time, which is shown above to be impossible unless $P = NP$. □

4.3.1 A Greedy Algorithm

Due to the hardness of approximation of Problem 1, no polynomial algorithm can provide any theoretical guarantees w.r.t. optimal overall regret. Still, instead of jumping to heuristics without any guarantee, we present an intuitive greedy algorithm (pseudo-code in Algorithm 4) with theoretical guarantees in terms of the total budget. It is worth noting that analyzing regret w.r.t. the total budget has real-world relevance, as budget is a concrete monetary and known quantity (unlike optimal value of regret) which makes it easy to understand regret from a business perspective.

The algorithm starts by initializing all the seed sets to be empty (line 1). It keeps selecting and allocating seeds until regret can no longer be minimized. In each iteration, it finds a user-advertiser pair (u, a_i) such that u's attention bound

is not violated (that is, $|\{S_i|u \in S_i\}| < \kappa_u$) and adding u to S_i (the seed set of a_i) yields the largest decrease in regret among all the valid pairs: clearly, we want to ensure that regret does not increase in an iteration (that is, $\mathcal{R}_i(S_i \cup \{u\}) < \mathcal{R}_i(S_i)$) (line 3). The user u is then added to S_i. If no such pair can be found, that is, regret cannot be reduced further, the algorithm terminates (line 4).

Before stating our results on bounding the overall regret achieved by the greedy algorithm, we identify extreme (and unrealistic) situations where no such guarantees may be possible.

Practical considerations. Consider a network with n users, one advertiser with a CPE of 1 and a budget $B \gg n$. Assume CTPs are all 1. Clearly, even if all n users are allocated to the advertiser, the regret approaches 100% of B, as most of the budget cannot be tapped.

At another extreme, consider a dense network with n users (e.g., clique), one advertiser with a CPE of 1 and a budget $B \ll n$. Suppose the network has high influence probabilities, such that assigning *any* one seed u to the advertiser will result in an expected revenue $\Pi(\{u\}) \gg B$. In this case, the allocation with the least regret is the empty allocation (!) and the regret is exactly B!

In many practical settings, the budgets are large enough that the marginal gain of any one node is a small fraction of the budget, and small enough compared to the network size, in that there are enough nodes in the network to allocate to each advertiser in order to exhaust or exceed the budget.

4.3.2 The General Case

In this subsection, we establish an upper bound on the regret achieved by Algorithm 4, when every candidate seed has essentially an unlimited attention bound. For convenience, we refer to the first term in the definition of regret (cf. Eq. 4.3) as *budget-regret* and the second term as *seed-regret*. The first one reflects the regret arising from undershooting or overshooting the budget and the second arises from utilizing seeds which are the host's resources. For a seed set S_i for ad i, the *marginal gain* of a node $x \in V \setminus S_i$ is defined as $MG_i(x|S_i) := \Pi_i(S_i \cup \{x\}) - \Pi_i(S_i)$. By submodularity, the marginal gain of any node is the greatest w.r.t. the empty seed set, i.e., $MG_i(x|\emptyset) = \Pi_i(\{x\})$. Let p_i be the maximum marginal gain of any node w.r.t. ad i, as a fraction of its budget B_i, i.e., $p_i := \max_{x \in V} \Pi_i(\{x\})/B_i$. As discussed at the end of the previous subsection, we assume that the network and the budgets are such that $p_i \in (0, 1)$, for all ads i. In practice, p_i tends to be a small fraction of the budget B_i. Finally, we define $p_{max} := \max_{i=1}^{h} p_i$ to be the maximum p_i among all advertisers.

Theorem 4.2. *Suppose that for every node u, the attention bound $\kappa_u \geq h$, the number of advertisers, and that $\lambda \leq \delta(u,i) \cdot cpe(i)$, $\forall$ user u and ad i. Then the regret incurred by Algorithm 4 upon termination is at most*

$$\sum_{i=1}^{h} \frac{p_i B_i + \lambda}{2} \; + \; \lambda \cdot \sum_{i=1}^{h} \left(1 + s^i_{opt} \lceil \ln \frac{1}{p_i/2 - \lambda/2B_i} \rceil \right),$$

where s^i_{opt} is the smallest number of seeds required for reaching or exceeding the budget B_i for ad i.

of Theorem 4.2. We establish a series of claims.

Claim 4.1. *Suppose S_i is the seed set allocated to advertiser a_i and $\Pi_i(S_i) < B_i$. Then the greedy algorithm will add a node x to S_i iff $|\Pi_i(S_i \cup \{x\}) - B_i| < |\Pi_i(S_i) - B_i|$ and $x = \text{argmax}_{w \in V \setminus S_i} (|\Pi_i(S_i) - B_i| - |\Pi_i(S_i \cup \{w\}) - B_i|)$, with ties broken arbitrarily.*

PROOF OF CLAIM: Let x be a node such that its addition to S_i strictly reduces the budget-regret and it results in the greatest reduction in budget-regret, among all nodes outside S_i. The contribution of every node outside S_i to the seed regret (i.e., the penalty term) is the same and is equal to λ. Thus, any node that achieves the maximum budget-regret reduction will have the maximum overall regret reduction. Furthermore, the overall regret reduction of adding such a node x to S_i will be non-negative, since its contribution to budget-regret reduction is at least $1 \cdot \delta(u,i) \cdot cpe(u,i) \geq \lambda$. So Greedy will add such a node x to S_i. Attention bound does not constrain this addition in anyway since $\kappa_u \geq h, \forall u$.
($\Longrightarrow$): Let x be the node added by Greedy to S_i. By definition, the addition of x to S_i results in a non-negative reduction in overall regret and it leads to the maximum overall regret reduction. By the argument in the "If" direction, x must also lead to the maximum reduction in the budget-regret, since seed-regret cannot discriminate between nodes. We will show that this reduction is strictly positive. Since Greedy added x to S_i, we have $\mathcal{R}(S_i \cup \{x\}) = |\Pi_i(S_i \cup \{x\}) - B_i| + \lambda \cdot (|S_i| + 1) \leq |\Pi_i(S_i) - B_i| + \lambda \cdot (|S_i|) = \mathcal{R}(S_i)$.
$\Longrightarrow |\Pi_i(S_i \cup \{x\}) - B_i| \leq |\Pi_i(S_i) - B_i| - \lambda$, that is,
$|\Pi_i(S_i \cup \{x\}) - B_i| < |\Pi_i(S_i) - B_i|$. This was to be shown. □

Claim 4.2. *The budget-regret of Greedy for advertiser a_i, upon termination, is at most $(p_i B_i + \lambda)/2$.*

PROOF OF CLAIM: Consider any iteration j. Let x be the seed allocated to advertiser a_i in this iteration. The following cases arise.

• Case 1: $\Pi_i(S_i \cup \{x\}) < p_i B_i$. By submodularity, for any node $y \in V \setminus (S_i \cup \{x\}) : MG_i(y|S_i \cup \{x\}) \leq MG_i(y|\emptyset) \leq p_i B_i$. Thus, from Claim 4.1, we know the algorithm will continue adding seeds to S_i until Case 2 (below) is reached.
• Case 2: $\Pi(S_i \cup \{x\}) \geq p_i B_i$.

• Case 2a: $\Pi(S_i \cup \{x\}) < B_i$. If x is the last seed added to S_i, then $\forall y \in V \setminus (S_i \cup \{x\}) : B_i - \Pi(S_i \cup \{x\}) + \lambda(|S_i| + 1) < \Pi_i(S_i \cup \{x\} \cup \{y\}) - B_i + \lambda(|S_i| + 2)$. Notice that upon adding any such y, a cross-over must occur w.r.t. B_i: suppose otherwise, then adding y would cause net drop in regret and the algorithm would just add y to $S_i \cup \{x\}$, a contradiction. Simplifying, we get $B_i - \Pi_i(S_i \cup \{x\}) < \Pi_i(S_i \cup \{x\} \cup \{y\}) - B_i + \lambda$. Also by submodularity, we have $\Pi_i(S_i \cup \{x\} \cup \{y\}) - \Pi_i(S_i \cup \{x\}) \leq p_i B_i$. Thus,
$\implies \Pi_i(S_i \cup \{x\} \cup \{y\}) - B_i + B_i - \Pi_i(S_i \cup \{x\}) \leq p_i B_i$.
$\implies 2(B_i - \Pi_i(S_i \cup \{x\})) - \lambda \leq p_i B_i$.
$\implies B_i - \Pi_i(S_i \cup \{x\}) \leq (p_i B_i + \lambda)/2$.

• Case 2b: $\Pi_i(S_i \cup \{x\}) > B_i$. Since Greedy just added x to S_i, we infer that $\Pi_i(S_i) < B_i$ and $[B_i - \Pi_i(S_i)] + \lambda|S_i| \geq \Pi_i(S_i \cup \{x\}) - B_i + \lambda(|S_i| + 1)$.
$\implies B_i - \Pi_i(S_i) \geq \Pi_i(S_i \cup \{x\}) - B_i + \lambda$. Clearly, x must be the last seed added to S_i, as any future additions will strictly raise the regret. By submodularity, we have

$\Pi_i(S_i \cup \{x\}) - \Pi_i(S_i) \leq p_i B_i$.
$\implies \Pi_i(S_i \cup \{x\}) - B_i + B_i - \Pi_i(S_i) \leq p_i B_i$.
$\implies 2(\Pi_i(S_i \cup \{x\}) - B_i) + \lambda \leq p_i B_i$.
$\implies \Pi_i(S_i \cup \{x\}) - B_i \leq (p_i B_i - \lambda)/2$.

By combining both cases, we conclude that the budget-regret of Greedy for a_i upon termination is $\leq (p_i B_i + \lambda)/2$. □

Next, define $\eta_0 = B_i$. Let S_i^j be the seed set assigned to advertiser a_i by Greedy after iteration j. Let $\eta_j := \eta_0 - \Pi_i(S_i^j)$, i.e., the shortfall of the achieved revenue w.r.t. the budget B_i, after iteration j, for advertiser a_i.

Claim 4.3. *After iteration j, $\exists x \in V \setminus S_i^j : \Pi_i(S_i \cup \{x\}) - \Pi_i(S_i) \geq 1/s_{opt}^i \cdot \eta_j$, where s_{opt}^i is the minimum number of seeds needed to achieve a revenue no less than B_i.*

PROOF OF CLAIM: Suppose otherwise. Let S_i^* be the seeds allocated to advertiser a_i by the optimal algorithm for achieving a revenue no less than B_i. Add seeds in $S_i^* \setminus S_i^j$ one by one to S_i^j. Since none of them has a marginal gain w.r.t. S_i that is $\geq 1/s_{opt}^i \cdot \eta_j$, it follows by submodularity that $\Pi_i(S_i^j \cup S_i^*) \leq \Pi(S_i^j) + s_{opt}^i \cdot 1/s_{opt}^i \cdot \eta_j < B_i$, a contradiction. □

It follows from the above proof that $\eta_j \leq \eta_{j-1} \cdot (1 - 1/s_{opt}^i)$, which implies that $\eta_j \leq 1/\eta_{j-1} \cdot e^{-1/s_{opt}^i}$. Unwinding, we get $\eta_j \leq \eta_0 \cdot e^{-j/s_{opt}^i}$. Suppose Greedy stops in ℓ iterations. We showed above that the budget-regret of Greedy,

for advertiser a_i, at the end of this iteration, is either at most $(p_i \cdot B_i + \lambda)/2$ or is at most $(p_i B_i - \lambda)/2$ depending on the case that applies. Of these, the latter is more stringent w.r.t. the #iterations Greedy will take, and hence w.r.t. the #seeds it will allocate to a_i. So, in iteration $\ell - 1$, we have $\eta_{\ell-1} \geq (p_i B_i - \lambda)/2$. That is,
$\eta_{\ell-1} = B_i \cdot e^{-(\ell-1)/s^i_{opt}} \geq (p_i B_i - \lambda)/2$, or
$\implies e^{-(\ell-1)/s^i_{opt}} \geq (p_i - \lambda/B_i])/2$.
$\implies \ell \leq 1 + s^i_{opt} \cdot \lceil \ln\{1/(p_i/2 - \lambda/2B_i)\}\rceil$. Notice that this is an upper bound on $|S^\ell_i|$.

We just proved that, when Greedy terminates, the seed-regret for advertiser a_i, upon termination, is at most $\lambda \cdot (1 + s^i_{opt} \cdot \lceil \ln\{1/(p_i/2 - \lambda/2B_i)\}\rceil)$. Combining all the claims above, we can infer that the overall regret of Greedy upon termination is at most $\sum_{i=1}^{h}(p_i B_i + \lambda)/2 + \lambda \sum_{i=1}^{h}[1 + s^i_{opt}(1 + \lceil \ln\{1/(p_i/2 - \lambda/2B_i)\}\rceil$. □

Discussion: The term $\delta(u, i) \cdot cpe(i)$ corresponds to the expected revenue from user u clicking on i (without considering the network effect). Thus, the assumption on λ, that it is no more than the expected revenue from any one user clicking on an ad, keeps the penalty term small, since in practice click-through probabilities tend to be small. Secondly, the regret bound given by the theorem can be understood as follows. Upon termination, the budget-regret from Greedy's allocation is at most $(1/2)B \cdot p_{max}$ (plus a small constant $\lambda/2$). The theorem says that Greedy achieves such a budget-regret while being frugal w.r.t. the number of seeds it uses. Indeed, its seed-regret is bounded by the minimum number of seeds that an optimal algorithm would use to reach the budget, multiplied by a logarithmic factor.

4.3.3 The Case of No Penalty

In this subsection, we focus on the regret bound achieved by Greedy in the special case that penalty term $\lambda = 0$, i.e., the overall regret is just the budget-regret. While the results here can be more or less seen as special cases of Theorem 4.2, it is illuminating to restrict attention to this special case. Our first result follows.

Theorem 4.3. *Consider an instance of* REGRET-MINIMIZATION *that admits a seed allocation whose total regret is bounded by a third of the total budget. Then Algorithm 4 outputs an allocation $\mathcal{S}$ with a total regret $\mathcal{R}(\mathcal{S}) \leq \frac{1}{3} \cdot B$, where $B = \sum_{i=1}^{h} B_i$ is the total budget.*

Proof. Consider an arbitrary iteration of Algorithm 4, where the algorithm assigns a node, say u, to advertiser a_i, i.e., it adds u to the seed set S_i. In particular, notice that u has been assigned to $< \kappa_u$ advertisers before this iteration, where κ_u is the attention bound of u. Three cases arise as shown in Figure 4.2.

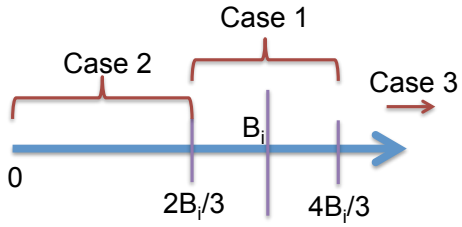


Figure 4.2: Interpretation of Theorem 4.3.

Case 1: $\frac{2}{3}B_i \leq \Pi_i(S_i \cup \{u\}) \leq \frac{4}{3}B_i$. In this case, clearly, the regret for this advertiser is $|\Pi_i(S_i \cup \{u\}) - B_i| \leq \min\{\frac{4}{3}B_i - B_i, B_i - \frac{2}{3}B_i\} \leq \frac{1}{3}B_i$.

Case 2: $\Pi_i(S_i \cup \{u\}) < \frac{2}{3}B_i$. Consider the next iteration in which another seed, say u', is assigned to the same advertiser a_i, i.e., u' is to S_i. Clearly, the marginal gain of u' w.r.t. $S_i \cup \{u\}$ cannot be more than $\frac{2}{3}B_i$, by submodularity. Thus, $\Pi_i(S_i \cup \{u, u'\}) < \frac{4}{3}B_i$. Now, if $\Pi_i(S_i \cup \{u, u'\}) \geq \frac{2}{3}B_i$, then by Case 1, we have that the regret of advertiser a_i is at most $\frac{1}{3}B_i$. Otherwise, $\Pi_i(S_i \cup \{u, u'\}) < \frac{2}{3}B_i$, and then it is similar to Case 2 condition, where u' is also added to S_i after u. In this case, subsequent iterations of the algorithm grow S_i until Case 1 is satisfied. A simple inductive argument shows that the regret for advertiser a_i is no more that $\frac{1}{3}B_i$.

Case 3: $\Pi_i(S_i \cup \{u\}) > \frac{4}{3}B_i$. The algorithm adds u to S_i only when $\Pi_i(S_i \cup \{u\}) - B_i < B_i - \Pi_i(S_i)$, which implies $\Pi_i(S_i \cup \{u\}) + \Pi_i(S_i) < 2B_i$.[3] However, since $\Pi_i(S_i \cup \{u\}) > \frac{4}{3}B_i$, this implies $\Pi_i(S_i) < \frac{2}{3}B_i$. This means the marginal gain of u w.r.t. S_i, i.e., $\Pi_i(S_i \cup \{u\}) - \Pi_i(S_i)$, is larger than $\frac{2}{3}B_i$. However, $\Pi_i(S_i) < \frac{2}{3}B_i$, which by submodularity, implies no subsequent seed can have a marginal gain of $\frac{2}{3}B_i$ or more, a contradiction. Thus, Case 3 is impossible.

We just showed that for any advertiser, the regret achieved by the algorithm is at most $\frac{1}{3}B_i$. Summing over all advertisers, we see that the overall regret is no more than $\frac{1}{3}B$. □

The regret bound established above is conservative, and unlike Theorem 4.2, does not make any assumptions about the marginal gains of seed nodes. In practice, as previously noted, most real networks tend to have low influence probabilities and consequently, the marginal gain of any single node tends to be a small fraction of the budget. Using this, we can establish a tighter bound on the regret achieved by Greedy.

[3] Since the algorithm makes the choice with lesser regret, we can assume w.l.o.g. that it adds u only when the addition will result in strictly lower regret than not adding it.

Theorem 4.4. *On any input instance that admits an allocation with total regret bounded by* $\min\{\frac{p_{max}}{2}, 1-p_{max}\} \cdot B$*, Algorithm 4 delivers an allocation* $\mathcal{S}$ *so that* $\mathcal{R}(\mathcal{S}) \leq \min\{\frac{p_{max}}{2}, 1-p_{max}\} \cdot B$.

Proof. The proof is similar to the proof of Theorem 4.3. Consider an arbitrary iteration of Algorithm 4. Suppose u is the seed that the algorithm assigned to a_i (i.e., added to seed set S_i) in this iteration. The following two cases arise.

Case 1: $\Pi_i(S_i \cup \{u\}) < p_i \cdot B_i$. Then, the algorithm will continue to add seeds to the seed set S_i, until the condition of Case 2 is met.

Case 2: $\Pi_i(S_i \cup \{u\}) \geq p_i \cdot B_i$. There can be two sub-cases in this scenario:

Case 2a: $\Pi_i(S_i \cup \{u\}) \leq B_i$. Clearly, regret is

$$B_i - \Pi_i(S_i \cup \{u\}) \leq B_i - p_i \cdot B_i = (1-p_i)B_i.$$

If u is the last seed added to the seed set S_i, then we have regret $\leq (1-p_i)B_i$. Moreover, u being the last seed also implies that for any other node $u' \notin S_i$, we have

$$B_i - \Pi_i(S_i \cup \{u\}) \leq \Pi_i(S_i \cup \{u, u'\}) - B_i,$$

since otherwise, the algorithm would have added u' to S_i to decrease the regret. Also, due to submodularity,

$$\begin{aligned}
& \Pi_i(S_i \cup \{u, u'\}) - \Pi_i(S_i \cup \{u\}) \leq p_i \cdot B_i, \\
\implies \quad & \Pi_i(S_i \cup \{u, u'\}) - B_i + B_i - \Pi_i(S_i \cup \{u\}) \leq p_i \cdot B_i, \\
\implies \quad & 2 \cdot (B_i - \Pi_i(S_i \cup \{u\})) \leq p_i \cdot B_i, \\
\implies \quad & B_i - \Pi_i(S_i \cup \{u\}) \leq \frac{p_i}{2} \cdot B_i.
\end{aligned}$$

Therefore, in Case 2a, if u is the last seed selected by the algorithm, then regret of advertiser a_i is $\min\{\frac{p_i}{2}, 1-p_i\} \cdot B_i$. Otherwise, the algorithm would continue with the next iteration and add seeds until Case 2a or Case 2b is satisfied.

Case 2b: $\Pi_i(S_i \cup \{u\}) > B_i$. Then regret for advertiser a_i is $\Pi_i(S_i \cup \{u\}) - B_i$.

In this case, u must be the last seed selected by the algorithm as adding another seed can only increase the regret. Therefore, it is clear that

$$\Pi_i(S_i \cup \{u\}) - B_i \leq B_i - \Pi_i(S_i).$$

Moreover, due to submodularity, we know that

$$\begin{aligned} & \Pi_i(S_i \cup \{u\}) - \Pi_i(S_i) \leq p_i \cdot B_i, \\ \implies \quad & \Pi_i(S_i \cup \{u\}) - B_i + B_i - \Pi_i(S_i) \leq p_i \cdot B_i, \\ \implies \quad & 2 \cdot (\Pi_i(S_i \cup \{u\}) - B_i) \leq p_i \cdot B_i, \\ \implies \quad & \Pi_i(S_i \cup \{u\}) - B_i \leq \frac{p_i}{2} \cdot B_i. \end{aligned}$$

Combining Cases 2a and 2b, and summing it over all advertisers, it is easy to see that total regret is $\leq \min(\frac{p_{max}}{2}, (1 - p_{max})) \cdot B$. □

We note that this claim generalizes Theorem 4.3. In fact, the two bounds: $\frac{p_{max}}{2}$ and $1 - p_{max}$ meet at the value of $1/3$ when $p_{max} = 2/3$. In practice, p_{max} may be much smaller, making the bound better.

4.4 Scalable Algorithms

Algorithm 4 (Greedy) involves a large number of calls to influence spread computations, to find the node for each advertiser a_i that yields the maximum decrease in regret $\mathcal{R}_i(S_i)$. Given any seed set S, computing its *exact* influence spread $\sigma(S)$ under the IC model is #P-hard [37], and this hardness trivially carries over to the topic-aware IC model [13] with CTPs. A common practice is to use Monte Carlo (MC) simulations to estimate influence spread [88]. However, accurate estimation requires a large number of MC simulations, which is prohibitively expensive and not scalable. Thus, to make Algorithm 4 scalable, we need an alternative approach.

In the influence maximization literature, considerable effort has been devoted to developing more efficient and scalable algorithms [22,37,42,86,132]. Of these, the IRIE algorithm proposed by Jung et al. [86] is a state-of-the-art heuristic for influence maximization under the IC model and is orders of magnitude faster than MC simulations. We thus use a variant of Greedy, GREEDY-IRIE, where IRIE replaces MC simulations for spread estimation. It is one of the strong baselines we will compare our main algorithm with in Section 4.5. In this section, we instead propose a scalable algorithm with guaranteed approximation for influence spread.

Recently, Borgs et al. [22] proposed a quasi-linear time randomized algorithm based on the idea of sampling *"reverse-reachable"* (RR) sets in the graph. It was improved to a near-linear time randomized algorithm – *Two-phase Influence Maximization (TIM)* – by Tang et al. [132]. Cohen et al. [42] proposed a sketch-based

design for fast computation of influence spread, achieving efficiency and effectiveness comparable to TIM. We choose to extend TIM as it is the current state-of-the-art influence maximization algorithm and is more adapted to our needs.

In this section, we adapt the essential ideas from Greedy, RR-sets sampling, and the TIM algorithm to devise an algorithm for REGRET-MINIMIZATION, called Two-phase Iterative Regret Minimization (TIRM for short), that is much more efficient and scalable than Algorithm 4 with MC simulations. Our adaptation to TIM is non-trivial, since TIM relies on knowing the exact number of seeds required. In our framework, the number of seeds needed is driven by the budget and the current regret and so is dynamic. We first give the background on RR-sets sampling, review the TIM algorithm [132], and then describe our TIRM algorithm.

4.4.1 Reverse-Reachable Sets and TIM

RR-sets Sampling: Brief Review. We first review the definition of RR-sets, which is the backbone of both TIM and our proposed TIRM algorithm. Conceptually speaking, a random RR-set R from G is generated as follows. First, for every edge $(u, v) \in E$, remove it from G w.p. $1 - p_{u,v}$: this generates a possible world X. Second, pick a *target* node w uniformly at random from V. Then, R consists of the nodes that can reach w in X. This can be implemented efficiently by first choosing a target node $w \in V$ uniformly at random and performing a breadth-first search (BFS) starting from it. Initially, create an empty BFS-queue Q, and insert all of w's in-neighbors into Q. The following loop is executed until Q is empty: Dequeue a node u from Q and examine its *incoming* edges: for each edge (v, u) where $v \in N^{in}(u)$, we insert v into Q w.p. $p_{v,u}$. All nodes dequeued from Q thus form a RR-set.

The intuition behind RR-sets sampling is that, if we have sampled sufficiently many RR-sets, and a node u appears in a large number of RR sets, then u is likely to have high influence spread in the original graph and is a good candidate seed.

TIM: Brief Review. Given an input graph $G = (V, E)$ with influence probabilities and desired seed set size s, TIM, in its first phase, computes a lower bound on the optimal influence spread of any seed set of size s, i.e., $OPT_s := \max_{S \subseteq V, |S|=s} \sigma^{ic}(S)$. Here $\sigma^{ic}(S)$ refers to the spread w.r.t. classic IC model. TIM then uses this lower bound to estimate the number of random RR-sets that need to be generated, denoted θ. In its second phase, TIM simply samples θ RR-sets, denoted $\mathbf{R}$, and uses them to select s seeds, by solving the Max s-Cover problem: find s nodes, that between them, appear in the maximum number of sets in $\mathbf{R}$. This is solved using a well-known greedy procedure: start with an empty set and repeatedly add a node that appears in the maximum number of sets in $\mathbf{R}$ that are not yet "covered".

TIM provides a $(1-1/e-\epsilon)$-approximation to the optimal solution OPT_s with high probability. Also, its time complexity is $O((s+\ell)(|V|+|E|)\log|V|/\epsilon^2)$, while that of the greedy algorithm (for influence maximization) is $\Omega(k|V||E| \cdot \mathrm{poly}(\epsilon^{-1}))$.

Theoretical Guarantees of TIM. Consider any collection of random RR-sets, denoted $\mathbf{R}$. Given any seed set S, we define $F_{\mathbf{R}}(S)$ as the fraction of $\mathbf{R}$ covered by S, where S covers an RR-set iff it overlaps it. The following proposition says that for any S, $|V| \cdot F_{\mathbf{R}}(S)$ is an unbiased estimator of $\sigma^{ic}(S)$.

Proposition 4.1 (Corollary 1, [132])**.** *Let $S \subseteq V$ be any set of nodes, and $\mathbf{R}$ be a collection of random RR sets. Then, $\sigma^{ic}(S) = \mathrm{E}[|V| \cdot F_{\mathbf{R}}(S)]$.*

The next proposition shows the accuracy of influence spread estimation and the approximation gurantee of TIM. Given any seed set size s and $\varepsilon > 0$, define $L(s,\varepsilon)$ to be:

$$L(s,\varepsilon) = (8+2\varepsilon)n \cdot \frac{\ell \log n + \log\binom{n}{s} + \log 2}{OPT_s \cdot \varepsilon^2}, \tag{4.5}$$

where $\ell > 0, \epsilon > 0$.

Proposition 4.2 (Lemma 3 & Theorem 1, [132])**.** *Let θ be a number no less than $L(s,\varepsilon)$. Then for any seed set S with $|S| \leq s$, the following inequality holds w.p. at least $1 - n^{-\ell}/\binom{n}{s}$:*

$$||V| \cdot F_{\mathbf{R}}(S) - \sigma^{ic}(S)| < \frac{\varepsilon}{2} \cdot OPT_s. \tag{4.6}$$

Moreover, with this θ, TIM returns a $(1-1/e-\epsilon)$-approximation to OPT_s w.p. $1-n^{-\ell}$.

This result intuitively says that as long as we sample enough RR-sets, i.e., $|\mathbf{R}| \geq \theta$, the absolute error of using $|V| \cdot F_{\mathbf{R}}(S)$ to estimate $\sigma^{ic}(S)$ is bounded by a fraction of OPT_s with high probability. Furthermore, this gives approximation guarantees for influence maximization. Next, we describe how to extend the ideas of RR-sets sampling and TIM for regret minimization.

4.4.2 Two-phase Iterative Regret Minimization

A straightforward application of TIM for solving REGRET-MINIMIZATION will not work. There are two critical challenges. First, TIM requires the number of

seeds s as input, while the input of REGRET-MINIMIZATION is in the form of monetary budgets, and thus we do not know the precise number of seeds that should be allocated to each advertiser beforehand. Second, our influence propagation model has click-through probabilities (CTPs) of seeds, namely $\delta(u, i)$'s. This is not accounted for in the RR-sets sampling method: it implicitly assumes that each seed becomes active w.p. 1.

We first discuss how to adapt RR-sets sampling to incorporate CTPs. Then we deal with unknown seed set sizes.

RR-sets Sampling with Click-Through Probabilities. Recall that in our model, when a node u is chosen as a seed for advertiser a_i, it has a probability $\delta(u, i)$ to accept being seeded, i.e., to actually click on the ad. For ease of exposition, in the rest of this subsubsection only, we assume that there is only one advertiser, and the CTP of each user u for this advertiser is simply $\delta(u) \in [0, 1]$. The technique we discuss and our results readily extend to any number of advertisers.

For clarity, we call the RR-sets generated with CTPs incorporated ***RRC-sets*** to distinguish them from normal RR-sets, which have no associated CTPs. The procedure for generating a random RRC-set is similar to that for generating a normal (random) RR-set. First, a root w is chosen uniformly at random from V. Let R_w denote the associated RRC-set being generated. Then, we enqueue w into a FIFO queue Q.

Until Q is empty, we repeat the following: dequeue the next node from Q, and let it be u. For all of its in-neighbors $v \in N^{in}(u)$, we first test the edge (v, u): it is live w.p. $p_{v,u}$, and blocked w.p. $1 - p_{v,u}$. If the edge is blocked, we ignore it and continue to the next in-neighbor, if any. If the edge is live, we further flip a biased coin, independently, for the node v itself: w.p. $\delta(v)$, we declare v live, and w.p. $1 - \delta(v)$, declare v blocked. The following two cases arise: (i). If v is live, then it can be a valid seed, and thus we add v to R_w as well as enqueue v into Q. (ii). If v is blocked, then it cannot be a valid seed itself, but it should still be added to Q, since its in-neighbors may still be valid seeds, depending on their own edge- and node-based coin flips.

Note that for the root w itself, the node test should also be performed using its CTP: w.p. $\delta(w)$, w is added to R_w. Again, even if this CTP test fails, which occurs w.p. $1 - \delta(w)$, the above procedure is still correct in terms of first enqueuing w into Q, since w's in-neighbors can be valid seeds to activate w.

Let $\mathbf{Q}$ be a collection of RRC-sets. Similar to $F_{\mathbf{R}}(S)$, for any set S, we define $F_{\mathbf{Q}}(S)$ to be the fraction of $\mathbf{Q}$ that overlap with S. Let $\sigma^{icctp}(S)$ be the influence spread of a seed set S under the IC model with CTPs. We first establish a similar result to Proposition 4.1 which says that $|V|F_{\mathbf{Q}}(S)$ is an unbiased estimator of $\sigma(S)$.

Lemma 4.2. *Given a graph $G = (V, E)$ with influence probabilities on edges, for any $S \subseteq V$, $\sigma^{icctp}(S) = \mathrm{E}[|V| \cdot F_{\mathbf{Q}}(S)]$.*

Proof. We show the following equality holds:

$$\sigma^{icctp}(S)/|V| = \mathrm{E}[F_{\mathbf{Q}}(S)]. \tag{4.7}$$

The LHS of (4.7) equals the probability that a node chosen uniformly at random can be activated by seed set S where a seed $u \in S$ may become live with CTP $\delta(u)$, while the RHS of (4.7) equals the probability that S intersects with a random RRC-set. They both equal the probability that a randomly chosen node is reachable by S in a possible world corresponding to the IC-CTP model. □

In principle, RRC-sets are those we should work with for the purpose of seed selection for REGRET-MINIMIZATION. However, note that by Equation (4.5) and Proposition 4.2, the number of samples required is inversely proportional to the value of the optimal solution OPT_s. However, in reality, click-through rates on ads are quite low, and thus OPT_s, taking CTPs into account, will decrease by at least two orders of magnitude (e.g., OPT_s with CTP 0.01 would become 100 times smaller than OPT_s with CTP 1). This in turn translates into at least two orders of magnitude more RRC-sets to be sampled, which ruins scalability.

An alternative way of incorporating CTPs is to pretend as though all CTPs were 1. We still generate RR-sets, and use the estimations given by RR-sets to compute revenue. More specifically, for any $S \subseteq V$ and any $u \in V \setminus S$, we compute the marginal gain of u w.r.t. S, namely $\sigma_C(S \cup \{u\}) - \sigma_C(S)$, by $\delta(u) \cdot |V| \cdot [F_{\mathbf{R}}(S \cup \{u\}) - F_{\mathbf{R}}(S)]$. This avoids sampling of numerous RRC-sets.

We can show that in expectation, computing marginal gain in IC-CTP model using RRC-sets is essentially equivalent to computing it under the IC model using RR-sets in the manner above.

Theorem 4.5. *Consider any $u \in S$ and any $S \subseteq V$. Let $\delta(u)$ be the probability that u accepts to become a seed. Let $\mathbf{R}$ and $\mathbf{Q}$ be a collection of RR-sets and of RRC-sets, respectively. Then,*

$$\delta(u)(\mathrm{E}[F_{\mathbf{R}}(S \cup \{u\})] - \mathrm{E}[F_{\mathbf{R}}(S)]) = \mathrm{E}[F_{\mathbf{Q}}(S \cup \{u\})] - \mathrm{E}[F_{\mathbf{Q}}(S)].$$

Proof. Consider a random RR-set X, and define an indicator function $\mathbb{I}_X(u, S)$,

which takes on 1 *if* $u \in X$ *and* $S \cap X = \emptyset$, and 0 *otherwise*. Then, we have:

$$\begin{aligned}
&\mathrm{E}[F_{\mathbf{R}}(S \cup \{u\})] - \mathrm{E}[F_{\mathbf{R}}(S)] \\
&= \sum_{X} \Pr[X] \cdot \mathbb{I}_X(u, S) = \sum_{X\,:\,\mathbb{I}_X(u,S)=1} \Pr[X],
\end{aligned} \tag{4.8}$$

where $\Pr[X]$ is the probability of sampling the RR-set X.

Note that the only difference between the generation of an RR-set and that of an RRC-set is the additional coin flips on nodes, with CTPs, which are all independent. Now, consider a fixed RR-set X that does contain u. If we were to generate an RRC-set — meaning that the outcomes of all edge-level coin flips would remain the same — then X may contain u w.p. $\delta(u)$. This is true since all edge- and node-level coin flips are independent. If u belongs to the RRC-set realization of X, we denote it by X_u.

Now, for the expected marginal gain of u under the model with CTPs, we have:

$$\begin{aligned}
&\mathrm{E}[F_{\mathbf{Q}}(S \cup \{u\})] - \mathrm{E}[F_{\mathbf{Q}}(S)] \\
&= \sum_{X_u} \Pr[X_u] = \sum_{X:\mathbb{I}_X(u,S)=1} \delta(u) \cdot \Pr[X] \\
&= \delta(u) \cdot (\mathrm{E}[F_{\mathbf{R}}(S \cup \{u\}) - \mathrm{E}[F_{\mathbf{R}}(S)]),
\end{aligned}$$

where we have applied (4.8) in the last equality. This completes the proof. □

This theorem shows even with CTPs, we can still use the usual RR-sets sampling process for estimating spread efficiently and accurately as long as we multiply marginal gains by CTPs. *This result carries over to the setting of multiple advertisers.*

Iterative Seed Set Size Estimation. As mentioned earlier, TIM needs the required number of seeds s as input, which is not available for the REGRET-MINIMIZATION problem. From the advertiser budgets, there is no obvious way to determine the number of seeds. This poses a challenge since the required number of RR-sets (θ) depends on s. To circumvent this difficulty, we propose a framework which first makes an initial guess at s, and then iteratively revises the estimated value, until no more seeds are needed, while concurrently selecting seeds and allocating them to advertisers.

For ease of exposition, let us first consider a single advertiser a_i. Let B_i be the budget of a_i and let s_i be the true number of seeds required to minimize the regret

Algorithm 5: TIRM

Input : $G = (V, E)$; attention bounds $\kappa_u, \forall u \in V$; items $\vec{\gamma}_i$ with $cpe(i)$ & budget $B_i, i = 1, \ldots, h$; CTPs $\delta(u, i), \forall u \forall i$

Output: $S_1, \cdots, S_h$

foreach $j = 1, 2, \ldots, h$ **do**
  $S_j \leftarrow \emptyset$; $Q_j \leftarrow \emptyset$; `// a priority queue`
  $s_j \leftarrow 1$; $\theta_j \leftarrow L(s_j, \varepsilon)$; $\mathbf{R}_j \leftarrow \mathsf{Sample}(G, \vec{\gamma}_j, \theta_j)$;
while *true* **do**
  foreach $j = 1, 2, \ldots, h$ **do**
    $(v_j, cov_j(v_j)) \leftarrow \mathsf{SelectBestNode}(\mathbf{R}_j)$; `// Algo 6`
    $F_{\mathbf{R}_j}(v_j) \leftarrow cov_j(v_j)/\theta_j$;
  $i \leftarrow \text{argmax}_{j=1}^{h} \mathcal{R}_j(S_j) - \mathcal{R}_j(S_j \cup \{v_j\})$ subject to: $\mathcal{R}_j(S_j \cup \{v_j\}) < \mathcal{R}_j(S_j)$; //`find the (user, ad) pair with max drop in regret.`
  if $i \neq$ **NULL then**
    $S_i \leftarrow S_i \cup \{v_i\}$;
    $Q_i.\mathsf{insert}(v_i, cov_i(v_i))$;
    $\mathbf{R}_i \leftarrow \mathbf{R}_i \setminus \{R \mid v_i \in R \wedge R \in \mathbf{R}_i\}$;
  //`remove RR-sets that are covered`;
  else return ;
  if $|S_i| = s_i$ **then**
    $s_i \leftarrow s_i + \lfloor \mathcal{R}_i(S_i)/(cpe(i) \cdot n \cdot \delta(v_i, i) \cdot F_{\mathbf{R}_i}(v_i)) \rfloor$;
    $\theta_i \leftarrow \max\{L(s_i, \varepsilon), \theta_i\}$;
    $\mathbf{R}_i \leftarrow \mathbf{R}_i \cup \mathsf{Sample}(G, \vec{\gamma}_i, \max\{0, L(s_i, \varepsilon) - \theta_i)\}$;
    $\Pi_i(S_i) \leftarrow \mathsf{UpdateEstimates}(\mathbf{R}_i, \theta_i, S_i, Q_i)$; //`revise estimates to reflect newly added RR-sets`;
    $\mathcal{R}_i(S_i) \leftarrow |B_i - \Pi_i(S_i)|$;

for a_i. We do not know s_i and estimate it in successive iterations as $\tilde{s}_i^t$. We start with an estimated value for s_i, denoted $\tilde{s}_i{}^1$, and use it to obtain a corresponding θ_i^1 (*cf.* Proposition 4.2). If $\theta_i^t > \theta_i^{t-1}$,[4] we will need to sample an additional $(\theta_i^t - \theta_i^{t-1})$ RR-sets, and use all RR-sets sampled up to this iteration to select $(\tilde{s}_i^t - \tilde{s}_i^{t-1})$ additional seeds. After adding those seeds, if a_i's budget B_i is not yet reached, this means more seeds can be assigned to a_i. Thus, we will need another iteration and we further revise our estimation of s_i. The new value, $\tilde{s}_i^{t+1}$, is obtained by adding to $\tilde{s}_i^t$ the floor function of the ratio between the current regret $\mathcal{R}_i(S_i)$ and the marginal revenue contributed by the $\tilde{s}_i^t$-th seed (i.e., the latest seed). This ensures we do not overestimate, thanks to submodularity, as future seeds have diminishing marginal gains.

[4] Assuming $\theta_i^0 = 0, i = 1, \ldots, h$.

Algorithm 6: SelectBestNode($\mathbf{R}_j$)

Output: $(u, cov_j(u))$

1 $u \leftarrow \text{argmax}_{v \in V} |\{R \mid v \in R \wedge R \in \mathbf{R}_j\}|$ subject to: $|\{S_l | v \in S_l\}| < \kappa_v$;

2 $cov_j(u) \leftarrow |\{R \mid u \in R \wedge R \in \mathbf{R}_j\}|$; //`find best seed for ad` a_j `as well as its coverage.`

Algorithm 7: UpdateEstimates($\mathbf{R}_i$, θ_i, S_i, Q_i)

Output: $\Pi_i(S_i)$

1 $\Pi_i(S_i) \leftarrow 0$;

2 **for** $j = 0, \ldots, |S_i| - 1$ **do**

3 $(v, cov(v)) \leftarrow Q_i[j]$;

4 $cov'(v) \leftarrow |\{R \mid v \in R, R \in \mathbf{R}_i\}|$;

5 Q_i.insert$(v, cov(v) + cov'(v))$;

6 $\Pi_i(S_i) \leftarrow \Pi_i(S_i) + cpe(i) \cdot n \cdot \delta(v, i) \cdot ((cov(v) + cov'(v))/\theta_i)$;

`//update coverage of existing seeds w.r.t. new RR-sets added to collection.`

Algorithm 5 outlines TIRM, which integrates the iterated seed set size estimation technique above, suitably adapted to multi-advertiser setting, along with the RR-set based coverage estimation idea of TIM, and uses Theorem 4.5 to deal with CTPs. Notice that the core logic of the algorithm is still based on greedy seed selection as outlined in Algorithm 4. Algorithm TIRM works as follows. For every advertiser a_i, we initially set its seed budget s_i to be 1 (a conservative, but safe estimate), and find the first seed using random RR-sets generated accordingly (line 3). In the main loop, we follow the greedy selection logic of Algorithm 4. That is, every time, we identify the valid user-advertiser pair (u, a_i) that gives the *largest decrease in total regret* and allocate u to S_i (lines 6 to 12), paying attention to the attention bound of u (line 1 of Algorithm 6). If $|S_i|$ reaches the current estimate of s_i after we add u, then we increase s_i by $\lfloor \mathcal{R}_i(S_i)/(cpe(i) \cdot n \cdot F_{\mathbf{R}_i}(u)) \rfloor$ (lines 15 to 20), as described above, as long as the regret continues to decrease. Note that after adding additional RR-sets, we should update the spread estimation of current seeds w.r.t. the new collection of RR-sets (line 19). This ensures that future marginal gain computations and selections are accurate. This is effectively a *lower bound* on the number of additional seeds needed, as subsequent seeds will not have marginal gain higher than that of u due to submodularity. As in Algorithm 4, TIRM terminates when all advertisers have saturated, i.e., no additional seed can bring down the regret. Note that in Algorithm 7, we update the estimated revenue (coverage) of existing seeds w.r.t. the additional RR-sets sampled, to keep them accurate.

Estimation Accuracy of TIRM. At its core, TIRM, like TIM, estimates the spread of chosen seed sets, even though its objective is to minimize regret w.r.t. a monetary budget. Next, we show that the influence spread of seeds estimated by TIRM enjoys bounded error guarantees similar to those chosen by TIM (see Proposition 4.2).

Theorem 4.6. *At any iteration t of iterative seed set size estimation in Algorithm* TIRM*, for any set S_i of at most $s = \sum_{j=1}^{t} s^j$ nodes, $|n \cdot F_{\mathbf{R}^t}(S_i) - \sigma_i(S_i)| < \frac{\varepsilon}{2} \cdot OPT_s$ holds with probability at least $1 - n^{-\ell}/\binom{n}{s}$, where $\sigma_i(S)$ is the expected spread of seed set S_i for ad i.*

Proof. When $t = 1$, our claim follows directly from Proposition 4.2. When $t > 1$, by definition of our iterative sampling process, the number of RR-sets, $|\mathbf{R}^t|$, is equal to $\max_{j=1,\ldots,t} L_j$, where $L_j = L\left(\sum_{a=1}^{j} s^a, \varepsilon\right)$. This means that at any iteration t, the number of RR-sets is always sufficient for Eq. (4.6) to hold. Hence, for the set S_i containing seeds accumulated up to iteration t, our claim on the absolute error in the estimated spread of S_i holds, by virtue of Proposition 4.2. □

4.5 Experiments

In this section, we conduct an empirical evaluation of the proposed algorithms [5]. The goal is manifold. First, we would like to evaluate the quality of the algorithms as measured by the regret achieved, the number of seeds they used to achieve a certain level of budget-regret, and the extent to which the attention bound (κ) and the penalty factor (λ) affect their performance. Second, we evaluate the efficiency and scalability of the algorithms w.r.t. advertiser budgets, which indirectly control the number of seeds required, and w.r.t. the number of advertisers. We measure both running time and memory usage.

Datasets. Our experiments are based on four real-world social networks, whose basic statistics are summarized in Table 4.1. Of the four datasets, we use FLIXSTER and EPINIONS for our quality experiments, and DBLP and LIVE-JOURNAL for scalability experiments. FLIXSTER is from a social movie-rating site (`http://www.flixster.com/`). The dataset records movie ratings from users along with their timestamps. We use the topic-aware influence probabilities and the item-specific topic distributions provided by the authors of [13], who learned the probabilities using maximum likelihood estimation for the TIC model with $K = 10$ latent topics. In our quality experiments, we set the number of advertisers h to be 10, and used 10 of the learnt topic distributions from Flixster

[5] The software is available from `https://github.com/aslayci/TIRM`

	FLIXSTER	EPINIONS	DBLP	LIVEJOURNAL
#nodes	30K	76K	317K	4.8M
#edges	425K	509K	1.05M	69M
type	directed	directed	undirected	directed

Table 4.1: Statistics of network datasets.

	Budgets			CPEs		
Dataset	mean	max	min	mean	max	min
FLIXSTER	375	200	600	5.5	5	6
EPINIONS	215	100	350	4.35	2.5	6

Table 4.2: Advertiser budgets and cost-per-engagement values.

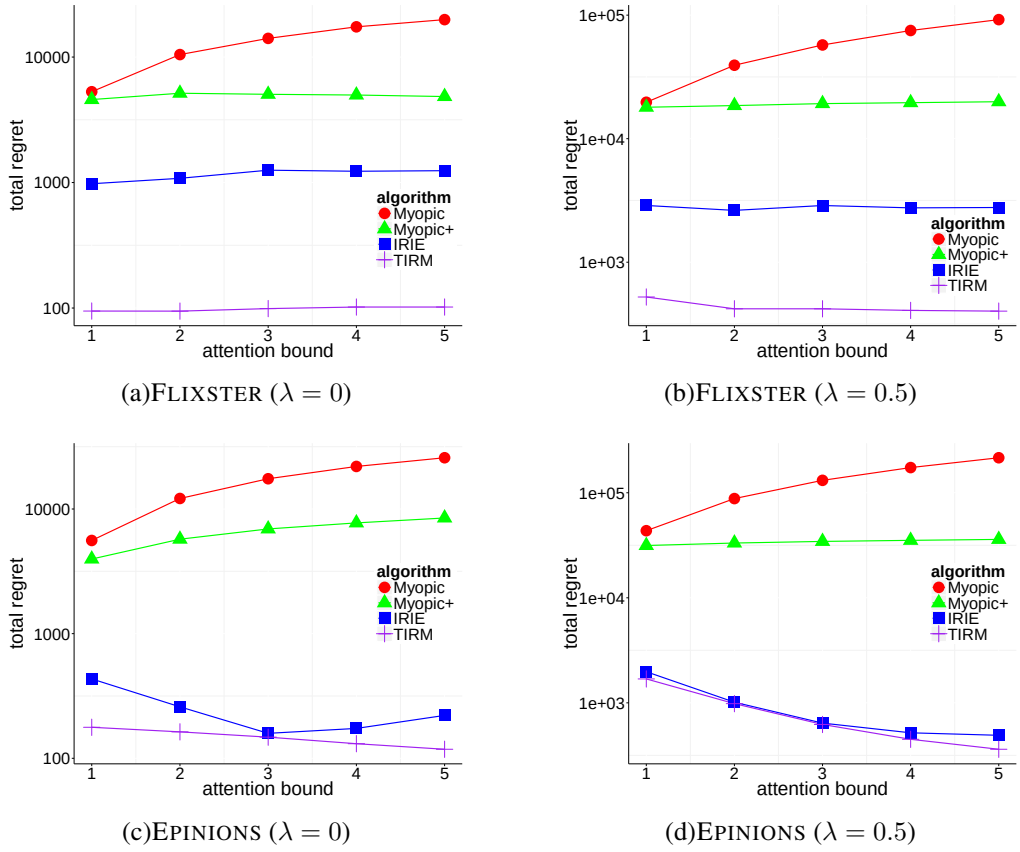
dataset, where for each ad i , its topic distribution $\vec{\gamma}_i$ has mass 0.91 in the i-th topic, and 0.01 in all others. CTPs are sampled uniformly at random from the interval $[0.01, 0.03]$ for all user-ad pairs, in keeping with real-life CTPs (see Section 4.1).

EPINIONS is a who-trusts-whom network taken from a consumer review website (`http://www.epinions.com/`). For Epinions, we similarly set $h = 10$ and use $K = 10$ latent topics. For each ad i, we use synthetic topic distributions $\vec{\gamma}_i$, by borrowing the ones used in FLIXSTER. For all edges and topics, the topic-aware influence probabilities are sampled from an exponential distribution with mean 30, via the inverse transform technique [53] on the values sampled randomly from uniform distribution $\mathcal{U}(0, 1)$.

For scalability experiments, we adopt two large networks, DBLP and LIVEJOURNAL (both are available at `http://snap.stanford.edu/`). DBLP is a co-authorship graph (undirected), where nodes represent authors, and there is an edge between two nodes if they have co-authored a paper indexed by DBLP. We direct all edges in both directions. LIVEJOURNAL is an online blogging site where users can declare which other users are their friends.

In all datasets, advertiser budgets and CPEs are chosen in such a way that the total number of seeds required for all ads to meet their budgets is less than n. This ensures no ads are assigned empty seed sets.Table 4.2 contains a statistical summary of the budgets and CPEs. Notice that since the CTPs are in the 1-3% range, the effective number of targeted nodes is correspondingly larger. We defer the numbers for DBLP and LIVEJOURNAL to Section 4.5.2.

All experiments were run on a 64-bit RedHat Linux server with Intel Xeon 2.40GHz CPU and 65GB memory. Our largest configuration is LIVEJOURNAL with 20 ads, which effectively has $69M \cdot 20 = 1.4B$ edges; this is comparable with [132], whose largest dataset has 1.5B edges (Twitter).



(a)FLIXSTER ($\lambda = 0$) (b)FLIXSTER ($\lambda = 0.5$)

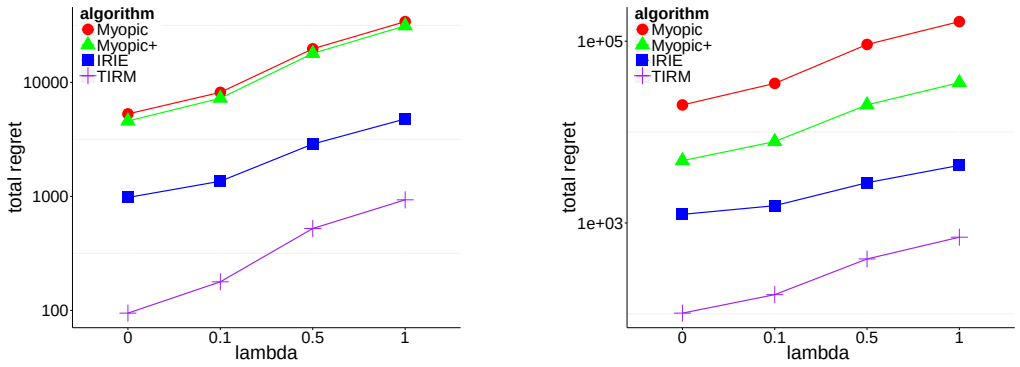
(c)EPINIONS ($\lambda = 0$) (d)EPINIONS ($\lambda = 0.5$)

Figure 4.3: Total regret (log-scale) vs. attention bound κ_u.

Algorithms. We test and compare the following four algorithms.

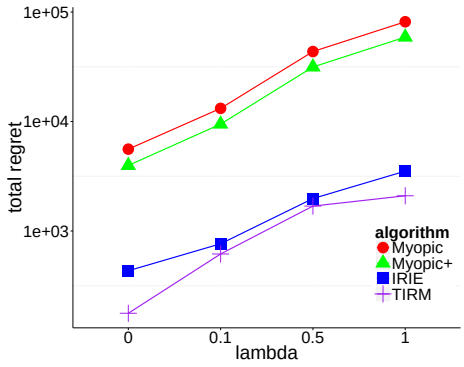
- MYOPIC: A baseline that assigns every user $u \in V$ in total κ_u most relevant ads i, i.e., those for which u has the highest expected revenue, not considering any network effect, i.e., $\delta(u, i) \cdot cpe(i)$. This baseline is called "myopic" as it solely focuses on CTPs and CPEs, and effectively ignores virality and budgets. Allocation $\mathcal{A}$ in Figure 4.1 follows this baseline.
- MYOPIC+: This is an enhanced version of MYOPIC which takes budgets, but not virality, into account. For each ad, it first ranks users w.r.t. CTPs, and then selects seeds using this order until budget is exhausted. User attention bounds are taken into account by going through the ads round-robin, and advancing to the next seed if the current node u is already assigned to κ_u ads.
- GREEDY-IRIE: An instantiation of Algorithm 4, with the IRIE heuristic [86] used for influence spread estimation and seed selection.

  IRIE has a damping factor α for accurately estimating influence spread in its

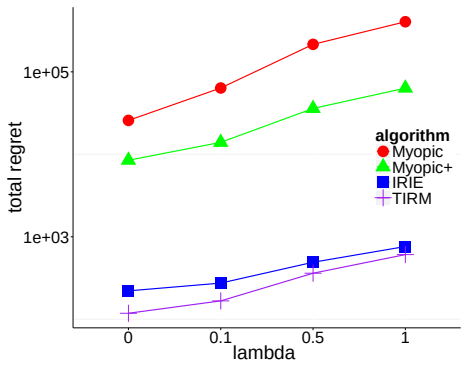


(a)FLIXSTER ($\kappa_u = 1$)

(b)FLIXSTER ($\kappa_u = 5$)



(c)EPINIONS ($\kappa_u = 1$)



(d)EPINIONS ($\kappa_u = 5$)

Figure 4.4: Total regret (log-scale) vs. λ.

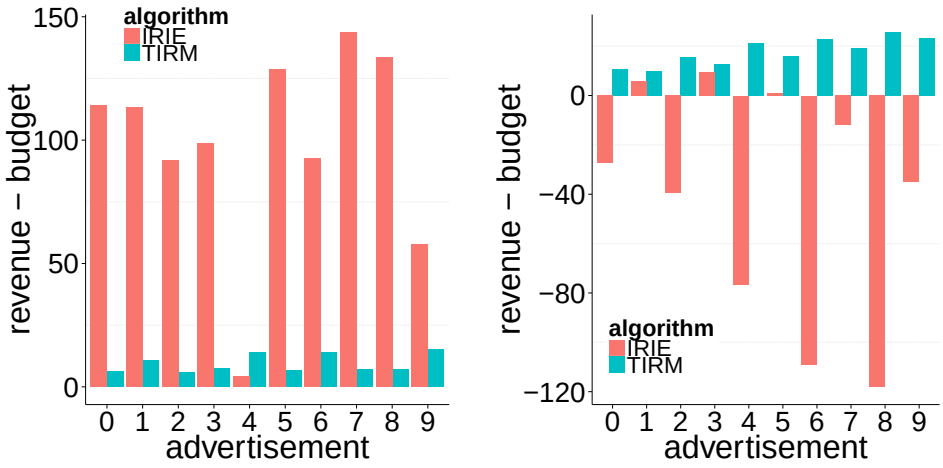
framework. Jung et al. [86] report that $\alpha = 0.7$ performs best on the datasets they tested. We did extensive testing on our datasets and found that $\alpha = 0.8$ gave the best spread estimation, and thus used 0.8 in all quality experiments.

- TIRM: Algorithm 5. We set ε to be 0.1 for quality experiments on FLIXSTER and EPINIONS, and 0.2 for scalability experiments on DBLP and LIVE-JOURNAL (following [132]).

For all algorithms, we evaluate the final regret of their output seed sets using Monte Carlo simulations (10K runs) for neutral, fair, and accurate comparisons.

4.5.1 Quality

Overall regret. First, we compare overall regret (as defined in Eq. (4.4)) against attention bound κ_u, varied from 1 to 5, with two choices 0 and 0.5 for λ. Figure 4.3 shows that the overall regret (in log-scale) achieved by TIRM and GREEDY-IRIE



(a)FLIXSTER (b)EPINIONS

Figure 4.5: Distribution of individual regrets ($\lambda = 0$, $\kappa_u = 5$).

are significantly lower than that of MYOPIC and MYOPIC+. For example, on FLIXSTER with $\lambda = 0$ and $\kappa_u = 1$, overall regrets of TIRM, GREEDY-IRIE, MYOPIC, and MYOPIC+, expressed relative to the total budget, are 2.5%, 26.1%, 122%, 141%, respectively. On EPINIONS with the same setting, the corresponding regrets are 6.5%, 15.9%, 145%, and 205%. MYOPIC, and MYOPIC+ typically always overshoot the budgets as they are not virality-aware when choosing seeds. Notice that even though MYOPIC+ is budget conscious, it still ends up overshooting the budget as a result of not factoring in virality in seed allocation. In almost all cases, overall regret by TIRM goes down as κ_u increases. The trend for MYOPIC and MYOPIC+ is the opposite, caused by their larger overshooting with larger κ_u. This is because they will select more seeds as κ_u goes up, which causes higher revenue (hence regret) due to more virality.

We also vary λ to be 0, 0.1, 0.5, and 1 and show the overall regrets under those values in Figure 4.4 (in log-scale), with two choices 1 and 5 for κ_u. As expected, in all test cases as λ increases, the overall regret also goes up. The hierarchy of algorithms (in terms of performance) remains the same as in Figure 4.3, with TIRM being the consistent winner. Note that even when λ is as high as 1, TIRM still wins and performs well. This suggests that the λ-assumption ($\lambda \leq \delta(u,i) \cdot cpe(i)$, $\forall$ user u and ad i) in Theorem 4.2 is conservative as TIRM can still achieve relatively low regret even with large λ values.

Drilling down to individual regrets. Having compared overall regrets, we drill down into the budget-regrets (see Section 4.3) achieved for different individual ads by TIRM and GREEDY-IRIE. Figure 4.5 shows the distribution of budget-regrets across advertisers for both algorithms. On FLIXSTER, both algorithms overshoot

FLIXSTER	$\kappa_u = 1$	2	3	4	5
TIRM	868	352	319	263	257
GREEDY-IRIE	3.7K	1.7K	1.5K	1237	1222
MYOPIC	29K	29K	29K	29K	29K
MYOPIC+	27K	13K	9.6K	7.5K	6.6K
EPINIONS	$\kappa_u = 1$	2	3	4	5
TIRM	4.4K	901	396	233	175
GREEDY-IRIE	3.1K	826	393	251	183
MYOPIC	76K	76K	76K	76K	76K
MYOPIC+	55K	28K	19K	15K	13K

Table 4.3: Number of nodes targeted vs. attention bounds ($\lambda = 0$).

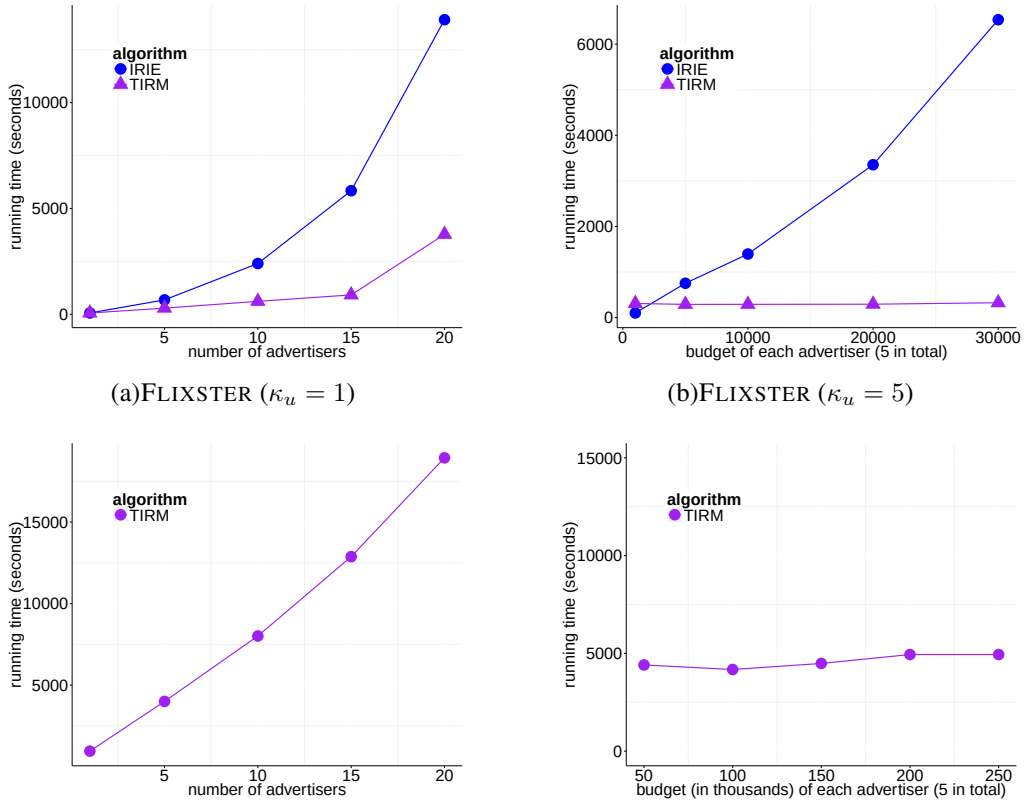
for all ads, but the distribution of TIRM-regrets is much more uniform than that of GREEDY-IRIE-regrets. E.g., for the fourth ad, GREEDY-IRIE even achieves a smaller regret than TIRM, but for all other ads, their GREEDY-IRIE-regret is at least 3.8 times as large as the TIRM-regret, showing a heavy skew. On EPINIONS, TIRM slightly overshoots for all advertisers as in the case of FLIXSTER, while GREEDY-IRIE falls short on 7 out of 10 ads and its budget-regrets are larger than TIRM for most advertisers. Note that MYOPIC and MYOPIC+ are not included here as Figures 4.3 and 4.4 have clearly demonstrated that they have significantly higher overshooting[6].

Number of targeted users. We now look into the distinct number of nodes targeted at least once by each algorithm, as κ_u increases from 1 to 5. Intuitively, as κ_u decreases, each node becomes "less available", and thus we may need more distinct nodes to cover all budgets, causing this measure to go up. The stats in Table 4.3 confirm this intuition, in the case of TIRM, GREEDY-IRIE, and MYOPIC+. MYOPIC is an exception since it allocates an ad to every user (i.e., all $|V|$ nodes are targeted). Note that on EPINIONS, TIRM targeted more nodes than GREEDY-IRIE. The reason is that GREEDY-IRIE tends to overestimate influence spread on EPINIONS, resulting in pre-mature termination of Greedy. When MC is used to estimate ground-truth spread, the revenue would fall short of budgets (see Figure 4.5). The behavior of GREEDY-IRIE is completely the opposite on FLIXSTER, showing its lack of consistency as a pure heuristic.

4.5.2 Scalability

We test the scalability of TIRM and GREEDY-IRIE on DBLP and LIVEJOURNAL. For simplicity, we set all CPEs and CTPs to 1 and λ to 0, and the values of these

[6]Their regrets are all from overshooting the budget on account of ignoring virality effects.



(a)FLIXSTER ($\kappa_u = 1$) (b)FLIXSTER ($\kappa_u = 5$)

(c)EPINIONS ($\kappa_u = 1$) (d)EPINIONS ($\kappa_u = 5$)

Figure 4.6: Running time of TIRM and GREEDY-IRIE on DBLP and LIVEJOURNAL.

parameters do not affect running time or memory usage. Influence probabilities on each edge $(u, v) \in E$ are computed using the Weighted-Cascade model [37]: $p^i_{u,v} = \frac{1}{|N^{in}(v)|}$ for all ads i. We set $\alpha = 0.7$ for GREEDY-IRIE and $\varepsilon = 0.2$ for TIRM, in accordance with the settings in [86, 132]. Attention bound $\kappa_u = 1$ for all users. We emphasize that our setting is fair and ideal for testing scalability as it simulates a fully competitive case: all advertisers compete for the same set of influential users (due to all ads having the same distribution over the topics) and the attention bound is at its lowest, which in turn will "stress-test" the algorithms by prolonging the seed selection process.

We test the running time of the algorithms in two dimensions: Figure 4.6(a) & 4.6(c) vary h (number of ads) with per-advertiser budgets B_i fixed (5K for DBLP, 80K for LIVEJOURNAL), while Figure 4.6(b) & 4.6(d) vary B_i when fixing $h = 5$. Note that GREEDY-IRIE results on LIVEJOURNAL (Figure 4.6(c) & 4.6(d)) are excluded due to its huge running time, details to follow.

DBLP	$h = 1$	5	10	15	20
TIRM	2.59	12.6	27.1	40.6	60.8
GREEDY-IRIE	0.16	0.30	0.48	0.54	0.84
LIVEJOURNAL	$h = 1$	5	10	15	20
TIRM	3.72	15.6	32.5	47.7	60.9

Table 4.4: Memory usage (GB).

At the outset, notice that TIRM significantly outperforms GREEDY-IRIE in terms of running time. Furthermore, as shown in Figure 4.6(a) & 4.6(c), the gap between TIRM and GREEDY-IRIE becomes larger as h increases. Furthermore, as shown in Figure 4.6(a), the gap between TIRM and GREEDY-IRIE on DBLP becomes larger as h increases. For example, when $h = 1$, both algorithms finish in 60 secs, but when $h = 15$, TIRM is 6 times faster than GREEDY-IRIE.

On LIVEJOURNAL, TIRM scales almost linearly w.r.t. the number of advertisers, It took about 16 minutes with $h = 1$ (47 seeds chosen) and 5 hours with $h = 20$ (4649 seeds). GREEDY-IRIE took about 6 hours to complete for $h = 1$, and did not finish after 48 hours for $h \geq 5$, thus we exclude it from Figure 4.6(c). When budgets increase (Figure 4.6(b)), GREEDY-IRIE's time will go up (super-linearly) due to more iterations of seed selections, but TIRM remains relatively stable (barring some minor fluctuations). On LIVEJOURNAL, TIRM took less than 75 minutes with $B_i = 50K$ (254 seeds), while GREEDY-IRIE could not finish in 48 hours, thus we exclude it from Figure 4.6(d). Note that once h is fixed, TIRM's running time depends heavily on the required number of random RR-sets (θ) for each advertiser rather than budgets, as seed selection is a linear-time operation for a given sample of RR-sets. Thus, the relatively stable trend on Figure 4.6(b) & 4.6(d) is due to the subtle interplay among the variables to compute $L(s, \varepsilon)$ (Eq. 4.5); similar observations were made for TIM in [132].

Table 4.4 shows the memory usage of TIRM and GREEDY-IRIE. As TIRM relies on generating a large number of random RR-sets for accurate estimation of influence spread, we observe high memory consumption by this algorithm, similar to the TIM algorithm [132]. The usage steadily increases with h. The memory usage of GREEDY-IRIE is modest, as its computation requires merely the input graph and probabilities. However, GREEDY-IRIE is a heuristic with no guarantees, which is reflected in its relatively poor regret performance compared to TIRM. Furthermore, as seen earlier, TIRM scales significantly better than GREEDY-IRIE on all datasets.

4.6 Discussion and Future Work

In this work, we build a bridge between viral marketing and social advertising, by drawing on the viral marketing literature to study influence-aware ad allocation for social advertising, under real-world business model, paying attention to important practical factors like relevance, social proof, user attention bound, and advertiser budget. In particular, we study the problem of regret minimization from the host perspective, characterize its hardness and devise a simple scalable algorithm with quality guarantees w.r.t. the total budget. Through extensive experiments we demonstrate its superior performance over natural baselines.

Our work takes a first step toward enriching the framework of social advertising by integrating it with powerful ideas from viral marketing and making the latter more applicable to real online marketing problems. It opens up several interesting avenues for further research. Studying continuous-time propagation models, possibly with the network and/or influence probabilities not known beforehand (and to be learned), and possibly in presence of hard competition constraints, is a direction that offers a wealth of possibilities for future work.

CHAPTER 5

SOCIAL ADVERTISING: REVENUE MAXIMIZATION

5.1 Introduction

The rise of online advertising platforms has generated new opportunities for advertisers in terms of personalizing and targeting their marketing messages. When users access a platform, they leave a trail of information that can be correlated with their consumption tastes, enabling better targeting options for advertisers. Social networking platforms particularly can gather larger amount of users' own shared information that stretches beyond general demographic and geographic data, hence, offers more advanced interest, behavioral, and connection-based targeting options, enabling a level of personalization that is not achievable by other online advertising channels. Hence, advertising on social networking platforms has been one of the fastest growing sectors in the online advertising landscape, further fueled by the explosion of investments in mobile ads: social advertising, a market that did not exist until Facebook launched its first advertising service in May 2005, is projected to generate 11 billion revenue by 2017, almost doubling the 2013 revenue.[1]

Social advertising. Social advertising models are typically employed by platforms such as Twitter, Tumblr, and Facebook through the implementation of the *promoted posts* that are shown in the "news feed" of their users.[2] A promoted

[1] http://www.unified.com/historyofsocialadvertising/

[2] According to a recent report, Facebook's news feed ads have 21 times higher click-through rates than standard web retargeting ads and 49 times the click-through rate of Facebook's right-hand side display ads. https://blog.adroll.com/trends/facebook-exchange-news-feed-numbers

post can be a video, an image, or simply a textual post containing an advertising message. Social advertising models of this type are usually associated with a *cost per engagement* (CPE) pricing scheme: the advertiser does not pay for the ad impressions, but pays to the platform owner (that hereafter we refer to as the *host*) only when a user actively engages with the ad. The *engagement* can be in the form of a social action such as *"like"*, *"share"*, or *"comment"*. In this chapter we blur the distinction between these different type of actions, and generically refer to all of them as *engagements* or *clicks* interchangeably.

Similar to organic (i.e., non-promoted) posts, promoted posts can propagate from user to user in the network,[3] potentially creating a virus-like contagion: whenever a user u engages with an ad i, the host is paid some fixed amount from the advertiser, moreover u's engagement with i appears in the feed of u's followers, that are hence exposed to the ad i and could in turn be influenced to engage with i, producing further revenue for the host [7, 134].

Incentivised social advertising. In this chapter we study the novel model of *incentivised social advertising*. Under this model, those users which are selected by the host to be the *seeds* for the campaign on a specific ad i, can take a "cut" on the social advertising revenue. These users are typically selected because they are influential or authoritative on the specific topic, brand, or market of i.

A recent report[4] indicates that Facebook is experimenting with the idea of incentivising users. YouTube launched a revenue-sharing program for prominent users in 2007. Twitch, the streaming platform of choice for gamers, lets partners make money through revenue sharing, subscriptions, and merchandise sales. YouNow, a streaming platform popular among younger users, earns money by taking a cut of the tips and digital gifts that fans give to its stars. On platforms without partner deals, including Twitter and Snapchat, celebrity users often strike sponsored deals to include brands in their posts, by-passing the platform host.[5]

In particular, in this chapter we consider incentives that are *proportional to the topical influence* of the seed users for the specific ad. More concretely, given an ad i, the financial incentive that a seed user u would get for engaging with i is proportional to the social influence that u has exhibited in the past in the topic of i. For instance, a user which often produces relevant content about long-distance running, capturing the attention of a relatively large audience, might be a good

[3] Tumblr's CEO David Karp reported (CES 2014) that a normal post is reposted on average 14 times, while promoted posts are on average reposted more than 10 000 times: `http://yhoo.it/1vFfIAc`.

[4] `http://www.theverge.com/2016/4/19/11455840/facebook-tip-jar-partner-program-monetization`

[5] `http://www.wsj.com/articles/more-marketers-offer-incentives-for-watching-ads-1451991600`

seed for endorsing a new model of running shoes. In this case, her past demonstrated influence on the topic would be taken in consideration when defining the lump amount for her engagement with the new model of running shoes. The same user could be selected as seed also for a new model of tennis shoes, but in that case the incentive would be lower, due to the lower past influence demonstrated.

The incentives model based on the past demonstrated social influence has several advantages. First of all, it captures in a neat uniform framework both the "celebrity-influencer", whose incentives are naturally very high (as her social influence), and who are typically preferred by more traditional types of advertising, such as TV ads; and the "ordinary-influencer" [8], a normal individual who is expert in some specific topic, thus, has a relatively restricted audience, or tribe, that trust her. Secondly, incentives do not only play their main role, that is to encourage the seed users to endorse an adverting campaign, but also, as a by-product, they incentivise the users of the social media platform to be influential in some topics by actively producing *good-quality content*, which also has an obvious direct benefit for the social media platform.

Revenue maximization. In this context, we study the fundamental problem of revenue maximization from the host perspective: an advertiser enters into a commercial agreement with the host to pay, following the CPE model, a fixed price p_i per each engagement with ad i. The agreement also specifies the finite budget B_i of the advertiser for the incentivised social advertising campaign for ad i. The host has to carefully select the seed users for the campaign: given that the budget B_i that it can receive from the advertiser is fixed, the host must try to achieve as many engagements on the ad i as possible, while spending little on the incentives for "seed" users. The host's task gets even more challenging by simultaneously accommodating many campaigns by different advertisers. As a quality guarantee for the advertising platform, for a fixed time window (say a 24-hours window), the host can tactically decide to select each user as the seed endorser for at most one ad: this constraint avoids the bad phenomenon of having, e.g., the same sport celebrity endorsing Nike and Adidas in the same time window. Therefore two ads i and j, which are in the same topical area, naturally compete for the influential users in that area.

We show that, keeping all important factors, such as topical relevance of ads, their propensity for social propagation, the topical influence of users, users incentives and advertisers budgets in consideration, the problem of revenue maximization in incentivised social advertising is NP-hard and it corresponds to the problem of monotone submodular function maximization subject to a partition matroid constraint on the ads-to-seeds allocation *and* submodular knapsack constraints on the advertisers' budgets. For this problem we devise two natural variants of the greedy algorithm for which we provide formal approximation guarantees. The

two algorithms differentiate on their sensitivity to advertisers' payment functions:

1. *Cost-Agnostic Greedy Algorithm* (CA-GREEDY), which greedily chooses the seed users solely based on the marginal gain in the revenue until the advertisers' budgets run out;

2. *Cost-Sensitive Greedy Algorithm* (CS-GREEDY), which greedily chooses the users based on the *rate* of marginal gain in revenue per marginal gain in the advertiser's payment, for each advertiser, until the advertisers' budgets run out.

Our results generalize the results of Iyer *et al.* [83,85] on submodular function maximization by (i) generalizing from a single submodular knapsack constraint to multiple submodular knapsack constraints, and (ii) by adding a partition matroid constraint.

5.1.1 Problem Definition

We first introduce the data model, then the business model, and finally we formally define the revenue maximization problem.

Data model, topic model, and propagation model. The social network platform, i.e., the host H, owns:

1. a *directed social graph* $G = (V, E)$, where an arc (u, v) means that v follows u, thus v can see u's posts and can be influenced by u;

2. a *topic model* for ads and users' interest, defined on a space of K topics;

3. a *topic-aware influence propagation model* defined on the social graph G and the topic model.

The topic model is defined by a hidden variable Z that can range among K states. Each topic (i.e., state of the latent variable) represents an abstract interest/pattern and intuitively models the underlying cause for each data observation (a user clicking on an ad). In our setting, the host owns a pre-computed probabilistic topic model. The topic model maps each ad i to a topic distribution $\vec{\gamma_i}$ over the latent topic space, formally:

$$\gamma_i^z = Pr(Z = z|i) \text{ with } \sum_{z=1}^{K} \gamma_i^z = 1.$$

The propagation model governs the way that ads propagate in the social network driven by social influence. In this work, we adopt the *Topic-aware Independent Cascade* model (TIC) introduce by Barbieri et al. [13] which extends the standard *Independent Cascade* (IC) model [88]: in TIC not only the ad is described by a distribution on the topic space, but also the strength of the social influence of user u over user v is topic-dependent, i.e., it is a probability $p^z_{u,v}$ for each topic z. Following the TIC model, when a node u clicks with an ad i, it has one chance of influencing each neighbor v, which has not yet clicked i, to do the same. This succeeds with a probability that is the weighted average of the arc probability w.r.t. the topic distribution of the ad i:

$$p^i_{u,v} = \sum\nolimits_{z=1}^{K} \gamma^z_i \cdot p^z_{u,v}. \tag{5.1}$$

Following the literature we denote with $\sigma_i(S_i)$ the *expected number of clicks* on ad i when S_i is the set of seed nodes, i.e., the nodes which are selected to endorse i and that get a financial incentive to do so. The influence value of a user u for ad i is defined as the *expected spread* of the singleton seed $\{u\}$ for the given ad description, under the TIC model i.e., $\sigma_i(\{u\})$: this is the quantity that is used to determine the incentive for user u to endorse the ad i.

Business model. An *advertiser*[6] enters into an agreement with the host for an incentivised social advertising campaign on his ad i: the advertiser agrees to pay the host a cost-per-engagement amount p_i for each click received by its ad i. The agreement also specifies the financial incentive for the seed users to endorse ad i: advertiser pays each seed user $u \in S_i$, an incentive

$$c_i(u) := \alpha \cdot \sigma_i(\{u\}),$$

where $\alpha > 0$ is a fixed amount in dollar cents set by the host. Abusing the notation slightly, we will denote the total cost of incentivising the users of the seed set S_i by $c_i(S_i)$, as the sum of the individual incentive costs of each seed user $u \in S_i$,

$$c_i(S_i) := \sum_{u \in S_i} c_i(u).$$

The advertiser's finite budget B_i limits the amount the advertiser can spend in

[6]We assume each advertiser has one ad to promote per time window, and use i to refer to the i-th advertiser and his ad interchangeably.

a social advertising campaign on ad i: when the seed set is S_i, the total payment advertiser i needs to make for the campaign on his ad i, denoted by $\rho_i(S_i)$, is the sum of his total costs for the ad-engagements, and incentivising the seed users:

$$\rho_i(S_i) := p_i \cdot \sigma_i(S_i) + c_i(S_i). \tag{5.2}$$

Let $\pi_i(S_i)$ denote the expected revenue of H from the total expected engagements to ad i when S_i is the seed set. Then, the expected revenue of H from ad i is given by

$$\pi_i(S_i) := p_i \cdot \sigma_i(S_i).$$

The revenue maximization problem. From the business model, the trade-off that the host faces when trying to maximize his own revenue is evident: on one hand, having a larger number of more influential seeds helps to increase the likelihood of having a successful campaign, or in other terms, it increases the expected number of clicks to ad i, hence the expected revenue of the host; on the other hand, each seed user has the associated cost of the financial incentives they receive, and the more influential they are, the larger is their incentive. The picture is made even more complex by the fact that, the host has to serve many advertisers at the same time, which could have potentially competitive ads (ads which are very close in the topic space).

Hereafter we assume a fixed time window (say a 24-hours period) in which the revenue maximization problem is defined. Within this time window we have h advertisers with ad description $\vec{\gamma_i}$, cost-per-engagement p_i, and budget B_i, $\forall i \in [h]$. We define an *allocation* $\vec{S}$ as a vector of h *pairwise disjoint* sets $(S_1, \cdots, S_h) \in 2^V \times \cdots \times 2^V$, where S_i is the seed set assigned to advertiser i to start the ad-engagement propagation process. Within the time window, each user in the platform can be selected to be seed for at most one ad, that is to say $S_i \cap S_j = \emptyset, \forall i, j \in [h]$.

We denote the total revenue of the host from h advertisers as the sum of all the ad-specific revenues:

$$\pi(\vec{S}) = \sum_{i \in [h]} \pi_i(S_i) \tag{5.3}$$

Next, we formally define the revenue maximization problem for incentivised social advertising from the host perspective.

Problem 5.1 (REVENUE-MAXIMIZATION (RM))**.** *Given a social graph $G = (V, E)$ with an instance of the TIC model, h advertisers with ad description $\vec{\gamma}_i$, cost-per-engagement p_i, and budget B_i, $\forall i \in [h]$, and ad-specific seed user incentive costs $c_i(u)$, $\forall u \in V$, $\forall i \in [h]$, find an allocation $\vec{S}$ such that the revenue of the host is maximized:*

$$\begin{aligned} \underset{\vec{S}}{\textit{maximize}} \quad & \pi(\vec{S}) \\ \textit{subject to} \quad & \rho_i(S_i) \leq B_i, \forall i \in [h], \\ & S_i \cap S_j = \emptyset, i \neq j, \forall i, j \in [h]. \end{aligned}$$

5.1.2 Contributions and Roadmap

The main contributions of this chapter can be summarised as follows:

- We initiate investigation in the area of *incentivised* social advertising, by formalizing the fundamental problem of revenue maximization from the host perspective, when the incentives paid to the seed users are proportional to their demonstrated past influence in the topic of the specific ad.
- We show that the problem is NP-hard and it corresponds to the problem of monotone submodular function maximization subject to partition matroid constraint on the ads-to-seeds allocation, and multiple submodular knapsack constraints on the budgets of advertisers. We devise 2 greedy algorithms with provable approximation guarantees, generalizing the results of Iyer *et al.* [83, 85].

The rest of the chapter is organized as follows. Section 5.3 presents the theoretical algorithms and prove their approximation guarantees. In Section 5.4 we provide a formal discussion of our ongoing efforts on devising scalable approximation algorithms. Relevant prior literature is surveyed in Section 5.2, while Section 5.6 concludes the chapter by discussing open challenges and future investigation.

5.2 Related Work

Computational advertising. As advertising on the web has become one of the largest businesses during the last decade, the general area of *computational advertising* has attracted a lot of research interest. The central problem of computational advertising is to find the "best match" between a given user in a given

context and a suitable advertisement. The context could be a user entering a query in a search engine ("sponsored search"), reading a web page ("content match" and "display ads"), or watching a movie on a portable device, etc. The most typical example is sponsored search: search engines show ads deemed relevant to user-issued queries, in the hope of maximizing click-through rates and in turn, revenue. Revenue maximization in this context is formalized as the well-known *Adwords* problem [103]. We are given a set Q of keywords and N bidders with their daily budgets and bids for each keyword in Q. During a day, a sequence of words (all from Q) would arrive online and the task is to assign each word to one bidder *upon its arrival*, with the objective of maximizing revenue for the given day while respecting the budgets of all bidders. This can be seen as a generalized online bipartite matching problem, and by using linear programming techniques, a $(1 - 1/e)$ competitive ratio is achieved [103]. Considerable work has been done in sponsored search and display ads [52, 62, 63, 68, 105]. For a comprehensive treatment, see a recent survey [102].

Social advertising. While computational advertising is a quite mature area, the sub-area of advertising on social network platforms is still in its infancy. Recent efforts, including Tucker [134] and Bakshy et al. [7], have shown, by means of field studies on sponsored posts in Facebook's News Feed, the importance and potentiality of keeping social influence in consideration when developing social advertising strategies. Nevertheless the literature concerning the exploitation of social influence for social advertising is rather limited.

Google's Bao and Chang have proposed *AdHeat* [11], a social ad model considering social influence in addition to relevance for matching ads to users. AdHeat diffuses hint words of influential users to others and then matches ads for each user with aggregated hints. Bao and Chang's experiments on a large online Q&A community show that AdHeat outperforms the relevance model on click-through-rate by significant margins. Wang et al. [140] also propose a new model for learning relevance, based on a heterogeneous social network approach, and apply it to the problem of selecting relevant ads for Facebook's users. In both [11] and [140] the proposed model is just assessed as a relevance model in terms of click-through-rate: neither viral propagation of ads nor revenue maximization are studied.

Our previous work [6] also study social advertising through the viral-marketing lens: in this work we study a different optimization problem and our business model is different as we do not consider an explicit monetary incentive to the seed users in [6].

Chalermsook et al. [33] take a viral-marketing perspective at the problem of social advertising, and study the revenue maximization problem for the host, when dealing with multiple advertisers at the same time. In their setting, each advertiser

a_i has to pay the host an amount c_i for each adoption of its product, up to a monetary budget B_i. However, an important difference from our setting is that in [33] each advertiser also specifies the maximum size s_i of its seed set. Thus, in practice, they have a double budget: one on the size of the seed set, one on the total CPE. Not having seed set size specified beforehand is a *significant challenge* we address in our work. Another important difference is that in our model both ads and social influence are *topic-aware*: this produces an interesting natural competition among ads which are close in a topic space for the attention of the users which are influential in the same area of the topic space. Instead Chalermsook et al. [33] adopt the simple IC model where all the ads are exactly the same.

Abbassi et al [1] also study social advertising through the viral-marketing lens. However, differently from our work and [6, 33] which are all based on CPE, they consider a CPM (cost-per-mille) pricing model: i.e., the advertiser enters in a contract with the host for its ad to be shown to a fixed number of users, agreeing to pay a certain CPM amount for every thousand impressions. Under this model the number of engagements (or clicks) that the ad receives, does not directly influences the revenue of the host. However, optimizing click-through-rate is nevertheless an important goal as it makes more likely that the advertiser will come back for another advertising campaign. Therefore the problem studied by Abbassi et al [1] is that of allocating ads to users so to optimize the number of clicks, for a predefined number of ads impressions, keeping in consideration social influence. Their results are mostly of theoretical interest and negative nature (i.e., hardness and strong inapproximability). None of these previous papers studies *incentivised social advertising* where the seed users are paid monetary incentives.

Viral marketing. As exemplified by the three papers discussed above, our work is also related to viral marketing, whose algorithmic optimization embodiment is the *influence maximization* problem [88], which we reviewed in detail in Chapter 2. The key difference between this literature and our setting, is that in the standard influence maximization the budget of an advertiser is modeled as a cardinality constraint on the number of free products to offer, hence the number of seed users to target [88]. Some work has studied the possibility to target the seed users non-uniformly, that is to say that different seeds might have a different cost, in which case the budget of an advertiser is modeled as a monetary amount that will be spend on the non-uniform costs of incentivising seed users [91, 111]. On the other hand, the real-world social advertisement models operate with monetary budgets, which is used not only for incentivising the seed users, but more generally for paying the CPE. Hence, the classic treatment of budgets in the optimization of a viral marketing campaign is inadequate for modeling the real-world social advertising scenarios, which is currently addressed by our work in this chapter.

5.3 Theoretical Analysis

In this section we we first show the correspondance of RM problem (formally defined in Problem 5.1) to the problem of monotone submodular function maximization subject to a partition matroid constraint on the ads-to-seeds allocation, *and* h submodular knapsack constraints on advertisers' payments for the total campaign costs. After showing that RM problem is NP-hard, we define the key concepts required for the theoretical analysis of RM problem, for which we devise two natural variants of the greedy algorithm, and provide formal approximation guarantees.

The monotonicity and submodularity of the ad-specific revenue function $\pi_i(\cdot)$ follows directly from the monotonicity and submodularity of the influence spread function $\sigma_i(\cdot)$. Being a linear combination of h monotone and submodular ad-specific revenue functions, the total revenue function $\pi(\vec{S}) = \sum_{i\in[h]} \pi_i(S_i)$, is also monotone and submodular. Similarly, the ad-specific payment function, $\rho_i(\cdot)$, being a non-negative linear combination of two monotone and submodular functions, $\sigma_i(\cdot)$ and $c_i(\cdot)$, is also monotone and submodular. Hence, the constraint that, for an advertiser i, the total cost of an incentivised social advertising campaign on his ad i should be less than his budget B_i, *i.e.*, $\rho_i(S_i) \leq B_i$, corresponds to a *submodular* knapsack constraint, for each $i \in [h]$.

Before we proceed further with our theoretical analysis, we first provide the required preliminary definitions.

Definition 5.1 (Independence System)**.** *A set system $(\mathcal{E}, \mathcal{I})$ defined with a finite ground set $\mathcal{E}$ of elements, and a family $\mathcal{I}$ of subsets of the ground set $\mathcal{E}$ is an independence system if it satisfies the following axiom:*

- $\emptyset \in \mathcal{I}$
- **Downward Closure***: If $X \subseteq Y$ and $Y \in \mathcal{I}$, then $X \in \mathcal{I}$*

Definition 5.2 (Matroid)**.** *An independence system $(\mathcal{E}, \mathcal{I})$ is a matroid $\mathfrak{M} = (\mathcal{E}, \mathcal{I})$ if it also satisfies the Augmentation axiom:*

- **Augmentation***: If $X \in \mathcal{I}$ and $Y \in \mathcal{I}$ and $|Y| > |X|$, then $\exists e \in Y \setminus X$: $X \cup \{e\} \in \mathcal{I}$.*

Definition 5.3 (Partititon Matroid)**.** *Let $\mathcal{E}_1, \cdots, \mathcal{E}_l$ be a partition of the ground set $\mathcal{E}$ into l* disjoint *sets. Let d_i be an integer, $0 \leq d_i \leq |\mathcal{E}_i|$. In a partition matroid $\mathfrak{M} = (\mathcal{E}, \mathcal{I})$, a set X is defined as independent when, for every index i, $|X \cap \mathcal{E}_i| \leq d_i$. Thus, $\mathcal{I}$ is defined as the following:*

$$\mathcal{I} = \{X \subseteq \mathcal{E} : |X \cap \mathcal{E}_i| \leq d_i, \forall i = 1, \cdots, l\} \tag{5.4}$$

In our RM problem, the constraint that the allocation $\vec{S} = (S_1, \cdots, S_h)$ should be composed of pairwise disjoint sets, *i.e.*, $S_i \cap S_j = \emptyset, i \neq j, \forall i, j \in [h]$, forms a partition matroid on the ground set of all possible node and advertiser pairings: given $G = (V, E)$, $|V| = n$, and a set $A = \{i : i \in [h]\}$ of advertisers, let $\mathcal{E} = V \times A$ denote the ground set of all possible $(node, advertiser)$ pairs, defined as:

$$\mathcal{E} = \{(u, i) : u \in V, i \in A\}.$$

Let $\mathcal{E}_u = \{(u, i) : i \in A\}, \forall u \in V$, and let $\{\mathcal{E}_u : \forall u \in V\}$ denote a *partition* of the ground set into n disjoint sets, *i.e.*, $\mathcal{E}_u \cap \mathcal{E}_v = \emptyset$, whenever $u \neq v$, and $\bigcup_{u \in V} \mathcal{E}_u = \mathcal{E}$. Let $\mathcal{X} \subseteq \mathcal{E}$ denote a feasible solution to the constraint that $\forall S_i, S_j \in \vec{S}$, $S_i \cap S_j = \emptyset$ when $i \neq j$: there is one-to-one correspondance between the pairwise disjoint seed sets $S_1, \cdots, S_h$ and the subsets $\mathcal{X} \in \mathcal{E}$ that satisfy $\mathcal{X} \cap \mathcal{E}_u \leq 1$, where the correspondance is given by

$$S_i = \{u : (u, i) \in \mathcal{X}\}, \forall i \in [h] \tag{5.5}$$

Then, the collection of independent subsets of $\mathcal{E}$ that provide feasible solutions to this constraint is defined as

$$\mathcal{I} = \{\mathcal{X} : \mathcal{X} \subseteq \mathcal{E}, |\mathcal{X} \cap \mathcal{E}_u| \leq 1, \forall u \in V\}$$

forming a partition matroid $\mathfrak{M} = (\mathcal{E}, \mathcal{I})$, on the ground set $\mathcal{E}$.

We have just showed that RM problem corresponds to the problem of submodular function maximization subject to partition matroid $\mathfrak{M} = (\mathcal{E}, \mathcal{I})$, *and* h submodular knapsack constraints. Next, we show that RM problem is NP-hard.

Theorem 5.1. RM *problem is NP-hard.*

Proof. When $h = 1$, RM problem has only one submodular knapsack constraint, and no partition matroid constraint that aries from the requirement of disjointness of the seed sets (as the solution is a single seed set). Hence, the NP-hard *Submodular-Cost Submodular-Knapsack (SCSK)* problem, which tackles the problem of submodular function maximization subject to a submodular knapsack constraint, as studied by Iyer *et al.* [85], is a special case of RM problem when $h = 1$. □

Now, we show that the constraints of the NP-hard RM problem form an an

independence system defined on the ground set $\mathcal{E}$ of $(node, advertiser)$ pairs. Given the partition matroid constraint $\mathfrak{M} = (\mathcal{E}, \mathcal{I})$, and h submodular knapsack constraints, let $\mathcal{C}$ denote the family of subsets, defined on the ground set $\mathcal{E}$ of $(node, advertiser)$ pairs, that are *feasible* solutions to RM problem. For each knapsack constraint $\rho_i(\cdot) \leq B_i$, let $\mathcal{F}_i \subseteq 2^V$ denote the collection of feasible subsets of the ground set V, defined as follows:

$$\mathcal{F}_i = \{S_i \subseteq V : \rho_i(S_i) \leq B_i\}.$$

The set system $(V, \mathcal{F}_i)$ defined by the set of feasible solutions to any knapsack constraint is *downward-closed*, hence is an *independence system*. Given $\mathcal{F}_i$, $\forall i \in [h]$ and $\mathfrak{M} = (\mathcal{E}, \mathcal{I})$, we can define the family of subsets defined on $\mathcal{E}$ that are feasible solutions to RM problem as follows:

$$\mathcal{C} = \{\mathcal{X} : \mathcal{X} \in \mathcal{I} \text{ and } S_i \in \mathcal{F}_i, \forall i \in [h]\}$$

where $S_i = \{u : (u, i) \in \mathcal{X}\}$. Let $\mathcal{X} \in \mathcal{C}$ and $\mathcal{X}' \subseteq \mathcal{X}$. In order to show that $\mathcal{C}$ is an independence system, we need to show that it satisfies the *downward closure* axiom as follows:

$$\mathcal{X} \in \mathcal{C} \text{ and } \mathcal{X}' \subseteq \mathcal{X} \implies \mathcal{X}' \in \mathcal{C}. \tag{5.6}$$

Let $S'_i = \{u : (u, i) \in \mathcal{X}'\}$, $\forall i \in [h]$, hence we have $S'_i \subseteq S_i$. As each single knapsack constraint $\rho_i(\cdot) \leq B_i$ is associated with the independence system $(V, \mathcal{F}_i)$, we have $S'_i \in \mathcal{F}_i$ for any $S'_i \subseteq S_i$, $\forall i \in [h]$. Similarly, as $\mathcal{X} \in \mathcal{I}$, we have $S_i \cap S_j = \emptyset$, hence, we should also show that $S'_i \cap S'_j = \emptyset$, which directly follows from one-to-one correspondence between S'_i and $\mathcal{X}'$, and $\mathcal{X}' \in \mathcal{I}$ due to the downward closure property of the partition matroid $\mathfrak{M}$. Hence, $\mathcal{C}$ satisfies the downward closure axiom depicted in Eq. 5.6, thus, is an independence system.

Our theoretical guarantees for the greedy approximation algorithms to RM problem depend on the notion of *curvature* of submodular functions, which was introduced by Conforti *et al.* [45] to the submodular function optimization literature: given a submodular function f, Conforti *et al.* [45] define the total curvature κ_f of f as:

$$\kappa_f = 1 - \min_{j \in V} \frac{f(\{j\} \mid V \setminus \{j\})}{f(\{j\})},$$

and the curvature $\kappa_f(S)$ of f *wrt* a set S as:

$$\kappa_f(S) = 1 - \min_{j \in S} \frac{f(\{j\} \mid S \setminus \{j\})}{f(\{j\})}.$$

where $f(\{j\} \mid S \setminus \{j\}) = f(V) - f(S \setminus \{j\})$.

In plain words, the curvature $0 \leq \kappa_f \leq 1$ measures the distance of f from *modularity*: $\kappa_f = 0$ iff for *modular* functions, and $\kappa_f = 1$ for totally normalized and saturated functions like matroid rank functions [85]. Noting that $\kappa_f = \kappa_f(V)$, curvature $\kappa_f(S)$ of f *wrt* a set S similarly reflects how much the marginal values $f(j \mid S)$ can decrease as a function of S, deviating from *modularity*. As studied in [45, 83–85, 138], there are several closely related forms of curvature defined in the literature, all of which provide improved bounds for submodular optimization problems. Iyer *et al.* [83] introduced the notion of average curvature $\hat{\kappa}_f(S)$ of f *wrt* a set S as

$$\hat{\kappa}_f(S) = 1 - \frac{\sum_{j \in S} f(\{j\} \mid S \setminus \{j\})}{\sum_{j \in S} f(\{j\})},$$

and demonstrated the following relation between these several forms of curvature:

$$0 \leq \hat{\kappa}_f(S) \leq \kappa_f(S) \leq \kappa_f(V) = \kappa_f \leq 1.$$

Next, we study the greedy approximation algorithms for RM problem. As we are dealing with submodular knapsack constraints on advertisers' total payments, *i.e.*, their total spendings on the costs of the incentivised social advertising campaign for their ads, we consider two natural variants of the greedy algorithm based on its mindfulness to the revenue produced for the money spent on the campaign costs: (*i*) Cost-Agnostic Greedy Algorithm (CA-GREEDY), which greedily chooses the seed users solely based on the marginal gain in the revenue until the advertisers' budgets run out; (*ii*) Cost-Sensitive Greedy Algorithm (CS-GREEDY), which greedily chooses the users based on the *rate* of marginal gain in revenue per marginal gain in the advertiser's payment, for each advertiser, until the advertisers' budgets run out.

Note that, Iyer *et al.* [83, 85], also consider these 2 greedy algorithm variants for their *Submodular-Cost Submodular-Knapsack (SCSK)* problem, that is the special single knapsack case of our RM problem when $h = 1$: their cost-agnostic greedy approximation results appears in [85], and their cost-sensitive

greedy approximation results later appears in [83], since, as stated in [85], they didn't have the theoretical results for cost-sensitive greedy approximation at the time of publication.

5.3.1 Cost-Agnostic Greedy Algorithm

Cost-Agnostic Greedy Algorithm (CA-GREEDY) for RM problem, depicted in Algorithm 8, chooses at each iteration a $(node, advertiser)$ pair that provides the maximum increase in the revenue of the host: let $\mathcal{X}_g \subseteq \mathcal{E}$ denote the greedy solution set of $(node, advertiser)$ pairs, returned by CA-GREEDY, having one-to-one correspondance with the greedy allocation $\vec{S}$, *i.e.*, $S_i = \{u : (u,i) \in \mathcal{X}_g\}$, $\forall S_i \in \vec{S}$. Also let $\mathcal{X}_g^t$ denote the greedy solution set after t iterations of CA-GREEDY. At each iteration t, CA-GREEDY first finds the $(node, advertiser)$ pair $(u^*, i^*) \leftarrow \underset{(u,i)\in\mathcal{E}^{t-1}}{\operatorname{argmax}} \pi_i(u \mid S_i^{t-1})$, and tests whether the addition of this pair to the current greedy solution set $\mathcal{X}_g^{t-1}$ would violate any matroid or knapsack independence constraint: if (u^*, i^*) is feasible, *i.e.*, $\mathcal{X}_g^{t-1} \cup \{(u^*, i^*)\} \in \mathcal{C}$, the pair (u^*, i^*) is added to the greedy solution as the t-th $(node, advertiser)$ pair. Otherwise, (u^*, i^*) is removed from the current ground set of $(node, advertiser)$ pairs $\mathcal{E}^{t-1}$, as it violates either the partition matroid constraint or the knapsack constraint. CA-GREEDY terminates when there is no feasible $(node, advertiser)$ pair left in the current ground set $\mathcal{E}^{t-1}$.

Observation 5.1. *The total revenue function $\pi(\vec{S})$ has a total curvature κ_π, defined on the ground set $\mathcal{E}$:*

$$\kappa_\pi = 1 - \min_{(u,i)\in\mathcal{E}} \frac{\pi_i(\{u\} \mid S_i \setminus \{u\})}{\pi_i(\{u\})}.$$

Proof. Let $g : 2^{\mathcal{E}} \mapsto \mathbb{R}_{\geq 0}$ be a monotone submodular function defined on the ground set $\mathcal{E}$. Then, the total curvature κ_g of g is defined as follows:

$$\kappa_g = 1 - \min_{x\in\mathcal{E}} \frac{g(\{x\} \mid \mathcal{E} \setminus \{x\})}{g(\{x\})},$$

where $x = (u, i) \in \mathcal{E}$. Using the one-to-one correspondance between the seed sets $S_1 \subseteq V, \cdots, S_h \subseteq V$, and the set of $(node, advertiser)$ pairs $\mathcal{X} \subseteq \mathcal{E}$ (Eq. 5.5), we can alternatively formulate RM problem as follows:

$$\begin{aligned} \underset{\mathcal{X} \subseteq \mathcal{E}}{\text{maximize}} \quad & g(\mathcal{X}) \\ \text{subject to} \quad & \mathcal{X} \in \mathcal{C}. \end{aligned}$$

where $g(\mathcal{X}) = \sum_{i \in [h]} \pi_i(S_i)$ with $S_i = \{u : (u, i) \in \mathcal{X}\}$.

Using this correspondance, we can rewrite κ_g as κ_π as follows:

$$\kappa_g = \kappa_\pi = 1 - \min_{(u,i) \in \mathcal{E}} \frac{\pi_i(\{u\} \mid S_i \setminus \{u\})}{\pi_i(\{u\})}.$$

□

Theorem 5.2. CA-GREEDY *obtains an approximation guarantee of*

$$\frac{1}{\kappa_\pi} \left[1 - \left(\frac{R - \kappa_\pi}{R} \right)^r \right]$$

where κ_π is the total curvature of the total revenue function $\pi(\vec{S})$ as defined in Observation 5.1, R is the upper rank of $\mathcal{C}$, i.e.*, the maximum cardinality of a maximal feasible set in $\mathcal{C}$, and r is the lower rank of $\mathcal{C}$,* i.e.*, the minimum cardinality of a maximal feasible set in $\mathcal{C}$, defined as follows:*

$$r = \min\{|X| : X \in \mathcal{C} \text{ and } X \cup \{(u,i)\} \notin \mathcal{C},\ \forall (u,i) \notin X\}$$

and

$$R = \max\{|X| : X \in \mathcal{C} \text{ and } X \cup \{(u,i)\} \notin \mathcal{C},\ \forall (u,i) \notin X\}.$$

Proof. Given that the family $\mathcal{C}$ of subsets that provide *feasible* solutions to the RM problem, is an independence system on the ground set $\mathcal{E}$ of $(node, advertiser)$ pairs as we have shown, the approximation guarantee of CA-GREEDY directly follows from the result of Conforti et al. [45] for submodular function maximization subject to an independence system constraint. □

Given the partition matroid $\mathfrak{M} = (\mathcal{E}, \mathcal{I})$, and h submodular knapsack constraints, $\rho_i(S_i) \leq B_i$, we can infer the following worst-case values for r and R: $r = h$ (where each advertiser i gets just one seed node u_i whose payment $\rho_i(\{u_i\})$ exhausts its budget B_i) and $R = n$.

Algorithm 8: CA-GREEDY

Input : $G = (V, E), \mathcal{E}, \mathcal{C}, \vec{B}, \vec{p}, \vec{\gamma}_i, \forall i \in [h], c(u), \forall u \in V$
Output: $\vec{S} = (S_1, \cdots, S_h)$

1 $t \leftarrow 1, \mathcal{E}^0 \leftarrow \mathcal{E}, \mathcal{X}_g^0 \leftarrow \emptyset$
2 $S_i^0 \leftarrow \emptyset, \forall i \in [h]$
3 **while** $\mathcal{E}^{t-1} \neq \emptyset$ **do**
4 $\quad (u^*, i^*) \leftarrow \underset{(u,i) \in \mathcal{E}^{t-1}}{\operatorname{argmax}} \pi_i(u \mid S_i^{t-1})$
5 $\quad$ **if** $(\mathcal{X}_g^{t-1} \cup \{(u^*, i^*)\}) \in \mathcal{C}$ **then**
6 $\quad\quad S_{i^*}^t \leftarrow S_{i^*}^{t-1} \cup \{u^*\}$
7 $\quad\quad S_j^t \leftarrow S_j^{t-1}, \forall j \neq i^*$
8 $\quad\quad \mathcal{X}_g^t \leftarrow \mathcal{X}_g^{t-1} \cup \{(u^*, i^*)\}$
9 $\quad\quad \mathcal{E}^t \leftarrow \mathcal{E}^{t-1} \setminus \{(u^*, i^*)\}$
10 $\quad\quad t \leftarrow t + 1$
11 $\quad$ **else**
12 $\quad\quad \mathcal{E}^{t-1} \leftarrow \mathcal{E}^{t-1} \setminus \{(u^*, i^*)\}$
13 $S_i \leftarrow S_i^{t-1}, \forall i \in [h]$
14 **return** $\vec{S} = (S_1, \cdots, S_h)$

5.3.2 Cost-Sensitive Greedy Algorithm

Cost-sensitive greedy algorithm (CS-GREEDY) for RM problem, at each iteration t, first finds the $(node, advertiser)$ pair $(u^*, i^*) \leftarrow \underset{(u,i) \in \mathcal{E}^{t-1}}{\operatorname{argmax}} \frac{\pi_i(u \mid S_i^{t-1})}{\rho_i(u \mid S_i^{t-1})}$, and tests whether the addition of this pair to the current greedy solution set $\mathcal{X}_g^{t-1}$ would violate any matroid or knapsack independence constraint: if (u^*, i^*) is feasible, *i.e.*, $\mathcal{X}_g^{t-1} \cup \{(u^*, i^*)\} \in \mathcal{C}$, the pair (u^*, i^*) is added to the greedy solution as the t-th $(node, advertiser)$ pair. Otherwise, (u^*, i^*) is removed from the current ground set of $(node, advertiser)$ pairs $\mathcal{E}^{t-1}$, as it violates either the partition matroid constraint or the knapsack constraint. CS-GREEDY terminates when there is no feasible $(node, advertiser)$ pair left in the current ground set $\mathcal{E}^{t-1}$. CS-GREEDY can be obtained by simply replacing Line 4 of Algorithm 8 with

$$(u^*, i^*) \leftarrow \underset{(u,i) \in \mathcal{E}^{t-1}}{\operatorname{argmax}} \frac{\pi_i(u \mid S_i^{t-1})}{\rho_i(u \mid S_i^{t-1})}$$

thus, we do not provide the pseudocode.

Theorem 5.3. CS-GREEDY *obtains an approximation guarantee of*

$$\left(1 + \frac{B \cdot K}{(1 - \max\limits_{i \in [h]} \hat{\kappa}_{\rho_i}(S_i^*)) \cdot \Delta\rho_{min}}\right)^{-1}$$

where $\vec{S^*} = (S_1^*, \cdots, S_h^*)$ *is the optimal allocation,* $\vec{S} = (S_1, \cdots, S_h)$ *is the greedy allocation,* $B = \sum_{i \in [h]} B_i$ *is the total of advertisers' budgets,* $K = \bigcup\limits_{i \in [h]} |S_i|$, $\Delta\rho_{min} := \min\limits_{i \in [h], t \in [1,K]} \rho_i(S_i^t) - \rho_i(S_i^{t-1})$, i.e., *minimum marginal gain in payment obtained in an iteration of the greedy algorithm, and* $\hat{\kappa}_{\rho_i}(S_i^*)$ *is the average curvature of* $\rho_i(\cdot)$ *wrt* S_i^*, $\forall i \in [h]$.

Fact 5.1. *(Fact* 1.10 *in [142]) For given positive numbers* $a_1, \cdots, a_h$ *and* $b_1, \cdots, b_h$*, the following always holds:*

$$\min_{i \in [h]} \frac{a_i}{b_i} \leq \frac{\sum_{i \in [h]} a_i}{\sum_{i \in [h]} b_i} \leq \max_{i \in [h]} \frac{a_i}{b_i}$$

Proof of Theorem 5.3. Let $\mathcal{X}^* \subseteq \mathcal{E}$ denote the optimal solution set of $(node, advertiser)$ pairs corresponding to the optimal allocation $\vec{S^*}$ such that $S_i^* = \{u : (u, i) \in \mathcal{X}^*\}$. Similarly, let $\mathcal{X}_g \subseteq \mathcal{E}$ denote the greedy solution set of $(node, advertiser)$ pairs, having one-to-one correspondance with the greedy allocation $\vec{S}$, *i.e.*, $S_i = \{u : (u, i) \in \mathcal{X}_g\}$, $\forall S_i \in \vec{S}$. Let $K = |\mathcal{X}_g| = |\mathcal{X}_g^K|$ denote the total size of the greedy solution of $(node, advertiser)$ pairs. Due to submodularity and monotonicity, we have:

$$\begin{aligned}\pi(\vec{S^*}) &\leq \pi(\vec{S}) + \sum_{(u,i) \in \mathcal{X}^* \setminus \mathcal{X}_g} \pi_i(u \mid S_i) \\ &\leq \pi(\vec{S}) + \sum_{(u,i) \in \mathcal{X}^*} \pi_i(u \mid S_i).\end{aligned}$$

At each iteration t, greedy algorithm first finds the $(node, advertiser)$ pair $(u^*, i^*) \leftarrow \operatorname{argmax}\limits_{(u,i) \in \mathcal{E}^{t-1}} \frac{\pi_i(u \mid S_i^{t-1})}{\rho_i(u \mid S_i^{t-1})}$, and tests whether the addition of this pair to the current greedy solution set $\mathcal{X}_g^{t-1}$ would violate any independence constraint: if (u^*, i^*) is feasible, *i.e.*, if $\mathcal{X}_g^{t-1} \cup \{(u^*, i^*)\} \in \mathcal{C}$, the pair (u^*, i^*) is added to

the greedy solution as the t-th $(node, advertiser)$ pair, which we will denote by (u_t, i_t). Otherwise, (u^*, i^*) is removed from the current ground set $\mathcal{E}^{t-1}$ since it is not feasible (violating at least one matroid or knapsack constraint). Let $\mathcal{U}^t$ denote the set of $(node, advertiser)$ pairs that the greedy algorithm *tested* to add to the greedy solution in the first $(t+1)$ iterations, before the addition of the $(t+1)$-st pair (u_{t+1}, i_{t+1}) into the greedy solution set $\mathcal{X}_g^t$. Then, $\forall (u,i) \in U^t \setminus U^{t-1}$, we have:

$$\frac{\pi_i(u \mid S_i^t)}{\rho_i(u \mid S_i^t)} \geq \frac{\pi_{i_{t+1}}(u_{t+1} \mid S_{i_{t+1}}^t)}{\rho_{i_{t+1}}(u_{t+1} \mid S_{i_{t+1}}^t)}.$$

since they had the maximum $\dfrac{\pi_i(u \mid S_i^t)}{\rho_i(u \mid S_i^t)}$, but they failed the independence test to be the $(t+1)$-st pair, and were removed from $\mathcal{E}^t$. Moreover, $\forall (u,i) \in U^t \setminus U^{t-1}$, we can also infer that

$$\frac{\pi_i(u \mid S_i^{t-1})}{\rho_i(u \mid S_i^{t-1})} \leq \frac{\pi_{i_t}(u_t \mid S_{i_t}^{t-1})}{\rho_{i_t}(u_t \mid S_{i_t}^{t-1})}.$$

since they they do not belong to $U^{t-1} \setminus U^{t-2}$, *i.e.*, they were not good enough to be tested during the selection of the t-th pair. Note that, the greedy algorithm terminates when there is no feasible pair left in the ground set. Hence after the K iterations of the greedy algorithm, $\mathcal{E}^K$ contains only the *infeasible* pairs that violate some matroid or knapsack constraint. Thus, we have $\mathcal{X}^* = \bigcup_{t=1}^{K} [\mathcal{X}^* \cap (U^t \setminus U^{t-1})]$. Let $\mathcal{U}_t^* := \mathcal{X}^* \cap (U^t \setminus U^{t-1})$. Then, we have:

$$\begin{aligned}
\pi(\vec{S^*}) &\leq \pi(\vec{S}) + \sum_{(u,i) \in \mathcal{X}^*} \pi_i(u \mid S_i) \\
&= \pi(\vec{S}) + \sum_{t=1}^{K} \sum_{(u,i) \in \mathcal{U}_t^*} \pi_i(u \mid S_i) \\
&\leq \pi(\vec{S}) + \sum_{t=1}^{K} \sum_{(u,i) \in \mathcal{U}_t^*} \frac{\pi_{i_t}(u_t \mid S_{i_t}^{t-1})}{\rho_{i_t}(u_t \mid S_{i_t}^{t-1})} \cdot \rho_i(u \mid S_i^{t-1}).
\end{aligned}$$

where the last inequality is due to, $\forall(u,i) \in \mathcal{U}_t^*$:

$$\pi_i(u \mid S_i) \leq \pi_i(u \mid S_i^{t-1}) \leq \frac{\pi_{i_t}(u_t \mid S_{i_t}^{t-1})}{\rho_{i_t}(u_t \mid S_{i_t}^{t-1})} \cdot \rho_i(u \mid S_i^{t-1}).$$

Continuing, we have:

$$\begin{aligned}\pi(\vec{S^*}) &\leq \pi(\vec{S}) + \sum_{t=1}^{K} \sum_{(u,i)\in\mathcal{U}_t^*} \frac{\pi_{i_t}(u_t \mid S_{i_t}^{t-1})}{\rho_{i_t}(u_t \mid S_{i_t}^{t-1})} \cdot \rho_i(u \mid S_i^{t-1}) \\ &= \pi(\vec{S}) + \sum_{t=1}^{K} \left(\frac{\pi_{i_t}(u_t \mid S_{i_t}^{t-1})}{\rho_{i_t}(u_t \mid S_{i_t}^{t-1})} \cdot \sum_{(u,i)\in\mathcal{U}_t^*} \rho_i(u \mid S_i^{t-1}) \right)\end{aligned}$$

$$\begin{aligned}\pi(\vec{S^*}) &\leq \pi(\vec{S}) + \left(\sum_{t=1}^{K} \frac{\pi_{i_t}(u_t \mid S_{i_t}^{t-1})}{\rho_{i_t}(u_t \mid S_{i_t}^{t-1})} \right) \cdot \left(\sum_{t=1}^{K} \sum_{(u,i)\in\mathcal{U}_t^*} \rho_i(u) \right) \\ &= \pi(\vec{S}) + \left(\sum_{t=1}^{K} \frac{\pi_{i_t}(u_t \mid S_{i_t}^{t-1})}{\rho_{i_t}(u_t \mid S_{i_t}^{t-1})} \right) \cdot \left(\sum_{(u,i)\in\mathcal{X}^*} \rho_i(u) \right).\end{aligned}$$

Being monotone and submodular, each $\rho_i(\cdot)$ has the following average curvature $\hat{\kappa}_{\rho_i}(S_i^*)$ defined wrt its optimal seed set S_i^*, $\forall i \in [h]$:

$$\hat{\kappa}_{\rho_i}(S_i^*) = 1 - \frac{\sum_{u\in S_i^*} \rho_i(u \mid S_i^* \setminus u)}{\sum_{u\in S_i^*} \rho_i(u)}.$$

Rearranging the terms, we get:

$$1 - \hat{\kappa}_{\rho_i}(S_i^*) = \frac{\sum_{u\in S_i^*} \rho_i(u \mid S_i^* \setminus u)}{\sum_{u\in S_i^*} \rho_i(u)}.$$

Using Fact 5.1, we have:

$$\min_{i\in[h]} \frac{\sum\limits_{u\in S_i^*} \rho_i(u \mid S_i^* \setminus u)}{\sum\limits_{u\in S_i^*} \rho_i(u)} \leq \frac{\sum\limits_{i\in[h]}\sum\limits_{u\in S_i^*} \rho_i(u \mid S_i^* \setminus u)}{\sum\limits_{i\in[h]}\sum\limits_{u\in S_i^*} \rho_i(u)} \leq \max_{i\in[h]} \frac{\sum\limits_{u\in S_i^*} \rho_i(u \mid S_i^* \setminus u)}{\sum\limits_{u\in S_i^*} \rho_i(u)}.$$

Thus, we have:

$$\min_{i\in[h]}(1-\hat{\kappa}_{\rho_i}(S_i^*)) \leq \frac{\sum\limits_{i\in[h]}\sum\limits_{u\in S_i^*} \rho_i(u \mid S_i^* \setminus u)}{\sum\limits_{i\in[h]}\sum\limits_{u\in S_i^*} \rho_i(u)} \leq \max_{i\in[h]}(1-\hat{\kappa}_{\rho_i}(S_i^*)).$$

Since $\hat{\kappa}_{\rho_i}(S_i^*) \in [0,1], \forall i \in [h]$, we can alternatively write:

$$1-\max_{i\in[h]} \hat{\kappa}_{\rho_i}(S_i^*) \leq \frac{\sum\limits_{i\in[h]}\sum\limits_{u\in S_i^*} \rho_i(u \mid S_i^* \setminus u)}{\sum\limits_{i\in[h]}\sum\limits_{u\in S_i^*} \rho_i(u)} \leq 1-\min_{i\in[h]} \hat{\kappa}_{\rho_i}(S_i^*).$$

Then, we have:

$$\begin{aligned}
\sum_{(u,i)\in\mathcal{X}^*} \rho_i(u) &= \sum_{i\in[h]}\sum_{u\in S_i^*} \rho_i(u) \\
&\leq \frac{\sum_{i\in[h]}\sum_{u\in S_i^*} \rho_i(u \mid S_i^* \setminus u)}{1-\max\limits_{i\in[h]} \hat{\kappa}_{\rho_i}(S_i^*)} \\
&\leq \frac{\sum_{i\in[h]} B_i}{1-\max\limits_{i\in[h]} \hat{\kappa}_{\rho_i}(S_i^*)}.
\end{aligned}$$

where the last inequality follows from the fact that

$$\sum_{i\in[h]}\sum_{u\in S_i^*} \rho_i(u \mid S_i^* \setminus u) \leq \sum_{i\in[h]} \rho_i(S_i^*) \leq \sum_{i\in[h]} B_i = B$$

due to submodularity and the knapsack constraints. Thus, continuing from where

we left, we have:

$$\pi(\vec{S^*}) \leq \pi(\vec{S}) + \left(\sum_{t=1}^{K} \frac{\pi_{i_t}(u_t \mid S_{i_t}^{t-1})}{\rho_{i_t}(u_t \mid S_{i_t}^{t-1})} \right) \cdot \left(\sum_{(u,i) \in \mathcal{X}^*} \rho_i(u) \right)$$
$$\leq \pi(\vec{S}) + \left(\sum_{t=1}^{K} \frac{\pi_{i_t}(u_t \mid S_{i_t}^{t-1})}{\rho_{i_t}(u_t \mid S_{i_t}^{t-1})} \right) \cdot \frac{B}{1 - \max\limits_{i \in [h]} \hat{\kappa}_{\rho_i}(S_i^*)}$$

$$\pi(\vec{S^*}) \leq \pi(\vec{S}) + \left(\sum_{t=1}^{K} \frac{\sum_{t=1}^{K} \pi_{i_t}(u_t \mid S_{i_t}^{t-1})}{\rho_{i_t}(u_t \mid S_{i_t}^{t-1})} \right) \cdot \frac{B}{1 - \max\limits_{i \in [h]} \hat{\kappa}_{\rho_i}(S_i^*)}$$
$$= \pi(\vec{S}) + \pi(\vec{S}) \left(\sum_{t=1}^{K} \frac{1}{\rho_{i_t}(u_t \mid S_{i_t}^{t-1})} \right) \cdot \frac{B}{1 - \max\limits_{i \in [h]} \hat{\kappa}_{\rho_i}(S_i^*)}$$
$$\leq \pi(\vec{S}) + \pi(\vec{S}) \cdot \frac{B \cdot k}{(1 - \max\limits_{i \in [h]} \hat{\kappa}_{\rho_i}(S_i^*)) \min\limits_{t \in [1,K]} \rho_{i_t}(u_t \mid S_{i_t}^{t-1})}$$
$$= \pi(\vec{S}) + \pi(\vec{S}) \cdot \frac{B \cdot K}{(1 - \max\limits_{i \in [h]} \hat{\kappa}_{\rho_i}(S_i^*)) \Delta\rho_{min}}.$$

□

Notice that, for handling the matroid constraint, our proof technique for Theorem 5.3 follows the reasoning applied by Fisher *et al.* [64] to the theoretical analysis of submodular function maximization subject to matroid constraints.

Similar to the findings of Iyer *et al.* [83] for the single knapsack version of our RM problem, our results also show that the approximation guarantee of the cost-sensitive greedy algorithm is unbounded when $\hat{\kappa}_{\rho_i}(S_i^*) = 1$, which is the case for matroid rank functions. However, combining the cost-sensitive and cost-agnostic results, we can obtain the bounded approximation guarantee.

5.4 Scalable Algorithms

In this section, we provide a formal discussion on our ongoing efforts for devising scalable versions of our approximation algorithms for RM problem.

Remember that, we consider two natural variants of the greedy algorithm for RM problem based on its mindfulness to the revenue produced for the money spent on the campaign costs: (*i*) CA-GREEDY which greedily chooses the seed users solely based on the marginal gain in revenue; (*ii*) CS-GREEDY, which greedily chooses the users based on the *rate* of marginal gain in revenue per marginal gain in payment. These 2 variants of the greedy algorithm involve a large number of calls to influence spread computations: at each iteration t, for each advertiser i, and for each node $u \in V \setminus S_i^{t-1}$, CA-GREEDY and CS-GREEDY need to compute $\pi_i(u \mid S_i^{t-1})$ and $\dfrac{\pi_i(u \mid S_i^{t-1})}{\rho_i(u \mid S_i^{t-1})}$ respectively.

Given any seed set S, computing its *exact* influence spread $\sigma(S)$ under the IC model is #P-hard [37, 38], and this hardness trivially carries over to the topic-aware IC model. A common practice is to use Monte Carlo (MC) simulations [88], however, accurate estimation requires a large number of MC simulations, which is prohibitively expensive and not scalable. Thus, to make CA-GREEDY and CS-GREEDY scalable, we need an alternative approach.

In the influence maximization literature, considerable effort has been devoted to developing more efficient and scalable algorithms, which we have previously reviewed in Chapter 2. The latest state-of-the-art influence maximization algorithms built on RIS framework [22, 112, 131, 132] provide significant improvements over using MC simulations. However, a straightforward application of these algorithms for solving RM problem would not work since their estimation procedures rely on knowing the exact number k of seed nodes required, which is the input to the influence maximization problem: the number of seed nodes needed to solve RM problem is driven by the current payments and the budgets of advertisers, hence is dynamic. Moreover, IMM [131], and SSA [112] are particularly very specialized for solving the influence maximization problem as they need to fine tune the many different parameters they use *w.r.t.* the $(1-1/e)$-approximation guarantee and optimal solution OPT_k of the influence maximization problem.

5.4.1 Scalable CA-GREEDY

Notice that CA-GREEDY follows a similar greedy framework like the influence maximization problem, *i.e.*, for each advertiser i, choosing the (*feasible*) seed set that provides the maximum revenue, hence, the maximum influence spread. This allows us to easily adjust the TIRM algorithm we propose in Chapter 4 for devising a scalable version of CA-GREEDY, which we will refer as *Two-phase Iterative Cost-Agnostic Revenue Maximization* (TI-CARM): TI-CARM directly adopts the RR-sets sampling framework of TIRM, and maximizes the marginal gain in revenue at each iteration as opposed to minimizing regret, while appropriately estimating the latent seed set size required for the estimation.

The estimation of the latent seed set size required by TI-CARM can be obtained as follows: for ease of exposition, let us first consider a single advertiser i. Let B_i be the budget of advertiser i and let s_i be the true number of seeds required to maximize the cost-agnostic revenue for advertiser i. We do not know s_i and we estimate it in successive iterations as $\tilde{s}_i^t$. Thus, we start with an estimated value for s_i, denoted $\tilde{s_i}^1$, and use it to obtain a corresponding θ_i^1. If $\theta_i^t > \theta_i^{t-1}$, we will need to sample an additional $(\theta_i^t - \theta_i^{t-1})$ RR-sets, and use all RR-sets sampled up to this iteration to select $(\tilde{s}_i^t - \tilde{s}_i^{t-1})$ additional seeds. After adding those seeds, if the current assigned payment $\rho_i(S_i)$ of i is still less than B_i, more seeds can be assigned to a_i. Thus, we will need another iteration and we further revise our estimation of s_i. The new value, $\tilde{s}_i^{t+1}$, is obtained by adding to $\tilde{s}_i^t$ the floor function of the ratio between the current unspent budget $B_i - \rho_i(S_i)$ and the sum of the marginal revenue contributed by the $\tilde{s}_i^t$-th seed and c_i^{max}, *i.e.*, the maximum seed user incentive cost specific for advertiser i, $c_i^{max} := \max_{v \in V} c_i(v)$. This ensures we do not overestimate, thanks to submodularity, as future seeds have diminishing marginal gains and lower incentive costs.

5.4.2 Scalable CS-GREEDY

Notice that the greedy node selection criteria employed by CS-GREEDY significantly deviates from the greedy framework of CA-GREEDY, hence, from the influence maximization problem. Consider the pair

$$(u^*, i^*) := \underset{(u,i) \in \mathcal{E}^{t-1}}{\operatorname{argmax}} \frac{\pi_i(u \mid S_i^{t-1})}{\rho_i(u \mid S_i^{t-1})}. \tag{5.7}$$

that has the maximum *rate* of marginal gain in revenue per marginal gain in payment at an iteration t of CS-GREEDY. In order to find the pair that satisfies Eq. 5.7 at an iteration t, CS-GREEDY needs to compute for each advertiser i:

$$u_i^t := \underset{v:(v,i) \in \mathcal{E}^{t-1}}{\operatorname{argmax}} \frac{\pi_i(u \mid S^{t-1})}{\rho_i(u \mid S^{t-1})}. \tag{5.8}$$

Finding the node u_i^t (Eq. 5.8) at an iteration t, for an advertiser i, requires to compute $\sigma_i(v|S_i^{t-1})$, $\forall v : (v,i) \in \mathcal{E}^{t-1}$, using MC simulations. Notice that, node u_i^t might even correspond to the node that has the *minimum* marginal gain in influence spread, since the rate defined in Eq. 5.8 is not necessarily monotonic. Thus, any effort to devise a scalable alternative to the MC simulations of CS-

GREEDY should estimate $\sigma_i(v|S_i^{t-1}), \forall v : (v,i) \in \mathcal{E}^{t-1}$.

None of the influence maximization algorithms built on RIS framework [22, 112, 131, 132] are capable of working as an influence spread oracle that can efficiently estimate the influence spread of any given node or set of nodes. Among these algorithms, TIM is the only algorithm that could be considered to adopt as an influence spread oracle, since the size of the sample that TIM uses is derived such that the influence spread of any set of at most k nodes can be accurately estimated: however, in practise, the nodes that are not very influential most likely do not appear in the random RR sets sampled, hence, making it impossible to obtain their estimates as we have experimentally witnessed. Due to this, adjusting TI-CARM to employ the cost-sensitive node selection criteria would similarly ignore the less influential node as it uses the same sampling framework as TIM.

Thus, if we want to find a "*fully*" cost-sensitive solution, *i.e.*, that computes Eq. 5.7 over "all" the pairs in $(u,i) \in \mathcal{E}^{t-1}$, we should either use MC simulations to be able to perform the marginal gain level estimation operations, or devise alternative estimation procedures, that can perform marginal gain level operations more efficiently than MC simulations.

To this end, as an alternative to MC simulations, next we introduce the integration of the sequential sampling design [49,139] into the RIS framework, which we refer shortly as SEQ-RIS. Then we formally demonstrate, by forming an equivalence between the possible world semantics and the RIS framework, how we can relate the marginal gain $\sigma(u|S)$ to a Bernoulli random variable with a parameter that is equal to the probability that a random RR set intersects the node u, but not with S. We then use this formalization for the estimation of $\sigma(u|S)$ using SEQ-RIS framework.

Sequential Sampling Design. For a given set S, the sample size required to achieve an influence spread estimate $\hat{\sigma}(S)$ that is (ϵ,δ)-approximation of $\sigma(S)$ is also a random variable: one of the classical tools that can be used to identify a lower bound on the required sample size is Chernoff Bounds [41]. However, Chernoff Bounds require the knowledge of the unknown mean μ^S (or OPT_k as in the case of TIM) which translates to additional estimation procedures, and loose bounds on the statistically required sample size. Alternatively, rather than fixing a sample size in advance, a sequential sampling design [139] allows to use the outcomes of previous experiments to decide, in accordance with a pre-defined stopping rule, whether the current sample provides an estimate that is (ϵ,δ)-approximation of $\sigma(S)$.

Based on sequential sampling design, Dagum *et al.* [49] propose, for any random variable X distributed in the interval $[0,1]$, a simple but powerful "Stopping Rule Algorithm" (SRA) that provides (ϵ,δ)-approximation of its mean μ^X. SRA, by using a pre-defined stopping rule $\Upsilon(\epsilon,\delta)$ (depicted in Eq. 5.9), sequentially per-

forms experiments by generating independent and identical copies $X_1, X_2, \cdots$ of the random variable X, until the first time the current number θ of experiments satisfies $\sum_{i=1}^{\theta} X_i \geq \Upsilon(\epsilon, \delta)$. Once this stopping condition is satisfied, SRA terminates with $\hat{\mu}^X := \Upsilon(\epsilon, \delta)/\theta$, that is (ϵ, δ)-approximation of μ^X. As stated in [49], if X is a Bernoulli random variable, then the expected number of experiments SRA runs is within a constant factor of the optimal.

$$\Upsilon(\epsilon, \delta) := 1 + (1 + \epsilon) \cdot 4(e - 2) \cdot \log(2/\delta)/\epsilon^2 \tag{5.9}$$

Notice that, SRA [49] can be directly employed, for a given S, to produce an estimate $\hat{\mu}^S$ that is (ϵ, δ)-approximation of μ^S: we start with an empty sample $\mathcal{R}$ at iteration 0; then, at each iteration, we generate a random RR set, and add into $\mathcal{R}$, until the first iteration θ_S in which the current $\mathcal{R}$ ($|\mathcal{R}| = \theta_S$) satisfies $\text{cov}_{\mathcal{R}}(S) = \sum_{i=1}^{\theta_S} X_i^S = \lceil \Upsilon(\epsilon, \delta) \rceil$ [7]. Once this stopping condition is satisfied, we can return $\hat{\mu}^S := \Upsilon(\epsilon, \delta)/\theta_S$, that is (ϵ, δ)-approximation of μ^S.

Next, we first briefly review the possible world semantics, which was introduced to the influence maximization literature by Kempe *et al.* [88], and the computation of marginal gains within this context.

Possible world semantics. Let $\mathcal{W}$ denote the set of all possible worlds that we can generate from $G = (V, E)$, by removing each encountered edge $(v, w) \in E$, with probability $1 - p_{v,w}$. Kempe *et al.* [88] show that, for a given set S, $\sigma(S)$ is the weighted average over all possible worlds:

$$\sigma(S) = \sum_{\omega \in \mathcal{W}} Pr[\omega] \cdot \sigma^{\omega}(S)$$

where

$$Pr[\omega] = \prod_{(u,v) \in \omega} p_{uv} \cdot \prod_{(u,v) \in G \setminus \omega} (1 - p_{uv})$$

and $\sigma^{\omega}(S)$ is the influence spread of S in the possible world ω. As stated by Kempe *et al.* [88], applying principle of deferred decisions, we can think of $\sigma^{\omega}(S)$ as the influence spread of S in the deterministic realization of the possible world

[7] Since X_i^S can only take values $\{0, 1\}$, the first time θ_S satisfies $\sum_{i=1}^{\theta_S} X_i^S \geq \Upsilon(\epsilon, \delta)$ corresponds to when $\sum_{i=1}^{\theta_S} X_i^S = \lceil \Upsilon(\epsilon, \delta) \rceil$

ω, hence, treat $\sigma^{\omega}(S)$ as a deterministic quantity, computed as:

$$\sigma^{\omega}(S) = \sum_{u \in V} path^{\omega}(S, v)$$

where $path^{\omega}(S, v)$ is the indicator variable that equals 1 if $\exists u \in S$ that can reach node $v \in V$ through a directed path in the deterministic world ω, and 0 otherwise. Let ω^T denote the deterministic graph obtained by reversing the edges of ω, and let $path^{\omega^T}(v, S)$ denote an indicator variable that equals 1 if there exists $u \in S$ that v can reach through a directed path in ω^T, and 0 otherwise. Then we can further interpret $path^{\omega}(S, v)$ as follows:

$$\begin{aligned}
\sigma^{\omega}(S) &= \sum_{u \in V} path^{\omega}(S, v) \\
&= \sum_{u \in V} path^{\omega^T}(v, S) \\
&= \sum_{u \in V} \mathbb{1}_{[S \cap R^{\omega}(v) \neq \emptyset]}
\end{aligned}$$

where $R^{\omega}(v)$ is the *deterministic* RR set rooted as node v in the deterministic world ω. Using this simple correspondance, and interpreting G as a distribution over unweighted directed graphs with a sample space $\mathcal{W}$, where each edge $(v, w) \in E$ is realized with probability $p_{v,w}$, Borgs *et al.* [22] formed the following equivalence that is the backbone of the RIS framework:

$$\begin{aligned}
\sigma(S) &= \sum_{\omega \in \mathcal{W}} Pr[\omega] \cdot \sigma^{\omega}(S) \\
&= \sum_{\omega \in \mathcal{W}} Pr[\omega] \cdot \sum_{v \in V} path^{\omega}(S, v) \\
&= \sum_{v \in V} \sum_{\omega \in \mathcal{W}} Pr[\omega] \cdot path^{\omega}(S, v) \\
&= \sum_{v \in V} \mathop{\mathbb{E}}_{g \sim G} [path_g(S, v)]
\end{aligned}$$

$$\begin{aligned}\sigma(S) &= \sum_{v \in V} \underset{g \sim G}{Pr} \left[path_g(S, v) = 1\right] \\ &= \sum_{v \in V} \underset{g \sim G}{Pr} \left[path_{g^T}(v, S) = 1\right] \\ &= \sum_{v \in V} \underset{g \sim G}{Pr} \left[S \cap R_g(v) \neq \emptyset\right]\end{aligned}$$

where the deterministic variables are accompanied by superscripts, and the random variables by subscripts. Following possible world semantics, for a given S and $u \notin S$, we can compute $\sigma(u|S) = \sigma(S \cup \{u\}) - \sigma(S)$ as follows:

$$\sigma(u|S) = \sum_{\omega \in \mathcal{W}} Pr[\omega] \cdot \sigma^{\omega}(u|S)$$

where

$$\begin{aligned}\sigma^{\omega}(u \mid S) &= \sum_{v \in V} path^{\omega}(S \cup \{u\}, v) - \sum_{v \in V} path^{\omega}(S, v) \\ &= |\{v : path^{\omega}(S \cup \{u\}, v) = 1| - |\{v : path^{\omega}(S, v) = 1\}| \\ &= |\ \{v : path^{\omega}(u, v) = 1 \text{ and } path^{\omega}(S, v) = 0\}\ |\end{aligned}$$

We can further extend this definition, and establish its equivalence using *deterministic* RR sets as follows:

$$\sigma^{\omega}(u \mid S) = |\ \{v : [\{u\} \cap R^{\omega}(v) \neq \emptyset] \text{ and } [S \cap R^{\omega}(v) = \emptyset]\}\ | \tag{5.10}$$

Marginal Gain Estimation. Now, we formally demonstrate how submodularity of the influence spread function translates to the decomposition of the indicator random variable X^S into mutually exclusive indicator random variables, all of which define compound events, *w.r.t.* the nodes or sets of nodes of S, that are directly associated to the definition of marginal gains.

Remember that, for a given set S, $X^S \sim$ Bernoulli(μ^S) is the indicator random variable for the event $[S \cap R \neq \emptyset]$, with success probability μ^S and failure probability $1 - \mu^S$, where μ^S is the probability that a random RR set R has non-empty intersection with S:

$$\mu^S = \frac{\sigma(S)}{n} = \textbf{Pr}\left[S \cap R \neq \emptyset\right].$$

Let $S' = S \cup \{u\}$, and let $X^{S'} \sim \text{Bernoulli}(\mu^{S'})$ denote the indicator random variable for the event $[(S \cup \{u\}) \cap R \neq \emptyset]$, succeeding with probability $\mu^{S'}$, failing with probability $1 - \mu^{S'}$, where $\mu^{S'}$ is the probability that a random RR set R has non-empty intersection with S':

$$X^{S'} = \begin{cases} 1, & \text{if } (S \cup \{u\}) \cap R \neq \emptyset \\ 0, & \text{otherwise} \end{cases}$$

Notice that, we can interpret the random variable $X^{S'}$ as follows:

$$X^{S'} = \begin{cases} 1, & \text{if } S \cap R \neq \emptyset \text{ or } \{u\} \cap R \neq \emptyset \\ 0, & \text{otherwise} \end{cases}$$

Hence, we can interpret the event $[(S \cup \{u\}) \cap R \neq \emptyset]$ as the union of 2 events:

$$[(S \cup \{u\}) \cap R \neq \emptyset] := [S \cap R \neq \emptyset] \text{ or } [\{u\} \cap R \neq \emptyset]$$

hence, using principle of inclusion and exclusion, we have:

$$\begin{aligned} \text{Pr}\left[(S \cup \{u\}) \cap R \neq \emptyset\right] = \text{Pr}\left[S \cap R \neq \emptyset\right] + \text{Pr}\left[\{u\} \cap R \neq \emptyset\right] - \\ \text{Pr}\left[(S \cap R \neq \emptyset) \text{ and } \{u\} \cap R \neq \emptyset\right] \end{aligned}$$

Notice that, $\text{Pr}\left[\{u\} \cap R \neq \emptyset\right] = \mu^u = \sigma(\{u\})/n$ is the probability that a random RR set R has non-empty intersection with node u. Using the basic difference rule of probability, we can further derive the following equivalence:

$$\text{Pr}\left[(S \cup \{u\}) \cap R \neq \emptyset\right] = \text{Pr}\left[S \cap R \neq \emptyset\right] + \text{Pr}\left[(\{u\} \cap R \neq \emptyset) \text{ and } (S \cap R = \emptyset)\right]$$

Realize that the event $[(\{u\} \cap R \neq \emptyset) \text{ and } (S \cap R = \emptyset)]$ directly follows the

definition of marginal gain, which we demonstrated in Eq. 5.10. Hence, we have:

$$\begin{aligned}\sigma(u|S) &= n \cdot (\sigma(S \cup \{u\}) - \sigma(S)) \\ &= n \cdot (\text{Pr}\left[(S \cup \{u\}) \cap R \neq \emptyset\right] - \text{Pr}\left[S \cap R \neq \emptyset\right]) \\ &= n \cdot (\text{Pr}\left[\{u\} \cap R \neq \emptyset \text{ and } S \cap R = \emptyset\right]).\end{aligned}$$

For a given set S and node u, let $X^{u|S} \sim \text{Bernoulli}(\mu^{u|S})$ denote the indicator random variable for the event

$$[\{u\} \cap R \neq \emptyset \text{ and } S \cap R = \emptyset],$$

succeeding with probability $\mu^{u|S}$ where $\mu^{u|S} := \sigma(u|S)/n$ is the probability that a random RR set intersects node u, but not with S. In order to estimate, $\sigma(u|S)$, we can simply follow any usual estimation procedure, by designing an experiment that produces independent copies of the random variable $X^{u|S}$, and use the average of the experiment outcomes as the estimate $\hat{\mu}^{u|S} := \dfrac{\hat{\sigma}(u|S)}{n}$. Next, we demonstrate how we can perform this operation within our SEQ-RIS framework.

For a given set S, node u, and sample $\mathcal{R}$ of random RR sets, let $\text{cov}_{\mathcal{R}}(u|S)$ denote the marginal gain in coverage of adding u to S in $\mathcal{R}$:

$$\text{cov}_{\mathcal{R}}(u|S) = |\{R \in \mathcal{R} \mid S \cap R = \emptyset \ \wedge \ u \in R\}| \tag{5.11}$$

In order to produce an estimate $\hat{\mu}^{u|S}$, that is (ϵ, δ)-approximation of $\mu^{u|S}$, SRA [49] can be directly employed. Algorithm 9 depicts the adaptation of SRA to our setting, with more efficient implementation that takes advantage of the monotonicity of the coverage function.

Application to Cost-Sensitive Greedy Algorithm. We have formally demonstrated the nice relation between the submodularity of the influence spread function and the compound events defining the marginal gain estimators. Now we provide remarks on its applicability to devise a scale alternative to CS-GREEDY.

Notice that, while MC simulations are computationally expensive, SRA itself is not "always" very efficient for estimating the influence spread of non-influential nodes: given an advertiser seed set S_i and a node u, assume the extreme scenario that $\sigma_i(u|S_i) = 1$, *i.e.*, node u can only influence himself given S_i. Using Wald's Equation [139], the expected size of the sample $\mathcal{R}_i$ that SRA needs to use to produce an estimate $\hat{\sigma}_i(u|S_i)$, that is (ϵ, δ)-approximation of $\sigma_i(u|S_i)$ can be com-

Algorithm 9: SEQ-RIS Marginal Gain Estimation

Input : $G = (V, E), S \subseteq V, u \in V \setminus S, \epsilon, \delta$
Output: $\hat{\sigma}(u|S)$

$\mathcal{R} \leftarrow \emptyset$;
$\Upsilon(\epsilon, \delta) := 1 + (1 + \epsilon) \cdot 4(e - 2) \cdot \log(2/\delta)/\epsilon^2$;
$\mathcal{R} \leftarrow \mathcal{R} \cup$ GenRandRRSets($\lceil \Upsilon(\epsilon, \delta) \rceil$) ;
$\text{cov}_{\mathcal{R}}(u|S) \leftarrow |\{R \in \mathcal{R} \mid S \cap R = \emptyset \ \wedge \ u \in R\}|$;
while $cov_{\mathcal{R}}(u|S) < \Upsilon(\epsilon, \delta)$ **do**
  $\mathcal{R} \leftarrow \mathcal{R} \cup$ GenRandRRSets($\lceil \Upsilon(\epsilon, \delta) \rceil - \text{cov}_{\mathcal{R}}(u|S)$) ;
  $\text{cov}_{\mathcal{R}}(u|S) \leftarrow |\{R \in \mathcal{R} \mid S \cap R = \emptyset \ \wedge \ u \in R\}|$;
$\hat{\sigma}(u|S) = n \cdot \Upsilon(\epsilon, \delta)/|\mathcal{R}|$;
return $\hat{\sigma}(u|S)$.

puted from:

$$\mathbb{E}[\theta] = \frac{\mu^{u|S_i}}{\mathbb{E}\left[\sum_{j=1}^{\theta} X_j^{u|S_i}\right]} \tag{5.12}$$

$$= \frac{1}{n \cdot \Upsilon(\epsilon, \delta)} \tag{5.13}$$

For a small directed graph of $n = 15229$ nodes, using $\epsilon = 0.1$, and $\delta = 1/n$, this translates to generating almost 50 million random RR sets just to estimate $\sigma_i(u|S_i)$.

Thus, for the problems in which the greedy node selection criteria does not follow the maximum coverage problem, like our cost-sensitive RM problem and the budgeted influence maximization problem [91, 111], scalability still remains an open challenge for finding a *fully* cost-sensitive greedy solution: we are currently working on devising an efficient and scalable algorithm to find a *fully* cost-sensitive greedy solution to RM problem.

Thus, for our experiments, rather than resorting on a *fully* cost-sensitive solution that computes $\sigma_i(v|S_i^{t-1})$, $\forall v : (v, i) \in \mathcal{E}^{t-1}$ at each iteration t and for each advertiser i, we will replace the MC simulations of CS-GREEDY with the SRA algorithm adapted for marginal gain estimation (Algorithm 9), and will inspect only a "window" of nodes that provide the highest w marginal gains in influence spread for applying cost-sensitive selection criteria. We will refer to this algorithm as CS-GREEDY-SRA. Notice that, this requires CS-GREEDY-SRA to efficiently approximate the top W nodes having the maximum marginal gains for each advertiser i at each iteration t. Next, in Theorem 5.4, we present our formal results

Algorithm 10: SEQ-RIS Maximum Marginal Gain Approximation

Input : $G = (V, E)$, $S \subseteq V$, $u \in V \setminus S$, ϵ, δ
Output: $\langle u, \hat{\sigma}(u|S) \rangle$

$\Upsilon := \Upsilon(\epsilon, \delta/(n - |S|))$;
$u \leftarrow \underset{v \in V \setminus S}{\text{argmax}} |\{R \in \mathcal{R} \mid S \cap R = \emptyset \ \wedge \ v \in R\}|$;
$\text{cov}_{\mathcal{R}}(u|S) \leftarrow |\{R \in \mathcal{R} \mid S \cap R = \emptyset \ \wedge \ u \in R\}|$;
while $cov_{\mathcal{R}}(u|S) < \Upsilon$ **do**
  $\mathcal{R} \leftarrow \mathcal{R} \cup \mathsf{GenRandRRSets}(\lceil \Upsilon \rceil - \text{cov}_{\mathcal{R}}(u|S))$;
  $u \leftarrow \underset{v \in V \setminus S}{\text{argmax}} |\{R \in \mathcal{R} \mid S \cap R = \emptyset \ \wedge \ v \in R\}|$;
  $\text{cov}_{\mathcal{R}}(u|S) \leftarrow |\{R \in \mathcal{R} \mid S \cap R = \emptyset \ \wedge \ u \in R\}|$;
$\hat{\sigma}(u|S) = n \cdot \Upsilon / |\mathcal{R}|$;
return $\langle u, \hat{\sigma}(u|S) \rangle$.

for efficiently approximating the node with the maximum marginal gain under the IC model, which directly applies to the TIC model, given their equivalence for an item $\vec{\gamma}_i$. We will use this approximation to select the pair (u^*, i^*) among the $W \cdot h$ pairs: notice that when $W = 1$, cost-sensitive greedy solution is equivalent to the cost-agnostic greedy solution.

Theorem 5.4. *Given a set $S \subseteq V$, let $u^* \in V \setminus S$ denote the node that provides the maximum marginal gain to the influence spread of S under the IC model:*

$$u^* = \underset{v \in V \setminus S}{\text{argmax}} \, \sigma(v|S).$$

Then, Algorithm 10 returns a node $\tilde{u}$ such that:

$$Pr\left[\sigma(u^*|S)(1-\epsilon) \leq \hat{\sigma}(\tilde{u}|S) \leq \sigma(u^*|S)(1+\epsilon)\right] > 1 - \delta.$$

Proof. First, notice that, in order to find a node $\tilde{u}$ that approximately provides the maximum marginal gain, we can run SRA simultaneously[8], with input parameters $(\epsilon, \delta/(n - |S|))$, to compute $(\epsilon, \delta/(n - |S|))$-approximation of $\sigma(v|S)$ for each $v \in V \setminus S$: generate a random RR set R into initially empty $\mathcal{R}$, and compute

$$\text{cov}_{\mathcal{R}}(v|S) = |\{R \in \mathcal{R} \mid S \cap R = \emptyset \ \wedge \ v \in R\}|,$$

[8]In the rest of the proof, we will refer to this algorithm, that executes SRA simultaneously for more than one random variable, using the same pool of random RR sets, as simultaneous-SRA.

for each $v \in V \setminus S$; whenever for some v at some iteration θ, $\text{cov}_{\mathcal{R}}(v|S) = \lceil \Upsilon(\epsilon, \delta/(n - |S|))\rceil$, compute its estimate $\Upsilon(\epsilon, \delta/(n - |S|)/\theta$, and continue the estimation procedure with the rest of the nodes, until simultaneous-SRA computes all the estimates. When simultaneous-SRA terminates, for each $v \in V \setminus S$, we have:

$$\text{Pr}\left[\hat{\mu}^{v|S} < (1-\epsilon)\mu^{v|S}\right] + \text{Pr}\left[\hat{\mu}^{v|S} > (1+\epsilon)\mu^{v|S}\right] \leq \frac{\delta}{n - |S|}. \tag{5.14}$$

Then, using union bound over these $n - |S|$ estimation failure scenarios, we can select the node $\tilde{u} := \underset{v \in V \setminus S}{\text{argmax}}\, \hat{\sigma}(v|S)$ as the node that approximately provides the maximum marginal gain with $1 - \delta$ probability.

Notice that, the size of the sample $\mathcal{R}$ when simultaneous-SRA generates a marginal gain estimate for a node v is the *stopping time* such that:

$$\Upsilon(\epsilon, \delta/(n - |S)) \leq \text{cov}_{\mathcal{R}}(v|S) < \Upsilon(\epsilon, \delta/(n - |S|)) + 1. \tag{5.15}$$

Using Wald's Equation [139]:

$$\mathbb{E}\left[\text{cov}_{\mathcal{R}}(v|S)\right] = \mathbb{E}\left[\theta_{v|S}\right] \cdot \mu^{v|S}. \tag{5.16}$$

Hence, during the execution of simultaneous-SRA, we can expect that the first node that could reach $\lceil \Upsilon(\epsilon, \delta/(n - |S|))\rceil$ coverage would be the node u^* since $\mu^{u^*|S}$ is the maximum among all the $n - |S|$ means:

$$\frac{\Upsilon(\epsilon, \delta/(n - |S|))}{\mu^{u^*|S}} \leq \mathbb{E}[\theta_{u^*}] \leq \cdots \leq \frac{\Upsilon(\epsilon, \delta/(n - |S|))}{\mu^{u_{min}|S}} \leq \mathbb{E}[\theta_{u_{min}}] \tag{5.17}$$

where $u_{min} = \underset{v \in V \setminus S}{\text{argmin}}\, \sigma(v|S)$.

Hence, thanks to the stopping rule that takes into account possible $n - |S|$ bad over- or under-estimation scenarios, rather than executing simultaneous-SRA until it generates all $n - |S|$ estimates, we can stop the estimation procedure early, as soon as a node, $\tilde{u}$, reaches $\Upsilon\left(\epsilon, \delta/(n - |S|)\right)$ coverage, and return it as the approximate solution with more than $1 - \delta$ probability, as depicted in Algorithm 10. □

	FLIXSTER	EPINIONS
#nodes	30K	76K
#edges	425K	509K
type	directed	directed

Table 5.1: Statistics of network datasets.

5.5 Experiments

In this section, we provide an empirical discussion on our algorithms. We test and compare the following algorithms that we formally discussed in Section 5.4.

- TI-CARM: For finding a cost-agnostic solution, we use the TI-CARM algorithm obtained from the adaptation of TIRM to RM problem.
- TI-CSRM: For finding a *restricted* cost-sensitive solution, we use TI-CSRM algorithm, simply derived from TI-CARM by applying the cost-sensitive node selection criteria for the selection of seed nodes.
- CS-GREEDY-SRA: For finding a *restricted* cost-sensitive solution, we use the CS-GREEDY-SRA algorithm that replaces the MC simulations of CA-GREEDY with SRA, and for each advertiser, computes the top W maximum marginal gains, using our maximum marginal gain approximation procedure (Algorithm 10) for SEQ-RIS. We will use several values of W to measure the affect of the window size on the total revenue and running time obtained.

Datasets. For our experiments, we use two real-world social networks, whose basic statistics are summarized in Table 5.1. FLIXSTER is from a social movie-rating site (`http://www.flixster.com/`). The dataset records movie ratings from users along with their timestamps. We use the topic-aware influence probabilities and the item-specific topic distributions provided by the authors of [13], who learned the probabilities using maximum likelihood estimation for the TIC model with $K = 10$ latent topics. In our quality experiments, we set the number of advertisers h to be 10, and used 10 of the learnt topic distributions from Flixster dataset, where for each ad i , its topic distribution $\vec{\gamma}_i$ has mass 0.91 in the i-th topic, and 0.01 in all others.

EPINIONS is a who-trusts-whom network taken from a consumer review website (`http://www.epinions.com/`). For Epinions, we similarly set $h = 10$ and use $K = 10$ latent topics. For each ad i, we use synthetic topic distributions $\vec{\gamma}_i$, by borrowing the ones used in FLIXSTER. For all edges and topics, the topic-aware influence probabilities are sampled from an exponential distribution with mean 30, via the inverse transform technique [53] on the values sampled randomly from uniform distribution $\mathcal{U}(0, 1)$.

In both datasets, advertiser budgets and CPEs are chosen in such a way that

	Budgets			CPEs		
Dataset	mean	min	max	mean	min	max
FLIXSTER	5700	2000	10,000	1.5	1	2
EPINIONS	6000	2000	10,500	1.9	1	2.5

Table 5.2: Advertiser budgets and cost-per-engagement values.

Algorithms	Revenue	Incentives	# Seeds	Runtime (min.s)
TI-CARM	42542.9	14251.4	840	2.8
TI-CSRM	42542.6	14371	867	2.9
CS-GREEDY-SRA ($W = 1$)	42571.2	14298.7	847	5.9
CS-GREEDY-SRA ($W = 10$)	42644.9	14206	883	15.1
CS-GREEDY-SRA ($W = 50$)	42973.5	13858.3	914	21.3

Table 5.3: Comparison on FLIXSTER dataset.

the total number of seeds required for all ads to meet their budgets is less than n. This ensures no ads are assigned empty seed sets.Table 5.2 contains a statistical summary of the budgets and CPEs.

For assigning ad-specific seed user incentives to nodes, we run MC simulations (10K runs) to obtain the singleton influence spread of each node per each ad, and multiply by $\alpha = 0.1$ to compute the seed user incentives.

For each of the considered algorithms, we evaluate the final revenue of their output seed sets using Monte Carlo simulations (10K runs) for neutral, fair, and accurate comparisons.

Table 5.3 summarizes our findings for FLIXSTER dataset. We had previously discussed that none of the influence maximization algorithms built on RIS framework [22, 112, 131, 132], as well as, our TI-CARM are capable of working as an influence spread oracle, thus, these algorithms would ignore less influential nodes. The indistinguishable results we obtain for TI-CARM and TI-CSRM experimentally verifies this behavior. On the other hand, although being *restricted* cost-sensitive, CS-GREEDY-SRA can obtain significantly higher revenue, with relatively lower money spent on seed user incentives, as it works as an influence spread oracle, and is mindful of the revenue obtained for the money spent. We can directly notice the increase in the revenue and decrease in the seed user incentives as the window applied to each advertiser increases. However, CS-GREEDY-SRA is not compatible with TI-CARM and TI-CSRM in terms of running time as it works as an influence spread oracle on the restricted set of nodes. We can see that the runtime significantly increases as we increase the window size W per each advertiser, while the revenue increases.

Table 5.4 summarizes our findings for EPINIONS dataset. Our results on EPINIONS dataset is similar to the trend we observe for the FLIXSTER dataset. Notice

Algorithms	Revenue	Incentives	# Seeds	Runtime (min.s)
TI-CARM	55464.4	3240.17	752	1.96
TI-CSRM	55468.9	3251.21	758	1.99
CS-GREEDY-SRA ($W = 1$)	55496.8	3246.65	755	18.3
CS-GREEDY-SRA ($W = 10$)	56116.9	3129.21	747	21.2
CS-GREEDY-SRA ($W = 50$)	56395.1	3141.17	757	31.4

Table 5.4: Comparison on EPINIONS dataset.

that, when $W = 1$, CS-GREEDY-SRA provides a cost-agnostic solution since it directly uses the node with the maximum marginal gain as in the case of TI-CARM. However, in both datasets, the runtime of CS-GREEDY-SRA even for $W = 1$ is not compatible with TI-CARM due to the marginal gain level estimation procedure it employs.

Our results on both datasets show that TI-CARM is a scalable and efficient alternative to CA-GREEDY. However, the total revenue obtained by TI-CARM is significantly outperformed by the total revenue obtained by the SEQ-RIS based CS-GREEDY-SRA that is capable of doing marginal gain level operations, even when using a *restricted* cost-sensitive node selection criteria. On the other hand, while the total revenue obtained by CS-GREEDY-SRA significantly increases as we inspect a higher window of nodes, the runtime of the algorithm also starts to increase. As we previously stated, for the problems in which the greedy node selection criteria does not follow the maximum coverage problem, like our cost-sensitive RM problem and the budgeted influence maximization problem [91,111], scalability still remains an open challenge: we are currently working on devising an efficient and scalable algorithm to find a *fully* cost-sensitive greedy solution to RM problem, by integrating the seed user incentives directly into the RIS estimation process, rather than limiting their presence to cost-sensitive node selection criteria.

5.6 Discussion and Future Work

In this work, we initiate investigation in the area of *incentivised* social advertising, by formalizing the fundamental problem of revenue maximization from the host perspective, when the incentives paid to the seed users are proportional to their demonstrated past influence in the topic of the specific ad. We show that, keeping all important factors, such as topical relevance of ads, their propensity for social propagation, the topical influence of users, users incentives and advertisers budgets in consideration, the problem of revenue maximization in incentivised social advertising is NP-hard and it corresponds to the problem of monotone submodular function maximization subject to a partition matroid constraint on the

ads-to-seeds allocation and submodular knapsack constraints on the advertisers' budgets. For this problem we devise two natural variants of the greedy algorithm w.r.t. their sensitivity to the advertisers' payment functions, and provide formal approximation guarantees. As scalability still remains an open challenge for devising algorithms that do not follow the greedy framework of influence maximization problem, like our CA-GREEDY and the budgeted influence maximization problem [91, 111], we provide a formal discussion on our ongoing efforts for devising an efficient and scalable version of CS-GREEDY that do not rely on computationally exhaustive MC simulations.

Our work takes a first step toward enriching the framework of incentivised social advertising by integrating it with powerful ideas from viral marketing and making the latter more applicable to real online marketing problems. It opens up several interesting avenues for further research. Capturing the auction dynamics of the real-world social advertising models by the integration of algorithmic mechanism design techniques to the allocation of ads, integrating hard competition constraints to the influence propagation process are directions that offer a wealth of possibilities for future work.

CHAPTER 6

CONCLUSIONS

In this thesis we develop techniques to take the algorithmic aspects of viral marketing out of the lab, and further enhance these aspects to account for the real world social advertisement models, by drawing on the viral marketing literature to study social influence aware ad allocation for social advertising. This chapter summarizes our contributions, open problems, and directions for future research.

6.1 Summary

Viral marketing, by "targeting" the most *influential* individuals, takes advantage of the networks of social influence to deliver a marketing message to a large portion of the social network. In addition to its popularity in the business literature, viral marketing has recently attracted substantial interest from the computer science community due to the fascinating computational challenges that it entails: influence maximization is the key algorithmic problem behind viral marketing, formally defined as a discrete optimization problem for the identification of the influential users [88].

Regardless the substantial research effort devoted to improve the efficiency and scalability of the influence maximization algorithms, their efficiency is still limited for applications that require milliseconds response time. Thus, in Chapter 3, we took a first step towards enabling social-influence online analytics in support of viral marketing decision making, and proposed an efficient influence indexing framework for a very general type of viral marketing queries: topic-aware influence maximization queries. Exploiting a tree-based index for similarity search in non-metric spaces, a clever approximate nearest neighbors search over the tree, and a weighted rank aggregation mechanism, we showed that our index can provide, in few milliseconds, a solution very similar to the one produced by

the standard offline influence maximization computation that usually takes hours or sometimes days, while achieving a similar expected influence spread.

Driven by the multi-billion dollar industry, the area of computational advertising has attracted a lot of interest during the last decade. Considerable work has been done in sponsored search and display advertising, mainly focusing on the central problem of finding the "best match" between a given user in a given context and a suitable advertisement. However, with the advent of social advertising, the standard interest-driven allocation of ads to users has become inadequate as it fails to leverage the potential of social influence. Thus, in Chapter 4, we initiate the investigation in the area of social advertising through the viral marketing lens. We assume a real-world business model in which the advertisers approach the host with a monetary budget, to pay for ad-engagements in return for the social advertising service provided by the host. In this context, we defined regret as the absolute value of the difference between the budget of an advertiser and the total cost paid by the advertiser to the host based on a cost-per-engagement pricing model, and formally studied the regret minimization problem for the allocation of ads under social influence. We showed that regret minimization problem is NP-hard and inapproximable w.r.t. any factor. However, we devised an algorithm that provides approximation guarantees w.r.t. the total budget of all advertisers. We also developed a scalable version of our approximation algorithm, which we extensively tested on four real-world data sets, confirming that our algorithm delivers high quality solutions, is scalable, and significantly outperforms several natural baselines.

In Chapter 5, we introduce the novel advertisement model of *incentivised* social advertising, where the users that are selected by the host to be the *seeds* for the campaign on a specific ad, can take a "cut" on the social advertising revenue. We assume a real-world business model in which an advertiser enters into a commercial agreement with the host to pay, following the cost-per-engagement pricing model, a fixed price per each engagement to his ad. In this context, we studied the fundamental problem of revenue maximization from the host perspective. We showed that the problem of revenue maximization for incentivised social advertising is NP-hard and it corresponds to the problem of monotone submodular function maximization subject to a partition matroid constraint on the ads-to-seeds allocation, and submodular knapsack constraints on the advertisers' budgets. We then devised two natural variants of the greedy approximation algorithm, based on their sensitivity to advertisers' payments, for which we provided formal approximation guarantees. We also presented experimental results and open problems.

6.2 Future Directions

While acknowledging that there are still a large number of problems that remain unexplored at the intersection of social influence propagation, viral marketing, and social advertising, in this thesis we took a first step to enable social influence analytics for viral marketing, and initiated the investigation on the area of social advertising through the viral marketing lens. Below we provide a discussion of directions that offer a wealth of possibilities for future work.

One immediate future direction regarding influence indexing frameworks is to study the automatic determination of the number of index items that is required for maintaining the accuracy of the framework. At the time of the publication of INFLEX [5], CELF++ [75] was the state-of-the-art influence maximization algorithm: on the problem instances that we consider in Chapter 3, the pre-processing step with CELF++ [75] took from few days to more than a week in order to extract a seed set of 50 nodes for a single item. Thus, our choice for the number of index items, query items, and the datasets to be used in the experimentation were limited, due to the extremely heavy computational burden of the standard influence maximization computation. However, with the latest advances on devising scalable influence maximization algorithms [22, 112, 131, 132], CELF++ can be directly replaced for the pre-computation of the seed sets, which would provide greater flexibility in choosing the number of index and query items, as well as, testing influence indexing techniques.

The latest advances on devising scalable influence maximization algorithms [22,112,131,132] provide very promising theoretical and practical results. However, their applicability to the problems that do not necessarily follow the greedy framework of the maximum coverage problem, as in the case of the budgeted influence maximization problem [91, 111], is still very limited. In Chapter 5, we have both theoretically and experimentally demonstrated the drawbacks of these influence maximization algorithms for finding a cost-sensitive greedy solution. Thus, further enhancing the Reverse Influence Sampling framework to efficiently and accurately solve the problems that deviate from the greedy framework of the maximum coverage problem is a promising future direction.

Another interesting direction is to adapt the influence indexing framework to handle the queries composed of *multiple items* in the presence of matroid, budgets, or hard competition constraints. Currently our INFLEX is designed for efficiently processing topic-aware influence maximization queries, where each query consists of a single item. Using a similar influence indexing framework adapted to handle the presence of multiple items in a single query, under budget and matroid constraints, would provide enormous efficiency for solving the social advertising problems we define in Chapters 4 and 5.

In Chapters 4 and 5, we initiated the research on the real world social adver-

tising models through the viral marketing lens to address the problems that viral marketing or computational advertising literature fail to address in isolation. However, to better capture the real world dynamics, we still have work to do. Currently, one of the important missing piece in these models is the auction-based dynamics of the real-world social advertising models. In our research, we assumed that the advertisers declare their preferred cost-per-engagement (CPE) value to the host, and the host charges this exact amount per engagement to their ads. However, real world social advertising platforms, such as Facebook and Twitter, implement ad auctions, which determine not only the allocation of ads, but also the CPE amount that advertisers should pay. The CPE value that advertisers declare is usually referred as their *valuation* regarding the maximum amount they are willing to pay to the host per engagement.

The determination of the price and allocation in an auction is widely studied under the name Algorithmic Mechanism Design [116]. Currently, within the context of incentivised social advertising that we study in Chapter 5, we are at the process of designing an *envy-free* revenue maximization mechanism: differently from our work in Chapter 5, we assume that the host runs ad auctions to determine both the ads-to-seeds allocation and the CPE for each advertiser based on the values they declare. We adopt the classic quasi-linear utility model for quantifying the advertisers' preferences and study the game-theoretic fairness notion of *envy-freeness*: the host should carefully determine the CPE and the seed set for each advertiser, with the rational goal of maximizing his revenue from the ad engagements, while ensuring that no advertiser envies another advertiser, *i.e.*, no advertiser can gain higher utility by exchanging his seed set and CPE assignment with another advertiser. At the moment, we have an envy-free revenue maximization mechanism, based on the dynamic programming formulation of envy constraints, for the case in which advertisers have uniform budgets. We will also be working on extending our results for the general case in which advertisers may have different budgets.

The design of envy-free and incentive compatible mechanisms lie at the intersection of computer science, economics, and game theory, and is widely studied by computational advertising researchers. Moreover, many real world online advertisement platforms, such as "Google's Sponsored Search" and "Facebook Ads", implement such mechanisms for operating their ad auctions. In addition to envy-freeness, many other game-theoretical aspects of ad allocation for social advertising offer a wealth of possibilities for future directions, such as *incentive compatible* mechanisms that ensure the *truthfulness* of advertisers in their CPE and budget declarations. Given the combinatorial nature of the problems we study in Chapters 4 and 5, introducing the real-world auction dynamics to these problems would require to handle combinatorial auctions in a virality-aware manner. However, this is not an easy task as the literature on combinatorial auctions is full

of impossibility results [128].

Lastly, the formal social advertising setting that we introduce in Chapters 4 and 5 provides tremendous directions for defining different combinatorial optimization problems that cannot be addressed by the influence maximization literature. As we studied in detail in this thesis, the classic model of treating the budgets in a viral marketing campaign does not model a real-world social advertising scenario: if the advertiser offers the same products to seed users as incentives in a viral marketing campaign, the advertiser's budget is modeled as a cardinality constraint on the number of free products to offer [88]. Alternatively, advertisers might choose to target the seed users non-uniformly, offering incentives of arbitrary costs, in which case the budget of an advertiser is modeled as a monetary amount that will be spend on the non-uniform costs of incentivizing seed users [91, 111]. On the other hand, in a social advertising campaign, the budget of an advertiser is used for paying the ad engagements, while these engagements might be due to the viral propagation of the ad. Moreover, in the case of an incentivised social advertising campaign, the budget is used for paying both the engagements and the seed user incentives. In this thesis, we defined two novel optimization problems, regret minimization and revenue maximization, addressing the inadequacies of the influence maximization literature for handling real-world social advertising scenarios. Building on our formal study, one can define many other interesting problems, such as finding the ads-to-seeds allocation that minimizes the money spent on seed user incentives, or minimizing the time for reaching a predefined engagement threshold. Finally, one of the most challenging but required piece is to design explore-exploit algorithms for these problems, along with theoretical guarantees, that can perform ads-to-seeds allocation in an online manner based on the real-time observed propagation traces.

BIBLIOGRAPHY

[1] Z. Abbassi, A. Bhaskara, and V. Misra. Optimizing display advertising in online social networks. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015*, pages 1–11, 2015.

[2] T. W. Anderson and D. A. Darling. Asymptotic theory of certain" goodness of fit" criteria based on stochastic processes. *The annals of mathematical statistics*, pages 193–212, 1952.

[3] S. Aral. Identifying social influence: A comment on opinion leadership and social contagion in new product diffusion. *Marketing Science*, 30(2):217–223, 2011.

[4] S. Aral and D. Walker. Creating social contagion through viral product design: A randomized trial of peer influence in networks. *Management science*, 57(9):1623–1639, 2011.

[5] C. Aslay, N. Barbieri, F. Bonchi, and R. A. Baeza-Yates. Online topic-aware influence maximization queries. In *EDBT*, pages 295–306, 2014.

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

[7] E. Bakshy, D. Eckles, R. Yan, and I. Rosenn. Social influence in social advertising: evidence from field experiments. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 146–161. ACM, 2012.

[8] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. Everyone's an influencer: quantifying influence on twitter. In *Proc. of the Forth Int. Conf. on Web Search and Web Data Mining (WSDM'11)*, 2011.

[9] L. Baltrunas, T. Makcinskas, and F. Ricci. Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 119–126. ACM, 2010.

[10] A. Banerjee, S. Merugu, I. Dhillon, and J. Ghosh. Clustering with Bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.

[11] H. Bao and E. Y. Chang. Adheat: An influence-based diffusion model for propagating hints to match ads. In *Proceedings of the 19th International Conference on World Wide Web*, WWW 10, 2010.

[12] N. Barbieri and F. Bonchi. Influence maximization with viral product design. SIAM, 2014.

[13] N. Barbieri, F. Bonchi, and G. Manco. Topic-aware social influence propagation models. In *2012 IEEE 12th International Conference on Data Mining*, pages 81–90. IEEE, 2012.

[14] J. Bartholdi, C. Tovey, and M. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and welfare*, 6(2):157–165, 1989.

[15] F. Bass. A new product growth model for consumer durables. management sciences. *Institute for Operations Research and the Management Sciences. Evanston, XV (5)*, 1969.

[16] J. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[17] J. L. Bentley. Multidimensional binary search trees in database applications. *IEEE Transactions on Software Engineering*, (4):333–340, 1979.

[18] S. Bharathi, D. Kempe, and M. Salek. Competitive influence maximization in social networks. In *International Workshop on Web and Internet Economics*, pages 306–311. Springer, 2007.

[19] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[20] S. Boltz, E. Debreuve, and M. Barlaud. High-dimensional statistical distance for region-of-interest tracking: Application to combining a soft geometric constraint with radiometry. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[21] J. C. Borda. Mémoire sur les élections au scrutin. 1781.

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

[23] A. Borodin, M. Braverman, B. Lucier, and J. Oren. Strategyproof mechanisms for competitive influence in networks. In *Proceedings of the 22nd international conference on World Wide Web*, pages 141–150. International World Wide Web Conferences Steering Committee, 2013.

[24] A. Borodin, Y. Filmus, and J. Oren. Threshold models for competitive influence in social networks. In *International Workshop on Internet and Network Economics*, pages 539–550. Springer, 2010.

[25] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[26] C. Budak, D. Agrawal, and A. El Abbadi. Limiting the spread of misinformation in social networks. In *WWW*, pages 665–674, 2011.

[27] R. S. Burt. Social contagion and innovation: Cohesion versus structural equivalence. *American journal of Sociology*, pages 1287–1335, 1987.

[28] B. Bustos and T. Skopal. Non-metric similarity search problems in very large collections. In *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, pages 1362–1365. IEEE, 2011.

[29] B. B. Cambazoglu, I. S. Altingovde, R. Ozcan, and Ö. Ulusoy. Cache-based query processing for search engines. *ACM Transactions on the Web (TWEB)*, 6(4):14, 2012.

[30] T. Carnes, C. Nagarajan, S. M. Wild, and A. Van Zuylen. Maximizing influence in a competitive social network: a follower's perspective. In *Proceedings of the ninth international conference on Electronic commerce*, pages 351–360. ACM, 2007.

[31] L. Cayton. Fast nearest neighbor retrieval for Bregman divergences. In *Proceedings of the 25th international conference on Machine learning*, pages 112–119. ACM, 2008.

[32] L. Cayton. Efficient Bregman range search. *Advances in Neural Information Processing Systems*, 22:243–251, 2009.

[33] P. Chalermsook, A. D. Sarma, A. Lall, and D. Nanongkai. Social network monetization via sponsored viral marketing. In *Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, 2015.

[34] E. Chávez, G. Navarro, R. Baeza-Yates, and J. L. Marroquín. Searching in metric spaces. *ACM Computing Surveys (CSUR)*, 33(3):273–321, 2001.

[35] S. Chen, J. Fan, G. Li, J. Feng, K.-l. Tan, and J. Tang. Online topic-aware influence maximization. *Proceedings of the VLDB Endowment*, 8(6):666–677, 2015.

[36] W. Chen, T. Lin, and C. Yang. Real-time topic-aware influence maximization using preprocessing. In *International Conference on Computational Social Networks*, pages 1–13. Springer, 2015.

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

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

[39] N. A. Christakis and J. H. Fowler. The spread of obesity in a large social network over 32 years. *New England journal of medicine*, 357(4):370–379, 2007.

[40] N. A. Christakis and J. H. Fowler. *Connected: The surprising power of our social networks and how they shape our lives*. Little, Brown, 2009.

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

[42] E. Cohen, D. Delling, T. Pajor, , and R. F. Werneck. Sketch-based influence maximization and computation: Scaling up with guarantees. In *CIKM*, 2014.

[43] J. S. Coleman, E. Katz, H. Menzel, et al. *Medical innovation: A diffusion study*. Bobbs-Merrill Company New York, NY, 1966.

[44] M. Condorcet. Éssai sur l'application de l'analyse à la probalité des décisions rendues à la pluralité des voix, 1785. *Zitiert auf den*, page 23, 1785.

[45] M. Conforti and G. Cornuéjols. Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the rado-edmonds theorem. *Discrete applied mathematics*, 7(3):251–274, 1984.

[46] A. Cont, S. Dubnov, and G. Assayag. On the information geometry of audio streams with applications to similarity computing. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):837–846, 2011.

[47] A. Copeland. A reasonable social welfare function. Technical report, mimeo, 1951. University of Michigan, 1951.

[48] D. Coppersmith, L. Fleischer, and A. Rudra. Ordering by weighted number of wins gives a good ranking for weighted tournaments. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 776–782. ACM, 2006.

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

[50] H. Daneshmand, M. Gomez-Rodriguez, L. Song, and B. Schoelkopf. Estimating diffusion network structures: Recovery conditions, sample complexity & soft-thresholding algorithm. In *ICML*, pages 793–801, 2014.

[51] S. Datta, A. Majumder, and N. Shrivastava. Viral marketing for multiple products. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 118–127. IEEE, 2010.

[52] N. R. Devanur, B. Sivan, and Y. Azar. Asymptotically optimal algorithm for stochastic adwords. In *EC*, pages 388–404, 2012.

[53] L. Devroye. Sample-based non-uniform random variate generation. In *Proceedings of the 18th conference on Winter simulation*, pages 260–265. ACM, 1986.

[54] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66. ACM, 2001.

[55] N. Du, Y. Liang, M. F. Balcan, and L. Song. Budgeted influence maximization for multiple products. *arXiv preprint arXiv:1312.2164*, 2014.

[56] N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha. Scalable influence estimation in continuous-time diffusion networks. In *Advances in neural information processing systems*, pages 3147–3155, 2013.

[57] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*, pages 613–622. ACM, 2001.

[58] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation revisited. *Manuscript*, 2(3):15, 2001.

[59] J. J. Egozcue, V. Pawlowsky-Glahn, G. Mateu-Figueras, and C. Barceló-Vidal. Isometric logratio transformations for compositional data analysis. *Mathematical Geology*, 35(3):279–300, 2003.

[60] R. Fagin, R. Kumar, and D. Sivakumar. Comparing top k lists. *SIAM Journal on Discrete Mathematics*, 17(1):134–160, 2003.

[61] R. Fagin, R. Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, SIGMOD '03, pages 301–312, New York, NY, USA, 2003. ACM.

[62] J. Feldman, M. Henzinger, N. Korula, V. S. Mirrokni, and C. Stein. Online stochastic packing applied to display ad allocation. In *ESA*, pages 182–194, 2010.

[63] J. Feldman, N. Korula, V. S. Mirrokni, S. Muthukrishnan, and M. Pál. Online ad assignment with free disposal. In *WINE*, pages 374–385, 2009.

[64] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions-ii. In *Polyhedral combinatorics*, pages 73–87. Springer, 1978.

[65] N. E. Friedkin. *A structural theory of social influence*, volume 13. Cambridge University Press, 2006.

[66] J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)*, 3(3):209–226, 1977.

[67] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[68] G. Goel and A. Mehta. Online budgeted matching in random input models with applications to adwords. In *SODA*, pages 982–991, 2008.

[69] J. Goldberger, S. Gordon, and H. Greenspan. An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 487–493. IEEE, 2003.

[70] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing letters*, 12(3):211–223, 2001.

[71] M. Gomez Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1019–1028. ACM, 2010.

[72] A. Goyal, F. Bonchi, L. V. Lakshmanan, and S. Venkatasubramanian. On minimizing budget and time in influence propagation over social networks. *Social Network Analysis and Mining*, 3(2):179–192, 2013.

[73] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan. Learning influence probabilities in social networks. In *WSDM*, pages 241–250, 2010.

[74] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan. A data-based approach to social influence maximization. *PVLDB*, 5(1):73–84, 2011.

[75] A. Goyal, W. Lu, and L. V. Lakshmanan. CELF++: optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on World wide web*, pages 47–48. ACM, 2011.

[76] A. Goyal, W. Lu, and L. V. S. Lakshmanan. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *ICDM*, pages 211–220, 2011.

[77] M. Granovetter. Threshold models of collective behavior. *American journal of sociology*, pages 1420–1443, 1978.

[78] M. S. Granovetter. The strength of weak ties. *American journal of sociology*, pages 1360–1380, 1973.

[79] A. Guttman. *R-trees: a dynamic index structure for spatial searching*, volume 14. ACM, 1984.

[80] G. Hamerly and C. Elkan. Learning the k in k-means. In *Advances in Neural Information Processing Systems*, volume 17, 2003.

[81] X. He, G. Song, W. Chen, and Q. Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. In *SDM*, pages 463–474, 2012.

[82] R. Iyengar, C. Van den Bulte, and T. W. Valente. Opinion leadership and social contagion in new product diffusion. *Marketing Science*, 30(2):195–212, 2011.

[83] R. Iyer. *Submodular Optimization and Machine Learning: Theoretical Results, Unifying and Scalable Algorithms, and Applications*. PhD thesis, University of Washington, Seattle, 2015.

[84] R. Iyer, S. Jegelka, and J. Bilmes. Fast semidifferential-based submodular function optimization. In *Proceedings of The 30th International Conference on Machine Learning*, pages 855–863, 2013.

[85] R. K. Iyer and J. A. Bilmes. Submodular optimization with submodular cover and submodular knapsack constraints. In *Advances in Neural Information Processing Systems*, pages 2436–2444, 2013.

[86] K. Jung, W. Heo, and W. Chen. Irie: Scalable and robust influence maximization in social networks. In *2012 IEEE 12th International Conference on Data Mining*, pages 918–923. IEEE, 2012.

[87] S. Jurvetson. What exactly is viral marketing. *Red Herring*, 78:110–112, 2000.

[88] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[89] M. Kimura, K. Saito, and H. Motoda. Efficient estimation of influence functions for SIS model on social networks. In *IJCAI*, pages 2046–2051, 2009.

[90] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5, 2007.

[91] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429. ACM, 2007.

[92] Y. Li, D. Zhang, and K.-L. Tan. Real-time targeted influence maximization for online advertisements. *Proceedings of the VLDB Endowment*, 8(10):1070–1081, 2015.

[93] C. X. Lin, Q. Mei, J. Han, Y. Jiang, and M. Danilevsky. The joint inference of topic diffusion and evolution in social communities. In *2011 IEEE 11th International Conference on Data Mining*, pages 378–387. IEEE, 2011.

[94] S. Lin, Q. Hu, F. Wang, and S. Y. Philip. Steering information diffusion dynamically against user attention limitation. In *2014 IEEE International Conference on Data Mining*, pages 330–339. IEEE, 2014.

[95] L. Liu, J. Tang, J. Han, M. Jiang, and S. Yang. Mining topic-level influence in heterogeneous networks. In *Proc. of the 19th ACM Conf. on Information and Knowledge Management (CIKM'10)*, 2010.

[96] W. Lu, F. Bonchi, A. Goyal, and L. V. Lakshmanan. The bang for the buck: fair competitive viral marketing from the host perspective. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 928–936. ACM, 2013.

[97] W. Lu, W. Chen, and L. V. Lakshmanan. From competition to complementarity: comparative influence diffusion and maximization. *Proceedings of the VLDB Endowment*, 9(2):60–71, 2015.

[98] W. Lu and L. V. Lakshmanan. Profit maximization over social networks. In *2012 IEEE 12th International Conference on Data Mining*, pages 479–488. IEEE, 2012.

[99] V. Mahajan, E. Muller, and F. M. Bass. New product diffusion models in marketing: A review and directions for research. In *Diffusion of technologies and social behavior*, pages 125–177. Springer, 1991.

[100] M. Mathioudakis, F. Bonchi, C. Castillo, A. Gionis, and A. Ukkonen. Sparsification of influence networks. In *KDD11*, pages 529–537, 2011.

[101] B. McFee and G. R. Lanckriet. Large-scale music similarity search with spatial trees. In *ISMIR*, pages 55–60, 2011.

[102] A. Mehta. Online matching and ad allocation. *Foundations and Trends in Theoretical Computer Science*, 8(4):265–368, 2013.

[103] A. Mehta, A. Saberi, U. V. Vazirani, and V. V. Vazirani. Adwords and generalized online matching. *J. ACM*, 54(5), 2007.

[104] T. Minka. Estimating a dirichlet distribution, 2000.

[105] V. S. Mirrokni, S. O. Gharan, and M. Zadimoghaddam. Simultaneous approximations for adversarial and stochastic online budgeted allocation. In *SODA*, pages 1690–1701, 2012.

[106] E. Mossel and S. Roch. On the submodularity of influence in social networks. In *STOC '07: Proc. of the thirty-ninth annual ACM symposium on Theory of computing*, pages 128–134, New York, NY, USA, 2007. ACM.

[107] S. A. Myers and J. Leskovec. Clash of the contagions: Cooperation and competition in information diffusion. In *2012 IEEE 12th International Conference on Data Mining*, pages 539–548. IEEE, 2012.

[108] R. Narayanam and A. A. Nanavati. Viral marketing for product cross-sell through social networks. In *Machine Learning and Knowledge Discovery in Databases*, pages 581–596. Springer, 2012.

[109] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions - i. *Mathematical Programming*, 14(1):265–294, 1978.

[110] P. Netrapalli and S. Sanghavi. Learning the graph of epidemic cascades. In *ACM SIGMETRICS Performance Evaluation Review*, volume 40, pages 211–222. ACM, 2012.

[111] H. Nguyen and R. Zheng. On budgeted influence maximization in social networks. *IEEE Journal on Selected Areas in Communications*, 31(6):1084–1094, 2013.

[112] H. T. Nguyen, M. T. Thai, and T. N. Dinh. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *Proceedings of the 2016 ACM SIGMOD International Conference on Management of Data*, 2016.

[113] F. Nielsen and R. Nock. Sided and symmetrized Bregman centroids. *Information Theory, IEEE Transactions on*, 55(6):2882–2904, 2009.

[114] F. Nielsen, P. Piro, and M. Barlaud. Bregman vantage point trees for efficient nearest neighbor queries. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pages 878–881. IEEE, 2009.

[115] F. Nielsen, P. Piro, M. Barlaud, et al. Tailored Bregman ball trees for effective nearest neighbors. In *Proceedings of the 25th European Workshop on Computational Geometry (EuroCG)*, pages 29–32, 2009.

[116] N. Nisan and A. Ronen. Algorithmic mechanism design. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, pages 129–140. ACM, 1999.

[117] N. Pathak, A. Banerjee, and J. Srivastava. A generalized linear threshold model for multiple cascades. In *2010 IEEE International Conference on Data Mining*, pages 965–970. IEEE, 2010.

[118] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 61–70. ACM, 2002.

[119] M. G. Rodriguez, J. Leskovec, D. Balduzzi, and B. Schölkopf. Uncovering the structure and temporal dynamics of information propagation. *Network Science*, 2(01):26–65, 2014.

[120] M. G. Rodriguez and B. Schölkopf. Influence maximization in continuous time diffusion networks. *arXiv preprint arXiv:1205.1682*, 2012.

[121] E. M. Rogers. *Diffusion of innovations*. Simon and Schuster, 2010.

[122] B. Ryan and N. C. Gross. The diffusion of hybrid seed corn in two iowa communities. *Rural sociology*, 8(1):15–24, 1943.

[123] K. Saito, M. Kimura, and H. Motoda. Discovering influential nodes for SIS models in social networks. In *International Conference on Discovery Science*, pages 302–316. Springer, 2009.

[124] K. Saito, M. Kimura, K. Ohara, and H. Motoda. Efficient estimation of cumulative influence for multiple activation information diffusion model with continuous time delay. In *Pacific Rim International Conference on Artificial Intelligence*, pages 244–255. Springer, 2010.

[125] K. Saito, R. Nakano, and M. Kimura. Prediction of information diffusion probabilities for independent cascade model. In *Proc. of the 12th Int. Conf. on Knowledge-Based Intelligent Information and Engineering Systems (KES'08)*, 2008.

[126] K. Saito, K. Ohara, Y. Yamagishi, M. Kimura, and H. Motoda. Learning diffusion probability based on node attributes in social networks. In *International Symposium on Methodologies for Intelligent Systems*, pages 153–162. Springer, 2011.

[127] F. Schalekamp and A. van Zuylen. Rank aggregation: Together we're strong. In *ALENEX*, pages 38–51, 2009.

[128] Y. Shoham and K. Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

[129] T. Skopal and B. Bustos. On nonmetric similarity search problems in complex domains. *ACM Computing Surveys (CSUR)*, 43(4):34, 2011.

[130] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 807–816. ACM, 2009.

[131] Y. Tang, Y. Shi, and X. Xiao. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1539–1554. ACM, 2015.

[132] Y. Tang, X. Xiao, and Y. Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. *SIGMOD*, 2014.

[133] M. Truchon. An extension of the Condorcet criterion and Kemeny orders. *Cahier*, 9813, 1998.

[134] C. Tucker. Social advertising. *Available at SSRN 1975897*, 2012.

[135] J. Uhlmann. Metric trees. *Applied Mathematics Letters*, 4(5):61–62, 1991.

[136] J. Uhlmann. Satisfying general proximity/similarity queries with metric trees. *Information processing letters*, 40(4):175–179, 1991.

[137] T. W. Valente. *Network Models of the Diffusion of Innovations (Quantitative Methods in Communication Series)*. Hampton Press (NJ)(January 10, 1995), 1995.

[138] J. Vondrák. Submodularity and curvature: the optimal algorithm. *RIMS Kokyuroku Bessatsu B*, 23:253–266, 2010.

[139] A. Wald. *Sequential analysis*. Wiley, 1947.

[140] C. Wang, R. Raina, D. Fong, D. Zhou, J. Han, and G. Badros. Learning relevance from heterogeneous social network and its application in online targeting. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, 2011.

[141] R. Weber, H. Schek, and S. Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the International Conference on Very Large Data Bases*, pages 194–205. INSTITUTE OF ELECTRICAL & ELECTRONICS ENGINEERS, 1998.

[142] D. P. Williamson and D. B. Shmoys. *The design of approximation algorithms*. Cambridge University Press, 2011.

[143] J. Woo, J. Son, and H. Chen. A SIR model for violent topic diffusion in social media. In *Intelligence and Security Informatics (ISI), 2011 IEEE International Conference on*, pages 15–19. IEEE, 2011.

[144] P. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, pages 311–321. Society for Industrial and Applied Mathematics, 1993.