STUDY IN BIG DATA HARNESSING AND RELATED PROBLEMS

by

Rong Jin

APPROVED BY SUPERVISORY COMMITTEE:

____

Weili Wu, Chair

____

Farokh B. Bastani

____

Latifur Khan

____

Xiaohu Guo

Dedicated

to

my beloved parents.

STUDY IN BIG DATA HARNESSING AND RELATED PROBLEMS

by

RONG JIN, BE, MS

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN

COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT DALLAS

August 2021

ACKNOWLEDGMENTS

First of all, I sincerely thank my advisor, Dr. Weili Wu, for her generous devotion on guidance, earnest encouragements, and strongest supports in every means throughout my PhD journey. This dissertation could not have been completed without her serving as the best advisor to me.

I really thank Dr. Ding-Zhu Du, who has also mentored me on various research problems in patience. His insightful and valuable suggestions are important for the line of research presented in this dissertation.

Thanks to Dr. Farokh B. Bastani, Dr. Latifur Khan, and Dr. Xiaohu Guo for being in my dissertation commitee members as well as for their great feedback and suggestions on my dissertation.

I want to thank the entire Data Communication and Data Management (DCDM) group members. Especially thank Dr. Qiufen Ni, Dr. Smita Ghosh, and Dr. Wenguo Yang for their collaborations. The outcome of the collaborations has made the very important pieces of this dissertation.

I also want to thank all professors and colleagues I have worked with in two research labs during my PhD studies, including Dr. Balakrishnan Prabhakaran, Dr. Kevin Desai, Dr. Suraj Raghuraman, and Dr. Uriel Haile Hernndez Belmonte in the Multimedia System Lab, and Dr. Modori Kitagawa, Dr. Paul Fishwick, Dr. Michael Kesden, Dr. Mary Urquhart, Dr. Rosanna Guadagno, Mike Tran, Aniket Raj, Baily Hale, and Ken Suura in the Creative Automata Lab. Their projects give me the chance of learning new technologies and helping me to identify my interested research areas to cultivate.

Thanks to the National Science Foundation and the Computer Science Department at UTD for their financial support of my work and scholarship.

Thanks to my friends who are always there when I need a company through all the ups and downs, success and trials I have faced along this path over these years. While I cannot list

them all here, I particularly want to mention Zheng Zhang, Zhen Cui, Rittika Shamsuddin, Yi Li, Ke Xu, Shuyang Gu, Yeqi Wang and Xiao Dong.

Finally, I deeply thank my parents. Without their love, understanding, and constant support, I would not have been able to make it this far. I thank them for always letting me live my life the way I want to.

June 2021

STUDY IN BIG DATA HARNESSING AND RELATED PROBLEMS

Rong Jin, PhD
The University of Texas at Dallas, 2021

Supervising Professor: Weili Wu, Chair

Social networks, such as Facebook and Twitter, have provided incredible opportunities for social communication between web users around the world. Social network analysis is an important problem in data harnessing. The analysis of social networks helps summarizing the interests and opinions of users (nodes), discovering patterns from the interactions (edges) between users, and mining the events that take place in online platforms. The information obtained by analyzing social networks could be especially valuable for many applications. Some typical examples include online advertisement targeting, viral marketing, personalized recommendation, health social media, social influence analysis, and citation network analysis. In this dissertation, we study two types of applications emerging from modern online social platforms in the view of social influence. One is influence maximization(IM) problem from a discount-based online viral marketing scenario, which aims at maximizing influence in the adoption of target products, and the other is online rumor source detection problem, in which the spread of misinformation is supposed to be minimized and the source is expected to be detected. We formulate them as set function optimization problems and design solutions with performance guarantees. In study of set function optimization, there is a challenge coming from the submodularity of objective function. That is, some of the practical problems are not submodular or supermodular, the existing greedy strategy cannot be directly applied to problems to get a guaranteed approximate solution. To solve those non-submodular and non-supermodular problems, one method called DS decomposition has been considered, in which given a set function, we decompose it to be representable as a difference between submodular functions. Based on this method, we further study a problem about how to find a DS decomposition efficiently and effectively. Then we propose a generalized framework that is made up of our novel algorithms under deterministic version and random version respectively to solve maximization of DS decomposition and show their performances under various combinatorial settings. In addition, we discuss our findings on the role of black-box, that has been an important component in study of computational complexity theory as well as has been used for establishing the hardness of problems, about its implied power and limitations in study of data-driven computation for proving solutions to some computational problems.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

Social networks have become major dissemination platforms for interchanging information and obviously promise to be fruitful data sources. With such *big* data being generated and collected, the analysis of network data has gained extremely wide attentions over the last two decades from researchers in computer science, social science, economics, and so on. The term *big* in this case does not signify just Volume, but also the Velocity, Variety and Veracity of data elements and sources, which is known as four V's of *big data*. If harnessed effectively and efficiently, these data can help to illuminate social problems and propel effective solutions.

Social network analysis is an important problem in data harnessing (or *data mining*). A critical step for analyzing social networks is to investigate into the process of information diffusion. Hence, various diffusion models have been studied to formulate how the information propagates in a social network since Kempe et al., [48] firstly defined the influence maximization (IM) problem on their proposed independent cascade (IC) model and linear threshold (LT) model respectively. For present purposes, we assume a social network as an interconnected set of individual people (i.e., nodes), and their interactions (i.e., edges) on social media (e.g., Facebook and Twitter), while a real social network is heterogeneous, where nodes are of different data types and can represent larger entities - communities of people connected by shared interests, roles, etc. Therefore, in our assumption, information diffusion can be treated as social influence diffusion in a social network. In general, there are two types of optimization problems on social influence. One is to seek maximal influence of positive information that can be described as the IM problem in social networks that has been found in many applications like product advertisement in viral marketing, active friending and content maximization, and the other is to minimize influence maximization of misinformation, which is also called misinformation countermeasures problem that has been studied for many applications, such as misinformation detection, misinformation source detection and rumor blocking. In this dissertation, we present our studies on the above-mentioned two kinds of influence optimization problems respectively in online social networks. All problems adopted the IC model in our work.

Optimization problems in social networks are known as submodular set function optimization whose objective functions have the "diminishing returns" property that the difference in the incremental value of the function that a single element makes when added to an input set decreases as the size of the input set increases. This property makes submodular functions suitable for many applications, including approximzation algorithms and game thoery (as function modeling user preferences), and so on. Submodularity of an objective function in the problem is a property that can be exploited algorithmically to obtain a good approximation to the problem. For example, the IM problem under the IC model can be solved approximately. However, many objective functions in practical problems are non-submodular, such as group profit maximization problem and rumor source detection problem, the greedy strategy cannot be directly applied to the problem to get a guaranteed approximate solution. To solve this non-submodular problem, we can consider a method called DS decomposition, because it has been proven true for that any non-submodular function could decompose as a difference of two submodular functions, specially the two submodular functions are monotone and non-decreasing [3]. However, how do we find the decomposition quickly? How do we solve DS decomposition efficiently and effectively? We will discuss it in this dissertation.

Regarding computational complexity of optimiztion problems, it is known that any generalization of an NP-hard problem is also NP-hard. For instance, we know that the IM problem under the IC model in a social network has been proven an NP-hard and computing the objetive function is $\#P$-hard, which is a special case of most formulated influence maximization problems such as profit maximization problems. Then, the formulated optimiation problem can be proved in a computational complexity class. Instead of computing in this way, can we obtain its solution by a data mining method, such as a machine learning method, in which the black box stores a large amount of data regarding solutions to problems through data training? Based on this idea, some observations are discussed in this dissertaion.

In Chapter 3, we study the problem of influence maximization discount allocation to a new online social network scenario where the nodes (i.e., users) and the edges (i.e., relationships) between nodes are determined but the states of edges between nodes are unknown. We can know the states of all the edges centered on a node only when it becomes active. To be specific, we consider a discount-based online viral marketing scenario, and we aim to minimize the discount cost that the marketer spends while ensuring the expected amount of users who adopt the target product in the end of influence diffusion process. In other words, the marketer needs to decide a set of initial users to offer their discounts through an online social network under the limited budget for cost of discounts. Investigating the details of this problem raises some questions. Which set of initial users should be selected to offer discounts? Whether these seed users are willing to accept the discount and to be influencers? How much should the discount be worth and be allocated by the marketer? These questions have been addressed in many studies as optimal selection problem. Different from previous work, we develop and model a new discount allocation for cost minimization problem through the classic IM problem. In this work, we adopt the IC model. We propose an online discount allocation policy to select seed users to spread the product information. The marketer initially selects one seed user to offer with a discount and observes whether the user accepts the discount. If the user adopts the discount, the marketer needs to observe how much discount has been accepted and how well this seed user contributes to the final adoption of the target product. The remaining seed users are chosen based on the feedback of diffusion results obtained by all previous selected seeds. We further propose two online discount allocation greedy algorithms under two different situations: uniform and non-uniform discounts allocation. In non-uniform discount allocation situation, we keep offering some pre-selected users with the discount starting from the lowest to the highest within a set of available options on discounts until the users accept the discount and become seed users, in this way marketers can save more compared to the method that marketers give free products to users. We present a theoretical analysis with bounded approximation ratios for the algorithms. Extensive experiments are conducted to evaluate the performance of the proposed online discount allocation algorithms on real-world online social networks datasets and the results demonstrate the effectiveness and efficiency of our methods.

We investigated into the problem of identifying rumor source, which is another important problem for rumor controlling in online social networks. In Chapter 4, we are first interested in surveying the existing work regarding rumor source detection problem. The prior works have seen various rumor diffusion models and rumor source estimator being assumed in detection of rumor source. We observe that very little previous review has been studied from these two perspectives on the rumor source detection problem in online social networks. Therefore, we firstly generalize currently three representative schemes of modeling the pattern of rumor propagation: the IC-based model, Epidemic-based models, and Learning-based models since their inception a decade ago. We further summarize three major existing schemes of rumor source estimator that have been constructed from one type of rumor diffusion models. We present a disscussion on the findings of three rumor spreading models. Next, in Chapter 5, we study a station-based approach for online rumor source detection in theoretical literature. We identify a single rumormonger based only on a set of observed infection stations under the Independent Cascade (IC) model in a complete snapshot of the network obtained at some time. We generate a k-station selection problem and develop a 2-approximation algorithm for it. Also, we derive that an estimator of the rumor source is chosen to be the root node associated with the infection path that most likely leads to the observed infection stations for tree networks. These results are well studied in this chapter.

In Chapter 6, we study a heat problem about how to design efficient and effective algorithms to solve maximization of DS decomposition in the set function optimizations. We propose a framework called Parameter Conditioned Greedy Algorithm which has a deterministic version and two random versions. To discuss in more details, this framework uses the difference with parameter decomposition function and combines non-negative condition. Besides, if we set the different parameters, the framework can return solutions with different approximation ratio. Also, we choose two special cases to show our deterministic algorithm gets $f(S_k) - (e^{-1} - c_g)g(S_k) \geq (1 - e^{-1})[f(OPT) - g(OPT)]$ and $f(S_k) - (1 - c_g)g(S_k) \geq (1 - e^{-1})f(OPT) - g(OPT)$ respectively for the cardinality constrained problem, where c_g is the curvature of monotone submodular set function. To speed up the deterministic algorithm, we introduce a random sample set which can represent optimal solution set as soon as possible. More importantly, it can also get the same approximation ratio as deterministic algorithm under expectation. Furthermore, for maximum DS decomposition without constraint, our another random algorithm gets $E[f(S_k) - (e^{-1} - c_g)g(S_k)] \geq (1 - e^{-1})[f(OPT) - g(OPT)]$ and $E[f(S_k) - (1 - c_g)g(S_k)] \geq (1 - e^{-1})f(OPT) - g(OPT)$ respectively. Because the Parameter Conditioned Algorithm is the general framework, different users can choose the parameters that fit their problems to get a better approximation.

In Chapter 7, we present several observations regarding the role of black box in the data-driven computation for solving computational complexity. Because black box has been an important tool in studying computational complexity theory and has been used for establishing the hardness of problems, we then discuss a frame of data-driven computation that has utilized black box as a tool for proving solutions to some computational problems.

The rest of the dissertation organization is as follows. Chapter 2 provides the preliminary background required for the problems in the following chapters. Chapter 8 concludes this dissertation and outlines a few open research themes in related fields.

CHAPTER 2

LITERATURE REVIEW

2.1 Social Networks and Notations

In this section, we define social networks and summarize some of the terminology that will be used in later chapters.

Formally, a social network is represented by a graph $G = (V, E)$, where vertices $v \in V$ represent users and E is the set of edges where each edge (u, v) represents relationship between users u and v. By convention, let $n = |V|$ and $m = |E|$. In this dissertation, we may use the term "node" and "user" interchangeably.

Neighbors. A vertex u is a neighbor of (or equivalently adjacent to) a vertex v in a graph $G = (V, E)$ if there is an edge $(u, v) \in E$. For a *directed* graph a vertex u is an *in-neighbor* of a vertex v if $(u, v) \in E$ and an *out-neighbor* if $(v, u) \in E$. We also say two edges or arcs are neighbors if they share a vertex.

Neighborhood. For an *undirected* graph $G = (V, E)$, the *neighborhood* $N_G(v)$ of a vertex $v \in V$ is its set of all neighbors of v, i.e., $N_G(v) = \{u|(u, v) \in E\}$. For a *directed* graph, let $N^+_G(v)$ indicate the set of out-neighbors and $N^-_G(v)$ indicate the set of in-neighbors of v. The neighborhood of a set of vertices $U \subseteq V$ is the union of their neighborhoods, e.g., $N^+_G(v) = \cup_{u \in U} N^+_G(u)$. If we use $N_G(v)$ for a directed graph, we mean the out neighbors. We will drop the subscript G when it is clear from the context which graph we are talking about.

Degree. The *degree* $d_G(v)$ of a vertex $v \in V$ in a graph $G = (V, E)$ is the size of the neighborhood $|N_G(v)|$. For directed graphs we use *in-degree* $d^-_G(v) = |N^-_G(v)|$ and *out-degree* $d^+_G(v) = |N^+_G(v)|$. Similar to neighborhood, the subscription of G will be dropped when it is clear from the context which graph it is reffered to.

Paths. A path in a graph is a sequence of adjacent vertices. More formally for a graph $G = (V, E)$, $Paths(G) = \{P \in V | 1 \leq i < |P|, (P_i, P_{i+1}) \in E\}$ is the set of all paths in G, where V indicates all (positive) length sequences of vertices. The length of a path is one less than the number of vertices in the path, i.e., it is the number of edges in the path. A *simple path* is a path with no repeated vertices. See the remark below, however.

Remark 2.1. *Some authors use the terms walk for path, and path for simple path. Even in this dissertation when it is clear from the context we will sometimes drop the simple from simple path.*

Distance. The *distance* $d_G(u, v)$ from a vertex u to a vertex v in a graph G is the shortest path, i.e., minimum number of edges, from u to v. It is also referred to as the *shortest path length* from u to v.

Eccentricity. The *eccentricity* of a graph G is the maximum distance over all pairs of vertices. That is, $e(v) = max\{d_G(v, w) : w \in V\}$.

Diameter. The *diameter* of a graph G is the maximum eccentricity among the vertices of G. Thus, $diam(G) = max\{e(v) : v \in V\}$.

Radius . The *radius* of a graph G is the minimum eccentricity among the vertices of G. Therefore, $radius(G) = min\{e(v) : v \in V\}$.

Cycles. In a directed graph a *cycle* is a path that starts and ends at the same vertex. A *simple cycle* is a cycle that has no repeated vertices other than the start and end vertices being the same. In an undirected graph a (simple) *cycle* is a path that starts and ends at the same vertex, has no repeated vertices other than the first and last, and has length at least three. As with paths, we may use cycles when we are talking about simple cycles.

Directed acyclic graphs. A directed graph with no cycles is a *directed acyclic graph* (DAG).

Trees and forests. An undirected graph with no cycles is a *forest* and if it is connected it is called a *tree.* A directed graph is a forest (or tree) when all edges are converted to undirected edges. A *rooted tree* is a tree with one vertex designated as the root. For a directed graph the edges are typically all directed toward the root or away from the root.

2.2 Influence Maximization under IC Model

This section will expose you the basic stochastic model of social infuence, i.e., the *Independent Cascade* (IC) model, and show how it can be used to find an influential set of nodes (also called *seeds*) to target in order to maximize the final adoption, i.e., the Influence Maximization problem.

2.2.1 Independent Cascade (IC) Model

The IC model was firstly introduced by Kempe et al.,[48] to model the dynamics of viral marketing and was inspired from the field of interacting particle systems. In this model, we start with an initial set S of active users. Each active user u has a single chance to activate each non-active neigbor $v \in N(v)$. But, the process of activation is deemed stochastic and succeeds with probability $p_{u,v}$ independently for each attempt. Hence, from an initial population of active users the activation process spreads in a cascading manner as newly activated users may activate new nodes that either previous attempts failed to activate or were not accessible before. To enable mathematical treatment of the model, we adopt an alternative view of the model utilizing the notion of reachability.

Definition 2.1 (Reachability). *Given a graph $G = (V, E)$ and a node u, define $R_G^E(u)$ the set of reachable nodes (including u) of V from u through the edges in E.*

In terms of the reachability of nodes via paths from the initial active set S, we can picture the process of a node u activating one of its neighbors v with probability $p_{u,v}$ as flipping a biased coin, and if it succeeds, that declares the edge *live*, otherwise declares the edge *blocked* (or *dead*). Moreover, without loss of generality, we use the *principle of deferred decision* and consider that all the coins are tossed before the process begins. Therefore, from the initial graph $G = (V, E)$, we get a graph $G = (V, E_{live})$ where we keep only live edges. Now, in this setting, all nodes that are reachable via a live path from the initial set S would become active when the cascade process terminated. This view is very helpful and will be used to solve our problem in Chapter 3.

Definition 2.2 (IC Model)**.** *Given a graph $G = (V, E)$ and non-negative edge probabilities $\{p_e\}_{e \in E}$, consider $\{X_e\}_{e \in E}$ independent uniform [0,1] random variables. Define the random set of active edges as $I = \{e \in E : p_e \leq X_e\}$. The IC model for the graph G and probabilities p defines for every initial set of active nodes S, the final set A of a function of the initial set S as $A_I(S) = \cup_{u \in S} R^I_G(u)$.*

We can think of $R^I_G(u)$ as the influence set of node u under random realization of edge activations I, where I is a random variable. From here on the graph G and edge probabilities are assumed implicitly given.

2.2.2 Influence Maximization (IM)

The end goal is to use the knowledge of the interactions to find a set of influential nodes. Because of the stochastic nature of the IC model, we consider the use of expectations to quantify the goodness of the initial set.

The problem, therefore, is given a social network, i.e., a set of nodes (users) and the edges (interactions) between them, to select the optimal "seed" of users to influence in the network, so that the expected number of active nodes is maximal for a "seed" set of size k after the activation process terminates.

Definition 2.3 (IM Problem)**.** *Given a graph $G = (V, E)$ and non-negative edge probabilities $\{p_e\}_{e \in E}$ and a positive integer k, the Influence Maximization problem asks for the initial set S of size k such that the total influence function for the IC model $\sigma(S) = \mathbb{E}[|A_I(S)|]$ is maximized.*

Theorem 2.1 (Kempe et al.,[48])**.** *The influence maximization problem is NP-hard for the IC model.*

2.2.3 Submodularity and IC Model

A key property of submodularity satisfied by the IC model will sidestep the hardness result and enable the algorithmic treatment of influence maximization (IM) problems.

Definition 2.4 (Submodularity)**.** *A set function $f\colon 2^V \to \mathbb{R}$ is called* submodular *if for all subsets $S \subseteq T \subseteq V$ and $u \in V \setminus T$, the following inequality holds:*

$$f(S \cap \{u\}) - f(S) \geq f(T \cap \{u\}) - f(T) \tag{2.1}$$

A function is called submodular if it satisfies the "diminishing returns" property. That is, the marginal gain by adding an element to a set S, it is at least as the marginal gain by adding an element to the superset T. In other words, the higher the ground value is, the smaller is the marginal gain of adding one element. Intuitively, submodularity is the set-function analog of concavity.

Definition 2.5 (Monotonicity)**.** *A set function f is* monotone nondecreasing *if for any two subsets S and T of V and $S \subseteq T \subseteq V$, we have:*

$$f(T) - f(S) \geq 0 \tag{2.2}$$

According to the Theorem 2.2 in Kempe et al.,[48] and the monotonicity of $A_I(S)$ in the Definition 2.2, we can have a consequence as follows.

Theorem 2.2. *The total influence function $\sigma(S)$ is monotone and submodular.*

2.2.4 Approximation Guarantee in the IC Model

Submodularity of the total influence function is a property that can be exploited algorithmically to obtain a good approximation to the IM problem. In particular, there is a hope that local optimal choices would result in good final spread. The following greedy hill-climbing algorithm proposed by Nemhauser et al.,[67, 28] approximates the optimum to within a factor of $(1 - 1/e)$, where e is the base of the natural logarithm.

Algorithm 2.1 Hill Climbing (Greedy) Algorithm

Input: a graph $G = (V, E)$, probabilities $\{p_e\}_{e \in E}$ and a positive integer k
Output: S_k

```
Initialize $S_0 \leftarrow \emptyset$
for $i = 1$ to $k$ do
    $s_i \leftarrow \arg\max_{u \in V \setminus S_{i-1}} \{\sigma(S_{i-1} \cup \{u\}) - \sigma(S_{i-1})\}$
    $S_i \leftarrow S_{i-1} \cup \{s_i\}$
end for
return $S_k$
```

Theorem 2.3 (Nemhauser et al.,[67, 28]). *The hill-climbing algorithm finds a set S such that $\sigma(S) \geq (1 - 1/e) \cdot \sigma(S^*)$, where S^* is the optimal set of cardinality k that maximizes the value of total influence function σ.*

2.3 DS Decomposition

In study of social networks, data mining, and machine learing, a lot of problems can be formulated into set function optimizations. The method of DS decomposition plays an important role in the set function optimization.

First, consider that Definition 2.6 is equivalent to Definition 2.4:

Definition 2.6. *A set function f: $2^V \to \mathbb{R}$ is submodular if for any subsets $S, T \subseteq V$, we have:*

$$f(S) + f(T) \geq f(S \cup T) + f(S \cap T) \tag{2.3}$$

As with Definition 2.5, f is monotone nondecreasing.

Next, the following theorem first proved that any set function can be expressed as the difference between two submodular set functions. Specially, the two submodular set function are monotone and nondecreasing.

Theorem 2.4 (Bilmes et al., [66, 41]). *Every set function $f : 2^V \to \mathbb{R}$ can be decomposed into the difference of two monotone nondecreasing submodular functions g and h, i.e., $f = g - h$.*

Definition 2.7 (Supermodularity)**.** *A set function* $f\colon 2^V \to \mathbb{R}$ *is* supermodular *if for any subsets* $S, T \subseteq V$*, we have:*

$$f(S) + f(T) \leq f(S \cup T) + f(S \cap T) \tag{2.4}$$

f *is monotone nondecreasing.*

Based on Definition 2.7, a variation of Theorem 2.4 was proved by [56] as follows.

Theorem 2.5. *Every set function* $f : 2^V \to \mathbb{R}$ *can be decomposed into the difference of two monotone nondecreasing supermodular functions* g *and* h*, i.e.,* $f = g - h$*.*

Although the proof of DS decomposition is constructive, in fact, it is still open whether there exists an efficient method to find a DS decomposition. This gives researchers a lot of chances to make their efforts on it, such as our related work in Chapter 6. Moreover, for a given set function, different DS decompositions may induce algorithms with different performances. Finding a DS decomposition for a given set function is very likely to be NP-hard.

2.4 Black Box

The term "Black Box" oftern appears in a variety of contexts within theoretical computer science, and happens to be extremely convenient to capture computations with restricted knowledge about or access to certain information. In its most basic form, a black box (known as an *oracle*) encodes a function f to which the computation may issue query x and get the response $f(x)$. We have no knowledge (or interest) on the implementation of f in the black box while f itself may be computationally hard or even not computable. From a programming perspective, this viewpoint is convenient when solving a problem using a subroutine for f that someone else has implemented and we are given its input-output specification only. From a theoretical perspective, the ability to efficiently solve a given computational problem g using an oracle to f may constitute a *reduction* from g to f, showing that computing g is not that harder than computing f. The most powerful example for black box reductions is in NP-completeness proofs. Nowadays, these ideas has surged in study of data-driven computation (see Chapter 7), machine learning, and other areas.

CHAPTER 3

DISCOUNT ALLOCATION FOR COST MINIMIZATION IN ONLINE SOCIAL NETWORKS[1]

3.1 Introduction

In recent years, Online Social Networks (OSNs) have gained popularity at a rapid pace and become an important part in our daily lives. People use OSN sites such as Twitter, Microblog, Facebook, and LinkedIn not only to stay in touch with friends but also to generate, spread and share various social contents. OSNs enable government agencies to post news and events as well as ordinary people to post contents from their own perspectives and experience.

As an important application of OSNs, viral marketing has become a focus of attention by many firms. It is an effective marketing strategy based on person-to-person recommendation within an OSN [47]. More and more firms promote their products through OSNs. We consider a problem that a firm has a new product they would like to advertise through OSNs. They want to offer some discounts to a set of initial users who will potentially introduce the new product to their friends. Only a limited number of users will get discounts because of the limited budget. So which set of initial users should be selected to provide the discount and whether they are willing to accept the discount to be active? How much discounts should be provided by the firm?

The above questions have been addressed in many studies researches [23], [73]. They formulate them as an optimal selection problem in which a good set of initial adopters, called seed users, is selected. A classical problem, called influence maximization [48], is to maximize the influence spread, i.e., the number (or expected number) of users who finally adopt the target product under influence initiated by seed users. In this chapter, we study

[1] ©2021. Reprinted, with permission, from Q. Ni, S. Ghosh, C. Huang, W. Wu, and R. Jin. Discount allocation for cost minimization in online social networks. Journal of Combinatorial Optimization 41, 213-233.

a minimization problem, i.e., to use the minimal cost for seeds to reach a certain level of the influence spread. For seed user selection, we propose an online discount allocation method, i.e., choosing seed users based on the observation of the previous seeds propagation results until a certain level of the influence spread is achieved.

Most of the existing research on influence maximization problem have strategies in three categories: zero-feedback model [48], full-feedback model [31] and partial feedback model [107]. The first one has to commit all the seed users at once in advance. The second one selects one seed or more seeds at a time and waits until the diffusion completes, then selects the next seed. The third one selects one seed or a batch of seeds and wait several slots but not the end of the propagating, which can balance the delay and performance tradeoff. Goyal et al., [32] show that most zero-feedback models may over-predict the actual spread. This model has no delay but poorer performance. We focus on the full-feedback model in discount allocation problem in online social networks. Instead of selecting all seeds at once in the influence maximization problem, we use greedy discount allocation policy to select one node at a time and offer him a discount, then we observe his state and how he propagates through the social networks. Based on the historical observation, we adaptively select the next seed user.

Currently, many influence diffusion models have been proposed, two most commonly used classical models are Independent Cascade (IC) and Linear Threshold (LT) models, which are proposed by Kempe et al., [48]. They prove that the expected number of influenced users, called *influence spread*, is monotone and submodular. They also propose a greedy algorithm to maximize the influence spread in the network. The maximization of the influence spread, i.e., the influence maximization, is NP-hard under IC and LT models. Tong et al., [88] and Wang et al., [96] showed that in some networks, the influence maximization is NP-hard under IC model while polynomial-time solvable in LT model. In this work, we adopt the Independent Cascade (IC) model.

We summarize the main contributions in this chapter as follows:

- We explore a new discount allocation scenario and use the online full-feedback setting to the discount allocation problem in the OSNs scenario. Then we divide the cascade into two stages: seed selection and information diffusion. Each selected initial user has to give feedback about whether he adopted the discount and the discount rate if he accepted the discount to become a seed. Each current decision depends on all the previous feedbacks and cascades.

- We introduce a new utility function which is to influence at least a certain number Q of people who adopt the product with the minimal cost in expectation, instead of maximizing the active users at the end of the diffusion process.

- We present two algorithms in uniform and non-uniform discount situations, respectively. In non-uniform discount allocation situation, we offer discount to selected users from lowest to highest in the discount rate set until the users become active, which saves the cost of firms comparing with the previous method that provides product to users for free.

- The performance guarantee is analyzed. An approximation ratio $\alpha ln(\frac{Q}{\beta})$ in uniform discount situation and an approximation ratio of the worst-case cost $\alpha(ln(\frac{Q}{\beta}+1))$ in non-uniform discount are got.

- We numerically validate the effectiveness of the proposed algorithm on real-world online social networks datasets.

The rest of this chapter is organized as follows. In Section 3.2 we begin by recalling some existing work. We introduce the problem description and influence diffusion model and process in Section 3.3. In Section 3.4, we propose the online full-feedback policy and present the greedy algorithm under uniform discount and non-uniform discount conditions.

We also give the theoretical proof of the algorithms in Section 3.5, and Section 3.6 presents the simulation results. Finally, the conclusion is presented in Section 3.7.

3.2 Related Work

In this chapter, we focus on using online discount allocation policy to get a minimum cost target with a certain number of market penetration of the product. Discount allocation of viral marketing in OSNs has been studied in many scenarios. Below we discuss recent related work on related topics.

Discount allocation in OSNs. Yang et al., [104] study the problem about what discount should be offered to users so that the expected number of adopted users is maximized within a predefined budget. They develop a coordinate descent algorithm and an engineering technology in practice. They illustrate that compared to the traditional influence maximization methods, continuous influence maximization can improve influence spread significantly. Abebe et al., [1] study how to make use of social influence when there is a risk of overexposure in viral marketing. They present a seeding cascade model that has benefits when reaching positive inclined customers and cost when reaching negative inclined customers. They show how it captures some qualitative phenomena related to overexposure. They provide a polynomial-time algorithm to optimally find a marketing strategy. Tang et al., [86] study the stochastic coupon probing problem in social networks. They adaptively offer coupons to some users and those users who accept the coupons will become seed users and then influence their friends. There are two constraints that have to be satisfied for a coupon probing policy that achieve the influence maximization: the set of coupons redeemed by users must meet inner constraints; the set of probed users must meet outer constraints. The proposed constant approximation policy for the stochastic coupon probing problem is suitable for any monotone submodular utility function. Yuan et al., [108] study the influence maximization discount allocation problem under both non-adaptive and adaptive policies.

They provide a limited budget of B to a set of initial users, and the target is to maximize the active users who adopt the product. They propose a greedy algorithm with a constant approximation ratio. Han et al., [35] study how to use influence propagation to optimize the 'pure gravy' of a marketing strategy. They consider that seed nodes only can be activated by the offered discounts probabilistically, and try to find discount allocation strategy to maximize the expected difference of revenue. They formulate this problem as a non-monotone and non-submodular optimization problem. A novel 'surrogate optimization' method and two randomized algorithms are presented. They prove the constant performance ratio for the proposed algorithms.

Feedback policy in OSNs. Salha et al.,[74] introduce a myopic partial observation policy in the influence maximization problem. The proposed optimal algorithm guarantees to provide a $(1-1/e)$-approximation ratio under a variant of the IC model. Yuan et al.,[107] also study the influence maximization problem under partial feedback model. They propose an α-greedy policy to capture the trade-off between delay and performance by adjusting the value of α. The algorithm guarantees a constant approximation ratio. Tong et al.,[90] consider the uncertainty of the diffusion process in real-world social networks because of high-speed data transmission and a large population of participants. They introduce a seeding strategy that seed nodes are only selected between spread rounds under the dynamic Independent Cascade model, this solution has a provable performance guarantee. An efficient heuristic algorithm is also provided for better scalability. Tang et al.,[85] study the optimal social advertising problem from the platform's perspective. Their goal is to maximize the expected revenue by finding the best ad sequence for each user. They integrate viral marketing into existing ad sequencing model and use zero-feedback and full-feedback ad sequencing policies to maximize the efficiency of viral marketing. Choi et al.,[15] study the problem of detecting the source of diffused information by the means of querying individuals. Two paid queries are asked: whether the respondent is the source or not; if not, which neighbor spreads the information to the respondent. The assumption is that respondents may lie. They design two kinds of algorithms: full-feedback and zero-feedback, which correspond to whether we adaptively select the next respondents based on respondents' previous answers. Their goal is to evaluate the budget to achieve the detection probability $1 - \delta$, $\forall\ 0 < \delta < 1$. Dhamal et al.,[22] focus on selecting seed users in multiple phases based on the observed historical diffusion under the IC model. They present a negative result but do not guarantee a better spread in more phases. They study the effect of diffusion in multiple phases on average and the standard deviation of the extent of diffusion, and how to reduce the uncertainty in diffusion with multiple phases. Singer et al., [84] survey the feedback based seeding methods for influence maximization. They discuss the algorithmic approaches, the friend paradox in random models and experiments on feedback based seeding. Han et al., [34] study the adaptive Influence Maximization(IM) problem, which selects seed users in batches of size b. They use full feedback strategy that the i-th batch can be selected after they observe the influence results of the $(i-1)$-th batch seed users. They propose two practical algorithms for $b = 1$ with an approximation ratio $1 - e^{(\xi - 1)}$ and $b > 1$ with $1 - e^{(\xi - 1 + 1/e)}$ approximation guarantee, where $\xi \in (0, 1)$. Tong et al.,[89] present a time-constrained adaptive influence maximization problem. They provide a $\frac{e^2-2}{e-1}$ lower bound on the adaptive gap for the time-constrained case. The adaptive gap measures the ratio between full feedback and the zero-feedback.

3.3 Network Model and Problem Formulation

3.3.1 The Network Model

In this model, the online social network is a directed graph $G(V, E)$, where each vertex in V is a person, and E is the set of social ties. In this model, the nodes and edges in the graph are deterministic, the states of edges between nodes are unknown. We can know the states of all the edges centered on that node through activating it. Assume that a marketer provides m discount rates $D = \{d_1, \cdots, d_m\}$ to each user for a product and the marketer offers a user only one discount at a time. We assume the discount as the amount taken off from the original price which are integers in our problem. We denote $c_v = d_i$ as that we provide discount d_i to user v. So in the graph, each user $v \in V$ has m choices for the discount rates but he only can accept one discount. We assume that each user v is independently associated with a discount adoption probability function $p_{vd_i} \in [0, 1]$, which models the probability that v accepts different discounts. Whether users would accept these discounts is uncertain. We assume that the adoption probability of any initial user is monotonically increasing with respect to discounts. So if $d_j \geq d_i$, $p_{vd_j} \geq p_{vd_i}$. The incoming neighbor set and the outgoing neighbor set of a node v are denoted as $N^-(v)$ and $N^+(v)$, respectively. Each edge $(u, v) \in E$ in the graph is associated with a probability $p_{uv} \in [0, 1]$ indicating the probability that node u independently influences node v once u has been influenced. If u activates v, the edge is in 'live' state, if u does not activate v successfully, the edge is in 'blocked' state. Influence can then spread from user u to his outgoing neighbors and so on according to the same process.

3.3.2 Diffusion Process

We compose the cascade into two phases: seeds selection phase and information diffusion phase.

In the seeds selection phase, we select some initial seed users to provide some discounts to each of them. After receiving these discounts, every user will decide whether to accept the discount or not. It is affected by factors such as users' preference for the products or the discount rates and so on. A user can only accept one discount at a time. If a user has accepted a discount d_i, it means that any larger discount is also acceptable for him. If a user rejects the highest discount, he will reject all offered discounts. We use $\phi(v)$ to define the active state of a user v. If a user v accepts the discount to become a seed, its state is active, and $\phi(v) = 1$, otherwise $\phi(v) = 0$.

In the information diffusion phase, every initial user who becomes the seed then has to propagate the product information to her neighbors across the social network under the Independent Cascade (IC) model. Independent Cascade (IC) model: each active seed user has a single chance to influence his uninfluenced neighbors with given probabilities. Each newly influenced user also has a single chance to influence his uninfluenced outgoing neighbors in the next step. If a node v has multiple newly activated incoming neighbors, their influences are sequenced in an arbitrary order. The diffusion completes until there is no new user who is influenced. The expected cascade of U denoted as $I(U)$, is the expected number of users influenced by seed set U.

We select one seed at a time and wait until the diffusion completes before selecting the next seed. We define the state $\psi(uv)$ as the state of edge (u, v), if v is activated by one of its incoming active neighbor u and edge(u, v) is 'live', $\psi(uv) = 1$, otherwise, edge (u, v) is 'dead', $\psi(uv) = 0$. So we can see that activating u will reveal the status of edge (u, v) (i.e., the value of $\psi(uv)$).

3.3.3 Problem Formulation

In the model, each user has two states: ϕ and ψ. We represent the user's state $\langle \phi, \psi \rangle$ as an *online realization*. In the full feedback model, every initial user needs to give feedback whether or not he adopted the discount. After activating u, we observe the set of out-edges from node u that become active or 'live'. These active nodes are those who are successfully activated by node u. After each pick, our observations so far can be represented as an *online partial realization*. We use the notation $dom(\phi, \psi)$ to refer to the domain of $\langle \phi, \psi \rangle$ (i.e., the set of states observed in $\langle \phi, \psi \rangle$). In the online seed selection policy, we choose the current user dynamically, it is up to the current state of observation $dom(\phi, \psi)$.

We define our online discount allocation strategy for picking users as a policy $\pi(\langle \phi, \psi \rangle)$, which is a function from a set of partial realization to current observation, specifying which user to probe in the next step under the known online partial realization and the resulting cascade. So we choose the next user to probe based on what seeds we have detected so far, whether they accept the discount or not, and their feedbacks about the network diffusions.

Let Φ and Ψ denote a random realization of ϕ and ψ. We assume that there is a known prior probability distribution $p(\phi) := P[\Phi = \phi]$ (and $p(\psi) := P[\Psi = \psi]$ resp.) over online seeding realization (and online diffusion realization resp.). Given an online realization $\langle \phi, \psi \rangle$, let $\pi(\phi, \psi)$ denote all users picked by policy π under online realization $\langle \phi, \psi \rangle$. $c(\pi)$ denotes the total amount of discounts that have been delivered by π under $\langle \phi, \psi \rangle$. The policy π terminates (stops probing users) upon online observation $\langle \phi, \psi \rangle$ until the resulting expected spread is above a given threshold.

We denote $S(\pi, \psi) \subseteq V$ as the seed node set that has been selected by π under realization ψ. The expected cascade of a policy π is defined as: $f(\pi) = \mathbb{E}[f(\pi; \Phi, \Psi)]$. Marketers want to get as many people as possible to buy the products at as low cost as possible. We specify a threshold Q of the expected cascade that we would like to obtain, and try to find the cheapest policy to achieve that spread goal. The policy π is to minimize the expected cost of the marketer under all possible online realization, and at least Q is achieved. We define the expected total cost $c(\pi) = \mathbb{E}[c(\pi; \Phi, \Psi)]$ we would like. The problem we want to solve can be described as follows:

Online Minimal Cost Target Seed Selection Problem: Given a directed probabilistic network $G = (V, E)$, the spread threshold is $Q > 0$, which is a certain fraction of the size of the OSNs. Find the optimal policy π^* that leads to the minimum cost, i.e., $\pi^* = arg \min\{c(\pi^*) | f(\pi^*) \geq Q\}$.

We use a toy social network in Fig. 3.1 to illustrate our online discount allocation policy π. There are six users $V = \{a, b, c, d, e, f\}$. The propagation probabilities are on the edges. The graph is unknown in advance, it can be partially revealed after some nodes become active. Our expected spread Q is 5. Possible discount set is $D = \{1, 2\}$. Assume that we

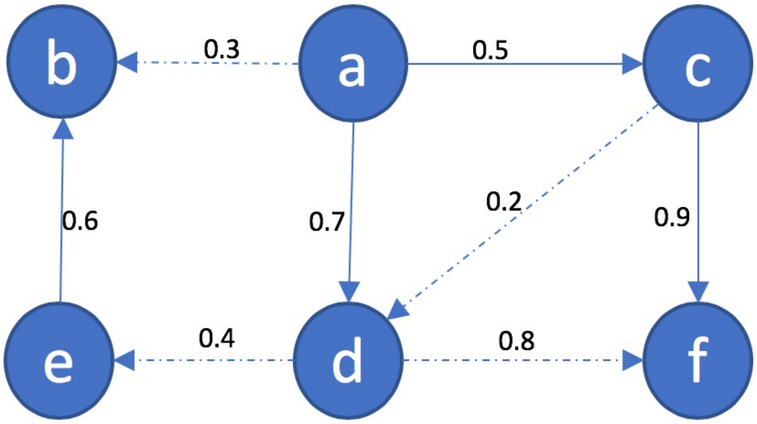

Figure 3.1: Illustration of a policy π.

select node a as the first seed node and set $\pi(\emptyset) = a$, which means that we probe $(a, 1)$ firstly, i.e., offering discount 1 to user a, and we observe the realization $\langle \phi, \psi \rangle$ of user a, $\phi(a, 1)$=1, i.e., a accepts the offer and become a seed; $\psi(ab) = 0$, $\psi(ac) = 1$, $\psi(cf) = 1$, $\psi(ad) = 1$, i.e., node c, f, d are influenced by seed a, b has not been influenced by a and e has not been influenced by d. In the graph, we use solid line (dotted line) to represent that a node successfully (resp. unsuccessfully) influences its outgoing neighbor nodes, that is to say, solid lines represent active edges and dotted lines represent inactive edges. After observing the state of edges of the active nodes, we can decide our policy in the next step: $\pi(\langle a, 1 \rangle) = e$. Because in this graph, only nodes e and b are inactive, we firstly offer discount 1 to e. We observe that $\phi(e, 1) = 0$, which means that e does not accept the discount and he is still inactive. As $\pi(\langle e, 0 \rangle) = b$, in the third step, we can probe user b with discount 1, b accepts the offer, i.e., $\phi(b, 1) = 1$. When b is active, he can not influence his neighbors since he does not have any outgoing neighbors, then the number of influenced users is 5 which achieved our threshold spread 5 and the cost is 2 eventually. But if we use the zero-feedback setting, we may choose nodes a and d as seed users in advance since these two nodes have the maximum outgoing neighbors, in which case we can only influence 4 nodes at most.

3.4 The Online Discount Allocation Greedy Policy

Before presenting our algorithms, we introduce the definition of the conditional expected marginal benefit of a seed.

Definition 3.1. *Given an online partial realization ψ and a user v, the conditional expected marginal benefit of v conditioned on having observed ψ is denoted by $\Delta(v|\psi)$ as*

$$\Delta(v|\psi) := \mathbb{E}[f(dom(\psi) \cup v) - f(dom(\psi)]$$

where the expectation is computed with respect to $p(\psi) := [\Psi = \psi]$.

In our seeds selection condition, $\Delta(v|\psi)$ quantifies the expected amount of additional increment by adding a seed user to the seed set to propagate the product information, in expectation over the posterior distribution $p(\Psi) := \mathbb{P}[\Psi]$ of how many users he will influence if he becomes a seed.

Definition 3.2. *Function f is online monotone with respect to distribution $p(\psi)$ if conditional expected marginal benefit of any seed is non-negative, i.e., for all ψ with $\mathbb{P}[\Psi \sim \psi] > 0$ and all $v \in V$ we have*

$$\Delta(v|\psi) \geq 0$$

Definition 3.3. *A function f is online submodular with respect to distribution $p(\psi)$ if for all ψ, and ψ' such that ψ is a sub-realization of ψ', and for all $v \in V \backslash dom(\psi')$, we have*

$$\Delta(v|\psi) \geq \Delta(v|\psi')$$

Assume π_{ag} is the online greedy policy which selects the node $v \in V \backslash dom(\psi)$ with the largest expected marginal gain $\Delta(v|\psi)$ under the given partial realization ψ, π_{opt} is the online optimal policy. Golovin et al., [31] proved that, the expected spread function f under edge level full feedback is monotone and submodular w.r.t. $p(\psi)$, and also prove that π_{ag} is an $(1-1/e)$- approximation of the adaptive optimal policy π_{opt}, $f_{avg}(\pi_{ag}) \geq (1-1/e)f_{avg}(\pi_{opt})$.

3.4.1 Greedy Policy under Uniform Discount

First, we focus on the case with uniform discount $c_v \equiv d$, $\forall v \in V$ in Algorithm 3.1. When the cost is uniform, we just need to select minimal number of seeds that can achieve the minimal sum cost while ensuring that a sufficient value $f \geq Q$ is obtained.

Algorithm 3.1 Uniform Discount Greedy Policy

```
Input: threshold $Q$, objective function $f$, prior distribution $p(\phi)$, $p(\psi)$, uniform discount $d$.
Output: $S$
$S = \emptyset$
select $v^* = arg\ max_{v \in V \backslash S} \Delta(v|\psi)$;
offer discount d to $v^*$
if $v^*$ accept the discount then
    $S \leftarrow S \cup v^*$;
    update the diffusion realization to $\psi_{t+1}$;
    if $f \geq Q$; then break;
    else continue;
    end if
else continue;
end if
return $S$
```

Given the observation of the current state $dom(\phi, \psi)$, we use $\Delta(v|\psi)$ to denote the expected marginal benefit of v in $V \backslash dom(\psi)$ where the condition is that v has been the seed. The greedy policy π myopically probes the seed which can get the maximal expected marginal gain ($v^* = arg\ max_{v \in V \backslash dom(\psi)} \Delta(v|\psi)$) at each iteration based on $dom(\phi, \psi)$. If there are multiple users in an iteration at the same time to obtain the maximal expected marginal gain, we just need to randomly choose one. If v^* accepts the discount, we remove v^* from V in the following round and add v^* to set S which is the selected seed set. Otherwise, if v^* rejects the discount, we put v^* aside and don't consider it any more. We have to update the new diffusion realization when a new seed is selected and completes his spread. This process iterates until reaching the spread threshold of Q. Our online greedy algorithm under the uniform discount is as Algorithm 3.1.

3.4.2 Greedy Policy under Non-uniform Discount

Algorithm 3.2 Non-uniform Discount Greedy Policy

```
Input: threshold $Q$, objective function $f$, prior distribution $p(\phi)$, $p(\psi)$, the discount rates
    $D = \{d_1, \cdots, d_m\}$.
Output: $S$
$S = \emptyset$
$i = 1$;
while $i \leq |D|$ do
    select $v^* = arg\ max_{v \in V \backslash S} \Delta(v|\psi)/d_i$;
    offer discount $d_i$ to $v^*$
    if $v^*$ accept discount $d_i$ then
        $S \leftarrow S \cup v^*$;
        update the diffusion realization to $\psi_{t+1}$;
        if $f \geq Q$; then break;
        else continue;
        end if
    else
        if $i = |D|$ then
            remove $v^*$ from $V$;
        else $i + +$;
        end if
    end if
end while
return $S$
```

In Algorithm 3.2, we consider the case when the cost is not uniform, and assume that the discounts are sorted in ascending order: $\forall 0 < i < j < |D|$: $d_i < d_j$. First, we probe user v^* who has the largest ratio of conditional expected marginal benefit to cost ($v^* = arg\ max_{v \in V \backslash S} \Delta(v|\psi)/d_i$) when offering him the lowest discount to each user. If there are multiple users in an iteration at the same time to obtain the maximal ratio of conditional expected marginal benefit to cost, we just randomly choose one. If he accepts this discount, we add v^* to selected seed set S, and then update the diffusion realization. If the influence can reach Q, we can terminate the selection of the seed. If $f < Q$, we have to go on selecting seeds. If v^* rejects the discount d_1, we can increase the discount rate. If the maximal discount can not be accepted by v^*, we simply don't consider it any more. The entire procedure will be repeated until function f obtains the value of Q.

3.5 Performance Analysis

Theorem 3.1. *When the discount is uniform and $c_v \equiv d$, let $\beta > 0$ be the spread shortfall for the online greedy discount allocation policy over its optimal discount allocation spread. The optimal discount allocation policy π^* achieves a target spread Q. The online greedy discount allocation policy π which is an α-approximate greedy policy and can achieve $Q - \beta$. Then we have the following relation between the cost of online greedy discount allocation policy $c_{avg}(\pi)$ and the cost of optimal discount allocation policy $c_{avg}(\pi^*)$*

$$c_{avg}(\pi) \leq \alpha c_{avg}(\pi^*) ln(\frac{Q}{\beta})$$

Theorem 3.2. *When the discount is non-uniform, let $\beta > 0$ be the spread shortfall for the online discount allocation greedy policy over its optimal discount allocation spread. Let π^*_{wc} be the optimal policy minimizing the worst-case cost c_{wc} while guaranteeing the maximum possible expected spread $f(\pi^*) = Q$. Let π be an α-approximate greedy policy, run until it achieves $f(\pi) \geq Q - \beta$. Then*

$$c_{wc}(\pi) \leq \alpha c_{wc}(\pi^*_{wc})(ln(\frac{Q}{\beta}) + 1)$$

Definition 3.4. *A policy π is an α-approximate greedy policy if $\Delta(v|\psi) > 0$, $\exists v \in V$ under online partial realization ψ,*

$$\pi(\psi) \in \{v : \frac{\Delta(v|\psi)}{c(v)}\} \geq \frac{1}{\alpha} \max_{v'}(\frac{\Delta(v'|\psi)}{c(v')})$$

π will terminate when observing a ψ has a negative conditional expected marginal benefit, that is $\Delta(v|\psi) \leq 0$ for all $v \in V$. An α-approximate greedy policy always obtains at least (1/α) of the maximal possible ratio of conditional expected marginal benefit to cost. It terminates when no more benefits can be obtained in expectation.

Lemma 3.1. *Suppose we have made online observations $\langle \phi, \psi \rangle$. Let π^* be any policy. Then for online greedy monotone submodular f:*

$$\Delta(\pi^*, \psi) \leq c(\pi^*|\psi) \max_v (\frac{\Delta(v|\psi)}{c(v)})$$

Proof. Consider policy π that attempts to select $v \in dom(\psi)$, terminating if $\Psi(v) \neq \psi(v)$, and then executing policy π^*. p_{vd_i} is the probability that v accept discount d_i and become a seed when running π, the online partial realization ψ' contains ψ as an online subrealization. By submodularity it implies $\Delta(v|\psi') \leq \Delta(v|\psi)$. So the total contribution of v to $\Delta(\pi^*|\psi)$ is upper bounded by $p_{vd_i}\Delta(v|\psi)$. By summing all $v \in V \backslash dom(\psi)$, we can get bound that

$$\Delta(\pi^*|\psi) \leq \sum_{v \in V \backslash dom(\psi)} p_{vd_i} \Delta(v|\psi)$$

For each $v \in V \backslash dom(\psi)$ contributes $p_{vd_i} c(v)$ cost to $c(\pi^*|\psi)$. So we have,

$$\sum_{v \in V \backslash dom(\psi)} p_{vd_i} c(v) \leq c(\pi^*|\psi)$$

Therefore,

$$\begin{aligned}\Delta(\pi^*|\psi) \leq \sum_{v \in V} p_{vd_i} \Delta(v|\psi) &\leq \sum_{v \in V} p_{vd_i} c(v) \max_{v \in V} \frac{\Delta(v|\psi)}{c(v)} \\ &\leq c(\pi^*|\psi) \max_{v \in V} \frac{\Delta(v|\psi)}{c(v)}\end{aligned}$$

□

Theorem 3.3. *Fix any $\alpha \geq 1$ and the discount that v accepts. Let $\pi^* \in arg \max_\pi f_{avg}(\pi_{[k]})$, where k is the number of selected seeds. If f is monotone and submodular with respect to the distribution $p(\psi)$, and π is an α-approximate greedy policy, then for all policy π^*, and positive integers h and l, we have*

$$f_{avg}(\pi_{[h]}) > (1 - e^{-h/\alpha l}) f_{avg}(\pi^*_{[l]})$$

Proof. The proof goes along the lines of the performance analysis and is an extension analysis for the α- approximate greedy algorithm. We assume without loss of generality that $\pi = \pi_{[h]}$ and $\pi^* = \pi^*_{[l]}$. Then for any $0 < i < l$, we can derive that:

$$f_{avg}(\pi^*) \leq f_{avg}(\pi_{[i]}) + \alpha l(f_{avg}(\pi_{[i+1]}) - f_{avg}(\pi_{[i]})) \tag{3.1}$$

This inequation follows from the monotonicity of f and Lemma 3.1. From Lemma 3.1 we also can get that

$$\mathbb{E}[\Delta(\pi^*|\psi)] \leq \mathbb{E}[c(\pi^*|\psi)] \max_v(\frac{\Delta(v|\psi)}{c(v)}),$$

since π^* has the form $\pi_{[k]}$ and k is the number of selected seeds. We have $\mathbb{E}[c(\pi^*|\psi)] \leq l$ for all ψ. It follows that $\mathbb{E}[\Delta(\pi^*|\psi)] \leq l \max_v(\frac{\Delta(v|\psi)}{c(v)})$. By the definition of an α-approximate greedy policy, π obtains at least $\frac{1}{\alpha} \max_v(\frac{\Delta(v|\psi)}{c(v)}) \geq \mathbb{E}[\Delta(\pi^*|\psi)]/\alpha l$ expected marginal benefit per unit cost in a step immediately following its observation of ψ.

Based on the monotonicity equivalence property in [31], given two policies π_1 and π_2, $\pi_1 @ \pi_2$ defined as the policy that after running π_1 then running policy π_2 ignoring the information gathered during the running of π_1. We can get that $f_{avg}(\pi_1) \leq f_{avg}(\pi_1 @ \pi_2)$.

Then, for a random partial realization Ψ, we can get the following inequality:

$$\begin{aligned} f_{avg}(\pi_{[i+1]}) - f_{avg}(\pi_{[i]}) &\geq \mathbb{E}[\frac{1}{\alpha} \max_v(\frac{\Delta(v|\Psi)}{c(v)})] \\ &\geq \mathbb{E}[\frac{\mathbb{E}[\Delta(\pi^*|\Psi)]}{\alpha l}] \\ &= \frac{f_{avg}(\pi_{[i]} @ \pi^*) - f_{avg}(\pi_{[i]})}{\alpha l} \end{aligned}$$

Now define $\Delta_i := f_{avg}(\pi^*) - f_{avg}(\pi_{[i]})$, so that inequation 3.1 implies $\Delta_i \leq \alpha l(\Delta_i - \Delta_{i+1})$, we can infer that $\Delta_{i+1} \leq (1 - \frac{1}{\alpha l})\Delta_i$ and hence $\Delta_h \leq (1 - \frac{1}{\alpha l})^h \Delta_0 \leq e^{-h/\alpha l}\Delta_0$, where for the second inequality we used the fact that $1 - x < e^{-x}$ for all $x > 0$. Hence $f_{avg}(\pi^*) - f_{avg}(\pi_{[h]}) < e^{-h/\alpha l}(f_{avg}(\pi^*) - f_{avg}(\pi_{[0]})) \leq e^{-h/\alpha l} f_{avg}(\pi^*)$ so $f_{avg}(\pi) > (1 - e^{-h/\alpha l}) f_{avg}(\pi^*)$. □

Proof. of Theorem 3.2. Let $\beta > 0$. Assume that l is the least seeds number that adaptive optimal policy π^* selected to achieve influence spread $f_{avg}(\pi^*) \geq Q$. Then running adaptive greedy policy π for h seeds, as $f_{avg}(\pi) \geq Q - \beta$, apply these two parameters to Theorem 3.3, we can get $h = \alpha l(lnQ/\beta)$. Let $l = c_{wc}(\pi^*_{wc})$, and apply parameters h and l to Theorem 3.3 we can get the following inequation

$$f_{avg}(\pi_{[h]}) \geq (1 - \frac{\beta}{Q}) f_{avg}(\pi^*_{[wc]}) \tag{3.2}$$

As π^*_{wc} can cover all the realization, $f_{avg}(\pi^*_{wc}) = \mathbb{E}[f(V(\pi, \psi), \Psi)] = Q$. The inequation 3.2 can denote as $Q - f_{avg}(\pi_{[h]}) \leq \beta$. Since f is monotonicity, $f_{avg}(\pi_{[h]}) \leq f_{avg}(\pi_{[h\rightarrow]})$, we can get that $Q - f_{avg}(\pi_{[h\rightarrow]}) \leq \beta$, this can infer that $Q - f_{avg}(\pi_{[h\rightarrow]}) = 0$, so $\pi_{[h\rightarrow]}$ covers every realization. It is known that π is an α-approximate greedy policy and π^*_{wc} covers all the online realizations at most l cost. From Lemma 3.1, we have:

$$\max_{v \in V} \frac{\Delta(v|\psi)}{c(v)} \geq \frac{\Delta(\pi^*_{wc}|\psi)}{c(\pi^*_{wc}|\psi)} \geq \frac{\Delta(\pi^*_{wc}|\psi)}{l} \tag{3.3}$$

We suppose that $\psi \in dom(\pi)$, we have $max_v \Delta(v|\psi) \leq \Delta(\pi^*_{wc}|\psi)$ as f is monotone submodular. Any user v with cost $c(v) > \alpha l$ also has $max_v \Delta(v|\psi)/c(v) \leq \Delta(\pi^*_{wc}|\psi)/\alpha l$, it can not be selected by any α-approximate greedy policy after observing ψ by inequation 3.3. The final user executed by $\pi_{[h\rightarrow]}$ has cost at most αl for any realization. So that $\pi_{[h\rightarrow]}$ has worst-case cost at most $h + \alpha l$, where $l = c^*_{wc}$. This completes the proof. □

Proof. of Theorem 3.1. When the cost $c_v = d$ for all seeds, the the number of seeds multiplies d equals the sum cost. So we conduct some algebraic manipulation in Theorem 3.3 to set $f_{avg}(\pi_{[h]}) = Q - \beta$, $f_{avg}(\pi^*_{[l]}) = Q$, we can get the conclusion of Theorem 3.1. □

3.6 Performance Evaluation

In this section, we will show the simulation of our proposed algorithms. The preparation of the experiment will be discussed in Section 3.6.1 with the datasets and parameter settings.

Table 3.1: Statistics of two datasets.

Dateset	**Nodes**	**Edges**	**Type**
Forum network	899	142760	directed
Newmans scientific collaboration network	16726	95188	directed

3.6.1 Experimental Setup

Datasets: In this chapter, we have used two datasets from [70], [68]. One of the dataset is a Forum Network which was collected from an online community very similar to the Facebook online social network. This dataset contained records of users activities in the forum. It is a collection of nodes and edges that depict the relationship among the 899 users. The other dataset is the Newmans scientific collaboration network which represents the co-authorship network based on preprints posted to Condensed Matter section of arXiv E-Print Archive between 1995 and 1999. This data represents the relationship among the co-authors. The details about the data is mentioned in the Table 3.1.

Propagation Probability: In both the datasets, the file containing the information about the node relation is used to build the graphs. The influence probability is assigned as 1/N where N is the number of in-degree of a given node. This method is widely used in previous literatures [94, 87, 104]. We use random numbers between 0 and 1 as the propagation probability threshold to find how many nodes are activated by each seed node. The generated random number is compared with the influence probabilities on the edges to count the number of nodes that can be activated in each iteration. If the probability on the edge is greater than the random number it will influence its neighbor; otherwise, it can not influence its neighbor.

Adoption Probability: We use $x \in [0, 1]$ to denote the percentage of the offered discount in the product price. The adoption probability $p_{v,x}$ of each node v for the discount rate x is set as: for the graph $G = (V, E)$, we randomly choose 85% of nodes to assign

$p_{v,x} = \sqrt[3]{x}$, these users are sensitive to discount, and 10% nodes to assign $p_{v,x} = x$, and 5% nodes to assign $p_{v,x} = x^2$ as their seed probability function, which means that those users are insensitive to discount. This setting is also used in some previous literatures [86].

Also, the random number between 0 and 1 is generated as the adoption probability threshold each time to determine if a node accepts the discount. If the probability of the node accepting the discount is greater than the random number it is assumed that it will accept the discount; otherwise, it will reject the discount.

3.6.2 Comparison Method

Zero-feedback greedy method. It selects seed node set which has the largest expected influence at once in advance, and probes these nodes with discounts in increasing order. We should make sure that these selected seed users can spread a target influence value Q.

We use the Monte Carlo sampling to get the mean of the expected influence function f for our problem and the Zero-feedback greedy method as well as the ±1 standard deviation intervals over 100 runs of each node.

3.6.3 Result Analysis

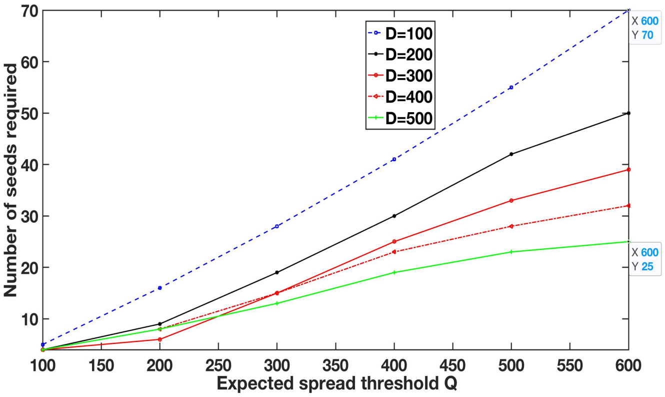

Figure 3.2: Results with zero-feedback greedy method on dataset 1

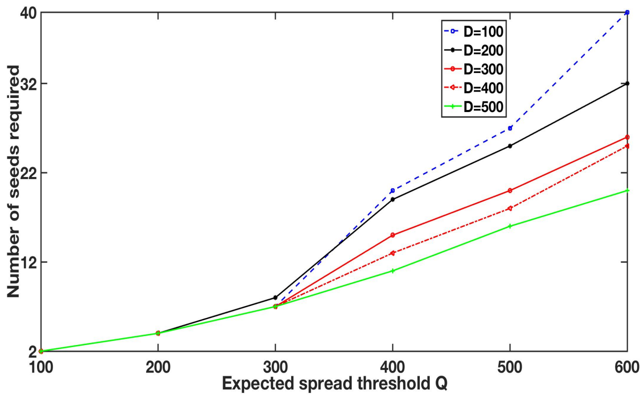

Figure 3.3: Results with online greedy method on dataset 1

Comparison with zero-feedback greedy method on dataset 1. This experiment is done on dataset 1 and the discount is uniform. In Figure 3.2, we show the variation of the number of seeds selected by changing the value of Q with the zero-feedback greedy method. The five different curves are in different discounts changing from 100 to 500. We can see that the smaller the discount is, the more the seeds are needed. But when the value of Q is less than 200, the number of seed users required is the same, as we can find influential users easily since just a small influence is needed. As the spread threshold of Q increases, we need to find more influential seeds, so larger discount will be more attractive for these users to be active. We can see in Figure 3.2, when the discount equals 100 and Q is over 200, it needs the most seeds in the five different discounts situations. When the discount equals 400 and 500, we can see the curves increase more slowly when the Q exceeds 400. It is because the larger discount can attract more high influential nodes, which can activate more nodes with fewer seed nodes. Although providing larger discount is more attractive, it can activate more influential seeds so that it just needs fewer seeds to reach the expected spread threshold Q, but it also has an obvious drawback of being expensive. We can calculate that when $D = 100$ and $Q = 600$, we need choose 70 seed nodes, it costs 7000, while when $D = 500$ and $Q = 600$, we need choose 25 seed nodes, it costs 12500, which is far greater than the uniform discount 100 situation, so bigger discount costs more.

In Figure 3.3, we do the experiment in online greedy method, we also set the uniform discount in five different values. We can see from the lines that the tendency of change is similar to Figure 3.2, they are also monotone increasing and the larger discounts need a smaller number of seeds. We can see that when the discount is 100, we need more seed users but when we calculate the cost, we can find that it takes the least cost. The reason is the same with zero-feedback greedy method above.

From Figure 3.2 and Figure 3.3, we can see that every curve is a gradient rise. And the growth ratio is larger when the discount is smaller. Because when the discount is smaller, many influential users would not accept the discounts to become seeds, we need to find some less influential users to accept the discounts. It can also be noted that the lines of the algorithms clearly illustrate the phenomenon of diminishing marginal returns, empirically illustrating monotonicity and submodularity. Comparing Figure 3.2 with 3.3, for a certain value of Q and the same discount offered, the online greedy method needs fewer seeds, because the online greedy algorithm can choose the next seed wisely based on real spread triggered by existing seeds. Although the larger discount can activate more influential users to become the seeds, it also needs a larger cost. On the contrary, the smaller discount needs to choose some less influential seeds but it saves the cost. So using a smaller discount, it can achieve our expected spread threshold Q and save cost at the same time.

Varying the discount rate set with proposed method. We do the experiment with the online greedy method in the non-uniform discount situation firstly, the results shows in Figure 3.4. The number of discount changes from three kinds to seven kinds, and we can see from the changes of lines that when the discounts are a combination of $D = \{100, 200, 300\}$, it needs the most seeds, and greater the discount combination is, the fewer the seeds we will select. The reason is the same as the results in Figure 3.2 and 3.3, because a greater discount is more attractive to the influential users. So for the combination $D = \{100 \sim 700\}$, it selects the least number of seeds. But based on the experimental results, we know that it

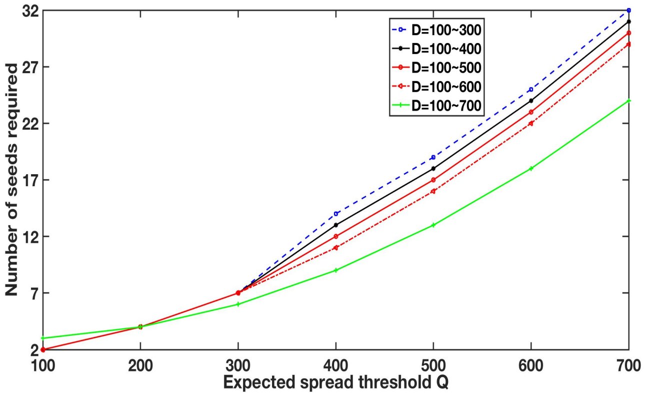

Figure 3.4: Results with online greedy method in non-uniform discount situation on dataset 1

costs the least with the combination of $D = \{100, 200, 300\}$, but it costs the most as the high influential seed would like to accept larger discount. So the smaller discount combination can achieve larger spread with minimum cost.

Comparing Figure 3.4 with Figure 3.2 and Figure 3.3, we can see that for a certain value Q, the non-uniform discount requires less seeds. Based on the experiment results in these three situations, non-uniform discount needs less cost than the uniform discount with the online greedy method as well as the uniform discount with the zero-feedback greedy method for the same Q. As in non-uniform cost, we have more discount choices, we would choose some high influential nodes who would accept low discounts.

In Figure 3.5, we use a dataset 2 to continue our experiments with the online greedy method under the uniform discount situation. Because the dataset 2 is larger than dataset 1, and the indegree of every node is larger than dataset 1, too, the influence probabilities are smaller than that in dataset 1. We can observe the change of the lines in Figure 3.5, for the same discount and Q, it needs more seeds than that in Figure 3.3, as the influence of the seed is smaller than that of dataset 1. But in the larger dataset 2, we can still get the result which is got in the Figure 3.3 with dataset 1 that the smallest discount 100 can save the most cost to get the spread threshold Q. So the results in 3.5 with dataset 2 are consistent

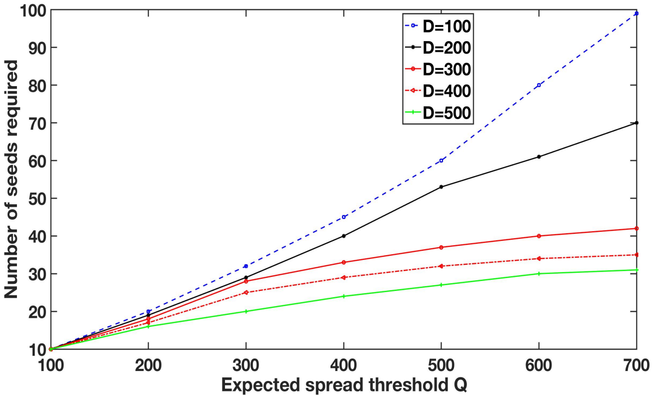

Figure 3.5: Results with online greedy method in uniform discount situation on dataset 2

with the results with dataset 1, which further confirms the correctness of the previous results .

Changes of running time with two proposed methods. The unit of the runtime is in seconds. The y-axis in Figure 3.6 and Figure 3.7 is the total runtime when the expected spread threshold Q increases from 100 to 800 in two different datasets with Online greedy uniform discount and non-uniform discount methods, respectively. In Figure 3.6, we set the uniform discount as 100, 200, 300 to do different simulations in dataset 1 and dataset 2 respectively, we can observe that the running time decreases with the increasing of uniform discount when the spread threshold is the same, it is because that when the discount increases, it is easier to find a seed user and more time saving. This result is verified in both dataset 1 and dataset 2. In non-uniform discount situation, we set discount combination as $D = \{100 \sim 300\}$, $D = \{100 \sim 500\}$, $D = \{100 \sim 700\}$, the results in Figure 3.7 show that when we increase the number of discount in the discount combination, the running time decreases, which is because the larger discount in the combination is easier to accept by a high influential seed node, this can save time to find a suitable seed node. This result is also shown in two different datasets. From the above results in Figure 3.3 and Figure 3.4, we know that it costs less when the uniform discount $D = 100$ than that in other larger uniform discount and the non-uniform discount situation with the online greedy method.

From Figure 3.6 and Figure 3.7, we can see that the uniform discount takes more running time when the dataset and spread threshold are same. Although the uniform discount with online greedy method saves more cost than the non-uniform discount with online greedy method, the non-uniform discount method is more time saving. Because when we choose a node with the maximal marginal gain, it may not accept the lower discount, and we can increase the discount as it has several discount choices in non-uniform discount method. The greater discount has a larger probability to be accepted by a user. It is more time saving than the uniform discount method as we have to find another seed again in the uniform situation once a node rejects a discount. And we also find that the dataset 2 needs more time to reach the same expected spread threshold with the same method, which is easy to understand since dataset 2 has much more nodes than dataset 1.

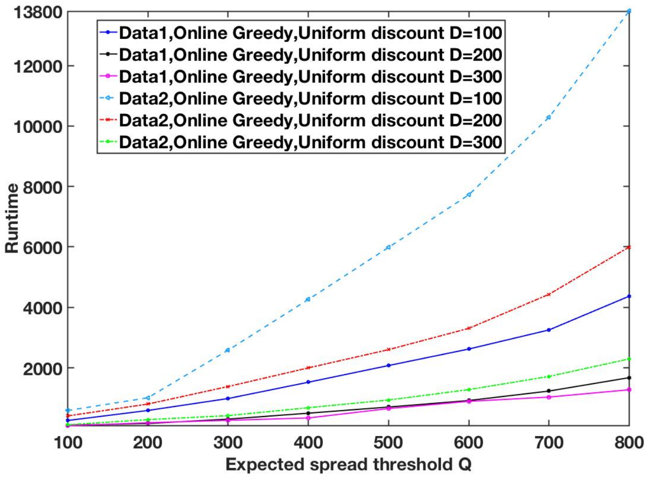

Figure 3.6: The runtime of different expected spread threshold Q with uniform discount method in two datasets

Results on sampling graph from dataset 1. At last, we take samples from the dataset 1. The graphs are sampled to give various subsets of the original graph. For building the sample graph, a random number is generated and if the edge probability is less than the

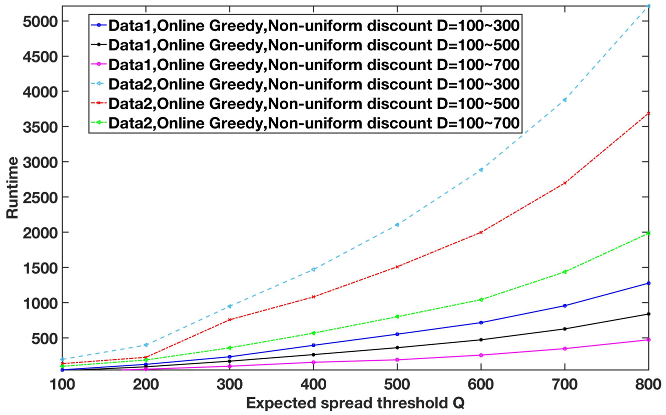

Figure 3.7: The runtime of different expected spread threshold Q with non-uniform method in two datasets

generated number, that edge will be removed from the original graph to give a resulting sample sub-graph. The Algorithm 3.1 is applied to each of the sample graphs to obtain the average results. We generate 100 sample graphs and iterate over all these sample graphs to obtain the results. Each time due to some random parameters in the algorithm, the solution is approximately the same thus the simulation converges. We observe that the trend of the lines in Figure 3.8 are also in accordance with previous experimental results. When the uniform discount is 300, it needs to selects least seed users but cost the most to achieve the spread threshold Q.

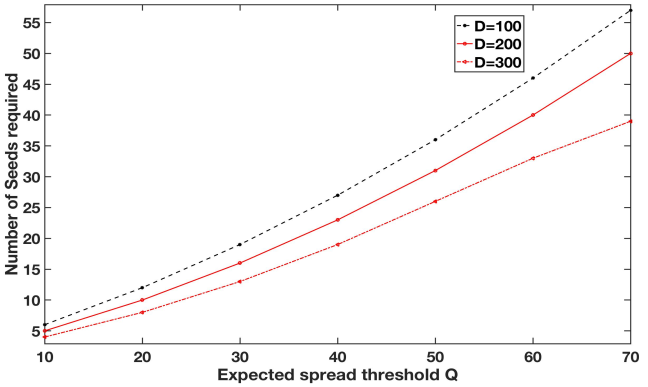

Figure 3.8: Results on sampling graph from dataset 1

From the above experimental results, we can get that the online greedy method is superior to the zero-feedback greedy method, and the lower uniform discount situation is more cost saving than the non-uniform discount situation. But the non-uniform discount situation is more time saving than the uniform discount situation.

3.7 Conclusion

We propose an online full-feedback setting model to solve the discount allocation problem in OSNs and divide the cascade into seed selection and information diffusion stages. Our goal is to minimize the cost that marketer spends while ensuring that the number of people who adopted the target product should not be less than a spread threshold of Q. We present two algorithms under uniform and non-uniform discount conditions and analyze the approximation ratios in these two situations. Finally, we use experiments on real-world OSNs to illustrate the empirical superiority of the proposed strategy. In future work, we intend to study a problem which evaluates a suitable discount for each seed but without using stochastic discount probing as used in this chapter. We also want to study another problem that the marketer has multiple products to advertise at the same time and different products have their specific features which are suitable for user group with different needs. How should we allocate the budget to each product to achieve influence maximization or profit maximization?

CHAPTER 4

SCHEMES OF PROPAGATION MODELS AND SOURCE ESTIMATORS FOR RUMOR SOURCE DETECTION IN ONLINE SOCIAL NETWORKS[1]

4.1 Introduction

Source detection plays a vital role in any network like Network of power grid, Network of peoples, and Online Social Networks, and so forth. Within these networks, source identification has been performed to find the origin or source, such as detecting the source of epidemics to control infection spreading [2], finding the source of a computer virus in a network [77], locating gas leakage source in wireless sensor network [83], identifying propagation sources in complex networks [16, 43], investigating the sources of misinformation in online social networks [69], and rumor source detection [78, 79] in online social media network. From existing works, such as [43] reviews source identification methods in accordance with three categories of network observation which is one of the major premises, [55] surveys various information diffusion models based on two categories, and then a comprehensive study about factors to be considered for source detection of rumor in social network by [82]. We observe that very little review has been done from the perspectives of diffusion patterns and various estimator for sources, on the rumor source detection problem in online social networks. Different models refer to different application domains in seeking propagation origins. Therefore, the objective of this chapter is to summarize and discuss existing approaches to rumor source detection in views of these two aspects. The roadmap of this chapter is depicted in Figure 4.1. To the best of our knowledge, this is the first work that focuses on the schemes of modeling rumor propagation and source estimation of seeking origins of rumor in online social networks.

[1] ©2021. Reprinted, with permission, from R. Jin and W. Wu. Schemes of propagation models and sourceestimators for rumor source detection in online social networks:A short survey of a decade of research. Discrete Mathematics, Algorithms and Applications 13 (04), 2130002.

This chapter is structured as below. In Section 4.2, we summarize existing three approaches to model rumor propagation and we introduce underlying mechanisms behind them. Section 4.3 shows three existing main schemes to estimate rumor sources. Evaluation metrics and datasets for rumor source detection problem are briefly presented in Section 4.4. Research open issues and challenges in source detection of rumor are stated as a part of conclusion in Section 4.5.

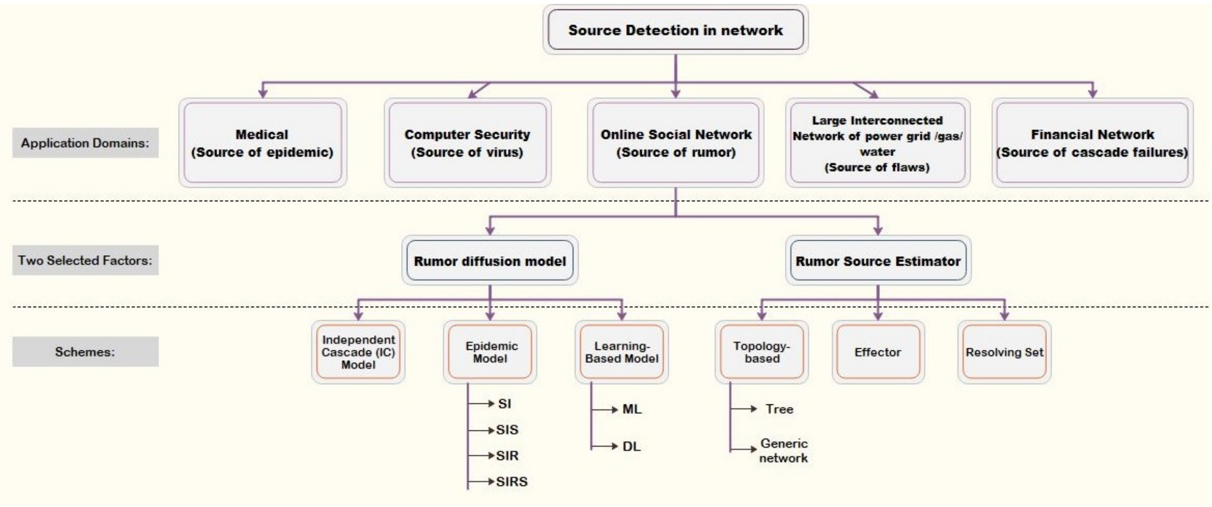
Source Detection in network
Application Domains:
Medical (Source of epidemic)
Computer Security (Source of virus)
Online Social Network (Source of rumor)
Large Interconnected Network of power grid /gas/ water (Source of flaws)
Financial Network (Source of cascade failures)
Two Selected Factors:
Rumor diffusion model
Rumor Source Estimator
Schemes:
Independent Cascade (IC) Model
Epidemic Model
Learning-Based Model
Topology-based
Effector
Resolving Set
SI
SIS
SIR
SIRS
ML
DL
Tree
Generic network

Figure 4.1: Roadmap of the chapter.

4.2 Rumor Propagation Model

Modeling how information spreads is utilized to help analyzing how misinformation spreads as well as to help stopping the spread of misinformation such as rumors [33]. In this section, we present fundamental knowledge of major rumor spreading models and we summarize how these models have been employed by researchers to their rumor source(s) detection problems. We broadly classify contemporary models of rumor propagation into three categories: the Independent Cascade-based model, the epidemic-based model and the state-of-the-art learning-based model.

4.2.1 Independent Cascade(IC) Model

The *Independent Cascade (IC) Model* is one of commonly used diffusion models [48]. In the IC model, the information passes in the network through cascades. It describes the process of information diffusion, from a set of initially activated nodes named sources, proceeding on a directed graph where each node can be activated or not with a monotonicity assumption, i.e., activated nodes are the one who believed the rumor and become infected, and activated nodes cannot deactivate [20]. When a node v becomes active under the IC model, it has a single chance of activating one of its currently inactive neighboring nodes, say a node w, with a numerical diffusion probability p_{vw} associated to the edge between nodes v and w. Researchers have assumed their diffusion process based on the IC model to seek rumor source(s).

Xu et al., [101] firstly studied the problem of identifying single rumor source detection with respect to the Independent Cascade (IC) model in online social networks when investigators have no relevant textual or content information. As cascades in the IC model are necessarily trees, the influence structure of a cascade is given by a directed tree T, which is contained in the directed graph G. And, each cascade in the IC model is associated with the propagation probability between any two neighboring nodes. Based on these two facts, Xu et al., defined Maximum Influence Path and Maximum Propagation Tree as well as proposed a polynomial time rumor source detection algorithm under partial observations.

Lim et al., [58] formulated and investigated a k-minimum distance rumor source detection problem, specially to find a small set of rumor candidates which can be used as initial seeds for further iterative query or investigation, by using the Independent Cascade (IC) model to analyze the likelihood of rumor sources. Since the propagation of rumor from one node to another usually incurs a certain amount of time delay, Lim et al., modified the classical IC mdoel which has extra property that each edge $(u, v) \in E$ has associated a time-varying probability $p(u, v, t)$, characterizing how much u can influence v at time t.

4.2.2 Epidemic based Model

Epidemic Models are the ways that describe the spreading, infection and recovery processes of diseases among the population. The basic epidemic models are used for finding the origin of viral disease [55]. Similarly, they are employed by researchers for finding the source(s) of rumors in the online social networks according to different scenarios. So far, four classical types of epidemic models are mainly characterized in rumor diffusion as discussed below.

1. *Susceptible-Infected (SI) Model*: In the SI model [71], there are two probable states of a node: susceptible (S) and infected (I). Nodes are initially susceptible and can be infected with spreading rumors. Susceptible nodes are uninfected nodes but having infected neighboring nodes. Once a susceptible node becomes infected due to contagion from its neighborhood, it remains infected forever. SI model only describes the infection process of $S \to I$ without taking into account that infected nodes can be recovered after having been infected. In terms of rumor diffusion in online social networks, infected nodes are those who have received or believed any rumor from their infected neighbors.

2. *Susceptible-Infected-Susceptible (SIS) Model*: in the SIS model, the two states of susceptible (S) and infected (I) are the same as that in the SI model. But in this model, when susceptible nodes become infected, after some time they can yet again be susceptible after being cured. This infection and recovery process of $S \to I \to S$ addresses impracticality in SI model.

3. *Susceptible-Infected-Recovered (SIR) Model*: The SIR model divides the total population into three categories: S, I, and R (recovered), where S and I represent the susceptible and infected nodes as described in the previous two models. Different from SI or SIS model, an infected node in SIR model may be recovered, it will not spread the information and remains in the recovered state further [19]. The diffusion process

can be characterized as $S \rightarrow I \rightarrow R$. When in an online social network, the recovered node is the one who recognizes a rumor, therefore it will either delete/ignore the rumor message or not pass that rumor to neighbors.

4. *Susceptible-Infected-Recovered-Susceptible (SIRS) Model*: In the SIRS model, it believes that a recovered node can become a susceptible node with probability α [46]. Thus, the diffusion process in this model is $S \rightarrow I \rightarrow R \rightarrow S$.

Shah and Zaman [78, 80] initially assumed that the rumor propagation follows the SI model originating from a single source. Dong et al., [25] extended study based on the setting of various suspect sources in the SI model. Luo et al., [64] later adopted the SI diffusion model for estimating multiple infection sources and their infection regions. Wang et al., [97] considered the value of diversity from multiple observations for single rumor source detection in the SI spreading. Choi et al., [14] studied the impact of querying in a highly generalized setup for a rumor source detection under consideration of the SI rumor spreading model. Beside that, the SIS model has been considered on inferring a single rumor source detection problem in Luo et al., [63] and Wang et al., [100]. Luo et al., derived an estimator based on estimating the most likely rumor source associated with the most likely infection path. Wang et al., proposed a rumor centrality based algorithm that leverages multiple observations to first construct a diffusion tree graph, and then use their union rumor centrality to find the rumor source. The SIR model was firstly integrated in detecting sources of computer viruses in networks by Shah and Zaman [77]. After that, the spread of information in Zhu and Ying [117] also follows the SIR model for studying single information source detection problem in a network. Locating multiple sources in a network under the SIR model has also been studied in Luo et al., [61], Zang et al., [109] and Jiang et al., [44]. So far, to best of my knowledge, the SIRS model has not yet been applied in rumor source identification methods. Future work may take it into consideration.

In the information diffusion, there are also new models developed based on classical models such as SEIR (Suceptible Exposed Infected Recovered) model [95], SEIRS (Suceptible Exposed Infected Recovered Susceptible) model [18], and MSEIR (Passive-immunity Suceptible Exposed Infected Recovered) [39], and so forth. As far as we know, these models have not been applied in rumor source(s) detection problems by researchers.

4.2.3 Learning based Model

Since it is usually difficult to acquire the actual underlying propagation model in practice, many studies have been proposed on an assumption that the underlying propagation model is known in advance and been given as input. For instance, the Independent Cascade(IC) model or one kind of epidemic based models has been widely utilized in rumor source detection. However, we know that this assumption may lead to impracticability on real data and it may limit its application range. Therefore, several approaches in machine learning methods [11] and deep learning methods [111] to graphs data have been studied and reviewed. In this chapter, we are only presenting learning based models in the rumor source(s) detection problem in online social networks.

Wang et al., [98] firstly studied on multiple sources detection without knowing the underlying propagation model by proposing a Label Propagation based Source Identification (LPSI) method, which is based on the idea of source prominence as well as inspired by a label propagation based semi-supervised learning method [113]. By setting an original label to the infection status of each node, positive labels (+1) to infected nodes and negative labels (-1) to uninfected nodes, the LPSI lets these labels iteratively propagate in a snapshot of a network, and finally predicts local peaks of the convergent node labels as source nodes. Nonetheless, their approach still suffers from the shortcoming that the node label is simply an integer which may restrict the prediction precision. To improve prediction precision on the same problem, Dong et al., [24] then firstly attempted to apply the Graph Convolutional Networks (GCN) technique on multiple rumor sources detection and then proposed a supervised learning based model named Graph Convolutional Networks based Source Identification (GCNSI). Basically, in the GCNSI, Dong et al., proposed an input generation algorithm to extend the LPSI integer label into a multi-dimentional vector for each node as training data, and applied GCN in capturing different features of a node based on two assumptions, one is source prominence from the LPSI, and the other is rumor centrality. Due to node representation utilizing its multi-order neighbors information by adopting spectral domain convolution so that their prediction precision on the sources is improved. Although these works are for multiple sources detection, they undoubtedly also fit into single source detection for sure. More recently, one work [81] also explored GCN for detecting possible suspicious users who are often involved in spreading the rumors on online social media. Additionally, another very recent work [76] revisited problem about locating the source of an epidemic, namely finding patient zero (P0) by using Graph Neural Networks (GNNs) to learn P0. Similarly, we believe that this can be applied in the rumor source(s) detection problem in online social networks.

To best of our knowledge, aforementioned works in this section are all existing studies of learning-based models regarding rumor source(s) detection problem in online social networks.

4.2.4 Discussion

Over the past decade of research on the methods of rumor source(s) detection, information propagation models and network structures have been taken into consideration. Works can be classified into two types: (1) infection status based analysis [78, 109]; (2) partial observations based analysis [62, 110, 112, 115]. In most of the methods on the former type, it is a given input that the underlying propagation model is known in advance. Obviously, it is difficult to obtain model information in practice. However, infection status based analysis has a broader application prospect since we cannot deploy observing nodes or we cannot acquire related information in some real-world situations. Those studies on the learning based models have been carried out for source identification without the requirement of knowing the underlying rumor propagation model.

Most of machine learning tasks on graphs - node classification, link prediction, learning over the whole graph, and community detection are very different from normal supervised/unsupervised learning. This is because graphs are interconnected with each other and data independence assumption fails. Researchers refer to it as semi-supervised learning.

Deep Learning techniques on graphs is about neural networks on data representation learning on graphs, which iteratively updates node representations by exchanging information between the neighbor nodes via relation paths and repeat the iteration until convergence and lets convolution layer capture different features of a node. In most cases, deep learning based models outperform among all baselines, but it spends time to train and tune parameters. Therefore, deep learning on graphs is non-trivial since several challenges exist in practical graphs such as irregular structures of graphs, diversity of graphs or large scale graphs, and so on [111].

4.3 Rumor Source Estimator

Rumor source estimator is the main challenge in rumor source detection problem. How to construct the rumor source estimator? In this section, we provide an overview of source estimators that have been used for existing rumor source detection problems in online social networks. We classify schemes of estimators into three categories as shown in Figure 4.1.

4.3.1 Topology based Approach

In this subsection, we summarize rumor source estimator developed by the center of certain type of graph. There are two main types of graphs: tree-like networks and generic networks.

1. *Regular Tree* : Shah and Zaman [77, 78] firstly proposed rumor centrality as an estimator for a single source. They showed that the node with maximum rumor centrality

(called rumor center) is the Maximum Likelihood Estimator of the rumor source if the underlying graph is a regular tree in social networks. After this work, many other variants of the problem have been studied. Dong et al., [25] utilized the same definition of rumor center and proposed a notion of local rumor center as the node with the highest rumor centrality in the priori set of suspects to identify a single source for regular trees. Luo et al., [63] proposed the multi-rumor-center method to identify multiple rumor sources in tree-structured networks. This method is too computationally expensive to be applied in large-scale networks as computational complexity of this method is $O(n^k)$, where n is the number of infected nodes and k is the number of sources. Zhu and Ying [117] proposed the Jordan center method, which utilizes a sample path based approach, to detect diffusion sources in tree networks with snapshot observations. Luo et al., [64] derived the Jordan center method using a different approach. Chen et al., [13] extended the Jordan center method from single source detection to the identification of multiple sources in tree networks. In addition, Shah and Zaman [78] proved that, even in tree networks, the rumor source identification problem is a *#P-complete problem*, which is at least as hard as the corresponding NP problem.

2. *Random Tree* : Shah and Zaman [80] also established the universality of rumor centrality for source detection for generic random trees without two limited settings of regular tree and the exponential spreading time setting in their previous work [77, 78]. Their results subsequently inspired Fuchs and Yu [29] to extend their work by following the definition of the rumor center for grown simple families of random trees, which contain binary search trees, recursive trees and plane-oriented recursive trees.

3. *Generic Network* : Later Shah and Zaman [79] extended their approaches to random graphs from tree-like networks. Additionally, Pinto et al., [72] employed BFS technique to reconstruct generic networks into trees, and then the origin is sought in the BFS trees.

4.3.2 Effector based Approach

An activation state in a social network with a certain influence diffusion shows the users who have been influenced. The effectors are nodes that can best explain the observed activation state, which indicates that identification of effectors is important in understanding the dynamics of influence diffusion. Specially, we can interpret effectors as the critical nodes that trigger or stop the spread of information, or we may consider effectors as the sources of influence diffusion. Thus, effector detection problem [54] helps in identifying the source of rumors or infections.

Tong et al., [91] tackled the effector detection problem from a novel perspective for social networks. Their approach is based on the influence distance that measures the chance that an active user can activate its neighbors. That is, for a certain pair of users, the shorter the influence distance, the higher probability that one can activate the other. When we are given an activation state, the effectors are expected to have short influence distance to active users while long influence distance to inactive users. By this idea, Tong et al., proposed the influence-distance-based effector detection problem for the IC model that was firstly studied in Lappas et al., [54]. Furthermore, Tong et al., provided a 3-approximation. In the problem, effectors can be played as rumor source estimators. However, for some activation states, it is challenging to find best effectors from active nodes. The criteria for selecting effectors depends not only on the activation state but also on the diffusion model. Thus, it is interesting to investigate whether a meaningful effector exists for a given activation state.

4.3.3 Resolving Set based Approach

In this section, we would like to present two novel rumor source(s) estimation approaches by Chen et al., [12] and Zhang et al., [112] in which they both study source detection problem from a deterministic point of view.

Chen et al., firstly discovered that the concept of doubly resolving set (DRS) can be employed to study the single source detection problem and presented an $O(\ln n)$-approximation algorithm for the minimum weight DRS problem.

After that, Zhang et al., found that the other concept of set resolving set (SRS) can be applied for estimating multiple rumor sources independent of diffusion models in networks with partial observations based analysis. They let G be a network of n nodes, a node subset K is an SRS of G if all detectable node sets are distinguishable by K. Then, the problem of multiple rumor sources detection in the network can be modeled as finding an SRS K with the smallest cardinality. Zhang et al., also gave a polynomial-time greedy algorithm for finding a minimum SRS in a general network with a performance ratio $O(\ln n)$. To the best of our knowledge, there is currently no any other continued and expanded work to study these approaches on different diffusion models or different applications.

4.4 Evaluation

4.4.1 Evaluation Metrics

An effective algorithm of finding the source of rumor spreading is a basic component of a rumor source detection system. In order to measure the performance of the algorithm, the choice of evaluation is usually measured by three different quality of localization metrics: the accuracy, the rank and the distance error. The accuracy is the empirical probability that a source found by the algorithm is the true source [64]. The rank is the true source position on the nodes list, which is sorted in descending order by likelihood of being the source [101]. The distance error is the shortest path distance between the real source and the source found by the algorithm [58].

4.4.2 Datasets

Datasets used for rumor source detection in social networks have been widely classified into synthetic datasets and real-world datasets [82].

Synthetic Datasets are primarily constructed in forms of tree and graph. The tree networks are usually represented by random d-regular trees. The graph networks are basically illustrated by Small-World (SW) networks and scale free networks.

Real Datasets are commonly datasets from Facebook, Twitter, or Wiki-vote, and so forth. They are freely accessible on Stanford Large Network Dataset Collection. There is also a popular Chinese micro blogging network, namely Sina Weibo used for rumor source detection. A comprehensive study of datasets used for rumor source detection problem in social networks has been widely investigated in [82].

4.5 Conclusion

The proliferation of data generated by a social network generates a number of real-world problems to be solved, and rumor source detection problem is one of them. This chapter aims to summarize and analyze existing schemes for modeling rumor diffusion process as well as schemes for estimating rumor source(s) in online social networks. In this chapter, except for the widely reviewed influence diffusion model (i.e., the IC model) and epidemic models, we additionally present all existing learning based models for rumor propagation in the rumor source detection problem. It has been studied and seen that data representation learning based approaches have shown promising results by modeling information diffusion process on graphs that leverage both graph structure and node feature information. Although this approach results in the best precision performance among all baselines, time of training and tuning parameters still requires more investigations to be reduced. We notice that this research direction is growing rapidly now for rumor or misinformation source detection problem in online social networks. Moreover, we notice that there is little work for investigating more efficient and effective schemes for estimating source(s) in the rumor source detection problem.

Overall, the objective of this literature study regarding schemes of rumor propagation models and rumor source estimators is expected to guide our research work as well as to provide a timely and worthwhile reference for similar work in other domain specific networks such as wireless sensor network, network of virus spread, epidemic network, and financial network, and so forth.

CHAPTER 5
STATION-BASED APPROACH FOR ONLINE RUMOR SOURCE DETECTION

5.1 Introduction

Information diffusion processes in online social networks refers to the spread of information throughout the networks and have been widely used to model many real-world phenomena such as the spreading of computer virus over the Internet, the outbreak of epidemics, and the spreading of misinformation over online social networks and so on. Because of its wide range of applications for many decades, the reverse of the information diffusion problem - the information source detection problem has also gained a lot of attention over the last decade since the seminal work regarding rumor source detection in online social network by Shah and Zaman [78]. The information source detection problem is to identify the source of information diffusion in networks based on some observations like the states of the nodes and the timestamps at which nodes adopted (or infected by) the information as well as assumed on different diffusion models such as the independent-cascade (IC) model, the epidemic models. This problem has been studied in many applications after Shah and Zaman's like detecting the source of epidemics to control infection spreading [2], finding the source of a computer virus in a network [77], locating gas leakage source in wireless sensor network [83], identifying propagation sources in complex networks [16], investigating the sources of misinformation in online social networks [69]. In this chapter, we are interested in one specific application domain that is actively triggered off on purpose for various reasons in current online media environment, which is rumor source detection problem. Existing research work about this problem has been widely investigated by recent surveys [43, 82], and Chapter 4 [45].

To understand the rumor propagation in online social networks, we consider the independent cascade (IC) model [48] for the spread of influence through online social networks.

The network is assumed to be an undirected graph, and each node has only two possible states: active (or called infected), and inactive (or called uninfected). Active nodes are users who adopted the rumor and cannot deactivate. We assume that initially all nodes are in the uninfected state except one infected node (called the rumor source). The rumor source then infects its currently inactive neighbors with a numerical propagation probability, and the rumor starts to spread in the network. Now given a complete snapshot of the network at some time, in which we can identify infected nodes and uninfected nodes. We pre-select a set of k infected nodes to be our observed stations among all infected nodes. Without knowing the first rumor infection time of each observed station nor the neighbor from which the rumor is adopted. The goal is to detect the rumor source based on the infection stations.

To overcome this problem, our main contributions of this chapter are summarized as follows.

Consider a social network $G = (V, E)$ with the IC diffusion model, we formally defined influence distance metric for each edge associated with a propagation probability that u can activate v after becoming active. This measurement allows the calculation of influence propagation becoming the computation of distance in the tree graphs. With the distance, we next propose the station-based source detection approach. We only consider the case where there is a single rumor source. We derive an estimator for the source node associated with the most likely infection path that yields a subset of infected nodes to be the Jordan infection center as well as we provide an efficient algorithm to find the proposed estimator in theory. Followed by the result, we propose a k-station selection problem to deal with the distribution of stations, and then we develop a greedy based 2-factor approximation algorithm to the problem. To the best of our knowledge, our station selection problem is the first work that solves the source detection problem in social networks. Moreover, the estimator using the eccentricity of a graph to identify rumor source is rarely discussed in the IC model, because it has only been used in epidemic models [117, 63, 62] before.

The chapter is organized as follows. We first discuss most relevant work in section 5.2. Section 5.3 formally formulates the problem under the IC model. Section 5.4 introduces the source estimator and stations deployment.

5.2 Related Work

There have been extensive studies on the rumor source detection problem in online social networks based on two classical diffusion models - epidemic models like SI, SIR, SIS or SIRS and the independent cascade (IC) model.

Shah and Zaman [78] firstly studied single rumor source detection problem, in which a new graph centrality called rumor centrality was proposed and proved to be the maximum likelihood estimator (MLE) on regular trees under the susceptible-infected (SI) model. Later, [117] proposed the sample path based approach for the single source detection problem and proved that the sample path based estimator on tree network is a Jordan infection center under the homogeneous SIR model with a complete snapshot. This approach has been extended to several directions under the same SIR model [116, 13] as well as under the other epidemic SIS model [63] for tree networks, respectively.

Besides the epidemic models, [101] firstly studied the problem of identifying single rumor source detection with respect to the IC model in online social networks where a monitor based approach was proposed and a maximum influence path with respect to maximum likelihood value as a rumor quantifier for ranking how likely nodes are the actual rumor source for tree graphs in the directed graph under the IC model. [101] also studied that various pairs of number of monitors and monitor selection methods affect the precision of source detection. Three monitor selection methods: Random, Incoming Degree (ID) and Betweenness Centrality (BC) were compared and experiments demonstrated that when the monitor are randomly chosen, the precision of locating the rumor source averagely increases with the monitor number increased from small number (e.g., 50) to large number (e.g., 2000).

However, the selection of monitors [101] is restricted to *representativeness* that contributes most to the rumor source detection problem. This recalls another problem, the propagation sources can be detected by injecting monitors (or sensors). When the monitors (or sensors) are injected into networks, they act as normal users while collecting the propagation information including states, infection time and so forth.

Several other related work under the independent cascade (IC) model include: [58] formulated and studied a k-minimum distance error rumor source detection problem with the objective of finding the optimal set of rumor-infected candidates under the size constraint of k for further minimizing their defined expected distance error with respect to the shortest hop distance from the rumor source to the k infected vertices, and they investigated the impact of the size of candidate set k on the average distance error. In addition, a similar problem under the IC model to minimize the L1 distance between the expected states and observed states of the nodes was studied and several heuristic algorithms based on a single snapshot of the tree network was proposed in [54].

5.3 Problem Formulation

5.3.1 IC Model for Rumor Spreading

A social network is represented by an undirected graph $G = (V, E)$, where vertices $v \in V$ represent users and E is the set of edges where each edge (u, v) represents interactions between users u and v. Each node has two possible states: active (or called infected) and inactive (or called uninfected). We assume that time is divided into discrete time slots. Nodes change their states at the beginning of each time-slot, and the state of node v in a time-slot t is denoted by $X_v(t)$. Let $X[0, t] = \{X_v(\tau) : 0 \leq \tau \leq t, v \in V\}$ to be a path of the infection process, which contains the states of all the nodes of the path from 0 to t.

Initially, all nodes are inactive except one single rumor source that is in the active state. At the beginning of each time slot t, an active node u attempts to activate it inactive neighbor

v. If u succeeds, then v becomes active at next time slot $t+1$, otherwise v remains inactive. The process of influence diffusion runs until no user can be further activated. Under the IC model, there is the weight associated with each edge, representing the success probability of the attempt, called the *infection probability* (or called *propagation probability*) of the edge and each attempt is independent of others. Each active node has only one chance to activate its inactive neighbors. Denote the infection probability of edge (u, v) by $p_{uv} \in [0, 1]$. Next, an inactive node v is infected with probability $1 - \prod_{u \in N_{active}(v)}(1 - p_{uv})$, where $N(v)$ denotes the neighbors of v. Note that $p_{uv} = p_{vu}$ as we consider the undirected graph. In addition, we assume an active node retains with the rumor forever once it is infected.

At a time-slot t, we observe a complete snapshot $O = \{X_v(t), v \in V\}$ such that,

$$X_v(t) = \begin{cases} O_a, & \text{if } v \text{ is active} \\ O_b, & \text{if } v \text{ is inactive} \end{cases}$$

We say O_a is a set of active nodes and the set O_b of inactive nodes. Our station-based rumor source detection problem is to detect the source s^* based on a subset $M \subset O_a$ of k explicit nodes (called *stations*) instead of observing O_a. We assume the observation time t is unknown.

5.3.2 Influence Distance from the IC Model

In a social network, there exists a concept of influence distance which measures the distance between users with respect to information propagation probability. That is, for two users u and v, the influence distance d_{uv} measures the chance that the influence can be propagated to v from u. The smaller value of d_{uv}, the higher probability with which u can activate v.

Based on that, we start by discussing how to extract the influence distance for each edge with propagation probability from the IC model in the graph G.

Suppose that at a time slot t, there is a weighted path

$$X_{v_1 v_k}[0, t] = \langle p_{12}, p_{23}, ..., p_{i(i+1)}, ..., p_{(k-1)k} \rangle$$

from node v_1 to v_k, in which $p_{i(i+1)}$ is infection probability associated with each edge of the path. Define the propagation probability of the path $X_{v_1 v_k}[0,t]$ to be,

$$\mathtt{Pr}(X_{v_1 v_k}[0,t]) = \prod_{i=1}^{k-1} p_{i(i+1)} \tag{5.1}$$

where the product is over a set of edges of the path. Intuitively, u can activate v via a path $X_{uv}[0,t]$ with the probability of $\mathtt{Pr}(X_{uv}[0,t])$, because all nodes along the path need to be activated. Let $\chi^G_{uv}(t)$ denote the set of all paths from u to v in the graph G at an observation time t, if there exists multiple paths connecting from u to v. Let $\bar{\mathtt{Pr}}(X_{uv}[0,t])$ be the path that has maximum propagation probability from u to v. Hence, $\bar{\mathtt{Pr}}(X_{uv}[0,t])$ is called *maximum propagation path* from u to v, and $\bar{\mathtt{Pr}}(X_{uv}[0,t]) \in \chi^G_{uv}(t)$.

Since there are many alternative joint paths from u to v, $\bar{\mathtt{Pr}}(X_{uv}[0,t])$ obviously cannot completely capture the influence diffusion. To handle this hardness, we borrow the k-th influence distance defined in [91].

Definition 5.1. *Let $\chi^k_{uv}(t)$ is the set of k independent paths that have maximum propagation probability from u to v at an observation time t. Define the k-th influence distance d^k_{uv} of the pair of users u and v to be,*

$$d^k_{uv} = -ln\left(1 - \prod_{X_{uv}[0,t]\in\chi^k_{uv}(t)} (1 - \mathtt{Pr}(X_{uv}[0,t]))\right) \tag{5.2}$$

By the construction of $\chi^k_{uv}(t)$, the paths are edge disjoint and thus u activates v independently through these paths. Therefore, $1 - \prod_{X_{uv}[0,t]\in\chi^k_{uv}(t)}(1 - \mathtt{Pr}(X_{uv}[0,t]))$ represents the probability that u can activate v through the paths in $\chi^k_{uv}(t)$. In this study, we assume maximum propagation path is always unique. Thus, by (5.2), we have influence distance for $k = 1$,

$$\begin{aligned} d^1_{uv} &= -ln\left(1 - \prod_{X_{uv}[0,t]\in\chi^1_{uv}(t)} (1 - \mathtt{Pr}(X_{uv}[0,t]))\right) \\ &= -ln(\bar{\mathtt{Pr}}(X_{uv}[0,t])) \end{aligned} \tag{5.3}$$

Lemma 5.1. *Given a graph $G = (V, E)$ under the IC model, let the path $X_{uv}[0, t]$ from node u and v is a set of edges with influence probability at an observation time t. The influence distance $d_{uv} = -ln(\mathtt{Pr}(X_{uv}[0, t]))$.*

Lemma 5.2. *Given a path $X_{uv}[0, t]$ with edge probability between two nodes u and v under the IC model in the graph. The influence distance d_{uv} is the sum over the influence distance of edges of the path from u to v.*

Proof. Take aforementioned weighted path $X_{v_1 v_k}[0, t]$ from node v_1 to v_k as an example, then by (5.1) we have,

$$\mathtt{Pr}(X_{v_1 v_k}[0, t]) = p_{12} \cdot p_{23} \cdot \ldots \cdot p_{(k-1)k}$$

Next, by Lemma 5.1, it turns out that,

$$\begin{aligned} d_{v_1 v_k} &= -ln(\mathtt{Pr}(X_{v_1 v_k}[0, t]) \\ &= -ln(p_{12} \cdot p_{23} \cdot \ldots \cdot p_{(k-1)k}) \\ &= -ln(p_{12}) - ln(p_{23}) \cdots - ln(p_{(k-1)k}) \\ &= d_{12} + d_{23} + \cdots + d_{(k-1)k} \end{aligned}$$

□

One can see that the calculation of influence probability becomes to compute the influence distance under the IC model in the graph. A short influence distance between two users implies the rumor is more easily to spread between them. Note that $d_{uv} = d_{vu}$ as we consider the undirected graph. We can think of *influence distance* as *distance*.

5.3.3 Infection Path Based Estimation

In the following, we use an infection path based approach to detect the path $X^M[0, t]$ that most likely leads to observed infection stations M in tree networks.

$$X^M[0, t] = \arg_t \max_{X[0,t] \in \chi(t)} \mathtt{Pr}(X[0, t]) \tag{5.4}$$

where $\chi(t) = \{X[0,t]|M \cap O_a\}$, and $\texttt{Pr}(X[0,t])$ is the probability to obtain the most likely infection path. The source node associated with $X^M[0,t]$ will be treated as the infection source.

5.4 Single Rumor Source Estimation for Tree Networks

In this section, we assume that the underlying network G is on trees with only one single source. We will derive that the source associated with the most likely infection path is a node with the minimum infection eccentricity. We first introduce the following definitions.

Definition 5.2. *Let d_{vu} denote the distance between nodes v and u in graph, where the distance is known as the shortest path between two nodes. Given a set M (—M—¿0) of explicit observed infection stations, define the largest distance between node v and any infected station node to be,*

$$\bar{d}(v, M) = \max_{u \in M} d_{vu} \tag{5.5}$$

where $\bar{d}(v, M)$ is called the infection eccentricity of node v, denote by $\tilde{e}_M(v)$. And the nodes with minimum infection eccentricity are defined as the Jordan infection centers of a graph.

Definition 5.3. *For each $v \in V$, let $X_v^M[0,t] \in \chi_v(t)$ to be the most likely infection path leading to M up to time t, given that v is the infection source,*

$$X_v^M[0,t] = \arg_t \max_{X[0,t] \in \chi_v(t)} \boldsymbol{\texttt{Pr}}(X[0,t]|s^* = v) \tag{5.6}$$

where $\chi_v(t)$ is viewed as the set of all possible infection paths starting with v and resulting in M at a time slot t and $\boldsymbol{\texttt{Pr}}(X[0,t]|s^* = v)$ is the likelihood of obtaining the path $X_v^M[0,t]$ given the source v.

5.4.1 Infection Path Propagation Time

Lemma 5.3. *Given a non-empty set M of explicit observed infection stations, suppose that v is the rumor source under the IC model. Then, the rumor spreading time associated with the most likely infection path conditioned on v is given by $t_v^M = \bar{d}(v, M)$.*

Proof. We analyze the time duration of the most likely infection path such that,

$$t_v^M = \arg_t \max_{X_v[0,t] \in \mathcal{X}_v(t)} \mathtt{Pr}(X_v[0,t] | s^* = v) \tag{5.7}$$

Which means that we want the time t_v^M maximizing the likelihood of obtaining the path covering the observed infection station set M.

As we assume time is divided into discrete time slots, the infection diffusion is at most one hop further from the source v in one time slot. If observation time $t < \bar{d}(v, M)$, the rumor infection can not reach the nodes in the M. Hence, it is not possible for every node in M to get infected. Therefore, we have the observation time $t \geq \bar{d}(v, M)$.

Next, intuitively, at an observation time t, a path with longer time duration involves more probabilistic infected nodes which contributes a factor of $(1 - p)$ in each time slot to the overall probability to remain active. Thus, the influence probability associated with the path $X_v^M[0, t]$ is monotonically decreasing with respect to t.

From above two results and according to Lemma 5.1, it can be proved that

$$t_v^M = \bar{d}(v, M)$$

□

We will have unique t_v^M for each $v \in V$.

5.4.2 Infection Source Estimator

The infected nodes form a connected *infection subgraph* under the IC model in the graph G. Since the source node must be an infected node, the detection of the rumor source can be restricted to the infection subgraph. We then have the following lemma.

Lemma 5.4. *Suppose the rumor infection spreading follows an IC model. Let g be the minimum infection subgraph of G that contains observed infection stations M, and let $t_v^M = \bar{d}(v, M)$ for any $v \in g$. Then, for any pair of neighboring nodes u and v in g, if $t_v^M < t_u^M$, we have,*

$$\boldsymbol{Pr}(X_v^M[0, t_v^M]) > \boldsymbol{Pr}(X_u^M[0, t_u^M]) \tag{5.8}$$

Proof. If $t_v^M < t_u^M$, by Lemma 5.3, we can easily have $\bar{d}(v, M) < \bar{d}(u, M)$. According to Definition 5.2, we have $\tilde{e}_M(v) < \tilde{e}_M(u)$ in the infection subgraph g. We further by Lemma 5.1, which finally indicates the path starting from a node with a smaller infection eccentricity is with a larger likelihood over edges of the path, which means the path is more likely to occur, we then can prove Lemma 5.4. □

Lemma 5.5 ([117])**.** *On a tree network with at least one infected node, there exists at most two Jordan infection centers. Furthermore, when the network exactly has two Jordan infection centers, these two centers must be adjacent.*

Based on Lemma 5.3, Lemma 5.4 and Lemma 5.5, we have the following theorem 5.1.

Theorem 5.1. *Suppose the rumor infection spreading follows an IC model. Consider an infinite tree network graph, and M is the observed infection stations. Then, the single infection source associated with $X^M[0, t]$ in (5.4) is estimated by,*

$$s^* = \arg\min_{v \in V} \tilde{e}_M(v) \tag{5.9}$$

Intuition: We will assume one Jordan infection center s^*. When there are two Jordan infection centers, based on Lemma 5.5, we can treat them as one node. We will show that, for any node $a \in g \setminus \{s^*\}$, there exists an infection path from a to s^*, along which the infection eccentricity monotonically decreases, which implies that s^* must be a Jordan infection center.

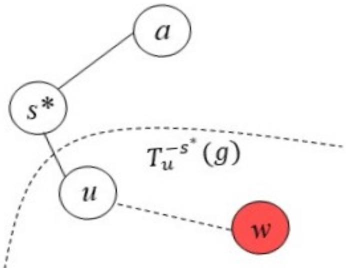

Figure 5.1: Example picture illustrating the proof, and node w is an observed infection station in M

Proof. First, on the infection path from a to s^*, assume u is the neighboring node of s^*. Let $T_u^{-s^*}(g)$ denote the subtree of g rooted at node u but without the branch from s^*, and the nodes $s^*, u \in g$.

From Figure 5.1, it is obvious that there exists an observed infection node w such that $d_{wu} = \tilde{e}_M(s^*) - d_{us^*}$, where $\tilde{e}_M(s^*) = d_{ws^*}$ is assumed, and the node $w \in T_u^{-s^*}(g) \cap M$. Consider a node $l \in T_u^{-s^*}(g) \cap M$, we will hold that $d_{lu} \leq d_{ws^*} - d_{us^*}$.

Now consider $a \in g \backslash \{s^*\}$, and assume $a \in g \backslash T_u^{-s^*}(g)$, then, for any node $l \in T_u^{-s^*}(g) \cap M$, we have

$$\begin{aligned} d_{al} &= d_{as^*} + d_{s^*u} + d_{ul} \\ &\leq d_{as^*} + d_{s^*u} + d_{ws^*} - d_{us^*} \\ &= d_{as^*} + d_{ws^*} \end{aligned}$$

Which denotes that $\tilde{e}(a) = d_{as^*} + d_{ws^*}$. Next, since $\tilde{e}(s^*) = d_{ws^*}$, we can claim that the infection eccentricity decreases along the infection path from nodes a to s^*. After that, by repeatedly using Lemma 5.4, we can conclude that the infection path rooted at node s^* is more likely to occur than that rooted at node a from a to s^*. Therefore, the theorem holds. □

5.4.3 Strategy for Station Selection

Now let us discuss how to distribute stations in a complete snapshot of the network. Given a finite set O_a of infected nodes, a subset $M \subset O_a$ of observed infection stations, and an integer $k = |M|$, we expect to the following formula to be smaller,

$$\min_{v \in V} \tilde{e}_{O_a}(v) - \min_{v \in V} \tilde{e}_M(v) \tag{5.10}$$

By Definition 5.2, where $\tilde{e}_{O_a}(v) = \max\limits_{u \in O_a} d_{vu}$, and $\tilde{e}_M(v) = \max\limits_{u \in O_a \cap M} d_{vu}$. And this consideration may ask set M to be a solution to k-center problem.

Definition 5.4 (k-center problem). *Given a set O_a of nodes and an integer $k \leq |O_a|$, select a set $M \subset O_a$ of k nodes that minimize the maximum distance of any node of O_a to its closest center in M, that is*

$$\min_{\substack{M \subset O_a \\ |M| = k}} \left(\max_{i \in O_a} d(i, M) \right) \tag{5.11}$$

where $d(i, M) = \min\limits_{j \in M} d_{ij}$, and $\max\limits_{i \in O_a} d(i, M)$ is also called radius r of any node of O_a to its closest center in M, denote by

$$r(M) = \max_{i \in O_a} d(i, M) \tag{5.12}$$

Let us view the k-center problem as a covering problem by balls. Given a node x and radius r, define the ball $b(x, r)$ to be the closed ball of radius r cnetered at x. Given any solution M to the k-center problem, if we now place balls of radius $r(M)$ about each node in M, it is easy to see that every node of O_a lies within the union of these balls. By definition of radius in (5.12), one of nodes of O_a will lie on the boundary of one of these balls. Otherwise, we could make $r(M)$ smaller. The neighborhood of each cluster will lie within its associated ball. See Figure 5.2(b).

Given this perspective, the k-center problem is equivalent to the following k-station selection problem.

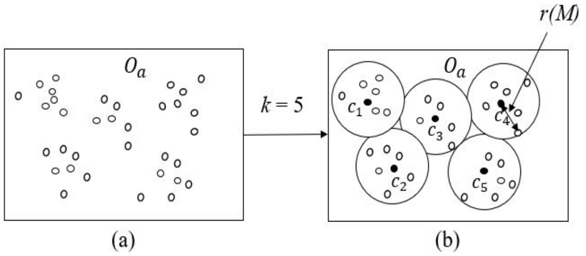

Figure 5.2: Illustration of the k-center problem

Definition 5.5 (k-station selection problem). *Given a set O_a of infected nodes and an integer $k \leq |O_a|$, find the minimum radius r and a set of balls of radius r centered at k infected station nodes of $M \subset O_a$ with $|M| = k$ such that O_a lies within the union of these balls.*

Hence, these pre-selected observed k infection stations can be treated as representatives for all infected nodes in the network.

Theorem 5.2. *The k-station selection problem is NP-hard.*

Proof. Like many clustering problems, the k-center problem is known to be NP-hard. Identically, our proposed k-station selection problem is to be NP-hard. □

Based on Theorem 5.2, we will not be able to solve the k-station selection problem exactly. Therefore, we develop a greedy based algorithm that can find an approximation to the optimum value of radius r in the problem efficiently in polynomial time in the worse case for general graphs.

Our greedy algorithm begins by selecting any node of O_a to be the initial center m_1. We then repeat the following process until we have k stations. Let $M_i = \{m_1, \ldots, m_i\}$ denote the current set of stations. Recall that $r(M_i)$ is the maximum distance of any infected node of O_a from its nearest station in M. Let node $u \in O_a$ be the node achieving this distance. Intuitively, u is the farthest user that adopts rumor by its closest pre-selected infection station. The greediest way to satisfy u is to put the next station directly at u. In other words, we set

$$m_{i+1} \leftarrow u$$

and

$$M_{i+1} \leftarrow M_i \cup m_{i+1}$$

The detailed description is presented in Algorithm 5.1. The value d_u denotes the distance from u to its closest station. To simplify, we start with M being empty. When we select the first station, all the nodes of O_a have infinite distances, so the initial choice is arbitrary.

Algorithm 5.1 Greedy k-station Selection

```
Input: $O_a \subset V$; integer $k$
Output: $M \subset O_a$ with $|M| = k$
 1: Initialize $M \leftarrow \emptyset$
 2: for $u \in O_a$ do
 3:     Initialize $d_u = \infty$
 4: end for
 5: for $i \leftarrow 1$ to $k$ do
 6:     Let $u \in O_a$ be the node such that $d_u$ is maximum
 7:     $m_{i+1} \leftarrow u$
 8:     $M_{i+1} \leftarrow M_i \cup m_{i+1}$
 9:     for $v \in O_a$ do                    ▷ //update distance to nearest station
10:         $d_v = \min(d_v, d_{vu})$
11:     end for
12:     $r = \max_{v \in O_a} d_v$  ▷ //update the radius $r$ of each station to its farthest neighboring node in $O_a$
13: end for
14: return $(M, r)$
```

The line 5 to line 8 of the algorithm is illustrated in Figure 5.3. Assuming that we have three stations $M = \{m_1, m_2, m_3\}$, let m_4 be the node that is farthest from it closest station, say m_1 in this case. In each step we create a station at m_4, so now $M = \{m_1, m_2, m_3, m_4\}$. In anticipation of the next step, we find the node, say m_5 in this case, that maximizes the distance to its nearest station, and if the algorithm continues, it will be the location of next station.

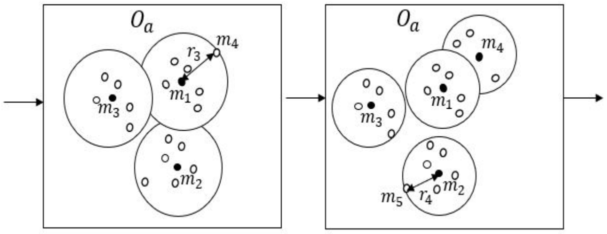

Figure 5.3: Illustration of greedy approximation to k-station selection

Theorem 5.3. *Greedy k-station Selection algorithm runs in $O(k \cdot |O_a|)$.*

Proof. According to the pseudocode in Algorithm 5.1, it is easy to see that its running time is $O(k \cdot |O_a|)$. In general $k \leq |O_a|$, so it is $O(|O_a|^2)$ in the worst case. □

Theorem 5.4. *Algorithm 5.1 is a 2-approximation solution for k-station selection problem.*

Proof. Let $M = \{m_1, \ldots, m_k\}$ denote the set of stations computed by the greedy algorithm, and let $r(M)$ denote the maximum radius over all the stations. Let $O = \{o_1, \ldots, o_k\}$ denote the optimum set of k stations such that $r(O)$ is the smallest possible. We will show that our greedy result is at most twice as bad as optimal, that is

$$r(M) \leq 2r(O)$$

Our approach is to determine a lower bound r_{min} on the optimum value that $r_{min} \leq r(O)$. Then we show that our greedy algorithm 5.1 produces a value that satisfies $r(M) \leq 2r_{min}$. Thus, It will follow that $r(M) \leq 2r(O)$.

Define M_i to be the set of stations after the ith execution of the greedy algorithm, and let $r_i = r(M_i)$ denote the overall radius that farthest neighboring node from its closest station in M_i. Algorithm 5.1 stops with m_k, but for the sake of analysis, we consider the next station to be added if we executed one more iteration. That is, define m_{k+1} to be the node of O_a that maximizes the distance (i.e., $r(M_k)$) to its closest station in M_k. Also, we define $M_{k+1} = \{m_1, \ldots, m_{k+1}\}$.

Claim 5.4.1. *For $1 \leq i \leq k+1$, $r_{i+1} \leq r_i$. That is, the sequence of radius is monotonically non-increasing.*

Proof. Whenever a new station is added, the distance to each node from its closest station will either be the same or will decrease. We also see the fact that the covering radius decrease with each step in Figure 5.3. □

Claim 5.4.2. *For $1 \leq i \leq k+1$, every pair of stations in M_i using greedy algorithm 5.1 is separated by a distance of at least r_{i-1}.*

Proof. Consider the ith step. By the induction hypothesis, the first $i-1$ stations are separated from each other by distance $r_{i-2} \geq r_{i-1}$. By definition, the ith station is at the location with distance r_{i-1} from its closest station, hence it is at distance at least r_{i-1} from all the other stations. □

Claim 5.4.3. *Let $r_{min} = r(M)/2$. For any set C of k cluster stations, $r(C) \geq r_{min}$.*

Proof. By definition of $r(C)$ in Definition 5.4, we know that every node of O_a lies within distance $r(C)$ of some point of C, and since $M \subset O_a$, this is true for M as well. Because $|M_{k+1}| = k+1$, by the pigeonhole principle, there exists at least two stations $m, m' \in M_{k+1}$ that are in the same neighborhood of some station $c \in C$, that is, $\max(d_{mc}, d_{m'c}) \leq r(c)$. Since $m, m' \in M_{k+1}$, by Claim 5.4.2, $d_{mm'} \geq r_k = r(M)$. We assume that the distance satisfies the typical properties of any natural distance function, then we apply them to the triple (m, c, m'), we have

$$\begin{aligned} r(M) &\leq d_{mm'} \\ &\leq d_{mc} + d_{cm'} (\text{by triangle inequality}) \\ &\leq d_{mc} + d_{m'c} (\text{by distance symmetry}) \\ &\leq r(c) + r(c) \\ &\leq r(C) + r(C) = 2r(C) \end{aligned}$$

Hence, as desired that, $r(C) \geq r(M)/2 = r_{min}$. □

Now, by applying Claim 5.4.3 to the optimum set O, we have $r(O) \geq r_{min}$. Since $r_{min} = r(M)/2$, we then have $r(M) \leq 2r(O)$. Based on above three claims, the proof is complete. □

5.4.4 Infection Path Based Source Detection

An efficient algorithm to find the Jordan infection center has been described in [117] under the SIR model, which we will call infection path based source detection algorithm under the IC model. The details are described in Algorithm 5.2.

Based on Theorem 5.1, we view the Jordan infection centers as possible candidates of the rumor source based on the structure properties of the most likely paths on trees.

The procedure of algorithm is to find the Jordan infection centers. First, the algorithm is to let every pre-selected observed infection station broadcast a rumor message containing its station ID to it neighbors, each node of which checks whether its first time to receives the station ID after receiving the rumor messages from its neighborhood. If the station ID not received before, the node records the station ID, and the time at which the rumor is received, say $t_u^{s_i}$, and then broadcasts the rumor messages containing the station ID to its neighbors. When a node receives all station IDs, we can claim it as the rumor source, and the algorithm terminates. If there are multiple nodes receive all station IDs at the same time, we then break ties based on the infection closeness, which is defined as the inverse of the sum of distances from a node to all infected nodes in [117] that measures the efficiency of a node to spread information to infected nodes. The algorithm selects a Jordan infection center with the largest infection closeness that breaks ties at random. In other words, the tie is broken by selecting the node with smallest $\sum t_u^{s_i}$. The set R denotes the set of candidate Jordan infection centers.

Algorithm 5.2 Infection Path Based Source Detection

Input: set M of k infection stations; $g = (V, E)$

Output: the estimated rumor source s^*

```
Set $t = 1$
for $i \in M$ do
    $i$ broadcasts rumor containing its station ID $s_i$ to its neighbors
    do
        for $u \in V$ do
            if $u$ receives $s_i$ for the first time then
                Set $t_u^{s_i} = t$ and continue broadcasting the rumor containing $s_i$ to its neighbors
            end if
        end for
        $t = t + 1$
    while no node receives $k$ distinct station IDs
end for
return ($s^* = \arg\min_{u \in R} \sum_{i \in M} t_u^{s_i}$), where $R$ is the set of nodes who receives $k$ distinct station IDs when the algorithm terminates. Ties are broken randomly.
```

Theorem 5.5. *The worse-case complexity of Algorithm 5.2 is equal to $O(k \cdot |E|)$.*

Proof. To implement Algorithm 5.2, we borrow some operations in [117]. We set an array of integers of size k and an integer counter to each node, and an integer index to each infection station. The values of that integer array are the distances from the node to the infection stations. The integer counter records the number of distinct station indices received. A rumor message only contains the index of a station. At each iteration, when a node not receiving the station index in the new message, it updates the value, which equals to the distance from the node to the infection station associated with its received station index, at the corresponding position of the array. Next, its integer counter increases by one and checks whether the value equals to k. Assuming that each edge is used to transmit at most k rumor messages in one direction, hence, there are at most $2k \cdot |E|$ messages. Note that the complexity of processing one message is $O(1)$. Therefore, our algorithm runs in $O(k \cdot |E|)$ in the worse case. □

CHAPTER 6

NOVEL ALGORITHMS FOR MAXIMUM DS DECOMPOSITION[1]

6.1 Introduction

In discrete optimization, submodular maximization for set functions is a hot topic, which attracts many scholars. In the past few decades, many important results for maximization of submodular set function have been proposed. Based on monotonicity and $f(\emptyset) = 0$, the pioneering work was done in 1978. Nemhauser et al., [67] firstly proposed a greedy algorithm which can get $1 - e^{-1}$ approximation ratio under cardinality constraint by adding a maximal marginal gains element to current solution in each iteration. In the same year, Fisher et al., [28] proved that the greedy strategy can get $(1 + p)^{-1}$ approximation with p matroid constraint. But, for the application, the quality of solution returned by greedy strategy is much better than the above approximation ratio. By constructing a metric called curvature $c \in [0, 1]$, Conforti et al., (1984) [17] proved that the greedy algorithm can get more tight approximation ratio $\frac{1-e^{-c}}{c}$ and $\frac{1}{1+c}$. That is why greedy strategy is excellent for monotone submodular maximization. These groundbreaking works have inspired a great deal of submodular optimization.

Beyond the monotonicity, the above algorithms are not good for submodular maximization. Until 2011, Feldman et al., [27] proposed a novel greedy algorithm which has e^{-1} approximation for non-monotone submodular maximization under matroids constraint. Further, by using double greedy, Buchbinder et al., (2014) [10] proved that non-monotone submodular maximization under cardinality constraint also has e^{-1} approximation polynomial algorithm. Besides, for maximization without constraints, Buchbinder et.al (2012) [9] also constructed an algorithm which can get $1/2$ approximation.

[1] ©2021. Reprinted, with permission, from S. Chen, W. Yang, S. Gao, and R. Jin. Novel algorithms for maximum DS decomposition. Theoretical Computer Science 857, 87-96.

Because these are some efficient algorithms, submodular optimization is widely utilized in data mining, machine learning, economics and operation research such as influence maximization (Kempe et al., (2003) [48]), active learning (Golovin et al., (2011) [31]), document summarization (Lin et al., (2011) [59]), and image segmentation (Jegelka et al., (2011) [42]).

In applications, unfortunately, a large number of objective functions are not submodular. Thus, how to optimize a general set function is the most important problem and has puzzled many researchers.

In this chapter, we study the non-submodular maximization problem based on DS decomposition about set functions, which is the difference between two monotone submodular functions. To solve this problem, we proposed Parameter Conditioned Greedy Algorithm by using the difference with parameter decomposition function and combining non-negative condition. There are general frameworks with a deterministic version and two random versions and users can choose rational parameters according to the property of problem.

6.1.1 Related Work

In this section, we introduce some previous works about non-submodular maximization problem which are related to our work directly.

On the one hand, some researchers proposed lots of definition about approximation of submodularity. Firstly, Krause et al., (2008) [53] constructed a metric called ϵ-Diminishing returns to evaluate what is the level of violation of marginal gains decreasing. On this basis, the authors proved that the standard greedy algorithm can get $f(X) \geq (1-e^{-1})(OPT-k\epsilon)$ approximation under cardinality constraint. Das and Kempe (2011) [21] introduced submodular ratio to measure violation about submodularity. Besides, they proved that the standard greedy strategy can get $f(X) \geq (1-e^{-\gamma})OPT$ approximation under cardinality constraint. Horel and Singer (2016) [40] proposed ϵ-approximation submodularity to limit non-submodular function. According to this metric, the authors proved that the standard greedy algorithm can return $f(X) \geq \frac{1}{1+\frac{4k\epsilon}{(1-\epsilon)^2}}(1 - e^{-1}(\frac{1-\epsilon}{1+\epsilon})^{2k}) \cdot OPT$ approximation under cardinality constraint. It is no surprise that computing these metrics is also a hard problem. It makes these results only theoretical meaning, but lack of practical application.

On the other hand, Lu et al., (2015) [60] proposed Sandwich Approach which chooses the best one solution of a submodular upper bound, a submodular lower bound and original problem. Because this approach can get a well-calculating parametric approximation, some researchers also applied Sandwich Approach to solve their application problems (Yang et al., (2020) [102], Yang et al., (2020) [103], Zhu et al., (2019) [114], Wang et al., (2017) [99]).

In addition, Iyer and Bilmes (2012) [41] proved that any set function can be expressed as the difference between two submodular set functions called DS decomposition. Specially, the two submodular set functions are monotone and non-decreasing. What's more, the authors proposed SubSup, SupSub, ModMod algorithms to solve the minimization of this problem.

Lemma 6.1 (Iyer and Bilmes (2012) [41])**.** *Any set function $h : 2^{\Omega} \to R$ can decompose the difference of two monotone non-decreasing submodular set functions f and g, i.e., $h = f - g$.*

This excellent result has been applied in many scenarios (Han et al., (2018) [35], Maehara et al., (2015) [65] and Yu et al., (2016) [106]). Because of the correspondence between submodularity and supermodularity, Li et al., (2020) [56] proved a variation of DS decomposition, i.e., any set function can be expressed as the difference of two monotone non-decreasing supermodular functions. Similarly, they also proposed greedy strategies like Iyer and Bilmes called ModMod and SupMod.

From the perspective of application, Bai and Bilmes (2018) [4] found that some set function can be expressed as the sum of a submodular and a supermodular function called BP decomposition, both of which are non-negative monotone non-decreasing. Interestingly, they proved that greedy strategy can get a $\frac{1}{k_f}[1 - e^{-(1-k_g)k_f}]$ and $\frac{1-k_g}{(1-k_g)k_f+p}$ approximation under cardinality and matroid constrained, where k_f, k_g are curvature about submodular and supermodular functions. Harshaw et al., (2019) [37] focused on the difference between a monotone non-negative submodular set function and a monotone non-negative modular set function. Besides, they proposed Distorted Greedy to solve this problem and get $f(S) - g(S) \geq (1 - e^{-\gamma})f(OPT) - g(OPT)$ approximation.

6.1.2 Contribution

Although DS decomposition has a bright application prospect, about general set function, there is a problem worth studying how to solve DS decomposition efficiently and effectively. Also, The Distorted Greedy strategy inspires our idea to solve maximization of DS decomposition. The major contribution of our work are as follows.

- Deterministic Parameter Conditioned Greedy is a deterministic framework under cardinality constrained problem. Under some rational assumption, the Deterministic Conditioned Greedy can get a polynomial approximation for maximum DS decomposition.

- To speed the Deterministic Parameter Conditioned Greedy, we use sampling to cover optimal solution set which can reduce ground set to a small set called Random Parameter Conditioned Greedy I. Besides, we prove it can get the same approximation ratio as Deterministic Parameter Conditioned Greedy.

- For maximization of DS decomposition under no constraint, we propose Random Parameter Conditioned Greedy II to solve it. Under some rational assumption, the Random Parameter Conditioned Greedy can get a polynomial approximation for maximum DS decomposition.

- We choose two special parameters to show our algorithms can get two novel approximations $f(S_k)-(e^{-1}-c_g)g(S_k) \geq (1-e^{-1})[f(OPT)-g(OPT)]$ and $f(S_k)-(1-c_g)g(S_k) \geq (1 - e^{-1})f(OPT) - g(OPT)$ respectively for cardinality constrained problem and also can get two novel approximations $E[f(S_k) - (e^{-1} - c_g)g(S_k)] \geq (1 - e^{-1})[f(OPT) - g(OPT)]$ and $E[f(S_k) - (1 - c_g)g(S_k)] \geq (1 - e^{-1})f(OPT) - g(OPT)$ respectively for unconstrained problem.

6.1.3 Organization

The rest of this chapter is organized as follow. In Section 6.2, we propose Deterministic Parameter Conditioned Greedy Algorithm to solve maximization of DS decomposition under cardinality constraint and use sample technique to speed our algorithm. As for without constraint, we design a Random Parameter Conditioned Greedy Algorithm II in Section 6.3. In Section 6.4, we choose some special parameters to analyze the efficiency of our framework. Conclusion and future works are in Section 6.5.

6.2 Maximization of DS Decomposition under Cardinality Constraint

In this section, we study the following problem, where f, g are the monotone non-decreasing submodular set functions, k is the cardinality constraint.

$$\begin{aligned} &\max_{X \subseteq \Omega} f(X) - g(X) \\ &s.t. \quad \|X\| \leq k \end{aligned} \tag{6.1}$$

6.2.1 Deterministic Parameter Conditioned Algorithm

Since Greedy Strategy is simple and efficient and gets some excellent constant approximation ratios for submodular optimization problem, many researchers use greedy strategy to deal with their problems. Therefore, we design a Deterministic Parameter Conditioned Greedy Algorithm to solve maximization DS decomposition with cardinality constraint. The details about Deterministic Parameter Conditioned Algorithm is in Algorithm 6.1.

From the Algorithm 6.1, Line 2-Line 8 is the main loop which needs to be executed

k times. Line 3 finds the element with maximal parameter decomposition marginal gains. Importantly, $A(i)$ and $B(i)$ are chosen by algorithm designers. In terms of practical applications, we assume $f(\emptyset) = g(\emptyset) = 0$, $A(i) \geq 0$ and $B(i) \geq 0$. Line 4 limits the maximal marginal gain must be positive, if negative means algorithm cannot add any elements in this loop because all elements may not have enough large marginal gain. It is dependent on our parameter how to choose. For example, when $A(i)$ is enough large and $B(i)$ is enough small, the condition of Line 4 is always satisfied. If designers choose well-defined parameters that are related to iteration round, it can get a wonderful approximation ratio about maximum DS decomposition. In this section, we assume $(1 - \frac{1}{k})A(i+1) - A(i) \geq 0$ and $B(i+1) - B(i) \geq 0$ specially. And then, we're going to go through the following process to explain why we're making this assumption.

Algorithm 6.1 Deterministic Parameter Conditioned Greedy

```
Input:Cardinality $k$, Parameters $A(i)$,$B(i)$
Output: $S_k$
 1: Initialize $S_0 \leftarrow \emptyset$
 2: for $i = 0$ to $k - 1$ do
 3:     $e_i \leftarrow \arg\max_{e \in \Omega} \{A(i+1)f(e|S_i) - B(i+1)g(e|\Omega \setminus e)\}$
 4:     if $A(i+1)f(e_i|S_i) - B(i+1)g(e_i|\Omega \setminus e_i) > 0$ then
 5:         $S_{i+1} \leftarrow S_i \cup \{e_i\}$
 6:     else
 7:         $S_{i+1} \leftarrow S_i$
 8:     end if
 9: end for
10: return $S_k$
```

To prove approximation ratio of Algorithm 6.1, firstly, we introduce a metric called curvature and two auxiliary functions which are useful in process of approximation proof.

Definition 6.1 (Conforti (1984) [17]). *Given a monotone submodular set function $f : 2^{\Omega} \to R$, the curvature of f is*

$$c_f = 1 - min_{e \in \Omega} \frac{f(e|\Omega \setminus e)}{f(e)}$$

Definition 6.2. *Define two auxiliary functions:*

$$\phi_i(T) = A(i)f(T) - B(i)\sum_{e\in T} g(e|\Omega \setminus e)$$
$$\psi_i(T,e) = \max\{0, A(i+1)f(e|T) - B(i+1)g(e|\Omega \setminus e)\}$$

From Definition 6.2, $\psi_i(T,e)$ is the condition of the Line 4 in Algorithm 6.1. What's more, $\phi_i(T)$ is the surrogate objective function. Clearly, if we take $\phi_i(T)$ as our objective function, Algorithm 6.1 is the classical greedy algorithm just combined non-negative condition. Next, we prove an important property about surrogate objective function.

Property 6.1. *In each iteration*

$$\phi_{i+1}(S_{i+1}) - \phi_i(S_i) = \psi_i(S_i, e_i) + [A(i+1) - A(i)]f(S_i) - [B(i+1) - B(i)]\sum_{e\in S_i} g(e|\Omega \setminus e)$$

Proof.

$$\begin{aligned}
&\phi_{i+1}(S_{i+1}) - \phi_i(S_i)\\
=& A(i+1)f(S_{i+1}) - B(i+1)\sum_{e\in S_{i+1}} g(e|\Omega \setminus e) - A(i)f(S_i) + B(i)\sum_{e\in S_i} g(e|\Omega \setminus e)\\
=& A(i+1)[f(S_{i+1} - f(S_i))] - B(i+1)g(e_i|\Omega \setminus e_i) + [A(i+1) - A(i)]f(S_i)\\
&- [B(i+1) - B(i)]\sum_{e\in S_i} g(e|\Omega \setminus e)\\
=& \psi_i(S_i, e_i) + [A(i+1) - A(i)]f(S_i) - [B(i+1) - B(i)]\sum_{e\in S_i} g(e|\Omega \setminus e)
\end{aligned}$$

□

Using the Property 6.1, we construct the relationship from Deterministic Parameter Conditioned Greedy to surrogate objective function. That is why we introduce the Definition 6.2. Coming from the proof of the approximation ratio of greedy algorithm in the submodular maximization, analyzing the marginal gain of surrogate objective function in each iteration is essential for approximation ratio for Algorithm 6.1. Interestingly, the condition of Algorithm 6.1 has a lower bound marginal gains and we prove it in Theorem 6.1.

Theorem 6.1. $\psi_i(S_i, e_i) \geq \frac{1}{k}A(i+1)[f(OPT) - f(S_i)] - \frac{1}{k}B(i+1)g(OPT)$

Proof.

$$\begin{aligned}
&k \cdot \psi_i(S_i, e_i) = k \cdot \max_{e \in \Omega}\{0, A(i+1)f(e|S_i) - B(i+1)g(e|\Omega \setminus e)\} \\
&\geq |OPT| \cdot \max_{e \in \Omega}\{0, A(i+1)f(e|S_i) - B(i+1)g(e|\Omega \setminus e)\} \\
&\geq |OPT| \cdot \max_{e \in OPT}\{A(i+1)f(e|S_i) - B(i+1)g(e|\Omega \setminus e)\} \\
&\geq \sum_{e \in OPT}[A(i+1)f(e|S_i) - B(i+1)g(e|\Omega \setminus e)] \\
&= A(i+1)\sum_{e \in OPT} f(e|S_i) - B(i+1)\sum_{e \in OPT} g(e|\Omega \setminus e) \\
&\geq A(i+1)[f(OPT \cup S_i) - f(S_i)] - B(i+1)g(OPT) \\
&\geq A(i+1)[f(OPT) - f(S_i)] - B(i+1)g(OPT)
\end{aligned}$$

The first inequality is $k \geq |OPT|$. The second inequality is $OPT \subseteq \Omega$. The third inequality is the maximum. The fourth inequality is submodularity, i.e.,$f(OPT \cup S_i) - f(S_i) \leq \sum_{e \in OPT} f(e|S_i)$ and $\sum_{e \in OPT} g(e|\Omega \setminus e) \leq g(OPT)$. The last inequality is monotony. □

Back to Deterministic Parameter Conditioned Algorithm, the following corollary is obvious by combining Property 6.1 and Theorem 6.1.

Corollary 6.1.

$$\begin{aligned}
&\phi_{i+1}(S_{i+1}) - \phi_i(S_i) \geq \frac{1}{k}A(i+1)f(OPT) - \frac{1}{k}B(i+1)g(OPT) + \\
&[(1 - \frac{1}{k})A(i+1) - A(i)]f(S_i) - [B(i+1) - B(i)]\sum_{e \in S_i} g(e|\Omega \setminus e)
\end{aligned}$$

According to Lemma 6.1, we have $\sum_{e \in S_i} g(e|\Omega \setminus e) \geq 0$ and $f(S_i) \geq 0$. That is to say the proper selection of $A(i)$ and $B(i)$ is the key to the final performance of the algorithm. If $[(1 - \frac{1}{k})A(i+1) - A(i)]f(S_i) - [B(i+1) - B(i)]\sum_{e \in S_i} g(e|\Omega \setminus e) \geq 0$, the result is trivial cause we can ignore them. As for other non-trivial parameter scenarios, it is an obstacle to analyze approximation ratio. That is why we assume $(1 - \frac{1}{k})A(i+1) - A(i) \geq 0$ and

$B(i+1) - B(i) \geq 0$. Therefore, we can get an approximation guarantee about maximum DS decomposition under cardinality constrained using Deterministic Parameter Conditioned Greedy.

Theorem 6.2. *S_k is the solution of Algorithm 6.1 after k iteration*

$$A(k)f(S_k) - (B(0) - c_g)g(S_k) \geq \frac{1}{k}\sum_{i=0}^{k-1}[A(i+1)f(OPT) - B(i+1)g(OPT)]$$

where c_g is the curvature of function g.

Proof. $\phi_0(S_0) = A(0)f(\emptyset) - B(0)g(\emptyset) = 0$ According the curvature $c_g = 1 - min\frac{g(e|\Omega\setminus e)}{g(e)}$, we have $\frac{g(e|\Omega\setminus e)}{g(e)} \geq 1 - c_g$,

$$\phi_k(S_k) = A(k)f(S_k) - B(k)\sum_{e\in S_k} g(e|\Omega \setminus e) \leq A(k)f(S_k) - B(k)(1-c_g)\sum_{e\in S_k} g(e)$$

$$\leq A(k)f(S_k) - B(k)(1-c_g)g(S_k)$$

Thus, we can rewrite an accumulation statement

$$A(k)f(S_k) - B(k)(1-c_g)g(S_k) \geq \phi_k(S_k) - \phi_0(S_0) = \sum_{i=0}^{k-1}[\phi_{i+1}(S_{i+1}) - \phi_i(S_i)]$$

$$\geq \sum_{i=0}^{k-1}[\frac{1}{k}A(i+1)f(OPT) - \frac{1}{k}B(i+1)g(OPT) + [(1-\frac{1}{k})A(i+1) - A(i)]f(S_i)]$$

$$- \sum_{i=0}^{k-1}[[B(i+1) - B(i)]\sum_{e\in S_i} g(e|\Omega \setminus e)]$$

$$\geq \frac{1}{k}\sum_{i=0}^{k-1}[A(i+1)f(OPT) - B(i+1)g(OPT)] - [B(k) - B(0)]\sum_{e\in S_k} g(e|\Omega \setminus e)$$

$$\geq \frac{1}{k}\sum_{i=0}^{k-1}[A(i+1)f(OPT) - B(i+1)g(OPT)] - [B(k) - B(0)]g(S_k)$$

Therefore, we can conclude

$$A(k)f(S_k) - (B(0) - c_g)g(S_k) \geq \frac{1}{k}\sum_{i=0}^{k-1}[A(i+1)f(OPT) - B(i+1)g(OPT)]$$

□

6.2.2 Random Parameter Conditioned Algorithm

Although greedy is an efficient algorithm to solve discrete problem, its complexity of computation is too high to accept in large scale problems such as social network influence maximization because it must consider all the elements in the ground set. To speed greedy, sampling is the widely utilized technique. Also, we use a sample set to cover the optimal solution for acceleration. Firstly, we prove an essential lemma for sample set.

Lemma 6.2. *$Pr[S \cap OPT \neq \emptyset] \geq (1-\epsilon)\frac{|OPT|}{k}$, if $|S| \geq \frac{-n\ln\epsilon}{k}$, where S is a sample set, OPT is the optimal solution set and n is the size of ground set.*

Proof. $Pr[S \cap OPT = \emptyset] \leq (1-\frac{|OPT|}{n})^{|S|} \leq e^{-|S|\frac{|OPT|}{n}} = e^{-\frac{|S|k}{n}\frac{|OPT|}{k}}$. The first inequality is the Bernoulli Distribution with $|S|$ times. The second inequality is $1-x \leq e^{-x}$. Thus, $Pr[S \cap OPT \neq \emptyset] \geq 1-e^{-\frac{|S|k}{n}\frac{|OPT|}{k}} \geq (1-e^{-\frac{|S|k}{n}})\frac{|OPT|}{k} \geq (1-\epsilon)\frac{|OPT|}{k}$ i.e $\epsilon \geq e^{-\frac{|S|k}{n}}$ where $1-e^{-ax} \geq (1-e^{-a})x$, if $x \in [0,1]$. Therefore, the sample size is $|S| \geq \frac{-n\ln\epsilon}{k}$. □

Algorithm 6.2 Random Parameter Conditioned Greedy I

```
Input: Cardinality $k$, Parameters $A(i)$,$B(i)$,$\epsilon$
Output: $S_k$
Initialize $S_0 \leftarrow \emptyset$
for $i = 0$ to $k-1$ do
    $S \leftarrow$ choose $|S|$ elements from ground set uniformly and randomly
    $e_i \leftarrow \arg\max_{e \in S} \{A(i+1)f(e|S_i) - B(i+1)g(e|\Omega \setminus e)\}$
    if $A(i+1)f(e_i|S_i) - B(i+1)g(e_i|\Omega \setminus e_i) > 0$ then
        $S_{i+1} \leftarrow S_i \cup \{e_i\}$
    else
        $S_{i+1} \leftarrow S_i$
    end if
end for
return $S_k$
```

Comparing Algorithm 6.1 and Algorithm 6.2, we speed algorithm just by reducing ground set to sample set. Intuitively, sample set is must smaller than ground set. For the proof of approximation ratio, we only need to prove the lower bound of the marginal gain. Other proofs are the same as the Algorithm 6.1.

Theorem 6.3. $E[\psi_i(S_i, e_i)] \geq \{\frac{1}{k}A(i+1)[f(OPT) - f(S_i)] - \frac{1}{k}B(i+1)g(OPT)\} \cdot (1-\epsilon)$

Proof.

$$\begin{aligned}
&E[\psi_i(S_i, e_i)] = E[\psi_i(S_i, e_i)|S \cap OPT \neq \emptyset]Pr[S \cap OPT \neq \emptyset] \\
&+ E[\psi_i(S_i, e_i)|S \cap OPT = \emptyset]Pr[S \cap OPT = \emptyset] \\
&\geq E[\psi_i(S_i, e_i)|S \cap OPT \neq \emptyset]Pr[S \cap OPT \neq \emptyset] \\
&= \max_{e \in S} \{0, A(i+1)f(e|S_i) - B(i+1)g(e|\Omega \setminus e)\}(1-\epsilon)\frac{|OPT|}{k} \\
&\geq \max_{e \in S \cap OPT} \{0, A(i+1)f(e|S_i) - B(i+1)g(e|\Omega \setminus e)\} \cdot (1-\epsilon)\frac{|OPT|}{k} \\
&\geq \frac{1}{|S \cap OPT|} \sum_{e \in S \cap OPT} [A(i+1)f(e|S_i) - B(i+1)g(e|\Omega \setminus e)] \cdot (1-\epsilon)\frac{|OPT|}{k} \\
&= \frac{1}{|OPT|} \sum_{e \in OPT} [A(i+1)f(e|S_i) - B(i+1)g(e|\Omega \setminus e)] \cdot (1-\epsilon)\frac{|OPT|}{k} \\
&= \{\frac{1}{k}A(i+1) \sum_{e \in OPT} f(e|S_i) - \frac{1}{k}B(i+1) \sum_{e \in OPT} g(e|\Omega \setminus e)\} \cdot (1-\epsilon) \\
&\geq \{\frac{1}{k}A(i+1)[f(OPT) - f(S_i)] - \frac{1}{k}B(i+1)g(OPT)\} \cdot (1-\epsilon)
\end{aligned}$$

The second inequality is $S \cap OPT \subseteq S$. The third inequality is that the largest value is greater than average value. And the following equality is that the sample set is chosen from ground set uniformly and randomly i.e., unbiased estimation. The third inequality is $k \geq |OPT|$. The fourth inequality is submodularity and monotonicity. □

Using the Theorem 6.3, we give the approximation ratio for Algorithm 6.2 without proof because this proof is the same as Theorem 6.2.

Theorem 6.4. *S_k is the solution of Algorithm 6.2 after k iteration*

$$E\{A(k)f(S_k) - (B(0) - c_g)g(S_k)\} \geq \frac{1}{k}\sum_{i=0}^{k-1}[(A(i+1) - \epsilon)f(OPT) - (B(i+1) - \epsilon)g(OPT)]$$

where c_g is the curvature of function g.

6.3 Maximization of DS Decomposition without Constraint

Maximization for non-monotone problem without constraint is an important problem in submodular optimization. The best approximation ratio for this problem is $\frac{1}{2}$ which is firstly proved by a random algorithm. Few years later, the deterministic algorithm also can get this ratio. Since randomness can bring a lot of uncertain factors and information, in some cases, random algorithms can get better approximation than deterministic algorithms. For non-monotone non-submodular maximization problem, so far, no algorithm has been able to give an acceptable approximation ratio. Therefore, we propose a Random Parameter Conditioned Greedy Algorithm II in this section. The following statement is the unconstraint problem for DS decomposition.

$$\max_{X\subseteq\Omega} h(X) = f(X) - g(X)$$

According to Definition 6.1, the Property 6.1 is also true for Random Parameter Conditioned Greedy II. We just have to modify the assumptions a little bit $(1-\frac{1}{n})A(i+1) - A(i) \geq 0$ to get the following theorems and corollary directly.

Theorem 6.5. $E[\psi_i(S_i, e)] \geq \frac{1}{n}A(i+1)[f(OPT) - f(S_i)] - \frac{1}{n}B(i+1)g(OPT)$

Proof.

$$\begin{aligned}
E[\psi_i(S_i, e_i)] &= \frac{1}{n} \cdot \sum_{e_i\in\Omega} \psi_i(S_i, e_i) \\
&\geq \frac{1}{n} \cdot \sum_{e_i\in OPT} [A(i+1)f(e_i|S_i) - B(i+1)g(e_i|\Omega \setminus e_i)] \\
&= \frac{1}{n}A(i+1) \sum_{e\in OPT} f(e_i|S_i) - \frac{1}{n}B(i+1) \sum_{e\in OPT} g(e_i|\Omega \setminus e_i) \\
&\geq \frac{1}{n}A(i+1)[f(OPT \cup S_i) - f(S_i)] - \frac{1}{n}B(i+1)g(OPT) \\
&\geq \frac{1}{n}A(i+1)[f(OPT) - f(S_i)] - \frac{1}{n}B(i+1)g(OPT)
\end{aligned}$$

□

Corollary 6.2. $E[\phi_{i+1}(S_{i+1}) - \phi_i(S_i)] \geq \frac{1}{n}A(i+1)f(OPT) - \frac{1}{n}B(i+1)g(OPT)$

$+ [(1-\frac{1}{n})A(i+1) - A(i)]f(S_i) - [B(i+1) - B(i)]\sum_{e \in S_i} g(e|\Omega \setminus e)$

Algorithm 6.3 Random Parameter Conditioned Greedy II

```
Input:Parameters $A(i)$,$B(i)$
Output: $S_n$
Initialize $S_0 \leftarrow \emptyset$
for $i = 0$ to $n-1$ do
    $e_i \leftarrow$ choose an elements from ground set uniformly and randomly.
    if $A(i+1)f(e_i|S_i) - B(i+1)g(e_i|\Omega \setminus e_i) > 0$ then
        $S_{i+1} \leftarrow S_i \cup \{e_i\}$
    else
        $S_{i+1} \leftarrow S_i$
    end if
end for
return $S_n$
```

Combined with Theorem 6.5 and Corollary 6.2, using the same method as Section 6.2, we can prove the Random Parameter Conditioned Greedy II can get the following approximation.

Theorem 6.6. *S_n is the solution of Algorithm 6.3 after n iteration*

$$E[A(n)f(S_n) - (B(0) - c_g)g(S_n)] \geq \frac{1}{n}\sum_{i=0}^{n-1}[A(i+1)f(OPT) - B(i+1)g(OPT)]$$

where c_g is the curvature of function g.

Proof.

$$
\begin{aligned}
&E[A(n)f(S_n) - B(n)(1-c_g)g(S_n)] \\
&\geq E[\phi_n(S_n)] - E[\phi_0(S_0)] = \sum_{i=0}^{n-1} E[\phi_{i+1}(S_{i+1})] - E[\phi_i(S_i)] \\
&\geq \sum_{i=0}^{n-1}[\frac{1}{n}A(i+1)f(OPT) - \frac{1}{n}B(i+1)g(OPT) + [(1-\frac{1}{n})A(i+1) - A(i)]f(S_i)] \\
&- \sum_{i=0}^{n-1}[B(i+1) - B(i)] \sum_{e\in S_i} g(e|\Omega \setminus e) \\
&\geq \frac{1}{n}\sum_{i=0}^{n-1}[A(i+1)f(OPT) - B(i+1)g(OPT)] - [B(n) - B(0)] \sum_{e\in S_n} g(e|\Omega \setminus e) \\
&\geq \frac{1}{n}\sum_{i=0}^{n-1}[A(i+1)f(OPT) - B(i+1)g(OPT)] - [B(n) - B(0)]g(S_n)
\end{aligned}
$$

Therefore, we can conclude

$$E[A(n)f(S_n) - (B(0) - c_g)g(S_n)] \geq \frac{1}{n}\sum_{i=0}^{n-1}[A(i+1)f(OPT) - B(i+1)g(OPT)]$$

□

From Theorem 6.5 and Theorem 6.6, we find an interesting phenomenon. If we decrease the number of iterations to k , this Random Parameter Conditioned Greedy II can also be used in problem with cardinality constrained. Since this proof is similar with Theorem 6.6, we just give the statement without proof.

Theorem 6.7. *S_k is the solution of Algorithm 6.3 after k iteration*

$$E[A(k)f(S_k) - (B(0) - c_g)g(S_k)] \geq \frac{1}{n}\sum_{i=0}^{k-1}[A(i+1)f(OPT) - B(i+1)g(OPT)]$$

where c_g is the curvature of function g.

6.4 Parameter Analyzing

In this section, we choose two special case to show our Parameter Conditioned Greedy framework. What's more, we compare the marginal gains under the different assumption about $A(i)$ and $B(i)$.

6.4.1 Case 1

In this case, we set $A(i) = B(i) = (1 - \frac{1}{k})^{(k-i)}$. Therefore, the Definition 6.1 becomes the following. Obviously, these settings satisfy all conditions and assumptions in Section 6.2 and Section 6.3. Hence, the following results are clearly. Because the proofs are similar to Section 6.2 and Section 6.3, we omit them here.

$$\phi_i(T) = (1 - \frac{1}{k})^{k-i}[f(T) - \sum_{e \in T} g(e|\Omega \setminus e)]$$

$$\psi_i(T, e) = \max\{0, (1 - \frac{1}{k})^{k-(i+1)}[f(e|S_i) - g(e|\Omega \setminus e)]\}$$

Property 6.2. *In each iteration*

$$\phi_{i+1}(S_{i+1}) - \phi_i(S_i) = \psi_i(S_i, e_i) + \frac{1}{k}(1 - \frac{1}{k})^{-1}\phi_i(S_i)$$

Theorem 6.8. *In Deterministic Parameter Conditioned Algorithm,* $\psi_i(S_i, e_i) \geq \frac{1}{k}(1 - \frac{1}{k})^{k-(i+1)}[f(OPT) - f(S_i) - g(OPT)]$. *In Random Parameter Conditioned Algorithm I,* $E[\psi_i(S_i, e_i)] \geq \{\frac{1}{k}(1 - \frac{1}{k})^{k-(i+1)}[f(OPT) - f(S_i) - g(OPT)]\} \cdot (1 - \epsilon)$. *In Random Parameter Conditioned Algorithm II,* $E[\psi_i(S_i, e_i)] \geq \frac{1}{n}(1 - \frac{1}{n})^{n-(i+1)}[f(OPT) - f(S_i) - g(OPT)]$

Theorem 6.9. *The Deterministic Parameter Conditioned Algorithm can return* $f(S_k) - (e^{-1} - c_g)g(S_k) \geq (1 - e^{-1})[f(OPT) - g(OPT)]$ *approximation ratio solution for cardinality constraint problem. The Random Parameter Conditioned Algorithm I can return* $E[f(S_k) - (e^{-1} - c_g)g(S_k)] \geq (1 - e^{-1} - \epsilon)[f(OPT) - g(OPT)]$ *approximation ratio solution for cardinality constraint problem. The Random Parameter Conditioned Algorithm II can return* $E[f(S_n) -$

$(e^{-1} - c_g)g(S_n)] \geq (1 - e^{-1})[f(OPT) - g(OPT)]$ for without constraint problem. Where c_g is the curvature of function g.

From Theorem 6.9, we find an interesting result and get the following corollary, when $c_g = 0$, i.e., the submodular function g is a modular function.

Corollary 6.3. *If $c_g = 0$, i.e., g is modular. For non-monotone submodular maximization with cardinality constraint, Algorithm 6.1 has $f(S_k) - e^{-1}g(S_k) \geq (1 - e^{-1})[f(OPT) - g(OPT)]$ approximation ratio, Algorithm 6.2 has $E[f(S_k) - e^{-1}g(S_k)] \geq (1 - e^{-1} - \epsilon)[f(OPT) - g(OPT)]$ approximation ratio. For non-monotonicity submodular maximization without constraint, Algorithm 6.3 has $E[f(S_n) - e^{-1}g(S_n)] \geq (1 - e^{-1})[f(OPT) - g(OPT)]$ approxiamtion ratio.*

6.4.2 Case 2

In this case, we set $A(i) = (1 - \frac{1}{k})^{(k-i)}$, $B(i) = 1$. Similarly, the Definition 6.1 become the following and these settings satisfy all condition in Section 2 and Section 3. Because the proofs are similar with Section 6.2 and Section 6.3, in this subsection, we have omitted all proofs.

$$\phi_i(T) = (1 - \frac{1}{k})^{k-i} f(T) - \sum_{e \in T} g(e|\Omega \setminus e)$$

$$\psi_i(T, e) = \max\{0, (1 - \frac{1}{k})^{k-(i+1)} f(e|S_i) - g(e|\Omega \setminus e)\}$$

Property 6.3. *In each iteration*

$$\phi_{i+1}(S_{i+1}) - \phi_i(S_i) = \psi_i(S_i, e_i) + \frac{1}{k}(1 - \frac{1}{k})^{k-(i+1)} f(S_i)$$

Theorem 6.10. *In Deterministic Parameter Conditioned Algorithm, $\psi_i(S_i, e_i) \geq \frac{1}{k}(1 - \frac{1}{k})^{k-(i+1)}[f(OPT) - f(S_i)] - \frac{1}{k}g(OPT)$. In Random Parameter Conditioned Algorithm I, $E[\psi_i(S_i, e_i)] \geq \{\frac{1}{k}(1 - \frac{1}{k})^{k-(i+1)}[f(OPT) - f(S_i)] - \frac{1}{k}g(OPT)\} \cdot (1 - \epsilon)$. In Random Parameter Conditioned II, $E[\psi_i(S_i, e_i)] \geq \frac{1}{n}(1 - \frac{1}{n})^{n-(i+1)}[f(OPT) - f(S_i)] - \frac{1}{n}g(OPT)$.*

Theorem 6.11. *The Deterministic Parameter Conditioned Algorithm can return $f(S_k) - (1 - c_g)g(S_k) \geq (1 - e^{-1})f(OPT) - g(OPT)$ approximation ratio solution for cardinality constraint. The Random Parameter Conditioned Algorithm I can return $E[f(S_k) - (1 - c_g)g(S_k)] \geq (1-e^{-1}-\epsilon)f(OPT)-(1-\epsilon)g(OPT)$ approxiamtion solution ratio for cardinality constraint. The Random Parameter Conditioned Algorithm II can return $E[f(S_n) - (1 - c_g)g(S_n)] \geq (1-e^{-1})f(OPT)-g(OPT)$ approximation ration for without constraint. Where c_g is the curvature of function g.*

Corollary 6.4. *If $c_g = 0$, i.e., g is modular. For non-monotone submodular maximization with cardinality constraint, Algorithm 6.1 has $f(S_k)-g(S_k) \geq (1-e^{-1})f(OPT)-g(OPT)$ approximation ratio, Algorithm 6.2 has $E[f(S_k) - g(S_k)] \geq (1 - e^{-1} - \epsilon)f(OPT) - (1 - \epsilon)g(OPT)$ approximation ratio. For non-monotone submodular maximization without constraint, Algorithm 6.3 has $E[f(S_n) - g(S_n)] \geq (1 - e^{-1})f(OPT) - g(OPT)$ approxiamtion ratio.*

Remark 6.1. *Clearly, if g is modular, then $h = f - g$ is submodular. The above approximations are different with submodular maximization under cardinality constrained problem $(1 - e^{-1})$. And also, the different parameters can also cause different approximations. We think first gap is caused by non-monotony, because h is not always monotonous. The second gap give us a clue that we can choose the appropriate parameters $A(i)$ and $B(i)$ according to the characteristics of the problem to get a better approximate ratio. The above approximations are different with non-monotone submodular maximization under unconstrained problem $1/2$. But we cannot measure which one is better than others. In some cases, our approximations may be better than $1/2$.*

Remark 6.2. *The conditions in Line 4 of Algorithm 6.1, 6.3 and Line 5 in Algorithm 6.2 are necessary. Firstly, if we don't add non-negative condition, we may add some bad elements which has the lower marginal gains to the solution. Combined the positive condition, it makes* *the large marginal gains element will be added into solution, because the conditions in Line 4 of Algorithm 6.1, 6.3 and Line 5 in Algorithm 6.2 are a lower bound of marginal gains. Also, we assume* $(1-\frac{1}{k})A(i+1)-A(i)\geq 0$ *and* $B(i+1)-B(i)\geq 0$ *i.e., the parameter of* $A(i)$ *and* $B(i)$ *is increasing with index* i *increase. If an element* e *is added into solution in loop* i*, the true marginal gains will be enough large because* $A(k)=B(k)=1$*.*

6.5 Conclusions

In this chapter, we propose Parameter Conditioned Greedy strategy with deterministic and random which are general frameworks for maximum DS decomposition under cardinality constrained and unconstrained respectively. Users can choose some rational parameters to fit special practical problems and get a wonderful approximation about the problem. Also, we choose two special cases to show that our strategy can get some novel approximation. In some situations, these novel approximations are better than the best approximation at the state of art.

In the future works, how to remove the curvature parameter in approximation ratio is important, because it can make the approximation much tight. Whats more, how to select $A(i)$ and $B(i)$ so that the algorithm can achieve the optimal approximation ratio is also an urgent problem to be solved.

CHAPTER 7

BLACK BOX AND DATA-DRIVEN COMPUTATION[1]

7.1 Introduction

Recently, data-driven is a trendy terminology in computational study. In data-driven computation, there exists a black box that contains a big amount of data, e.g., solutions for a certain problem. When the computation needs to solve a problem, its solution can be obtained from the black box by a data mining method, such as a machine learning method, instead of computing from the scratch. Today, this frame of computation is available due to the study of big data such that the technology for big data management and mining is already successfully established. The current data-driven computation is an application of such technologies.

It is not the first time that the black box appears in studying computation. However, previously, the black box plays a different role; it usually represents something unable to compute or hard to compute. Now, it represents something already known. What makes this change of its role? In machine learning, it is made by data training, that is, when the black box accumulates a large enough amount of data through preprocessing, the black box can be used for solving a problem with a machine learning method. In this short chapter, we would like to present some observations based on this idea.

There are many concepts and notations in computational complexity theory, which will be used in this chapter. Readers can find these concepts and notations in [26].

[1] ©2021. Reprinted, with permission, from R. Jin, W. Wu, M.T. Thai, DZ. Du. Black-Box and Data-Driven Computation. In: Pardalos P.M., Rasskazova V., Vrahatis M.N. (eds) Black Box Optimization, Machine Learning, and No-Free Lunch Theorems. Springer Optimization and Its Applications, vol 170, pp 163-168. Springer, Cham.

7.2 Black Box and Oracle

An oracle in a Turing machine is a black box, which is like a subroutine in a computer program. Usually, the oracle is represented by a language or a function. If it is represented by a language A, then the oracle would be able to give a solution of the membership problem about A, that is, given an input string x, the oracle can tell whether x does belong to A or not. If an oracle is represented by a function f, then it can compute the function f, that is, given an input string x, the oracle returns a string $f(x)$. The oracle is considered as a black box because we do not care how the oracle obtains the output from the input and, hence, computational process inside of the oracle is black.

When analyzing the running time complexity and the space usage of an oracle Turing machine, only the computation outside the oracle is considered. In this way, fixed a language oracle A, a sequence of computational complexity classes P^A, NP^A, $PSPACE^A$, ..., called *relativized* classes, can be defined similarly to the well-known classes P, NP, $PSPACE$, ... The first interesting result on relativized classes came from Baker et al., [5] as follows.

Theorem 7.1. *There exists a language oracle A such that $P^A = NP^A$. There also exists a language oracle B such that $P^B \neq NP^B$.*

What is the significance of this result? To explain it, let us consider the following hierarchy theorem [26].

Theorem 7.2 (Time Hierarchy Theorem). *If t_2 is a fully time-constructible function, $t_2(n) \geq t_1(n) \geq n$, and*

$$\lim_{n\to\infty} \frac{t_1(n)\log(t_1(n))}{t_2(n)} = 0$$

then $DTIME(t_2(n)) \setminus DTIME(t_1(n)) \neq \emptyset$.

This theorem is proven using the diagonalization argument and is an important tool for separating complexity classes. However, Theorem 7.1 indicates that the time hierarchy theorem cannot succeed to separate classes P and NP. The reason is as follows: With the same argument, the time hierarchy theorem for relativized complexity classes can also be established.

Theorem 7.3. *If t_2 is a fully time-constructible function, $t_2(n) \geq t_1(n) \geq n$, and*

$$\lim_{n\to\infty} \frac{t_1(n)\log(t_1(n))}{t_2(n)} = 0$$

then $DTIME^C(t_2(n)) \setminus DTIME^C(t_1(n)) \neq \emptyset$ where C is any language oracle.

If the time hierarchy theorem can solve the $P - NP$ problem, then following the same logic pattern, Theorem 7.3 would be able to derive either $P^C = NP^C$ or $P^C \neq NP^C$. However, by Theorem 7.1, in case $C = A$, $P^C = NP^C$, and in case $C = B$, $P^C \neq NP^C$, a contradiction.

Actually, there exists many variations of diagonalization argument in the recursive function theory. Theorem 7.1 indicates that they are all not powerful enough to solve the $P-NP$ problem.

There exists many efforts made following the work of Baker et al., [5]. Among them, it is worth mentioning those about relativized polynomial-time hierarchy class HP^A. Is there a language oracle A such that $PH^A \neq PSPACE^A$? This problem was identified as a hard problem and was solved in two steps.

In the first step, [30] established a relationship between this problem and the circuit complexity of parity function

$$p(x_1, x_2, ..., x_n) = x_1 \oplus x_2 \oplus \cdots \oplus x_n$$

where $\oplus$ is the exclusive-or operation, i.e.,

$$x \oplus y = \begin{cases} 1 & \text{if there is exactly one 1 in } \{x, y\}; \\ 0 & \text{otherwise.} \end{cases}$$

The language defined by the parity function is

$$P = \{x_1x_2\cdots x_n \mid p(x_1, x_2, ..., x_n) = 1\}.$$

They showed that if the language P does not belong to class AC^0, then there exists a language oracle A such that $PH^A \neq PSPACE^A$.

In the second step, Yao [105] showed that the language P does not belong to AC^0. This result was one of the two big results appearing in 1985 about circuit complexity. Yao also claimed that he proved an existence of a language oracle A such that $\Sigma_k^{p,A} \neq \Sigma_{k+1}^{p,A}$ for any natural number k. However, his proof did not get a chance to be published due to Hastads work. Hastad [38] simplified Yao's proof in [105] and, meanwile, published his simplified proof for Yao's claim.

Later, Ko [50, 49, 51, 52] showed a sequence of results about relativized classes about polynomial-time hierarchy, e.g., for any integer $k \geq 1$, there exists a language oracle A such that $PSPACE^A \neq PH^A = \Sigma_k^{p,A} \neq \Sigma_{k-1}^{p,A}$ [50].

7.3 Reduction

An important part of computational complexity theory is to study the complete problem in each complexity class, and the completeness is usually established through certain reduction. The reduction can be seen as an application of black box. For example, initially, the NP-complete problem was established through the polynomial-time Turing reduction, also called Cook-reduction today. For two languages A and B, A is said to be Cook-reducible to B if the membership problem of A can be solved in polynomial-time with a black box B. A problem in NP is called an NP-complete problem B if every problem A in NP is Cook-reducible to B.

The complete problem in each complexity class is the hardest problem in the class, related to certain type of computational problems. For example, the NP-complete problem is hardest with respect to nondeterministic polynomial-time computation. The P-complete problem is hardest with respect to efficient parallel computation. Therefore, the reduction is used for establishing the hardness of problems in the past.

Data-driven computation brings a new role for reduction, which is used not only for establishing the hardness of a problem, but also for providing solutions to some problems. For example, if a black box collects a large amount of data about solutions of an NP-complete problem, then all problems in NP would get "efficient" (in certain sense) solutions through the polynomial-time reduction. What new issues would this new role of reduction be introduced? Let us present some of our observations in the next section.

7.4 Data-Driven Computation

When the black box is opened by data-driven computation, the following new issues may need to be considered.

Choice of Black Box. First, note that due to current technology on data management, a black box may accumulate a very large amount of data; however, the quantity is still finite and cannot be infinitely large. Therefore, it cannot contain solutions of a problem for all inputs. This means that the data-driven algorithm aims mainly at practical solutions, not at theoretical. In a practical point of view, "polynomial-time" is not enough to describe the efficiency. For example, nobody would like to implement an algorithm with a running time $O(n^{100})$ to solve a practical problem. Possibly, an algorithm with running time at most $O(n^3)$ may be more practical. A question is motivated from this consideration: Is there an NP-complete problem A such that for any problem B in class NP, B can be reduced to A via a reduction with the running time at most $O(n^k)$ for a fixed number k? Following theorem gives us a negative answer.

Theorem 7.4. *For any integer $k \geq 1$, there is no NP-complete problem A such that for every problem B in NP, there exists a reduction, running in time at most $O(n^k)$, from B to A.*

Proof. Consider an NP-complete problem A. Note that

$$\lim_{n \to \infty} \frac{n^k \log n^k}{n^{2k}} = 0.$$

By Theorem 7.3, there exists a language $B \in DTIME^A(n^{2k}) \setminus DTIME^A(n^k)$. This means that B can be Cook-reducible to A. However, the reduction cannot be running in time $O(n^k)$. Therefore, A cannot be the NP-complete problem satisfying the condition in the theorem. □

This theorem indicates that there does not exist an NP-complete problem that can be used for solving all problems in NP practically through reduction and using data-driven method. Therefore, we have to make a proper choice of black box for each small class of the real-world problems.

Speed Up Reduction. There already exists many reductions in the literature for the proof of NP-completeness. In the past, these reductions only need to have the polynomial running time. Under the new setting of its new roles, we may want to speed up the reduction methods. Is it always possible to speed them up? Following negative answer is a corollary of Theorem 7.4.

Corollary 7.1. *For any integer $k \geq 1$ and any NP-complete problem A, there exists a problem in NP such that any reduction from B to A cannot run within time $O(n^k)$.*

Multiple Black Boxes. A computer program may contain more than one subroutine. When data-driven technique is used, it is possible to use more than one black box. In the study of computational complexity theory for discrete problems, no effort has been made on more than one oracle. However, with the similar way, a time hierarchy theorem may be established regarding to more than one language oracle, by which the following result can be proven with a similar argument to the proof of Theorem 7.4.

Theorem 7.5. *For any integer $k \geq 1$, there do not exist a finite number of NP-complete languages A_1, A_2, ..., A_h such that the membership problem of every language B in NP can be solved by a $O(n^k)$-algorithm with language oracles A_1, A_2, ..., A_h.*

It is worth mentioning that more than one black boxes may be found in the information-based complexity theory for continuous problems [92]. Therefore, we may get some research ideas from there.

Complexity Issues. To build a model for the data-driven computation, we have to face a computation with mixed of uniform and nonuniform computations, that is, carry out uniform computation outside the black box while implementing nonuniform computation inside the black box. This may bring to new issues in the study of computational complexity theory.

CHAPTER 8

CONCLUSION AND FUTURE WORK

8.1 Conclusion

In the dissertation, we have studied social network analysis in big data harnessing (known as *data mining*) and related optimization problems in the following aspects.

1. We have studied two types of practical optimization problems on social influence in homogeneous online social networks respectively as follows.

- For the influence maximization discount allocation problem in a practical online viral marketing scenario, we proposed an online full-feedback setting model to solve the discount allocation problem. To be specific, we divided the cascade into seed selection and information diffusion stages. We then proposed an online discount allocation policy for seed selection and introduced a monotone and submodular utility function which aims to influence at least a certain number of people who adopt the target product with the minimal cost in expectation for marketers. We further proposed two online discount allocation greedy algorithms and analyzed performance gaurantee with an approximation ratio under uniform and non-uniform discount situations, respectively. The effectiveness of the proposed algorithms has been evaluated on real-world online social networks datasets in Section 3.6.

- For the influence minimization on rumor controlling problem, we focused mainly on the rumor source detection problem in online social networks. We firstly analyzed existing main schemes for modeling rumor diffusion process as well as schemes for estimating rumor source based on the specific diffusion model and network graphs in the problems. We then theoretically studied a station-based approach for online rumor source detection under the influence cascade (IC) model. We derived a rumor source estimator based on the concept of eccentricity in graph theory [36] and employed the concept of k-center problem to establish the k-station selection problem. We also developed two algorithms to find the estimator and k stations respectively for regular tree networks.

2. We have developed novel algorithms for maximization of DS decomposition in the study of set function optimization. These algorithms can be general frameworks and users can choose rational parameters according the property of problems. More specifically, we proposed a Deterministic Conditioned Greedy Algorithm 6.1 to solve maximization of DS decomposition under cardinality constraint in Section 6.2, and we also proposed a Random Conditioned Greedy Algorithm 6.3 to solve maximization of DS decomposition without constraint, in which we used the difference with parameter decomposition function. Under each algorithm, we carried out special cases that can get novel approximation ratios.

3. We have discussed a frame of data-driven computation that has utilized black box as a tool for proving solutions to some computational problems, because black box has been an important tool in studying computational complexity theory and has been used for establishing the hardness of problems. Through our findings, we believe that some new reserach ideas can be raised from here and they are worth to study.

8.2 Future Work

There are many challenging research fields within the study of data harnessing (or called *data mining*) in social networks. Here we illustrate a few of them.

Submodularity of Objective Function. Related to optimiztion problems in social networks, most of the objective functions of the problems are shown as non-submodular. Although there are existing literature to solve it such as [3] proved that any non-submodular function could decompose as a difference of two submodular functions. Another strategy named “sandwich approximation” was introduced by [60], which approximates the objective function by formulating its lower bound and upper bound. Later, [75] presented strong inapproximability results for the problem objective function that was non-submodular, which is called the 2-quasi-submodular. In fact, the modularity of objective function is defined with respect to various practical problems. How do we deal with non-submodularity of the problems upon or beyond the existing methods? This is a challenge research work for future.

Diffusion Analysis in Heterogeneous Social Networks. Regarding diffusion analysis in social networks, a majority of existing network models have been assumed as homogeneous social networks, where nodes are objects of the same entity type (e.g., person) and links are relationships from the same relation type (e.g., friendship or interaction). Interesting results have been generated from such studies with numerous influential applications, such as the well-known Google PageRank algorithm [8]. However, most real-world networks are heterogeneous, where nodes and edges are of different types of data. For example, the network of Facebook consists of persons as well as other types of data, like posts, schools, movies, photos, and so on. In addition to interactions between persons, there are other types of links, such as the person-school relationship, the person-post relationship, and so on. Therefore, harnessing diffusion on homogeneous networks may miss important semantic information of data. In the future, it is highly required to model diffusion based on heterogeneous networks that can better represent the real-world networks.

Representation Learning for Social Network Analysis. Another central problem in social network data mining is how to extract useful features from non-Euclidean structured networks, then in turn to enable the setting of downstream models for specific analysis. For example, in the case of viral marketing in an online social network, the main challenge might be how to embed network nodes into a low-dimensional space so that the closeness between nodes could be easily measured with distance metrics. To deal with it, earlier efforts primarily rely on labeled features, such as graph kernels [93], node classification in graph

[7], or link prediction on node proximity [57]. However, such feature processing has turned out to be very time-consuming and ineffective for many real-world applications. Fortunately, representation learning has been proposed to avoid this limitation as it can automatically learn feature representations that capture various information data in networks (Bengio et al., [6]). The key component in the representation learning is a transformation function that maps nodes, or subgraphs as vectors to a low-dimensional feature space, where the spatial relations between the vectors reflect the structures or contents in the original network. Subsequently, some machine learning models like clustering models and detection models could be directly used to target network applications by given projected low-dimensional feature vectors from high-dimensional space. In recent years, although there has seen a surge in leveraging representation learning techniques for information network analysis, existing embedding methods mainly deal with homogeneous networks. It is because, as mentioned, that heterogeneous networks extracted from real-life applications have multiple types of nodes or edges. In this case, it is hard to evaluate semantic proximity between different network elements in the low-dimensional space. There still remains many tasks on learning embeddings for heterogeneous networks, such as techniques that are required to fully capture the relations between different types of network objects to comprehensively model practical applications. This direction is worth further exploring.

Data-driven Computational Social Networks. With the surge of big Web-data currently available, this allows us to study social interactions and investigate about social behavior on a scale thorough this wealth of data. The goals are to generate a greater understanding of social interactions and social behaviors in online networks through data analysis, to develop reliable and scalable data-driven framework that can model social processes, and ultimately to address the target optimization problem in this data-driven framework. How to create such data-driven models for social network analysis is an interesting work for future research.

REFERENCES

[1] Abebe, R., L. A. Adamic, and J. Kleinberg (2018). Mitigating overexposure in viral marketing. In *Thirty-Second AAAI Conference on Artificial Intelligence.*

[2] Antulov-Fantulin, N., A. Lančić, T. Šmuc, H. Štefančić, and M. Šikić (2015, Jun). Identification of patient zero in static and temporal networks: Robustness and limitations. *Phys. Rev. Lett. 114*, 248701.

[3] Bach, F. (2013). Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning 6*(2-3), 145–373.

[4] Bai, W. and J. Bilmes (2018, 10–15 Jul). Greed is still good: Maximizing monotone Submodular+Supermodular (BP) functions. In *Proceedings of the 35th International Conference on Machine Learning*, Volume 80 of *Proceedings of Machine Learning Research*, pp. 304–313. PMLR.

[5] Baker, T., J. Gill, and R. Solovay (1975). Relativizations of the p=pnp question. *SIAM Journal on Computing 4*(4), 431–442.

[6] Bengio, Y., A. Courville, and P. Vincent (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*(8), 1798–1828.

[7] Bhagat, S., G. Cormode, and S. Muthukrishnan (2011). *Node Classification in Social Networks. In: Aggarwal C. (eds) Social Network Data Analytics.*, pp. 115–148. Boston, MA: Springer US.

[8] Brin, S. and L. Page (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems 30*(1), 107–117. Proceedings of the Seventh International World Wide Web Conference.

[9] Buchbinder, N., M. Feldman, J. Naor, and R. Schwartz (2012). A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, pp. 649–658.

[10] Buchbinder, N., M. Feldman, J. S. Naor, and R. Schwartz (2014). Submodular maximization with cardinality constraints. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, USA, pp. 14331452. Society for Industrial and Applied Mathematics.

[11] Chami, I., S. Abu-El-Haija, B. Perozzi, C. Ré, and K. Murphy (2020). Machine learning on graphs: A model and comprehensive taxonomy. *ArXiv abs/2005.03675.*

[12] Chen, X., X. Hu, and C. Wang (2016, January). Approximation for the minimum cost doubly resolving set problem. *Theor. Comput. Sci. 609*(P3), 526–543.

[13] Chen, Z., K. Zhu, and L. Ying (2016). Detecting multiple information sources in networks under the sir model. *IEEE Transactions on Network Science and Engineering 3*(1), 17–31.

[14] Choi, J., S. Moon, J. Woo, K. Son, J. Shin, and Y. Yi (2017). Rumor source detection under querying with untruthful answers. In *IEEE INFOCOM 2017*, pp. 1–9.

[15] Choi, J. and Y. Yi (2018). Necessary and sufficient budgets in information source finding with querying: Adaptivity gap. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pp. 2261–2265. IEEE.

[16] Comin, C. H. and L. da Fontoura Costa (2011, Nov). Identifying the starting point of a spreading process in complex networks. *Phys. Rev. E 84*, 056105.

[17] Conforti, M. and G. Cornujols (1984). 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.

[18] Cooke, K. and P. van den Driessche (1996, December). Analysis of an seirs epidemic model with two delays. *Journal of mathematical biology 35*(2), 240–260.

[19] Da, L. (2014). Microblog information diffusion:simulation based on sir model. *Journal of Beijing University of Posts and Telecommunications*, 28–33.

[20] D'Angelo, G., L. Severini, and Y. Velaj (2016). Influence maximization in the independent cascade model. *CEUR Workshop Proceedings 1720*, 269–274.

[21] Das, A. and D. Kempe (2011). Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, Madison, WI, USA, pp. 10571064. Omnipress.

[22] Dhamal, S. (2018). Effectiveness of diffusing information through a social network in multiple phases. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7. IEEE.

[23] Domingos, P. and M. Richardson (2001). Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 57–66. ACM.

[24] Dong, M., B. Zheng, N. Quoc Viet Hung, H. Su, and G. Li (2019). Multiple rumor source detection with graph convolutional networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, New York, NY, USA, pp. 569–578. Association for Computing Machinery.

[25] Dong, W., W. Zhang, and C. W. Tan (2013). Rooting out the rumor culprit from suspects. In *2013 IEEE International Symposium on Information Theory*, pp. 2671–2675.

[26] Du, D.-Z. and K.-I. Ko (2014). *Theory of Computational Complexity (2nd Ed)*. New York, NY: John Wiley and Sons.

[27] Feldman, M., J. Naor, and R. Schwartz (2011). A unified continuous greedy algorithm for submodular maximization. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pp. 570–579.

[28] Fisher, M. L., G. L. Nemhauser, and L. A. Wolsey (1978). An analysis of approximations for maximizing submodular set functionsii. *Mathematical Programming 8*(1), 73–87.

[29] Fuchs, M. and P.-D. Yu (2015). Rumor source detection for rumor spreading on random increasing trees. *Electronic Communications in Probability 20*, 1–12.

[30] Furst, M., J. B. Saxe, and M. Sipser (1984). Parity, circuits, and the polynomial-time hierarchy. *Mathematical systems theory 17*(1), 13–27.

[31] Golovin, D. and A. Krause (2011). Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research 42*(1), 427–486.

[32] Goyal, A., F. Bonchi, and L. V. Lakshmanan (2011). A data-based approach to social influence maximization. *Proceedings of the VLDB Endowment 5*(1), 73–84.

[33] Guille, A., H. Hacid, C. Favre, and D. A. Zighed (2013, July). Information diffusion in online social networks: A survey. *SIGMOD Rec. 42*(2), 17–28.

[34] Han, K., K. Huang, X. Xiao, J. Tang, A. Sun, and X. Tang (2018). Efficient algorithms for adaptive influence maximization. *Proceedings of the VLDB Endowment 11*(9), 1029–1040.

[35] Han, K., C. Xu, F. Gui, S. Tang, H. Huang, and J. Luo (2018, 06). Discount allocation for revenue maximization in online social networks. pp. 121–130.

[36] Harary, F. (1969). *Graph Theory*. Reading, Massachusetts: Addison-Wesley.

[37] Harshaw, C., M. Feldman, J. Ward, and A. Karbasi (2019). Submodular maximization beyond non-negativity: Guarantees, fast algorithms, and applications. 36th International Conference on Machine Learning, pp. 2634–2643.

[38] Hastad, J. (1986). Almost optimal lower bounds for small depth circuits. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC '86, New York, NY, USA, pp. 620. Association for Computing Machinery.

[39] Hethcote, H. W. (2000). The mathematics of infectious diseases. *SIAM Review 42*(4), 599–653.

[40] Horel, T. and Y. Singer (2016). Maximization of approximately submodular functions. In *Advances in neural information processing systems*.

[41] Iyer, R. and J. Bilmes (2012). Algorithms for approximate minimization of the difference between submodular functions, with applications. Arlington, Virginia, USA. AUAI Press.

[42] Jegelka, S. and J. Bilmes (2011). Submodularity beyond submodular energies: Coupling edges in graph cuts. In *CVPR 2011*, pp. 1897–1904.

[43] Jiang, J., S. Wen, S. Yu, Y. Xiang, and W. Zhou (2017). Identifying propagation sources in networks: State-of-the-art and comparative studies. *IEEE Communications Surveys and Tutorials 19*(1), 465–481.

[44] Jiang, J., S. Wen, S. Yu, Y. Xiang, and W. Zhou (2018). Rumor source identification in social networks with time-varying topology. *IEEE Transactions on Dependable and Secure Computing 15*(1), 166–179.

[45] Jin, R. and W. Wu (2021). Schemes of propagation models and source estimators for rumor source detection in online social networks: A short survey of a decade of research. *Discrete Mathematics, Algorithms and Applications 13*(04), 2130002.

[46] Jin, Y., W. Wang, and S. Xiao (2007). An sirs model with a nonlinear incidence rate. *Chaos, Solitons & Fractals 34*(5), 1482 – 1497.

[47] Jurvetson, S. (2000). What exactly is viral marketing. *Red Herring 78*, 110–112.

[48] Kempe, D., J. Kleinberg, and E. Tardos (2003). Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, New York, NY, USA, pp. 137–146.

[49] Ko, K.-I. Separating and collapsing results on the relativized probabilistic polynomial-time hierarchy. *J. ACM (1990) 37*(2), 415–438.

[50] Ko, K.-I. (1988). Relativized polynomial time hierarchies having exactly k levels. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, New York, NY, USA, pp. 245–253. Association for Computing Machinery.

[51] Ko, K.-I. (1990). A note on separating the relativized polynomial time hierarchy by immune sets. *RAIRO - Theoretical Informatics and Applications 24*(3), 229–240.

[52] Ko, K.-I. (1991). Separating the low and high hierachies by oracles. *Information and Computation 90*(2), 156–177.

[53] Krause, A., A. Singh, and C. Guestrin (2008). Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research 9*(3), 235–284.

[54] Lappas, T., E. Terzi, D. Gunopulos, and H. Mannila (2010). Finding effectors in social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, New York, NY, USA, pp. 1059–1068. Association for Computing Machinery.

[55] Li, M., X. Wang, K. Gao, and S. Zhang (2017). A survey on information diffusion in online social networks: Models and methods. Volume Information 8(4).

[56] Li, X., H. G. Du, and P. M. Pardalos (2020). A variation of ds decomposition in set function optimization. *Journal of Combinatorial Optimization*.

[57] Liben-Nowell, D. and J. Kleinberg (2007). The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology 58*(7), 1019–1031.

[58] Lim, S., J. Hao, Z. Lu, X. Zhang, and Z. Zhang (2018). Approximating the k-minimum distance rumor source detection in online social networks. In *2018 27th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–9.

[59] Lin, H. and J. Bilmes (2011). A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, USA, pp. 510–520. Association for Computational Linguistics.

[60] Lu, W., W. Chen, and L. V. S. Lakshmanan (2015, October). From competition to complementarity: Comparative influence diffusion and maximization. *Proc. VLDB Endow. 9*(2), 6071.

[61] Luo, W. and W. P. Tay (2012). Identifying multiple infection sources in a network. In *2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pp. 1483–1489.

[62] Luo, W. and W. P. Tay (2013). Estimating infection sources in a network with incomplete observations. In *2013 IEEE Global Conference on Signal and Information Processing*, pp. 301–304.

[63] Luo, W. and W. P. Tay (2013). Finding an infection source under the sis model. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2930–2934.

[64] Luo, W., W. P. Tay, and M. Leng (2013). Identifying infection sources and regions in large networks. *IEEE Transactions on Signal Processing 61*(11), 2850–2865.

[65] Maehara, Takanori, Murota, and Kazuo (2015). A framework of discrete dc programming by discrete convex analysis. *Mathematical Programming*.

[66] Narasimhan, M. and J. Bilmes (2005, July). A submodular-supermodular procedure with applications to discriminative structure learning. In *Uncertainty in Artificial Intelligence (UAI)*, Edinburgh, Scotland. Morgan Kaufmann Publishers.

[67] Nemhauser, G. L., L. A. Wolsey, and M. L. Fisher (1978). An analysis of approximations for maximizing submodular set functionsi. *Mathematical Programming 14*(1), 265–294.

[68] Newman, M. E. (2001). The structure of scientific collaboration networks. *Proceedings of the national academy of sciences 98*(2), 404–409.

[69] Nguyen, D. T., N. P. Nguyen, and M. T. Thai (2012). Sources of misinformation in online social networks: Who to suspect? In *MILCOM 2012 - 2012 IEEE Military Communications Conference*, pp. 1–6.

[70] Opsahl, T. (2013). Triadic closure in two-mode networks: Redefining the global and local clustering coefficients. *Social Networks 35*(2), 159–167.

[71] Pastor-Satorras, R. and A. Vespignani (2001, Apr). Epidemic spreading in scale-free networks. *Phys. Rev. Lett. 86*, 3200–3203.

[72] Pinto, P. C., P. Thiran, and M. Vetterli (2012, Aug). Locating the source of diffusion in large-scale networks. *Phys. Rev. Lett. 109*, 068702.

[73] Richardson, M. and P. Domingos (2002). Mining knowledge-sharing sites for viral marketing. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 61–70. ACM.

[74] Salha, G., N. Tziortziotis, and M. Vazirgiannis (2018). Adaptive submodular influence maximization with myopic feedback. In *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 455–462. IEEE.

[75] Schoenebeck, G. and B. Tao (2019, April). Beyond worst-case (in)approximability of nonsubmodular influence maximization. *ACM Trans. Comput. Theory 11*(3).

[76] Shah, C., N. Dehmamy, N. Perra, M. Chinazzi, A. Barab'asi, A. Vespignani, and R. Yu (2020). Finding patient zero: Learning contagion source with graph neural networks. *ArXiv abs/2006.11913*.

[77] Shah, D. and T. Zaman (2010). Detecting sources of computer viruses in networks: Theory and experiment. Volume 38, pp. 203–214.

[78] Shah, D. and T. Zaman (2011). Rumors in a network: Who's the culprit? *IEEE Transactions on Information Theory 57*, 5163–5181.

[79] Shah, D. and T. Zaman (2012). Rumor centrality: A universal source detector. *SIGMETRICS Perform. Eval. Rev. 40*(1), 199–210.

[80] Shah, D. and T. Zaman (2016, June). Finding rumor sources on random trees. *Oper. Res. 64*(3), 736–755.

[81] Sharma, S. and R. Sharma (2020). A graph neural network based approach for detecting suspicious users on online social media. *ArXiv abs/2010.07647*.

[82] Shelke, S. and V. Attar (2019). Source detection of rumor in social network - a review. *Online Social Networks and Media 9*, 30–42.

[83] Shu, L., M. Mukherjee, X. Xu, K. Wang, and X. Wu (2016). A survey on gas leakage source detection and boundary tracking with wireless sensor networks. *IEEE Access 4*, 1700–1715.

[84] Singer, Y. (2016). Influence maximization through adaptive seeding. *ACM SIGecom Exchanges 15*(1), 32–59.

[85] Tang, S. When social advertising meets viral marketing: Sequencing social advertisements for influence maximization. In *Thirty-Second AAAI Conference on Artificial Intelligence, 2018*.

[86] Tang, S. (2018). Stochastic coupon probing in social networks. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 1023–1031. ACM.

[87] Tang, Y., X. Xiao, and Y. Shi (2014). Influence maximization: Near-optimal time complexity meets practical efficiency. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pp. 75–86. ACM.

[88] Tong, G., R. Wang, X. Li, W. Wu, and D.-Z. Du (2019). An approximation algorithm for active friending in online social networks. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1264–1274. IEEE.

[89] Tong, G., R. Wang, C. Ling, Z. Dong, and X. Li (2020). Time-constrained adaptive influence maximization. *arXiv preprint arXiv:2001.01742*.

[90] Tong, G., W. Wu, S. Tang, and D.-Z. Du (2017). Adaptive influence maximization in dynamic social networks. *IEEE/ACM Transactions on Networking (TON) 25*(1), 112–125.

[91] Tong, G. A., S. Li, W. Wu, and D. Du (2016). Effector detection in social networks. *IEEE Transactions on Computational Social Systems 3*(4), 151–163.

[92] Traub, J. F. and A. G. Werschulz (1998). *Complexity and Information.* Cambridge: Cambridge University Press.

[93] Vishwanathan, S., N. N. Schraudolph, R. Kondor, and K. M. Borgwardt (2010). Graph kernels. *Journal of Machine Learning Research 11*(40), 1201–1242.

[94] Wang, C., W. Chen, and Y. Wang (2012). Scalable influence maximization for independent cascade model in large-scale social networks. *Data Mining and Knowledge Discovery 25*(3), 545–576.

[95] Wang, C., K. Xu, and G. Zhang (2013). A seir-based model for virus propagation on sns. In *2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies*, pp. 479–482.

[96] Wang, H., B. Liu, X. Zhang, L. Wu, W. Wu, and H. Gao (2016). List edge and list total coloring of planar graphs with maximum degree 8. *Journal of Combinatorial Optimization 32*(1), 188–197.

[97] Wang, Z., W. Dong, W. Zhang, and C. W. Tan (2015). Rooting our rumor sources in online social networks: The value of diversity from multiple observations. *IEEE Journal of Selected Topics in Signal Processing 9*(4), 663–677.

[98] Wang, Z., C. Wang, J. Pei, and X. Ye (2017). Multiple source detection without knowing the underlying propagation model. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pp. 217–223. AAAI Press.

[99] Wang, Z., Y. Yang, J. Pei, L. Chu, and E. Chen (2017). Activity maximization by effective information diffusion in social networks. *IEEE Transactions on Knowledge and Data Engineering 29*(11), 2374–2387.

[100] Wang, Z., W. Zhang, and C. W. Tan (2015). On inferring rumor source for sis model under multiple observations. In *2015 IEEE International Conference on Digital Signal Processing (DSP)*, pp. 755–759.

[101] Xu, W. and H. Chen (2015). Scalable rumor source detection under independent cascade model in online social networks. In *2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, pp. 236–242.

[102] Yang, W., S. Chen, S. Gao, and R. Yan (2020). Boosting node activity by recommendations in social networks. *Journal of Combinatorial Optimization*.

[103] Yang, W., Y. Zhang, and D. zhu Du (2020). Influence maximization problem: properties and algorithms. *Journal of Combinatorial Optimization*.

[104] Yang, Y., X. Mao, J. Pei, and X. He (2016). Continuous influence maximization: What discounts should we offer to social network users? In *Proceedings of the 2016 international conference on management of data*, pp. 727–741. ACM.

[105] Yao, A. C.-C. (1985). Separating the polynomial-time hierarchy by oracles. In *Proc. 26th Annual Symposium on Foundations of Computer Science*, pp. 110. IEEE Press.

[106] Yu, J. and M. B. Blaschko (2016). A convex surrogate operator for general non-modular loss functions. *ArXiv abs/1604.03373*.

[107] Yuan, J. and S. Tang (2016). No time to observe: Adaptive influence maximization with partial feedback. *arXiv preprint arXiv:1609.00427*.

[108] Yuan, J. and S.-J. Tang (2017). Adaptive discount allocation in social networks. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 22. ACM.

[109] Zang, W., P. Zhang, C. Zhou, and L. Guo (2015). Locating multiple sources in social networks under the sir model: A divide-and-conquer approach. *Journal of Computational Science 10*, 278 – 287.

[110] Zejnilovi, S., J. Gomes, and B. Sinopoli (2013). Network observability and localization of the source of diffusion based on a subset of nodes. In *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 847–852.

[111] Zhang, Z., P. Cui, and W. Zhu (2020). Deep learning on graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering 14*(8), 1–1.

[112] Zhang, Z., W. Xu, W. Wu, and D.-Z. Du (2017). A novel approach for detecting multiple rumor sources in networks with partial observations. *Journal of Combinatorial Optimization 33*(1), 132–146.

[113] Zhou, D., O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf (2003). Learning with local and global consistency. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'03, Cambridge, MA, USA, pp. 321–328. MIT Press.

[114] Zhu, J., S. Ghosh, J. Zhu, and W. Wu (2019). Near-optimal convergent approach for composed influence maximization problem in social networks. *IEEE Access PP*(99), 1–1.

[115] Zhu, K., Z. Chen, and L. Ying (2017). Catch'em all: Locating multiple diffusion sources in networks with partial observations. In *AAAI*, pp. 1676–1683.

[116] Zhu, K. and L. Ying (2014). A robust information source estimator with sparse observations. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 2211–2219.

[117] Zhu, K. and L. Ying (2016). Information source detection in the sir model: A sample-path-based approach. *IEEE/ACM Transactions on Networking 24*(1), 408–421.

BIOGRAPHICAL SKETCH

Rong Jin was born in Nanjing, the capital of Jiangsu province of China. She attended Nanjing Zhonghua High School. She then joined Nanjing University of Posts & Telecommunications for four years and graduated with her Bachelor of Engineering in communication engineering in July 2011. She started the graduate program in computer science at The University of Texas at Dallas in August 2013. She earned a master's degree in computer science in May 2015 and then completed a doctorate in computer science under the supervision of Dr. Weili Wu in August 2021. Her research interests include graph data mining, computational social networks, multimedia systems, and technology in STEM education.

CURRICULUM VITAE

Rong Jin

June 16, 2021

Contact Information:

Department of Computer Science
The University of Texas at Dallas
800 W. Campbell Rd.
Richardson, TX 75080-3021, U.S.A.

Email: `rong.jin@utdallas.edu`

Educational History:

B.E., Communication Engineering, Nanjing University of Posts & Telecommunications, 2011
M.S., Computer Science, The University of Texas at Dallas, 2015
Ph.D., Computer Science, The University of Texas at Dallas, 2021

Study in Big Data Harnessing and Related Problems
Ph.D. Dissertation
Computer Science Department, The University of Texas at Dallas
Advisors: Dr. Weili Wu

Employment History:

Instructor of Computer Science, Emporia State University, January 2021 – May 2021
Research/Teaching Assistant, The University of Texas at Dallas, February 2015 – December 2020
Software Test Engineer, Infosys China, July 2011 – February 2013

Professional Recognitions and Honors:

ACM SIGSIM-PADS Student Travel Grant, 2019
Grace Hopper Conference Scholarship, UTD, 2017
CRA-W Graduate Cohort Scholarship, 2017
Academic Performance Fellowship, NJUPT, 2007 –2011

Professional Memberships:

Institute of Electrical and Electronics Engineers (IEEE), 2016–present
Association of Computing Machinery (ACM), 2017–present