INFLUENCE OPTIMIZATION PROBLEMS IN SOCIAL NETWORKS

by

Shuyang Gu

APPROVED BY SUPERVISORY COMMITTEE:

Dr. Weili Wu, Chair

Dr. Farokh B. Bastani

Dr. Bhavani Thuraisingham

Dr. Latifur Khan

Copyright © 2020

Shuyang Gu

All rights reserved

Dedicated to my father.

INFLUENCE OPTIMIZATION PROBLEMS IN SOCIAL NETWORKS

by

SHUYANG GU, BS

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

May 2020

ACKNOWLEDGMENTS

First of all, I would like to thank Dr. Weili Wu, my Ph.D. advisor. I am so lucky to have Dr. Wu as my advisor. It was a memorable experience working with Dr. Wu, who gave me a lot of help in both my research and my life in Dallas. Dr. Wu's vigour and enthusiasm affected me significantly, no matter what difficulty I face, as long as I talk with her, I could feel powerful and fearless. It is no exaggeration to say Dr. Wu changed my life, her personality will be my lifetime treasure.

I would like to thank Dr. Ding-zhu Du, who gave me important advice on my research too. It is an honor that I could discuss research problems with Dr. Du, who is a top scientist in computer science and mathematics. Dr. Du encouraged me to stick on my academic career, and I benefited from his guide in my research and life plan.

I would like to thank Dr. Farokh B. Bastani, Dr. Bhavani Thuraisingham and Dr. Latifur Khan for their guidance and patience in my dissertation work. I also thank my colleagues and friends who have made my life in Dallas a great time. They are Dr. Chuangen Gao, Ruiqi Yang, Dr. Yi Li, Dr. Yuqi Fan, Smita Ghosh, Jianxiong Guo, and many other friends and visiting scholars at UTD.

Finally, I would like to thank my Mom, Minghua, and my son, Cooper. Thanks to my Mom I could keep chasing my dream. It is my son who makes my Ph.D. study special and happy.

March 2020

INFLUENCE OPTIMIZATION PROBLEMS IN SOCIAL NETWORKS

Shuyang Gu, PhD
The University of Texas at Dallas, 2020

Supervising Professor: Dr. Weili Wu, Chair

Online social networks have been developing and prosperous during the last two decades, my dissertation focus on the study of social influence. Several practical problems about social influence are formulated as optimization problems.

First, users of online social networks such as Twitter, Instagram have a nature of expanding social relationships. Thus, one important social network service is to provide potential friends to a user that he or she might be interested in, which is called friend recommendation. Different from friend recommendation, which is a passive way for an user to connect with a potential friend, in my work, I tackle a different problem named active friending as an optimization problem about how to friend a person in social networks taking advantage of social influence to increase the acceptance probability by maximizing mutual friends influence.

Second, the influence maximization problem has been studied extensively with the development of online social networks. Most of the existing works focus on the maximization of influence spread under the assumption that the number of influenced users determines the success of product promotion. However, the profit of some products such as online game depends on the interactions among users besides the number of users. We take both the number of active users and the user-to-user interactions into account and propose the interaction-aware influence maximization problem.

Furthermore, due to the uncertainty in edge probability estimates in social networks, we propose the robust profit maximization problem to have the best solution in the worst case of probability settings.

TABLE OF CONTENTS

ACKNOWLEDGMENTS v
ABSTRACT vi
LIST OF FIGURES xi
LIST OF TABLES xii
CHAPTER 1 INTRODUCTION 1
CHAPTER 2 FRIENDING 3
2.1 Active Friending 4
2.2 Target Friending 9
2.3 Group Friending 9
CHAPTER 3 INTERACTION-AWARE INFLUENCE 12
3.1 Related Work 15
3.2 Problem Formulations 17
3.2.1 Activity Maximization 17
3.2.2 Interaction-aware Influence Maximization 19
3.3 A Method for Interaction-aware Influence Maximization 21
CHAPTER 4 A GENERAL METHOD OF ACTIVE FRIENDING IN DIFFERENT DIFFUSION MODELS IN SOCIAL NETWORKS 24
4.1 Introduction 25
4.1.1 Motivation 25
4.1.2 Problem Description 26
4.1.3 Related Work 26
4.1.4 Our Contributions 28
4.2 Problem Formulation 29
4.2.1 Acceptance Probability in LT Model 29
4.2.2 Acceptance Probability in IC Model 31
4.2.3 Hardness Results 33
4.3 Problem Conversion 35
4.3.1 Problem Conversion in LT Model 35

4.3.2 Problem Conversion in IC Model . . . 37
4.4 Algorithms . . . 41
4.4.1 Iterated Submodular Cost Knapsack Algorithm . . . 42
4.4.2 Greedy Algorithm . . . 47
4.5 Performance Evaluation . . . 48
CHAPTER 5 INTERACTION-AWARE INFLUENCE MAXIMIZATION AND ITERATED SANDWICH METHOD . . . 51
5.1 Introduction . . . 52
5.2 Related Works . . . 53
5.3 Problem Formulation . . . 55
5.3.1 Interaction-aware Influence Maximization . . . 55
5.3.2 Modularity of Objective Function . . . 56
5.3.3 Hardness Result . . . 57
5.4 Sandwich Theory . . . 57
5.4.1 Preliminary . . . 58
5.4.2 Sandwich Theory . . . 59
5.4.3 DS decomposition . . . 60
5.5 Algorithms . . . 64
5.5.1 Iterated Sandwich Algorithm . . . 64
5.5.2 Analysis . . . 67
5.6 Experiment . . . 69
5.6.1 Settings . . . 69
5.6.2 Effectiveness . . . 71
CHAPTER 6 ROBUST PROFIT MAXIMIZATION WITH DOUBLE SANDWICH ALGORITHMS IN SOCIAL NETWORKS . . . 72
6.1 Introduction . . . 73
6.2 Related work . . . 75
6.3 Problem Formulation . . . 77
6.3.1 Robust Profit Maximization Problem . . . 77
6.3.2 Profit Maximization Problem . . . 79

6.4 Strategy for Profit Maximization Problem 81
6.4.1 Upper bound 81
6.4.2 Lower bound 83
6.4.3 Sandwich Strategy 83
6.5 Strategy for Robust profit maximization problem 87
6.5.1 Complexity 87
6.5.2 Double Sandwich Algorithm 88
6.5.3 Double Sandwich Algorithm with Uniform Sampling 90
6.6 Experiment 95
6.6.1 Settings 95
6.6.2 Effectiveness and Analysis 97
CHAPTER 7 CONCLUSION 99
REFERENCES 100
BIOGRAPHICAL SKETCH 108
CURRICULUM VITAE

LIST OF FIGURES

3.1 Counter example 21
4.1 Counter Examples 32
4.2 Reduction for NP hardness in LT model 33
4.3 Experimental Results for Random Pair of Nodes 48
4.4 Experimental Results for Random Pair of Nodes (node 1 to node 5) 49
5.1 Counter example 57
5.2 Iterated sandwich algorithms flow 66
5.3 The relationship between profit and seed set size 70
6.1 Counter example 80
6.2 The experiment results 95

LIST OF TABLES

4.1 Notation 30
5.1 The statistics of the data sets 70

CHAPTER 1

INTRODUCTION

With the advancements in information science in the last two decades, online social networks find important applications in viral marketing, under this circumstance, influence maximization becomes a very popular research direction, which could be described as the problem of finding a small set of most influential nodes in a social network so that the number of influenced nodes under certain diffusion model in the network is maximized.

A large number of effort has been made in this research topic since Kempe *et al.* (Kempe et al., 2003a) first defined the problem and obtained plentiful results in many ways. The topic is also extended to other optimization problems other than influence maximization, the core of this research is to study how to employ social influence in order to achieve some practical goals. In this dissertation, we are going to study three influence optimization problems: active friending, interaction-aware influence maximization, and robust profit maximization.

Active friending. friending in social networks is the activity of building new relationships. In Chapter2, we will introduce the basics of friending. Active friending is a problem that is to assist a user to build the relationships to a target user by sending invitations to a set of intermediate users taking advantage of social influence, the goal is to maximize the acceptance probability at the target node taking advantage of the social influence through the network formed by the intermediate nodes. In this dissertation, we convert the original formulated active friending problem of nonsubmodular maximization subject to cardinality constraint into a submodular cost submodular knapsack problem in IC model, we show that the two problems are equivalent. We similarly make the conversion on the active friending in LT model. Then we give a general combinatorial optimization algorithm to solve active friending problems in both IC model and LT model with a guaranteed approximation. We analyze the computational complexity of the problem and the algorithm performance. The

effectiveness of the generalized method is verified on real data sets. This method will be shown in Chapter 4.

Interaction-aware Influence Maximization. This problem extends the classical influence maximization problem. Most of the existing work about influence maximization focus on the maximization of influence spread under the assumption that the number of influenced users determines the success of product promotion. However, the profit of some products such as online game depends on the interactions among users besides the number of users. In Chapter 3, we review the problems we discover about interaction-aware influence, where we take both the number of active users and the user-to-user interactions into account and propose the interaction-aware influence maximization problem. . In this dissertation, to address this practical issue, we analyze its complexity and modularity, propose the sandwich theory which is based on decomposing the non-submodular objective function into the difference of two submodular functions and design iterated sandwich algorithm which is guaranteed to get data-dependent approximation solution. This problem will be demonstrated in Chapter 5.

Robust Profit Maximization. The goal of this problem is also interaction-aware influence maximization, but it considers the uncertainty of influence propagation probability in social networks, we propose the robust profit maximization problem to have the best solution in the worst case of probability settings. We design a double sandwich algorithm to this problem and further improve the algorithm with a sampling method such that it increases the robustness of the output. Through real data sets, we verify the effectiveness of our proposed algorithm. This part will be shown in Chapter 6.

The rest of the dissertation proceeds as follows: Chapter 2 gives the preliminaries to friending, Chapter 3 discusses the interaction-aware social influence. Through Chapters 4 to 6, we study the problems of active friending, interaction-aware influence maximization, and robust profit maximization respectively. Chapter 7 concludes this dissertation.

CHAPTER 2

FRIENDING[1]

Authors – Shuyang Gu, Hongwei Du, My T. Thai and Ding-Zhu Du

The Computer Science Department, EC 31

The University of Texas at Dallas

800 West Campbell Road

Richardson, Texas 75080-3021

[1] Reprinted by permission from Shuyang Gu, Hongwei Du, My T. Thai, Ding-zhu Du: Springer Nature, Nonlinear Combinatorial Optimization, Friending, Shuyang Gu, Hongwei Du, My T. Thai, Ding-zhu Du, ©2019

2.1 Active Friending

If you have a LinkedIn or Facebook account, then you may frequently receive a message like this "Xuefei Zhang added connections you may know", which reminds you that you may know someone or someone is your friend's friend. If you open the message, then you may find a link to login your account and from your account, you may find some names who invited you to be their friend and a list of names whom you may consider to invite for your friends. These activities are called friending.

The active friending is the first optimization problem appeared in the literature (Yang et al., 2013) about friending. The problem can be described as follows:

Definition 1 (Active Friending)**.** *Consider a social network represented as directed graph $G = (V, E)$ with an information diffusion model m. Suppose S is the list of existing friends of a node s and t is a target node that s wants to include in his friend list. Given an integer $r > 0$, the problem is to find a subset R with at most r nodes to maximize the success probability $Prob(s, S, R, t)$, i.e., the probability that node t is activated through subgraph induced by $R \cup S \cup \{s, t\}$ when initially set up all nodes in $S \cup \{s\}$ to be active.*

There are two popular information diffusion models studied in the literature, the independent cascade (IC) model and the linear threshold (LT) model. They are defined as follows.

The IC model: Each node has two states, active and inactive. Every arc (u, v) is labeled with a probability p_{uv} which means that if u is active and v is inactive, then the event that v accepts the influence of u, i.e., v becomes active because of active u occurs with probability p_{uv}. Before the process starts, all nodes are inactive. Initially, choose a subset of nodes, called seeds, and activate them. In each subsequence step, every fresh-active node tries to influence its inactive out-neighbors where a node is fresh-active if it becomes active in the step right before the current step. If an inactive node v gets influenced by more than one,

say k, fresh-active nodes $u_1, u_2, ..., u_k$ at the same step, then all k events that u_i influences v successfully are treated as k independent events. This process ends if no fresh-active node is produced.

The LT model: Each node has two states, active and inactive. Every arc (u, v) is labeled with a positive weight w_{uv} such that for any node v, $\Sigma_{u \in N^-(v)} w_{uv} \leq 1$ where $N^-(v) = \{u|(u, v) \in E\}$. Before the process starts, all nodes are inactive. Initially, choose a subset of nodes, called seeds, and activate them; meanwhile, each node u choose a threshold θ_u uniformly and randomly from $[0, 1]$. In each of subsequence steps, every inactive node v evaluates the total weight of w_{uv} for u overall active in-neighbors. If this total weight is at least θ_v, then v becomes active; otherwise, v keeps inactive. This process ends if no fresh-active node is produced.

The following is proved in (Yang et al., 2013) by using dynamic programming.

Theorem 1. *For an arborescence directed to t with the IC model, the active friending can be solved in polynomial-time.*

Using this result, they also designed a heuristic by, first, approximating the general network with an in-arborescence with root t. This arborescence is the union of all the most influential paths from each $S \cup \{s\}$ to t where the most influential path from $s' \in S \cup \{s\}$ to t is the shortest path when we consider -logp_{uv} as the distance from node u to v and p_{uv} is the probability that node v accepts the influence from u in the IC cascade model.

Kempe, Kleiburg, and Tardos (Kempe et al., 2003a) generalized the LT model and the IC model to the general threshold model and the general cascade model, and proved that every general threshold model is equivalent to a general cascade model, vice versa. For this equivalence, the LT model is equivalent to a general cascade model, called the mutually-exclusive cascade (MC) model. The MC model can be defined in the same way as that of the IC model, except that when k fresh-active nodes $u_1, u_2, ..., u_k$ try to influence an inactive

node v at the same step, this is considered as that k mutually-exclusive events occur. In the equivalence relation between the LT model and the MC model, $w_{uv} = p_{uv}$. The MC model (of course, the LT model, too) has an important property. Consider a social network with four nodes u, v, x, y and three arcs $(u, x), (v, x), (x, y)$ in the MC model. Suppose u and v are seeds. Then the probability that y becomes active is,

$$(p_{ux} + p_{vx})p_{xy} = p_{ux}p_{xy} + p_{vx}p_{xy}$$

that is, this probability is the sum of the probability that y accepts the influence of u through the path from u to y and the probability that y accepts the influence of v through the path from v to y. In general, this property can be stated in the following lemma.

Lemma 1. *In the LT model, a node v accepts the influence from a seed set S with probability equal to*

$$\sum_{P \in \mathcal{P}} Prob(P)$$

where $\mathcal{P}$ is the set of paths from S to v and $Prob(P)$is the probability that v accepts the influence of a seed in S along path P.

This property makes that the problem in the LT model sometimes is easier than that in the IC model. For example, the influence maximization in arborescence directed to the root is polynomial-time solvable in the LT model (Wang et al., 2016), however NP-hard in the IC model (Lu et al., 2017). (This result was first conjectured in (Bharathi et al., 2007) and then proved in (Lu et al., 2017).)

With this special property of the LT model, Yuan et al. (Yuan et al., 2017) proved the following result about $Prob(s, S, R, t)$.

Theorem 2. *$Prob(s, S, R, t)$ is a monotone nondecreasing, supermodular function with respect to R for social network G in the linear threshold model, that is, for any $R' \in R$, $Prob(s, S, R', t) \leq Prob(s, S, R, t)$, and for any R and R',*

$$Prob(s, S, R, t) + Prob(s, S, R', t) \leq Prob(s, S, R \cup R', t) + Prob(s, S, R \cap R', t)$$

Proof. It is easy to see the property of monotone nondecreasing. We next show the supermodularity. Before doing so, let us first recall a special property proved in (Wang et al., 2016) that the linear threshold model is equivalent to the mutually-exclusive cascade model in which when k fresh-active nodes influence an inactive node, this event is considered as a composed event of k mutually-exclusive events. This property yields that in the linear threshold model, $Prob(s, S, R, t)$ is equal to the sum of accepting probabilities each of which is the probability that t accepts the invitation from anode $s' \in S \cup \{s\}$ along a path p to t where p is over all paths from a node in $S \cup \{s\}$ to t and with all nodes in R. Let $P(R)$ denote the set of all such paths p. □

Now, we compare $P(R) \cup P(R')$ with $P(R \cup R')$ and $P(R \cap R')$. Clearly, both $P(R)$ and $P(R')$ are subsets of $P(R \cup R')$. Moreover, if a path p appears in both $P(R)$ and $P(R')$, then p must appear in $P(R \cap R')$. Therefore,

$$Prob(s, S, R, t) + Prob(s, S, R', t) \leq Prob(s, S, R \cup R', t) + Prob(s, S, R \cap R', t)$$

By Theorem 5, the active friending with the LT model can be formulated into following problem:

$$\begin{aligned} \max \quad & Prob(s, S, R, t) \\ \text{subject to} \quad & |R| \leq r, \end{aligned}$$

that is, a monotone supermodular maximization with size constraint. This formulation suggests that the discrete Lagrangian method (Shang and Wah, 1998) is suitable to solve the active friending problem for the LT model. The greedy algorithm in (Bai and Bilmes, 2018) can also be used. However, the estimation of the curvature is trouble, which may be done possibly only for some special networks, such as power-law graphs.

Next, we move our attention to the IC model. Let P be the set of all paths from $\{s\} \cup S$ to t. Denote by $Prob(R; P)$ the probability that the randomized subgraph induced

by $R \cup S \cup \{s,t\}$ containing all paths in P . Denote $P_i = \sum_{|P|=i, P \subseteq \mathcal{P}} Prob(R;P)$. By the inclusive-exclusive formula,

$$Prob(s,S,R,t) = P_1 - P_2 + P_3 - P_4 + \cdots + (-1)^{|\mathcal{P}|} P_{|\mathcal{P}|}$$

By an argument similar to that in the proof of Theorem 2, we can show following result.

Lemma 2. *$Prob(R;P)$ is monotone nondecreasing supermodular with respect to R.*

Proof. It is clear that $Prob(R;P)$ is monotone nondecreasing. Next, we show the supermodularity. Consider two node subsets R_1 and R_2. Note that the randomized subgraph induced by $(R_1 \cup R_2) \cup S \cup \{s,t\}$ contains those paths contained by the randomized subgraph induced by $R_j \cup S \cup \{s,t\}$ for $j = 1, 2$. In addition, it also contains those paths contained by union of these two randomized subgraphs. Therefore,

$$Prob(R_1;P) + Prob(R_2;P) \leq Prob(R_1 \cup R_2;P) + Prob(R_1 \cap R_2;P),$$

that is, $Prob(R;P)$ is supermodular. □

By above lemma, the following holds.

Theorem 3. *In the IC model, $Prob(s,S,R,t)$ can be represented as a difference of two nonnegative monotone nondecreasing supermodular functions, i.e.,*

$$Prob(s,S,R,t) = (P_1 + P_3 + \cdots) - (P_2 + P_4 + \cdots).$$

By Theorem 6, we may employ the sandwich method (Lu et al., 2015a; Chen et al., 2016a; Wang et al., 2017a; Tong et al., 2018), the submodular-supermodular method (Narasimhan and Bilmes, 2005a), the modular-modular method (Iyer and Bilmes, 2012a), and the iterated sandwich method (Wu et al., 2018) to solve the active friending problem for the IC model.

2.2 Target Friending

The second optimization problem on friending is the target friending described as follows.

Definition 2 (Target Friending)**.** *Consider a social network represented as directed graph $G = (V, E)$ with an information diffusion model m. Suppose S is the list of existing friends of a node s and t is a target node that s wants to include in his friend list. Given an integer $0 < \rho < 1$, the problem is to find a minimum node subset R such that $Prob(s, S, R, t) \geq \rho$.*

By Theorem 5, the target friending for the LT model is a supermodular cover problem as follows.

$$\begin{aligned} \min \quad & |R| \\ \text{subject to} \quad & Prob(s, S, R, t) \geq \rho, \end{aligned}$$

The target friending for the IC model is a generalization of the well-known submodular cover problem (Wolsey, 1982) the same as above except that $Prob(s, S, R, t)$ is a nonsubmodular and nonsupermodular function in the cover constraint. It is an interesting research subject to see how to generalize the approximation analysis for the submodular cover problem. In fact, there are so many different proofs for the same theorem regarding to the approximation performance ratio of a greedy algorithm for the submodular cover (Wolsey, 1982; Du et al., 2011; Wan et al., 2010). None of them is able to give a generalization for above nonsubmodular cover problem so far.

2.3 Group Friending

The group friending was first studied in (Chen et al., 2014). They consider a romantic scenario as follows: A boy found an attractive girl. However, they do not really know each other. The boy worries that he may get rejected if he asks her directly. Hence, he wants

to influence her friends at the first stage. Thus, her friends form target group for friending. The objective in this problem is the expected number of her friends who become his friends after friending process. This problem has no much difference from active friending.

Definition 3 (Active Group Friending)**.** *Consider a social network represented as directed graph $G = (V, E)$ with an information diffusion model m. Suppose S is the list of existing friends of a node s and T is a set of target nodes that s wants to include in his friend list. Given an integerr > 0, the problem is to find a subset R with at most r nodes to maximize the expected number of active nodes in T , which are activated through subgraph induced by $R \cup S \cup \{s, t\}$ when initially set up all nodes in $S \cup \{s\}$ to be active.*

The mathematical formulations are similar respectively to that of active friending in the LT model and the IC model.

Shen et al. (Shen et al., 2015) proposed another formulation based on a quite different scenario. Suppose we want to organize a social activity with at lease p persons, in order to make new friendship between members in a big social organization. Two factors are very important for us, the existing friendship between members and potential friendship between numbers. To evaluate the success of the activity, we may give each potential friendship a positive weight in $(0, 1]$ and a measure of making new friends which is the ratio between the total weight and group size.

Definition 4 (Hop-bounded Group Friending)**.** *Consider a heterogeneous social graph $G = (V, E, R)$ with edge weight $w : R \to (0, 1]$, where V is the set of nodes, E is the set of friend edges, and R is the set of potential friend edges. Given a hop constraint h and a group size constraint p, find a subset of at least p nodes, H, such that every pair of nodes u and v are within distance h in the graph with node set V and edge set E and $\sigma(H)$ reaches the maximum, where $\sigma(H) = w(H)/|H|$ and $w(H)$ is the total weight of potential friend edges in the subgraph induced by H.*

This problem has been proved to be NP-hard and has no polynomial-time approximation with a performance ratio $\rho < 1$ unless NP=P (Shen et al., 2015).

Finding a cohesive group from a social network with existing friend edges is an important research topic in the literature. However, before (Shen et al., 2015), all efforts are based on existing friendship (Wasserman and Faust, 1994; Feige et al., 2001; Mokken, 1979; Yang et al., 2011, 2012; Zhu et al., 2014; Shuai et al., 2013; Surian et al., 2011) and no "friending"is involved. In order to have "friending"involved, the potential friend edges are employed in the hop-bounded group friending. How to know the potential friend edges? The link prediction methods are used (Kashima and Abe, 2006; Liben-Nowell and Kleinberg, 2007; Clauset et al., 2008; Kunegis and Lommatzsch, 2009; Leung et al., 2010). They analyze the features, the similarity, and/or the interactive patterns to make recommendation for a potential friendship.

In the community expansion (Bi et al., 2014, 2013,?) , each community consists of all customers for a certain business which always wants to expanse their service. Therefore, a different type of "friending"problems are raised. They can all be formulated into nonlinear combinatorial optimization problems.

CHAPTER 3

INTERACTION-AWARE INFLUENCE [1]

Authors – Shuyang Gu, Chuangen Gao, Weili Wu

The Computer Science Department, EC 31

The University of Texas at Dallas

800 West Campbell Road

Richardson, Texas 75080-3021

[1]Reprinted by permission from Shuyang Gu, Chuangen Gao, Weili Wu: Springer Nature, Nonlinear Combinatorial Optimization, Interaction-aware Influence Maximization in Social Networks, Shuyang Gu, Chuangen Gao, Weili Wu, ©2019

With the advancements in information science in the last two decades, online social networks find important applications in viral marketing, under this circumstance, influence maximization becomes a very popular research direction, which could be described as the problem of finding a small set of most influential nodes in a social network so that the number of influenced nodes under certain diffusion model in the network is maximized.

Kempe *et al.* (Kempe et al., 2003a) first formulate it as the influence maximization problem: A social network is modeled as a graph with vertices representing individuals and edges representing a relationship between two individuals. Influence is propagated in the network according to a stochastic cascade model. One of the most popular cascade models is independent cascade (IC) model: Each edge (u, v) in the graph is associated with a propagation probability $p(u, v)$, which is the probability that node u independently activates node v at step $t+1$ if u is activated at step t. Given a social network graph, the IC model, and a number k, the influence maximization problem is to find k nodes in the graph (referred to as seeds) such that under the influence cascade model, the expected number of nodes activated by the k seeds (referred to as the influence spread) is the largest possible. Kempe et al. prove that the optimization problem is NP-hard, and they present a greedy algorithm which guarantees that the influence spread is within $(1 - 1/e - \epsilon)$ of the optimal influence spread, where e is the base of natural logarithm, and ϵ depends on the accuracy of their Monte-Carlo estimate of the influence spread given a seed set.

A large number of effort has been made in this research topic since Kempe *et al.* (Kempe et al., 2003a) first defined the problem and obtained plentiful results in many ways. Most of the works focus on maximization of the spread of influence, which considers the number of users influenced by viral marketing or the "word-of-mouth"effect in online social networks. These works are based on the assumption that the number of influenced users determines the profit of the product. However, some types of products earn profit in a continuous way besides the sales of the product itself. The revenue model of online games is a good example,

the sales of a game is just one source of a game company's profit, another important part of revenue depends on the participation and interaction of players who have already bought the game.

The interactions among users contribute to game profit in several ways. First, the interactive users play games in an online manner, which will attract more in-game advertising. In-game advertising allows advertisers to pay to have their name or products featured in games, in 2017, $109 billion was spent on in-game advertising. Second, the virtual goods transactions in games depend on players' interactions. In 2009, the sale of virtual goods brought in $1 billion.

We analyze such a revenue model and define a novel problem of interaction-aware influence maximization. Since the first part of revenue, sales of a game, depends on the spread of influence, the objective is the same as the classical influence maximization. The second part of revenue hinges on the interactions among users. We use interaction profit to represent such revenues related to the strength of interactions among players. We then define an interaction-aware profit maximization problem, which is how to select a seed set to maximize both the number of influenced users and the interaction profits among active nodes.

For the traditional influence maximization problem, since its submodularity, the greedy algorithm can achieve a guaranteed approximation with $1-1/e$. But unfortunately, interaction-aware influence maximization problem is not submodular, thus the greedy strategy can't be directly applied to our problem to get a guaranteed approximate solution. To solve this problem, we propose a new method called decomposition strategy in which we decompose our objective function as a difference of two submodular functions. And based on the decomposition we replace them with the modular functions which are upper or lower bound of them to address the non-submodularity part of the problem and design an iterated sandwich algorithm.

3.1 Related Work

Influence maximization was first described as an algorithm problem by Domingos and Richardson (Domingos and Richardson, 2001a) (Richardson and Domingos, 2002a), they model the problem using Markov random fields and propose heuristic solutions. Kempe *et al.* (Kempe et al., 2003b) formulated the influence maxmization problem from the view of combinatorial optimization and showed that the problem is NP-hard under both the IC and LT models, they propose a simple greedy algorithm with an approximation ratio of $1 - 1/e$. However, a drawback of their work is the scalability of the greedy algorithm. Since then a number of efficient heuristic algorithms have been proposed in many works to address the issue, one direction is to improve the greedy algorithm, the other is to propose effective heuristics. In (Leskovec et al., 2007), Leskovec *et al.* present a "lazy-forward" optimization in selecting new seeds, which exploits submodularity and greatly reduces the number of evaluations on the influenced nodes, the main idea is that the marginal gain of a node in the current iteration cannot be better than its marginal gain in the previous iterations.

In (Chen et al., 2009), Chen *et al.* improved the greedy algorithm by combining with the CELF optimization proposed in (Leskovec et al., 2007), they also propose a degree discount heuristics under the independent cascade model. The main idea of degree discount heuristics is when selecting a node based on its degree, the degree does not include the neighbors that are already activated. In(Chen et al., 2010a), they show that computing influence spread in the independent cascade model is $\#P$-hard, they propose a heuristic algorithm to use local arborescence structures of each node to approximate the influence propa- gation. The heuristic algorithm restricts computations on the local influence regions of nodes. Moreover, by tuning the size of local influence regions, this heuristic can achieve a tunable tradeoff between efficiency (in terms of running time) and effectiveness (in terms of influence spread). In (Goyal et al., 2011a), Goyal *et al.* introduce CELF++ that further

optimizes CELF by exploiting submodulairty, the advantage of the algorithm CELF++ is that it avoids unnecessary re-computation of marginal gains incurred by CELF.

The influence maximization problem has also been extended to practical scenarios in recent works. (Chen et al., 2015) studies the topic-aware influence maximization problem which considers user interests. In real-world social networks, users have their interests (topics) and are more likely to be influenced by their friends with similar topics. To address this problem, they study topic-aware influence maximization, that is, given a topic-aware influence maximization (TIM) query, finds k seeds from a social network such that the topic-aware influence spread of the k seeds is maximized.

In (Li et al., 2015), a keyword-based targeted influence maximization is proposed, where users who are relevant to a given advertisement are targeted. In (Han et al., 2016), the problem of privacy reserved influence maximization in both cyber-physical and online social networks is studied, they propose a model that merges both GPS data and relationship data from a social network. Bharathi et al.(Bharathi et al., 2007) study the game of innovation diffusion with multiple competing innovations, for example, multiple companies market competing for products using viral marketing. In (Chen et al., 2011), Chen et al. propose an extension to the independent cascade model that incorporates the emergence and propagation of negative opinions, the new model has a quality factor to model the natural behavior of people turning negative to a product due to product defects.

Most of the works only consider the number of activated users or the nodes of social graphs but few work considers interactions among users in viral marketing. The interaction activities between users is first processed by (Wang et al., 2017a). They consider a specific problem of how one can stimulate the discussion about a topic in a social network as much as possible within a budget. They model the problem as *activity maximization*. Given a propagation network, which records user interaction activity strength along each edge, the goal is to find an optimal set of seed users under a given budget, such that starting

information propagation from the seed users leads to the maximum sum of activity strengths among the influenced users. They show that the activity maximization problem is NP-hard under IC model and LT model. The objective function of the problem is proved neither submodular nor supermodular.

Activity maximization does not include maximizing the influence spread in the meantime and only count activity strength of the directly connected users. We propose a different problem - interaction-aware influence maximization, which takes both parts into consideration, in the following section we will go through the formulations of these two problems and then we will discuss a method to solve interaction-aware influence maximization.

3.2 Problem Formulations

In this section, let us introduce the different formulations on influence maximization problems that consider activity/interactions among users.

3.2.1 Activity Maximization

This problem was first processed by (Wang et al., 2017a). Consider a social network represented by a directed graph $G = (V, E)$, together with an information diffusion model m. In this model, each node has two states, active and inactive. Initially, all nodes are in an inactive state. The influence diffusion consists of discrete steps. At the beginning, a set of nodes are activated. Nodes in this set are called *seeds*. At each subsequent step, every inactive node v evaluates its status and decides whether it should be activated or not, based on the rule in the model m. The process ends at a step in which no more inactive node becomes active.

Let S denote the set of seeds and $I_m(S)$ the set of active nodes at the end of the diffusion process. Suppose that for each pair of active nodes $u, v \in I_m(S)$, if (u, v) is an edge of G,

i.e., $(u, v) \in E$, then an activity profit $A(u, v)$ will be generated where $A : E \to R_+$ is a nonnegative activity profit function. The activity maximization is the following problem:

$$\begin{aligned} \max \alpha(S) = & \sum_{(u,v)\in E: u,v\in I_m(S)} A(u, v) \\ \text{subject to} \quad & |S| \leq k \\ & S \subseteq V \end{aligned} \tag{3.1}$$

This problem has been proved to be NP-hard in (Wang et al., 2017a). There are also counterexamples in (Wang et al., 2017a), which show that $\alpha(S)$ is neither submodular nor supermodular. However, Wang *et al.* (Wang et al., 2017a) introduced two monotone nondecreasing submodular set functions $\beta : 2^V \to R_+$ and $\gamma : 2^V \to R_+$ such that for any $S \in 2^V$, $\beta(S) \geq \alpha(S) \geq \gamma(S)$. These two set functions are defined as follows.

$$\beta(S) = \sum_{(u,v)\in E: u\in I_m(S)} A(u, v)$$

and

$$\gamma(S) = \sum_{s\in S} \sum_{(u,v)\in E: u,v\in I_m(\{s\})} A(u, v).$$

By a theorem of Nemhauser and Wolsey (Nemhauser et al., 1978), there is a greedy algorithm which can find $(1 - e^{-1})$-approximation solutions for the following two problems.

$$\begin{aligned} & \max \beta(S) \\ \text{subject to} \quad & |S| \leq k, \\ & S \subseteq V \end{aligned} \tag{3.2}$$

$$\begin{aligned} & \max \gamma(S) \\ \text{subject to} \quad & |S| \leq k, \\ & S \subseteq V. \end{aligned} \tag{3.3}$$

Let S_β and S_γ be $(1 - e^{-1})$-approximation solutions for problem 3.2 and 3.3, respectively. Let S_α be a feasible solution for problem 3.1. Choosing the best one from S_α, S_β, and S_γ, we would obtain a data-dependent approximation solution for problem (α), i.e., the data-dependent approximation solution is

$$S_{data} = \text{argmax}_{S \in \{S_\alpha, S_\beta, S_\gamma\}} \alpha(S).$$

3.2.2 Interaction-aware Influence Maximization

The goal of interaction-aware influence maximization is to find a set of initial users to maximize total profit related to both the number of the influenced nodes and the interaction among the influenced nodes.

Again a social network is represented as directed graph $G = (V, E)$ to represent a social network, where V is the set of users and E is the set of social relations between users. Each edge $(u, v) \in E$ is assigned with a probability p_{uv} so that when u is active, v is activated by u with probability p_{uv}. And the benefit related to the interaction between nodes is represented by a nonnegative function $b : V \times V \to \mathbb{R}_{\geq 0}$, in which $b(u, v) = b(v, u)$ for the unordered pair $\{u, v\}$ of node u and v. Note that for each $\{u, v\}$, we only compute once the benefit between them, i.e., $b(u, v)$ or $b(v, u)$ instead of $b(u, v) + b(v, u)$.

Consider a moment in the propagation process under IC model, when node u has just become active, and it attempts to activate its neighbor v, succeeding with probability $p_{u,v}$. We can view the outcome of this random event as being determined by flipping a coin of bias $p_{u,v}$. With all the coins flipped in advance, the edges in G for which the coin flip indicated a successful n activation are declared to be live; the remaining edges are declared to be blocked(Kempe et al., 2003a). We use g to represent the outcome of this process which is called a live graph of G since it consists of all edges declared to be live. We denote as $g \sim D$, where D is the distribution of g. For any seed set S, denote by $I_g(S)$ the set of all active

nodes at the end of the cascade process in live graph g. Its cardinality is represented by $|I_g(S)|$.

The total expected benefit would be defined as

$$\begin{aligned} f(S) &= \mathbb{E}_{g\sim D}[\alpha \cdot |I_g(S)| + \beta \cdot \textstyle\sum_{\{u,v\}\subseteq I_g(S)} b(u,v)] \\ &= \textstyle\sum_g Prob[g] \cdot (\alpha \cdot |I_g(S)| + \beta \cdot \sum_{\{u,v\}\subseteq I_g(S)} b(u,v)) \end{aligned}$$

The benefit consists of two parts, the first part denoted as $\alpha \cdot I_g(S)$ is related to the number of nodes that are finally activated, and the second part $\beta \cdot \sum_{\{u,v\}\subseteq I_g(S)} b(u,v)$ is related to the strength of the interaction between the active nodes. The parameters α , β are used to balance the weight of the two parts of the profits, and $\{u,v\} \subseteq I(S)$ denotes the all unordered pair in the set $I(S)$. Note that for each unordered pair $\{u,v\}$, since $b(u,v) = b(v,u)$, we only compute once the benefit between them. The expectation is respected to g.

The interaction-aware influence maximization is the following problem: Given a social network $G = (V,E)$, a propagation probability p_{uv} for each edge (u,v) under the IC model, a benefit function $b : V \times V \to \mathbb{R}_{\geq 0}$, and a positive integer k, find a set S of k seeds to maximize the expected profit through influence propagation:

$$\max f(S)$$
$$s.t. |S| \leq k$$

This problem can be proved NP-hard by showing a special case of interaction-aware influence maximization problem is NP-hard, where $\alpha = 0$, since a problem being NP-hard in a special case implies NP-hardness in the general case. The seeds size equals k. Then the problem is transferred to seek k seeds that maximize the benefit between activated nodes. Now we prove by reducing from the set cover problem, which is NP-complete (Alon et al., 2003a). Given a ground set $U = \{u_1, u_2, \ldots, u_n\}$ and a collection of sets $\{S_1, S_2, \ldots, S_m\}$

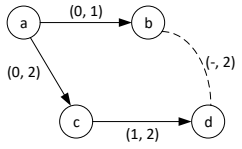


Figure 3.1. Counter example

whose union equals the ground set, the set cover problem is to decide if there exist k sets in S so that the union equals U. Given an instance of the set cover problem, we construct a corresponding graph with $m + 2n$ nodes as follows. For each set S_i we create one node p_i, and for each element u_j we create two nodes q_j and q'_j. If the S_i contains the element u_j, then we create two edges (p_i, q_j) and (p_i, q'_j). Note that each edge is live which means the probability is 1. Now we design the benefit function over pairs of nodes. For the pairs $\{q_j, q'_j\}$, the benefit equals to 1, and the other pairs equal to 0. Then the set cover problem is equivalent to deciding if there is a set S of k nodes such that the benefit of S equals to n. The NP-hardness follows immediately.

There are also counter examples which show that $f(S)$ is neither submodular nor supermodular. We prove by the counter example shown in Fig.6.1. The first element in the tuple tied on each edge represents the propogation probability, and the second one denote the benefit between its two end nodes. For pairs $\{u, v\}$ between which there is no edge set $b(u, v) = 0$ except pair $\{b, d\}$. In Fig.6.1, $(0, 1)$ on edge (a, b) means propogation probability $p_{ab} = 0$ and $b(a, b) = 1$, then we have $f(\{a\}) = 1+0 = 1$, $f(\{a, b\}) = 2+1 = 3$, $f(\{a, d\}) = 2+0 = 2$ and $f(\{a, b, d\}) = 3 + 3 = 6$. Thus, $f(\{a, d\}) - f(\{a\}) < f(\{a, b, d\}) - f(\{a, b\})$, which implies $f(S)$ is not submodular. Also, we have $f(\{c\}) = 2+2 = 4$, $f(\{d, c\}) = 2+2 = 4$, $f(\{d\}) = 1$. Thus, $f(\{c\}) - f(\emptyset) > f(\{d, c\}) - f(\{c\})$ which implies $f(S)$ is not supermodular.

3.3 A Method for Interaction-aware Influence Maximization

We have the following theoretical result leading us to a new method to solve our non-submodular problem.

Theorem 4. *For any set function $f : 2^X \to R$ and any set $Y \subset X$, there are two modular/submodular/supermodular funtions $m_f^u : 2^X \to R$ and $m_f^l : 2^X \to R$ such that $m_f^u(X) \geq f(X) \geq m_f^l(X)$ and $m_f^u(Y) = f(Y) = m_f^l(Y)$.*

We can apply the theorem as long as we have a decomposition of the objective function into two submodular functions. This decomposition sometimes can be obtained trivially from the set function structure (or problem structure). However, in general, it is conjectured to be NP-hard(Narasimhan and Bilmes, 2012). In our case, it is not trivial, but we successfully found a decomposition with a special technique and moreover, we made obtained submodular functions computationally possible.

The following shows how we decompose our objective function $f(S)$ as the difference of $f_1(S)$ and $f_2(S)$ both of which are submodular proved as following, i.e., $f(S) = f_1(S) - f_2(S)$.

Given a seed set S and a live graph g, we define the $B_1(S)$ as a benefit between activated users $I_g(S)$ and all users V, and define $B_2(S)$ as the benefit among all activated users $I_g(S)$ plus the benefit between the activated users $I_g(S)$ and the non-activated users $V \setminus I(S)$, which are formulated as follows:

$$\begin{aligned} B_1(S) &= \sum_{u \in I_g(S)} \sum_{v \in V} b(u,v) \\ &= \textstyle\sum_{\{u,v\} \subseteq I_g(S)} 2 \cdot b(u,v) + \sum_{u \in I_g(S)} \sum_{v \in V \setminus I_g(S)} b(u,v) \end{aligned}$$

$$B_2(S) = \textstyle\sum_{\{u,v\} \subseteq I_g(S)} b(u,v) + \sum_{u \in I_g(S)} \sum_{v \in V \setminus I_g(S)} b(u,v)$$

Thus we have

$$\begin{aligned} B(S) &= B_1(S) - B_2(S) \\ &= \textstyle\sum_{\{u,v\} \subseteq I_g(S)} b(u,v) \end{aligned}$$

And given a seed set S, we define the following functions

$$f_1(S) = \mathbb{E}_{g\sim D}[\alpha \cdot |I_g(S)| + \beta \cdot B_1(S)]$$

$$f_2(S) = \mathbb{E}_{g\sim D}[\beta \cdot B_2(S)]$$

Then we have

$$\begin{aligned} f(S) &= \mathbb{E}_{g\sim D}[\alpha \cdot |I_g(S)| + \beta \cdot \textstyle\sum_{\{u,v\}\subseteq I(S)} b(u,v)] \\ &= \mathbb{E}_{g\sim D}[\alpha \cdot |I_g(S)| + \beta \cdot (B_1(S) - B_2(S))] \\ &= \mathbb{E}_{g\sim D}[\alpha \cdot |I_g(S)| + \beta \cdot B_1(S)] - \mathbb{E}_{g\sim D}[\beta \cdot B_2(S)] \\ &= f_1(S) - f_2(S) \end{aligned}$$

Thus $f(S)$ is decomposed as a difference between function f_1 and f_2, and both of them are submodular. According to theorem 4 and our decomposed submodular functions, we can design an iterated sandwich algorithm to solve the Interaction-aware Influence Maximization problem. The main idea of our algorithm is to find the upper bound function and lower bound function based on the current seed set, then solve three functions: the upper bound function, the lower found function, and the objective function. then we choose the best solution from those three solutions, this best solution is then the seed set for generation of upper and lower bound functions in the next iteration. The procedure iterates until converged.

CHAPTER 4

A GENERAL METHOD OF ACTIVE FRIENDING IN DIFFERENT DIFFUSION MODELS IN SOCIAL NETWORKS

Authors – Shuyang Gu, Chuangen Gao, Ruiqi Yang,

Weili Wu, Hua Wang, and Dachuan Xu

The Computer Science Department, EC 31

The University of Texas at Dallas

800 West Campbell Road

Richardson, Texas 75080-3021

4.1 Introduction

Online social networks have been developing and prosperous during the last two decades, users of online social networks such as Twitter, Instagram have the nature of expanding social relationships. Thus one important social network service is to provide potential friends to a user that he or she might be interested in, such service is called friend recommendation. Different from friend recommendation, which is a passive way for a user to connect with a potential friend, in this paper, we tackle a different problem named active friending, which is the first studied in (Yang et al., 2013) as an optimization problem about how to friend a person in social networks. In short, the goal of the problem is how to assist a person to take proactive actions to make friends online with a target person by sending multiple invitations to a group of users, the probability that the target node is connected with the initiator is influenced by the common friends of those two users.

4.1.1 Motivation

There are lots of scenarios that people is interested in making friends with a person who is far away in his/her social network topology, a student may want to get connected with a professional who just gave a talk on campus, a fan might want to make friends with her idol, etc. Of course, a user has friending need can directly send an invitation to the target person, however, since that person does not know him/her at all, the probability that the friending request is accepted is low. Active friending exploits the social influence to maximize the friending acceptance probability, it assumes that the number of common friends between the initiator and a friending invitation receiver affects the probability that the receiver accepts the invitation. For a user to friend a distant target, the initiator may approach the friending target by making friends with a set of intermediate users, if the initiator succeed to friend some target's direct friends, the probability that the target accepts friending request is increased, to make friends with target's direct friends, the initiator might first get acquainted

with their friend and so on. Since an individual user does not know the network topology and the influence probability of the edges, the service provider will recommend two-hop away users to the initiator at each step to help the initiator approach the target.

4.1.2 Problem Description

Suppose we have a directed graph $G = (V, E)$ that represents a social network with an influence propagation model m. (In the studies of the influence of social networks, two influence propagation models are often adopted, which we will introduce in the following section.) Given a friending initiator s and his target node t as well as the set S which is the existing friends of s, the problem is to find a subset of nodes R to maximize the acceptance probability at the target t with a constraint that $|R| \leq r$, r is an integer. The acceptance probability at t is the probability that t can be activated through the subgraph induced by $\{s\} \cup S \cup R$, at the beginning only the nodes in $\{s\} \cup S$ are active. We call a node active in active friending problem if it is a friend of s or it accepts the friending invitation from s, and the probability that a node is activated depends on both the number of his/her active friends who influence him/her and the influence propagation probability in IC model or weight on edges in LT model which will be specified later.

4.1.3 Related Work

Social networks are studied extensively as a platform for viral marketing, a most famous problem in this area is influence maximization (Chen et al., 2009; Bharathi et al., 2007; Chen et al., 2010a; Tang et al., 2014a; Goyal et al., 2011a,b), which studies under some propagation model how to select the initial seed nodes such that the total number of influenced users is maximized. Domings and Richardson (Domingos and Richardson, 2001b; Richardson and Domingos, 2002b) first study the influence maximization as an optimization problem. Later, (Kempe et al., 2003a) adopts Linear Threshold (LT) Model and Indepence Cascade

(IC) Model as the influence propagation models and formulate influence maximization as a discrete optimization problem. The active friending problem was studied under a single propation model in previous studies, IC model studied in (Yang et al., 2013) and LT model studied in (Yuan et al., 2017). In this paper we find a general method to solve it in both models, in the meantime, we consider the general graph as input.

Active friending problem is first studied by (Yang et al., 2013), the problem is formulated in IC model and an algorithm is proposed based on an abstraction of the social graph structure between source and target as a tree and in-node aggregation, the abstraction consists of the most influential paths from S to t, where S is the set of current friends of s, setting $\log p_{uv}$ as the weight on edge (u, v), the abstracted graph contains the shortest path for each $s' \in S$ to t. p_{uv} denotes the acceptance probability for a node v who is influenced by node u, that is, the probability that the node v accepts the friending invitation from s given u is a friend of s.

The algorithm proposed in (Yang et al., 2013) employs the tree structure for efficiency but with a trade-off of effectiveness. The reason it sacrifices effectiveness is that for each node v in S, which is adopted as a leaf node in (Yang et al., 2013), only the most influential path between v and t is considered, however, in this case, if there exists a path that has a lot of common nodes with a most influential path, and both of them have the same source and destination, of course, such paths are not taken into account. Unfortunately, these paths are valuable because if we send invitations to both the nodes on the most influential path and a few extra nodes to form another path, we have two possible ways to friend the target yet the cost is not increased much. In addition, the most influential path is the one that has the most influential probability between v and t, however, the cost, the number of hops between them is not taken into account, there might be some path has lower influential probability but a small number of hops, this kind of paths deserve attention too. In this paper, we tackle the same active friending problem with general graph structure.

In (Yuan et al., 2017), the active friending with budget constraint under LT model is studied, the objective function is proved supermodular, and then effective algorithms based on superdifferentials are designed. Recently (Wu et al., 2018) studied a related problem under LT model, the problem is to find a minimal set of nodes in the network to send invitations to such that the acceptance probability at the target node could reach a threshold, the authors explore there exists relationship between the minimum subset cover problem and the formulated problem,, then they propose a randomized algorithm with an approximation factor of $O(\sqrt{n})$. Another problem related to active friending is group friending, the problem was proposed in (Chen et al., 2014), which studies the problem how to friend a group of users at the same time, the objective is to maximize the average number of active nodes on the list of target nodes.

The aforementioned studies are all on active friending, for the passive friending problem, a major direction is friend recommendation. (Silva et al., 2010) proposes a system recommending friends based on the topology of the network graphs. In (Wang et al., 2015), a recommendation system for users according to their lifestyles rather than social graphs is proposed. (Xie, 2010) explores interest-based features in friend recommendation. (Kwon and Kim, 2010) proposes a friend recommendation scheme using physical and social information.

4.1.4 Our Contributions

For all we know, it is novel to tackle the active friending problem in both the independent cascade model (IC) and the linear threshold model (LT) on general social graph. The contributions of this paper are summarized as follows:

- In this paper, active friending problem on the general graphs is studied in both IC and LT models. We prove the objective function of acceptance probability under the IC model is monotone nondecreasing nonsubmodular. We further prove the problem in LT model is NP-hard.

- By defining the objective function with respect to set of paths instead of set of nodes, we convert the problem in LT model from a supermodular maximization subject to cardinality constraint problem into a modular maximization subject to submodular knapsack constraint problem, and we convert the problem in IC model from a non-submodular maximization subject to cardinality constraint problem into a submodular maximization subject to submodular knapsack constraint problem.

- To solve these two problems, we propose a general method, the iterated submodular cost knapsack algorithm, which iteratively solves the problem by replacing the submodular knapsack constraint by a modular upper bound and then uses a greedy algorithm to solve the submodular cost modular knapsack problem until convergence.

- The experiments on real data sets verify that our proposed algorithms are effective.

4.2 Problem Formulation

4.2.1 Acceptance Probability in LT Model

In a social graph $G = (V, E)$, en edge $(u, v) \in E$ represents that node v is a friend of u. In LT model there is a positive weight w_{uv} associated with the edge (u, v) , which indicates the probability that node u could influence v. In active friending problem, w_{uv} represents the probability that node v would accept the invitation from a stranger s if node u is s's friend, and also u is a friend of v. Note that in LT model the sum of weights of all incoming edges to a node is less than one, $\Sigma_{u \in N^{in}(v)} w_{uv} \leq 1$ where $N^{in}(v) = \{u|(u, v) \in E\}$. At the beginning, only nodes in S are friends of s. Then node s sends invitations to all the nodes in set R that are also friends of s's friends and never received the friending invitation before, as the number of s's friends increases, the procedure continues until there are no nodes in R left have not receive friending invitations or there are no nodes in R are two-hop friends of s. In LT model, the friends of a node v has mutual exclusive social influence on the end

Table 4.1. Notation

Notation	Description
s	The initiator who has friending needs.
t	The target node that s is willing to friend.
S	The initiator's current friends set.
R	The intermediate nodes set selected for sending friending invitations.
P	The set of paths from s to t.
$a(R,t)$	The acceptance probability at t with invitations sent to nodes in R in LT model.
$b(R,t)$	The acceptance probability at t with invitations sent to nodes in R in IC model.
$c(X,t)$	The acceptance probability at t with invitations sent to nodes on paths in X in LT model.
$d(X,t)$	The acceptance probability at t with invitations sent to nodes on paths in X in IC model.
τ_p	The vertices on path p which are not s's direct friends.
$\mathcal{P}_i$	The probability that exactly i paths are contained in the random subgraph induced by $\{s,t\} \cup S \cup R$.
$\Omega(X,A)$	The probability that the randomized subgraph induced by nodes in X containing all paths in A.
$N^{in}(v)$	The set of nodes that are incoming neighbors of v.
$f(X)$	The number of nodes on the paths in X which are not in $s \cup S$.

node v to accept the friending quest. For a node v, the sum of the weights on the edges connecting any friend of s is the probability that v will accept the friending invitation. Note that each node in R could only get one friending invitation, which prevents the acceptance probabilities varies from cycling friending process. The formal definition of the acceptance probability is as follows.

Definition 5. *In LT model, the acceptance probability at t given invitation set R is defined as* $a(R,t) =$

$$\begin{cases} 1 & \text{if } t \in S \\ 0 & \text{if } t \notin R \cup S \text{ or } N^{in}(t) = \emptyset \\ \sum\limits_{u \in N^{in}(t)} a(R,u) \cdot w_{ut} & \text{otherwise} \end{cases} \tag{4.1}$$

For a set function f, let P be the ground set, f is submodular if it holds that $\Delta_j f(S) \geq \Delta_j f(T)$ for all $S \subseteq T$ and $j \notin T$, where $\Delta_j f(S) = f(S \cup j) - f(S)$ is defined as the gain of $j \in P$ added to a set $S \subseteq P$. On the other hand, f is considered supermodular if and only if $\Delta_j f(S) \leq \Delta_j f(T)$ for all $S \subseteq T$ and $j \notin T$. Function f is monotone when $\Delta_j f(S) \geq 0, \forall j \notin S, S \subseteq P$. The following theorem is about the supermodularity of the acceptance probability function proved in (Yuan et al., 2017).

Theorem 5. *In LT model the acceptance probability function $a(R,t)$ with respect to R is monotone nondecreasing supermodular.*

Definition 6. *(Active Friending with Respect to Set of Intermediate Nodes in LT model, AFIN-LT):*

$$\max \quad a(R,t) \tag{4.2}$$

$$s.t. \quad |R| \leq r, \tag{4.3}$$

By Theorem 5, the active friending problem in LT model is maximizing a monotone supermodular function with cardinality constraint.

4.2.2 Acceptance Probability in IC Model

In active friending problem, each node is in one of two states, either a friend of s or a stranger for s. In IC model every edge (u,v) is associated with a real number p_{uv} indicating if u is

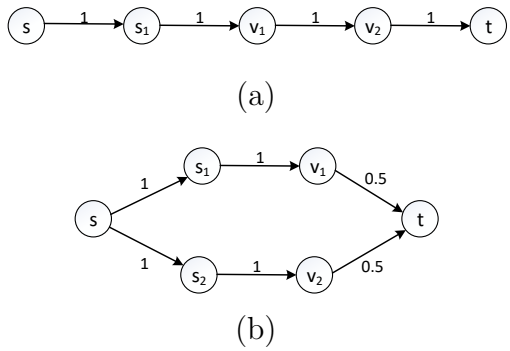


Figure 4.1. Counter Examples

s's friend, node v will accept the invitation from s influenced by u with the probability p_{uv}. Before the friending invitations are sent, only nodes in S are s's friends. Initially, we choose a subset of nodes R to send invitation to, but instead of sending the invitations all at once, in each step we send invitation to nodes in R that are two-hop friends of s. In each step, every new-friend node is going to influence its out-neighbors who is not friend of s and also get friending invitation at current step, where a node is new-friend if it accepts invitation from s in the very last step. If a stranger node v is being influenced by multiple nodes $u_1, u_2, ..., u_k$ at the same time, then those k events that u_i influences v are considered to be independent.

Definition 7. *In IC model, the acceptance probability at t is* $b(R,t) =$

$$\begin{cases} 1 & \text{if } t \in S \\ 0 & \text{if } t \notin R \cup S \text{ or } N^{in}(t) = \emptyset \\ 1 - \prod_{u \in N^{in}(t)} (1 - b(R,u) \cdot p_{ut}) & \text{otherwise} \end{cases} \tag{4.4}$$

We have the following result about the non-submodularity of the function $b(R,t)$.

Theorem 6. *$b(R,t)$ is a monotone nondecreasing, neither submodular nor supermodular with respect to R in IC model.*

Proof. It is obvious $b(R,t)$ is monotone nondecreasing. We prove it is non-submodular and non-supermodular by counter examples. For the example in figure 1(a), in which the number

on each edge indicates the weight associated with that edge. We consider a special case that the probability of a node accepting the invitation is 1 given the previous node on the path is friend of s. Initially, s only has one friend s_1. $b(\{t, v_1\} \cup \{v_2\}, t) - b(\{t, v_1\}, t) = 1 - 0 = 1 > b(\{t\} \cup \{v_2\}, t) - b(\{t\}, t) = 0 - 0 = 0$, which proves $b(R, t)$ is not submodular. For the example in figure 1(b), initially, s has only two friends s_1 and s_2, the probability at the last edge to the target t is 0.5. We have $b(\{t, v_1\} \cup \{v_2\}, t) - b(\{t, v_1\}, t) = 3/4 - 1/2 = 1/4 < b(\{t\} \cup \{v_2\}, t) - b(\{t\}, t) = 1/2 - 0 = 1/2$, which proves $b(R, t)$ is not supermodular. □

Now that we have the formal definition of the acceptance probability in IC, we define the active friending problem based on this function as follows.

Definition 8. *(Active Friending with respect to Set of Intermediate Nodes Problem in IC model, AFIN-IC):*

$$\max \quad b(R, t) \tag{4.5}$$

$$s.t. \quad |R| \leq r, \tag{4.6}$$

The active friending problem in IC model is monotone non-submodular maximization with cardinality constrain by Theorem 6.

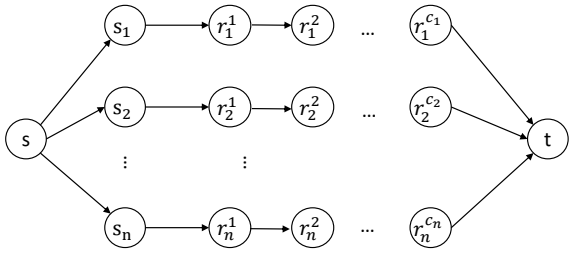


Figure 4.2. Reduction for NP hardness in LT model

4.2.3 Hardness Results

Let us assess the computational complexity of the active friending problem in different diffusion models.

Theorem 7. *Both AFIN-IC and AFIN-LT are NP-hard.*

Proof. Active friending problem is NP-hard under the independent cascade (IC) model, which was proved in (Yang et al., 2013). To prove it is NP-hard in LT model, we reduce 0-1 knapsack problem to our problem 4.2. The decision version of our problem is to determine if a subset R of nodes with $|R| \leq r$ exists such that $a(R,t)$ is at least δ. The decision version of 0-1 knapsack problem is given a group of items $U = \{u_1, u_2, \ldots, u_n\}$, each item u_i has a weight c_i and a value v_i, also we are given a maximum total weight W, to determine if there exists a subset of the items $U' \subseteq U$ such that total value of them achieves Q while the total weight of these items is within W. Note that 0-1 means that one item can be used at most once. We assume the weight c_i is an integer, which does not affect the NP-hardness of the problem. For an instance of 0-1 knapsack problem, we construct a graph as follows: create an initiator node s and a target node t, for each item $u_i \in U$ create a direct neighbor s_i for s, create a chain of c_i nodes $r_i^1, r_i^2, \ldots, r_i^{c_i}$ between s_i and t. The weight on edge $(r_i^{c_i}, t)$ is assigned $v_i / \sum_{i=1}^n v_i$, the other edges are assigned weight 1. The construction is illustrated as Figure 4.2. We show that there exists a subset U' of items that the total weight is at most W and the total value is at least Q in the 0-1 knapsack problem if and only if there exists a subset $R \subseteq V$ with $|R| \leq W + 1$ for s to send invitations to such that $a(R,t)$ is at least $Q / \sum_{i=1}^n v_i$.

To show the sufficient condition holds, if there exists a subset U' that the corresponding total weight is within W and total value is at least Q, selecting the set of nodes on the corresponding paths, denoted by R , will have the acceptance probability $a(R,t) = Q / \sum_{i=1}^n v_i$ at target t under LT model, note that since t is always included on the invitation list, so the total number of invitations is $W+1$. To show the "only if" part, suppose there exists a subset $R \subseteq V$ with$|R| \leq W+1$, to make the acceptance probability at t be at least $Q / \sum_{i=1}^n v_i$, then the nodes must be on some of the paths from s to t, since each path corresponds to an item in U, choose the corresponding items to form a subset U', then the total weight is within

W and the total value is at least Q. Thus finding a subset of nodes with size constraint at the meantime maximizing acceptance probability at the destination node is NP-hard in LT model. Thus the theorem follows. □

4.3 Problem Conversion

4.3.1 Problem Conversion in LT Model

Kempe, Kleiburg, and Tardos (Kempe et al., 2003a) proved that LT model can be viewed as a mutual-exclusive cascade (MC) model. MC model has similar definition to IC model, the difference is that when multiple newly-active users try to activate one user at a certain step, these events are considered mutual-exclusive, that is the total probability is the sum of probability of each event. The LT model is equivalent to the MC model when $w_{uv} = p_{uv}$. In these two models influence from different nodes to one node is mutual-exclusive, for example let us consider a simple social network with $V = \{x, y, z, u, v\}$ and $E = \{(x, u), (y, u), (z, u), (u, v)\}$. Assume x, y and z are active. By the mutual-exclusive property the probability that v could be activated is

$$(p_{xu} + p_{yu} + p_{zu})p_{uv} = p_{xu}p_{uv} + p_{yu}p_{uv} + p_{zu}p_{uv}$$

which shows the activating probability at the end node v is the sum of the activation probabilities from all active nodes along different paths. In general, we have the following lemma about this property in LT model. Let $Prob(p)$ be the probability that t is activated along path p originated from a seed in S, let P be the set of all paths between nodes in S and the target node t.

Lemma 3. *In LT model, assume nodes in a set S is active at the beginning, then after the propagation process ends a node t can be activated with probability equal to*

$$\sum_{p \in P} Prob(p).$$

The property can simplify some problems in LT model. For instance, the influence maximization problem in LT can be solved in polynomial time (Wang et al., 2016) if the topology is arborescence, while this problem is NP-hard in IC model (Lu et al., 2017; Bharathi et al., 2007). For active friending problem, we can get a similar property. Let P be the ground set of all paths from s to t in G, let P^R be the set of paths from s to t contained in the subgraph induced by $\{s\} \cup S \cup R$, note that set R must include node t for t to accept the friending quest in the end. Let τ_p be the vertices on path p which are not s's direct neighbors which results in cost in our constraint, and $a(\{\tau_p\}, t)$ be the probability that t accepts the invitation of s by s making friends with all nodes along path p. A path p can be viewed as a sequence of nodes or edges, here we represent a path p as a set of edges, we have $a(\{\tau_p\}, t) = \prod_{(u,v)\in p} w_{uv}$.

In LT model, given the initiator s, the neighbor set S of the initiator s, and the set of nodes R that receive invitations from s, we have the following property for the active friending problem.

Lemma 4. *In LT model, the acceptance probability at t equals to the sum of the acceptance probability along each path.*

$$a(R, t) = \sum_{p \in P^R} a(\{\tau_p\}, t)$$

Let $c(X, t) : (2^P, t) \to [0, 1]$ be the acceptance probability function on subset X of paths from s to t given ground set P, let $f(X) : 2^P \to \mathbb{N}$ be the function that maps a set of paths from s to t to the number of vertices in the graph induced by the union of the selected paths that are not in $s \cup S$. By lemma 4, we could convert the active friending problem 4.2 to the problem as follows:

Definition 9. *(Active Friending with Respect to Set of Paths in LT model, AFP-LT)*

$$\max \quad c(X, t) \tag{4.7}$$

$$s.t. \quad f(X) \leq r, \tag{4.8}$$

We can have the following theorem easily.

Theorem 8. *AFP-LT is equivalent to AFIN-LT.*

Proof. By lemma 4, only nodes that contribute to forming intact paths from s to t are effective in active friending problem, thus the problem of selecting a group of paths maximizing the acceptance probability is equivalent to selecting a group of effective nodes that achieves the same acceptance probability. □

Theorem 9. *The function $c(X,t)$ is modular with respect to X.*

Proof. Since $c(X,t) = \sum_{p\in X} c(\{p\},t)$, the linear combination of modular functions is still modular, so $c(X,t)$ is modular with respect to X. □ □

Theorem 10. *The function $f(X)$ is monotone nonnegative submodular with respect to X.*

Proof. To prove $f(X)$ is submodular, let $S \subseteq T \subseteq P$, $u \in V \backslash T$. We have $f(S\cup\{u\}) - f(S) \geq f(T\cup\{u\}) - f(T)$, the reason is the path u has more overlapped nodes with set T than with S since $S \subseteq T$, thus the submodularity follows. □

From the above two theorems, we have the problem 4.7 is a submodular cost knapsack(modular) problem. □

4.3.2 Problem Conversion in IC Model

In IC model, the influences for a certain node on different edges are independent. Let $Prob(R;A)$ be the probability that the random subgraph induced by $\{s,t\}\cup S\cup R$ contains all paths in A, where A is a subset of paths. Let $\mathcal{P}_i = \sum_{|A|=i,A\subseteq P} Prob(R;A)$, which is the probability that exactly i paths are contained in the random subgraph induced by $\{s,t\} \cup S \cup R$. The acceptance probability can be obtained by probabilistic version of inclusive-exclusive principle.

Lemma 5. *In IC model, given the subset of nodes R that receive the invitations, by inclusion–exclusion principle, a node t accepts the friending invitation from node s who has a direct friends set S with acceptance probability equal to*

$$b(R,t) = \mathcal{P}_1 - \mathcal{P}_2 + \mathcal{P}_3 - \mathcal{P}_4 + \cdots + (-1)^{|P|-1}\mathcal{P}_{|P|}.$$

Let $d(X,t) : (2^P, t) \to [0,1]$ be the acceptance probability function on set X of paths from s to t given ground set P. By lemma 5, we could convert the active friending problem with respect to the set of intermediate nodes 4.5 to the problem as follows.

Definition 10. *(Active Friending with respect to Set of Paths in IC model, AFP-IC)*

$$\max \quad d(X,t) \tag{4.9}$$

$$s.t. \quad f(X) \leq r, \tag{4.10}$$

Similar to theorem 8, we have the following theorem and omit the proof.

Theorem 11. *Problem AFP-IC is equivalent to problem AFIN-IC.*

We have converted the problem from selecting a subset of intermediate nodes between s to t to selecting a set of paths between s and t by the fact that only the nodes forming paths contribute to acceptance probability. Now that the overall acceptance probability is the probability that t is reachable from s in the randomized subgraph induced by the nodes in X, let us denote by $\Omega(X,A)$ the probability that the randomized subgraph induced by nodes in X containing all paths in A, then we can rewrite the formula of $\mathcal{P}_i$ as $\mathcal{P'}_i = \sum_{|A|=i, A\subseteq P} \Omega(X,A)$. Note that in the new problem 4.9, if $\mathcal{P}_i = \sum_{|A|=i, A\subseteq P} Prob(\bigcup_{p\in X} \tau_p, A)$, then $\mathcal{P}_i = \mathcal{P'}_i$, thus we have

$$d(X,t) = \mathcal{P}_1 - \mathcal{P}_2 + \mathcal{P}_3 - \mathcal{P}_4 + \cdots + (-1)^{|P|-1}\mathcal{P}_{|P|} \tag{4.11}$$

Theorem 12. *$d(X,t)$ is monotone nonnegative submodular with respect to X.*

Proof. Let us denote $d(X,t)$ as $d(X)$ for short. To prove $d(X)$ is submodular, we need to prove $d(A)+d(B) \geq d(A\cup B)+d(A\cap B)$, given $A, B \subseteq P$. Let $A \setminus B = X$, $B \setminus A = Y$, $A \cap B = Z$. Then $d(A)+d(B) = d(X\cup Z)+d(Y\cup Z)$. Let $\mathcal{P}_i^H = \sum_{|L|=i, L\subseteq P} \Omega(H,L)$. Then we have

$$d(X) = \mathcal{P}_1^X - \mathcal{P}_2^X + \cdots + (-1)^{|X|-1}\mathcal{P}_{|X|}^X = \sum_{i=1}^{|X|}(-1)^{i-1}\mathcal{P}_i^X$$

$$d(Z) = \mathcal{P}_1^Z - \mathcal{P}_2^Z + \cdots + (-1)^{|Z|-1}\mathcal{P}_{|Z|}Z = \sum_{i=1}^{|Z|}(-1)^{i-1}\mathcal{P}_i^Z$$

$$d(Y) = \mathcal{P}_1^Y - \mathcal{P}_2^Y + \cdots + (-1)^{|Y|-1}\mathcal{P}_{|Y|}^Y = \sum_{i=1}^{|Y|}(-1)^{i-1}\mathcal{P}_i^Y$$

And we have

$$\begin{aligned} d(X\cup Z) &= \mathcal{P}_1^{X\cup Z} - \mathcal{P}_2^{X\cup Z} + \cdots + (-1)^{|X\cup Z|-1}\mathcal{P}_{|X\cup Z|}^{X\cup Z} \\ &= \sum_{i=1}^{|X\cup Z|}(-1)^{i-1}\mathcal{P}_i^{X\cup Z} \\ d(Y\cup Z &= \mathcal{P}_1^{Y\cup Z} - \mathcal{P}_2^{Y\cup Z} + \cdots + (-1)^{|Y\cup Z|-1}\mathcal{P}_{|Y\cup Z|}^{Y\cup Z} \\ &= \sum_{i=1}^{|Y\cup Z|}(-1)^{i-1}\mathcal{P}_i^{Y\cup Z}. \end{aligned}$$

We have $\mathcal{P}_1^{X\cup Z} = \mathcal{P}_1^X + \mathcal{P}_1^Z$ because X and Z are disjoint sets of paths.

$$\mathcal{P}_2^{X\cup Z} = \mathcal{P}_2^X + \mathcal{P}_2^Z + \sum_{|L|=2, L\setminus X\neq\emptyset, L\setminus Z\neq\emptyset} \Omega(X\cup Z, L)$$

$$\mathcal{P}_3^{X\cup Z} = \mathcal{P}_3^X + \mathcal{P}_3^Z + \sum_{|L|=3, L\setminus X\neq\emptyset, L\setminus Z\neq\emptyset} \Omega(X\cup Z, L)$$

$$\mathcal{P}_i^{X\cup Z} = \mathcal{P}_i^X + \mathcal{P}_i^Z + \sum_{|L|=i, L\setminus X\neq\emptyset, L\setminus Z\neq\emptyset} \Omega(X\cup Z, L)$$

$$\begin{aligned} d(X\cup Z) &= \mathcal{P}_1^{X\cup Z} - \mathcal{P}_2^{X\cup Z} + \cdots + (-1)^{|X\cup Z|}\mathcal{P}_{|X\cup Z|}^{X\cup Z} \\ &= d(X) + d(Z) + \cdots + (-1)^{i-1} \sum_{|L|=i, L\setminus X\neq\emptyset, L\setminus Z\neq\emptyset} \Omega(X\cup Z, L) + \ldots \\ &+ (-1)^{|X\cup Z|} \sum_{|L|=|X\cup Z|, L\setminus X\neq\emptyset, L\setminus Z\neq\emptyset} \Omega(X\cup Z, L), i \geq 2 \end{aligned}$$

Similarly we have the formula for $d(Y \cup Z)$ and $d(X \cup Y \cup Z)$. So we have

$$d(A) + d(B) = d(X \cup Z) + d(Y \cup Z) = d(X) + d(Y) + 2d(Z) + \dots$$
$$+(-1)^{i-1}[\sum_{|L|=i, L\setminus X \neq \emptyset, L \setminus Z \neq \emptyset} \Omega(X \cup Z, L) + \sum_{|L|=i, L\setminus Y \neq \emptyset, L \setminus Z \neq \emptyset} \Omega(Y \cup Z, L)]$$

And we have

$$\begin{aligned} d(A \cup B) + d(A \cap B) = d(X \cup Y \cup Z) + d(Z) = d(X) + d(Y) + 2d(Z) + \dots \\ + (-1)^{(i-1)}[\sum_{|L|=i, L\setminus X \neq \emptyset, L \setminus Z \neq \emptyset} \Omega(X \cup Z, L) + \\ \sum_{|L|=i, L\setminus Y \neq \emptyset, L \setminus Z \neq \emptyset} \Omega(Y \cup Z, L) + \\ \sum_{|L|=i, L\setminus X \neq \emptyset, L \setminus Y \neq \emptyset} \Omega(X \cup Y \cup Z, L) + \\ \sum_{|L|=i, L\setminus X \neq \emptyset, L \setminus Y \neq \emptyset, L \setminus Z \neq \emptyset} \Omega(X \cup Y \cup Z, L)]. \end{aligned} \tag{4.12}$$

From the previous equations for $d(A \cup B) + d(A \cap B)$ and $d(A) + d(B)$, we have

$$\begin{aligned} & d(A \cup B) + d(A \cap B) - d(A) - d(B) \\ & = \sum_{i=2}^{|P|} (-1)^{i-1}[\sum_{|L|=i, L\setminus X \neq \emptyset, L \setminus Y \neq \emptyset} \Omega(X \cup Y \cup Z, L) \\ & + \sum_{|L|=i, L\setminus X \neq \emptyset, L \setminus Y \neq \emptyset, L \setminus Z \neq \emptyset} \Omega(X \cup Y \cup Z, L)] \leq 0 \end{aligned} \tag{4.13}$$

The reason is that the term with $i = 2, 4, 6, \dots$ is negative and the absolute value is greater than the term with $i + 1$, which is positive, so the summation is always less than zero. The key of the result is the probability that a set S of paths is contained in a randomized graph is always greater than the probability that a superset of S is contained in a randomized graph, and the difference between $d(A \cup B) + d(A \cap B)$ and $d(A) + d(B)$ contains the terms with cardinality of paths set greater or equal to 2, by inclusion-exclusion principle, the term with even cardinality is negative, and every other term is positive, but the overall value is

always less than 0. Therefore we have $d(A) + d(B) \geq d(A \cup B) + d(A \cap B)$, which proves the submodularity. □

Hence, the problem 4.9 is a submodular cost with submodular knapsack constraint problem. Now we have converted the problem in LT model from a supermodular maximization with cardinality constraint problem into a modular maximization with submodular knapsack constraint problem, and we converted the problem in IC model from a nonsubmodular maximization with cardinality constraint problem into a submodular maximization with submodular knapsack constraint problem. Of course both problems are NP-hard since when both the objective function and constraint function are modular, the problem becomes knapsack problem which is NP-hard, thus the general form is also NP-hard. Since the modular function can be treated as a special form of submodular function, the two problems might be solved by a general method.

4.4 Algorithms

Before we propose our algorithm, we want to point out that in order to get the input of our algorithm for the problem with respect to a subset of paths, we need to preprocess our data which is a graph $G = (V, E)$ to have the paths from the initiator to the target node. That is to say, our first step is to have all paths from s to t as our ground set P for the problem AFP-IC and AFP-LT. Note that finding one single path can be done by DFS in $O(|V| + |E|)$ time, but the number of simple paths can be very large, e.g. $O(n!)$.

Fortunately in our problem, we only need to search the paths whose lengths are less than $r + 2$, the longer paths are not going to be selected since we have the constraint of nodes number. And the path that visits multiple nodes in S are also not going to be selected, the reason is as follows. Suppose we want to select a path that visit multiple nodes in S, i.e. $path1 = (s, s_1, \ldots, s_2, \ldots, t)$, then there always exists a path $path2 = (s, s_2, \ldots, t)$, which is

shorter and the overall probability along path 2 must be larger than path 1. Therefore, we do not need to consider the paths whose length are larger or equal to $r + 2$.

4.4.1 Iterated Submodular Cost Knapsack Algorithm

Next we propose Iterated Submodular Cost Submodular Knapsack Algorithm (ISCK), which is inspired by (Iyer and Bilmes, 2013). The main idea is iteratively choosing a modular upper bound of the submodular constraint function f, which makes the problem become maximizing a submodular set function subject to a modular knapsack constrain, for such problem we get in each iteration a greedy algorithm provides a $1 - e^{-1}$ approximation (Sviridenko, 2004). The iterations continue until convergence. Note that function g in the algorithm stands for either c or d depending on the propagation diffusion model is LT or IC, function c is modular, the submodular generalization can also handle it.

To have the modular upper bound function of the constraint function f, we use the technique proposed in (Jegelka and Bilmes, 2011). For any subset Y, we can obtain the modular function which is an upper bound of f tight at Y in the following two ways (when referring either one, we use m_Y^f):

$$m_{Y,1}^f(X) \triangleq f(Y) - \sum_{j \in Y \setminus X} \Delta_j f(Y \setminus j) + \sum_{j \in X \setminus Y} \Delta_j f(\emptyset) \tag{4.14}$$

$$m_{Y,2}^f(X) \triangleq f(Y) - \sum_{j \in Y \setminus X} \Delta_j f(P \setminus j) + \sum_{j \in X \setminus Y} \Delta_j f(Y) \tag{4.15}$$

For the greedy algorithm to solve the problem $max\{g(X)|m_Y^f(X) \leq r\}$ in each iteration, we use the method in (Sviridenko, 2004) and propose Algorithm 2. This algorithm consists of two parts. The first is to try all the subsets with cardinality of 1 or 2, store the set which provides the best solution to the problem. The second part of the algorithm finds all sets with cardinality of 3, and then for each set add elements to the set greedily with

Algorithm 1 Iterated Submodular Cost Submodular Knapsack Algorithm

initialize $t \leftarrow 1$, $I^0 \leftarrow \emptyset$
repeat
  compute a modular upper bound $m^f_{I^{t-1}}(X)$ for f
  $I^t \leftarrow \text{argmax}_X\{g(X)|m^f_{I^{t-1}}(X) \leq r\}$;
  $t \leftarrow t + 1$;
until *converged, i.e.,* $I^t = I^{t-1}$
return I_t

Algorithm 2 Greedy Algorithm for Submodular Maximization subject to a Knapsack Constraint

Input: $G(V, E)$, function g, m^f_Y, integer r, paths set P.
initialize $Z^{max} \leftarrow \emptyset$
for $\forall U \subset P, |U| = 1$ *or* 2 **do**
  $Z^{max} \leftarrow \text{argmax}_{|U|=1 or 2} g(U), m^f_Y(U) \leq r$
end for
for $\forall U \subset P, |U| = 3$ **do**
  let $Z^0 \leftarrow U, t \leftarrow 1, P^0 \leftarrow P$
  repeat
    $i_t \leftarrow \text{argmax}_{i \in P^{t-1} \setminus Z^{t-1}} \frac{g(Z^{t-1} \cup \{i\}) - g(Z^{t-1})}{m^f_Y(i)}$
    if $\sum_{i \in Z^{t-1} \cup \{i_t\}} m^f_Y(i) \leq r$ **then**
      $Z^t \leftarrow Z^{t-1} \cup \{i_t\}$, $P^t \leftarrow P^{t-1}$
    else
      $Z^t \leftarrow Z^{t-1}$, $P^t \leftarrow P^{t-1} \setminus \{i_t\}$
    end if
    $t \leftarrow t + 1$
  until $P^t \setminus Z^t = \emptyset$
  $Z^{max} \leftarrow \text{argmax}\{g(Z^{max}), g(Z^t)\}$
end for
return Z^{max}

the solution feasible with respect to the knapsack constraint, store the best solution to the second part. Compare the two sets from the two parts, the algorithm outputs the one with greater objective function value.

it is easy to see that to evaluate the submodular constraint function f is just to calculate the cardinality of the corresponding nodes set from the union of paths sets, which is simple. The greedy algorithm assumes that it can evaluate the underlying objective function exactly. Note that the function $c(X)$ in LT model could be evaluated efficiently and exactly since it is a linear combination of probability product along a set of paths. However, for the problem in IC model, the acceptance probability $d(X)$ is difficult to evaluate. Fortunately, by simulating the friending influence process and sampling the result of reachability between s and t on the randomized subgraph formed by the selected nodes for sending invitations, we are able to obtain very close approximations to the probability $d(S)$.

Our approximation analysis for the proposed algorithm ISCK is based on the concept of curvature, which was defined to tighten the approximation performance for submodular maximization problems. The approximation ratio of the method of monotone submodular maximization subject to a cardinality constraint proposed in (Conforti and Cornuéjols, 1984), and monotone submodular maximization subject to matroid constraints (Vondrák, 2010), are improved from $(1-1/e)$ to $\frac{1}{\kappa_f}(1-e^{-\kappa_f})$, where κ_f is the curvature of the objective function f. Next let us give the definition of curvature. The total curvature κ_f of a submodular function f and the curvature $\kappa_f(S)$ with respect to a set $S \subseteq V$, as defined in (Iyer et al., 2013) are following.

$$\kappa_f = 1 - \min_{j \in V} \frac{\Delta_j f(P \setminus j)}{f(j)} \tag{4.16}$$

$$\kappa_f(S) = 1 - \min_{j \in S} \frac{\Delta_j f(S \setminus j)}{f(j)} \tag{4.17}$$

We also define an alternate notion of curvature as (Iyer et al., 2013).

$$\hat{\kappa}_f(S) = 1 - \frac{\sum_{j \in S} \Delta_j f(S \setminus j)}{\sum_{j \in S} f(j)} \tag{4.18}$$

Note that these three curvatures have the following relation.

Proposition 1. *Given that the set function f is monotone submodular, for set $S \subseteq P$ we have,*

$$\hat{\kappa}_f(S) \leq \kappa_f(S) \leq \kappa_f$$

Proof. It is easy to see that $\kappa_f(S) \leq \kappa_f(P) = \kappa_f$, since $\kappa_f(S)$ is a monotone-decreasing set function. To prove that $\hat{\kappa}_f(S) \leq \kappa_f(S)$, note that

$$1 - \kappa_f(S) = \min_{j \in S} \frac{\Delta_j f(S \setminus j)}{f(j)} \leq \frac{\Delta_j f(S \setminus j)}{f(j)}, \forall j \in S$$

Also note that

$$\begin{aligned} 1 - \hat{\kappa}_f(S) &= \frac{\sum_{j \in S} \Delta_j f(S \setminus j)}{\sum_{j \in S} f(j)} \\ &\geq \frac{\sum_{j \in S} (1 - \kappa_f(S)) f(j)}{\sum_{j \in S} f(j)} \\ &\geq 1 - \kappa_f(S) \end{aligned}$$

Hence, $\hat{\kappa}_f(S) \leq \kappa_f(S)$. □

Therefore $\hat{\kappa}_f(S)$ is the tightest notion of curvature. Intuitively, κ_f measures how close the function f is to modular function. If curvature $\kappa_f = 0$, the function f is modular. We further define a parameter that measures the maximum size of a feasible solution as

$$K_f = \max\{|X| : f(X) \leq r\}. \tag{4.19}$$

Given the definition of curvature and the definition of the modular upper bound function for a submodular function f. We have the following lemma for the relationship between the value of modular upper bound function and the value of the original submodular function of a feasible solution.

Lemma 6. *For a monotone submodular function f, and an modular upper bound function defined as $\hat{f}^m(X) \triangleq \sum_{j\in X} f(j)$, assume $\tilde{X}$ is a feasible solution s.t. $\hat{f}^m(\tilde{X}) \leq r$, it holds that*

$$f(\tilde{X}) \leq \hat{f}^m(\tilde{X}) = \sum_{j\in\tilde{X}} f(j) \leq \frac{K_f}{1+(K_f-1)(1-\kappa_f)} f(\tilde{X}).$$

Proof. We first show the inequality holds for $\hat{\kappa}_f(\tilde{X})$, and since $\kappa_f \geq \hat{\kappa}_f(\tilde{X})$, the inequality will hold for κ_f. By the property of submodularity and the curvature definition, we have following two facts

$$\text{Fact 1} : (1-\hat{\kappa}_f(\tilde{X}))\sum_{j\in\tilde{X}} f(j) = \sum_{j\in\tilde{X}} \Delta_j f(\tilde{X}\setminus j) \tag{4.20}$$

$$\text{Fact 2} : f(\tilde{X}) - f(k) \geq \sum_{j\in\tilde{X}\setminus k} \Delta_j f(\tilde{X}\setminus j), \forall j \in \tilde{X}. \tag{4.21}$$

Sum for all elements in $\arg\max_X\{|X| : f(X) \leq r\}$ in fact 1. Note that $K_f = \max\{|X| : f(X) \leq r\}$ by definition, then from Fact 1, we have

$$\begin{aligned}
&K_f f(\tilde{X}) - \sum_{k\in\tilde{X}} f(k) \\
&\geq \sum_{k\in\tilde{X}} \sum_{j\in\tilde{X}\setminus k} \Delta_j f(X\setminus j) \\
&\geq \sum_{k\in\tilde{X}} \sum_{j\in\tilde{X}} \Delta_j f(X\setminus j) - \sum_{k\in\tilde{X}} \Delta_j f(X\setminus k) \\
&\geq (K_f-1) \sum_{j\in\tilde{X}\setminus k} \Delta_j f(X\setminus j) \\
&\geq (K_f-1)(1-\hat{\kappa}_f(\tilde{X})) \sum_{k\in\tilde{X}} f(k)
\end{aligned}$$

Therefore, we have

$$\sum_{j\in\tilde{X}} f(j) \leq \frac{K_f}{1+(K_f-1)(1-\hat{\kappa}_f(\tilde{X}))} f(\tilde{X})$$

Since $\hat{\kappa}_f(\tilde{X}) \leq \kappa_f$, we have,

$$\sum_{j\in\tilde{X}} f(j) \leq \frac{K_f}{1+(K_f-1)(1-\kappa_f)} f(\tilde{X}).$$

□

We further obtain the following theorem about the performance of our proposed algorithm ISCK based on lemma 6.

Theorem 13. *Given* $\tilde{X}$ *as the optimal solution of* $\max\{g(X)|f(X) \leq \frac{r(1+(K_f-1)(1-\kappa_f))}{K_f}\}$*, it is guaranteed that the ISCK algorithm obtains a solution* X^t *such that* $g(X^t) \geq (1-1/e)g(\tilde{X})$*, where* $K_f = \max\{|X| : f(X) \leq r\}$.

Proof. The algorithm starts with $I^0 = \emptyset$, it chooses the upper bound of f as $m^f_{I^0}(X) = \sum_{j \in X} f(j)$, and then solve the corresponding problem by the greedy algorithm and iteratively continue this procedure until convergence. For the first iteration, the algorithm returns a set X^1 that $g(X^1) \geq (1-1/e)g(\tilde{X})$, where $\tilde{X}$ is the optimal solution of $\max\{g(X)|\sum_{j \in X} f(j) \leq r\}$. By lemma 6, we are also guaranteed that $\tilde{X}$ is the solution of $\max\{g(X)|f(X) \leq \frac{r(1+(K_f-1)(1-\kappa_f))}{K_f}\}$. The following iterations would improve the solution since this is an ascent algorithm. □

4.4.2 Greedy Algorithm

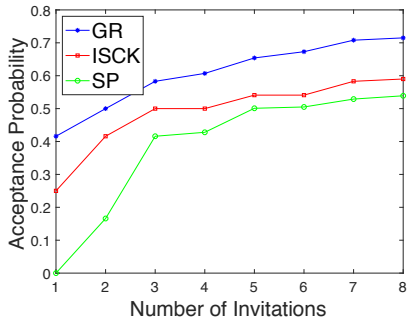
The other algorithm we propose is greedy algorithm, which starts with an empty set and then chooses path one by one $j \notin S : f(S \cup \{j\}) \leq r$ that maximizes $\Delta_j g(S)$.

Algorithm 3 Greedy Algorithm for Submodular Cost Submodular Knapsack

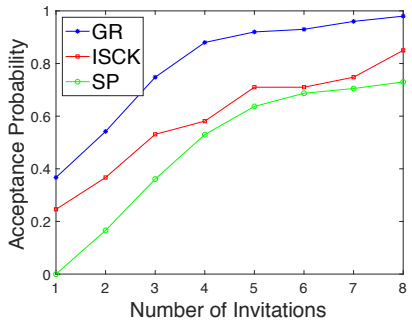
```
initialize S ← ∅
u ← argmax_{j∈P\S} Δ_j g(S);
while f(S ∪ {u}) ≤ r do
    S ← S ∪ {u};
    u ← argmax_{j∈P\S} Δ_j g(S);
end while
return S
```

The algorithm obtains an approximation factor of $\frac{1}{\kappa_g}(1-(\frac{K_f-\kappa_g}{K_f})^{k_f}) \geq \frac{1}{K_f}$, where $K_f = \max\{|X| : f(X) \leq r\}$ and $k_f = \min\{|X| : f(X) \leq r, \forall j \notin X, f(X \cup j) \geq r\}$ (Iyer and

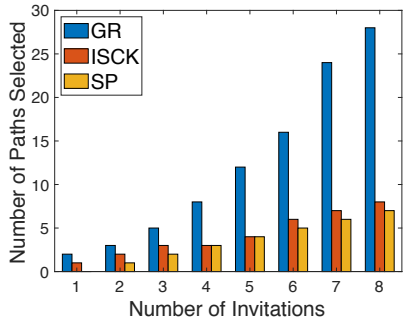
Bilmes, 2013). In the worst case when $k_f = 1$ and $K_f = n$, the approximation factor can be as bad as $1/n$.



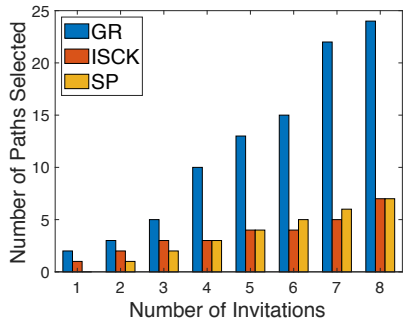
(a) Acceptance Probability in LT



(b) Acceptance Probability in IC



(c) The number of paths selected in LT



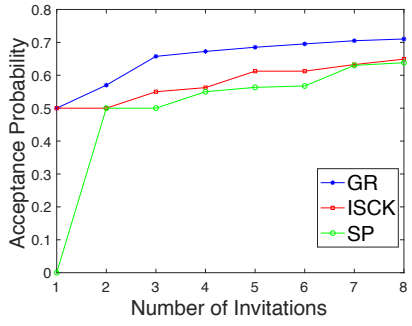
(d) The number of paths selected in IC

Figure 4.3. Experimental Results for Random Pair of Nodes

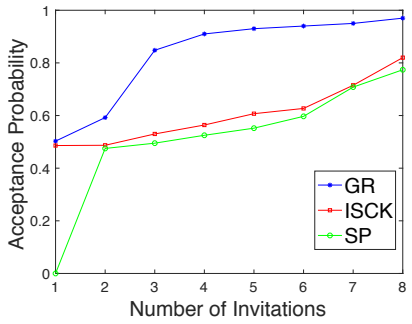
4.5 Performance Evaluation

We use the dataset Adolescent health for experiments, which encodes the friendship relation among adolescent students. The dataset is public available, which can be obtained from the website KONECT (http://konect.uni-koblenz.de). We set the propagation probability on each edge connecting node v for IC model as $1/degree(v)$, we adopt the same propagation probability as the weight on each edge for LT model, both of the settings are widely used in the simulations of other literature (Chen et al., 2010a; Kempe et al., 2003a).

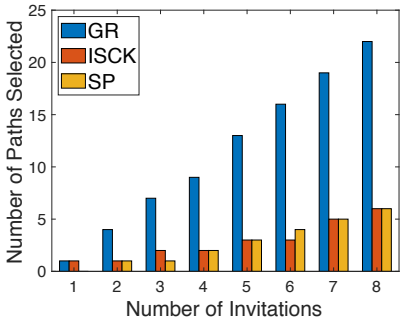
We compare the two algorithms proposed and design a heuristic Shortest Path Greedy algorithm for comparison. The Shortest Path Greedy algorithm(SP) first selects nodes on the



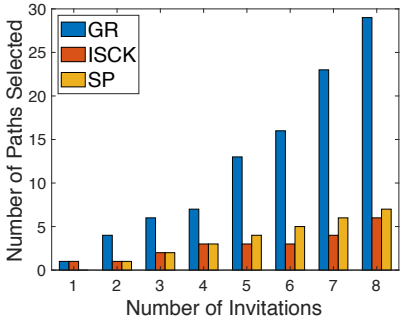
(a) Acceptance Probability in LT



(b) Acceptance Probability in IC



(c) Number of paths selected in LT



(d) Number of paths selected in IC

Figure 4.4. Experimental Results for Random Pair of Nodes (node 1 to node 5)

shortest path to send invitations if it does not violate the constraint, and then the nodes on the second shortest path but not ever been sent invitations to are selected if the invitations are still within r, the greedy selection goes on until the total number of invitations reaches the limit.

We implement the three algorithms In different influence propagation models LT and IC, we compare the acceptance probability against the number of invitations constraint r for different algorithms. We also record the number of paths selected by these three algorithms respectively.

Figures 4.3 and 4.4 respectively demonstrate the results for two pairs of random nodes selected as the initiator and the friending target. We can see that the acceptance probability increases as the number of invitations increase for all three algorithms, which conforms to the intuition. Our proposed algorithm ICSK and Greedy outperform SP significantly. The reason ICSK is worse than Greedy in the acceptance probability since it uses the modular

upper bound function as the knapsack constraint, which is stricter than the real submodular knapsack constraint, however, Greedy has a poor worst case guarantee. In figure 4.4, the ISCK has acceptance probability close to SP, the reason is that ISCK uses a stricter function as the knapsack constraint, which makes the probability lower.

The number of the selected paths by Greedy is much greater than the other two algorithms, however, the acceptance probability does not show that significant increase. This is consistent with the submodularity of our objective function, which has a diminishing return property, the increasing of the number of paths gains less on a larger set than on a small set. The number of paths selected by ISCK and that selected by the shorted path greedy algorithm is almost the same, in some cases, the shortest path greedy selects even slightly more than ISCK, but ISCK still performs better than the shortest path greedy algorithm, the reason is that ISCK selects path simultaneously considering the submodularity of the path cost, which is how many new invitations a path might increase, however, the shortest path greedy algorithm only considers the length of the path.

CHAPTER 5

INTERACTION-AWARE INFLUENCE MAXIMIZATION AND ITERATED SANDWICH METHOD[1]

Authors – Chuangen Gao, Shuyang Gu, Ruiqi Yang,

Jiguo Yu, Weili Wu, and Dachuan Xu

The Computer Science Department, EC 31

The University of Texas at Dallas

800 West Campbell Road

Richardson, Texas 75080-3021

[1]Reprinted by permission from Chuangen Gao, Shuyang Gu, Ruiqi Yang, Jiguo Yu, Weili Wu, and Dachuan Xu: Interaction-aware Influence Maximization and Iterated Sandwich Method, Theoretical Computer Science, ©2020 Elsevier

5.1 Introduction

Viral marketing has long been acknowledged as an effective marketing strategy. The development of online social networks such as Facebook and Twitter provide opportunities for large-scale online viral marketing in social networks. Under this circumstance, influence maximization (Kempe et al., 2003b) becomes a very popular research direction in the past decade, which could be described as the problem of finding a small set of most influential nodes so that the spread of influence in the network is maximized.

Most of the works focus on maximization of the spread of influence, which considers the number of users influenced by "word-of-mouth"effect in online social networks. These works are based on the assumption that the number of influenced users determines the profit of product. However, some types of products earn profit in a continuous way besides the sales of product itself. The online game is a good example. The game company's revenue usually comes from two parts, one is the revenue from selling the game product itself, and the other is from the proceeds of advertising and virtual item products. For the first part of revenue, the value of a single game product itself is fixed. The more game players buy game products, the more they earn. For the second part of the revenue is related to the interaction of the game player. When multiple people enter the same game scene online, the advertisement will be displayed and browsed. The more frequent the interaction between players, the more times an advertisement is presented and viewed, which will lead to more advertising revenue. In addition, when players participate in the game, they will use some props to complete the task. These items are virtual equipment, which can increase the experience and fun of the game players. These virtual products will also bring certain benefits(Fox et al., 2018).

We analyze such revenue model and define the interaction-aware influence maximization problem selecting a seed set to maximize the revenue dependent on the number of the influenced users and the interaction between activated nodes. The interaction-aware influence maximization problem is not submodular, thus the greedy strategy can't be directly applied

to our problem to get a guaranteed approximate solution. To solve this problem, we propose a sandwich theory which is based on the decomposition strategy that represents objective function as a difference of two submodular functions. And based on the sandwich theory and the decomposition we design two iterated sandwich algorithms.

The contributions of this paper are summarized as follows.

- We propose a new problem named interaction-aware influence maximization and we prove it is NP-hard and non-submodular.
- To solve this non-submodular problem, we propose the sandwich theory that for any set function there are a modular/submodular/supermodular upper bound and a modular/submodular/supermodular lower bound respectively. The sandwich theory is mainly based on the fact that any set function can be expressed as a difference between two submodular functions. And we successfully decompose our objective function into the difference of two submodular functions which are monotone nondecreasing.
- Based on the sandwich theory and the decomposed submodular functions mentioned above, we design two iterated sandwich algorithms to solve the interaction-aware influence maximization problem, which can get a data-dependent approximate solution.
- Through real data sets, we verify the effectiveness of our proposed algorithms.

5.2 Related Works

Influence maximization was first described as an algorithm problem by Domingos and Richardson (Domingos and Richardson, 2001a) (Richardson and Domingos, 2002a), they model the problem using Markov random fields and propose heuristic solutions. Kempe *et al.* (Kempe et al., 2003b) formulated the influence maximization problem from the view of combinatorial optimization and showed that the problem is NP-hard under both the IC and LT models,

they propose a simple greedy algorithm with an approximation ratio of $(1-1/e)$. However, a drawback of their work is the scalability of the greedy algorithm. Since then a number of efficient heuristic algorithms have been proposed in many works (Chen et al., 2009) (Rodriguez and Schölkopf, 2012) (Jung et al., 2012) (Chen et al., 2011)(Han et al., 2016)(Li et al., 2015). In (Leskovec et al., 2007), Leskovec *et al.* present a "lazy-forward" optimization in selecting new seeds, in which submodularity is exploited.

The influence maximization problem has been extended to practical scenarios in recent work. (Chen et al., 2015) studies the topic-aware influence maximization problem which considers user interests. In (Li et al., 2015), a keyword-based targeted influence maximization is proposed, where users who are relevant to a given advertisement are targeted. In (Han et al., 2016), the problem of privacy reserved influence maximization in both cyber-physical and online social networks is studied, they propose a model that merges both GPS data and relationship data from a social network. Bharathi et al.(Bharathi et al., 2007) study the game of innovation diffusion with multiple competing innovations, for example, multiple companies market competing products using viral marketing. In (Chen et al., 2011), Chen *et al.* propose an extension to the independent cascade model that incorporates the emergence and propagation of negative opinions, the new model has a quality factor to model the natural behavior of people turning negative to a product due to product defects.

Most of the works only consider the number of activated users, and the activities between users are first processed by (Wang et al., 2017b). However, their work does not maximize the influence spread in the meantime and only count activity strength of the directly connected users. In this paper, we propose the interaction-aware influence maximization problem which takes both parts into consideration.

5.3 Problem Formulation

In this section, we formulate the interaction-aware influence maximization problem in IC model formally and prove it is neither submodular nor supermodular by counter examples. For complexity, we prove it is NP-hard by a special case of the problem.

5.3.1 Interaction-aware Influence Maximization

In this paper, we use the directed graph $G = (V, E)$ to represent a social network, where V is the set of users and E is the set of social relations between users. Each edge $(u, v) \in E$ is assigned with a probability p_{uv} so that when u is active, v is activated by u with probability p_{uv}. And the benefit related to the interaction between nodes is represented by a nonnegative function $b : V \times V \to \mathbb{R}_{\geq 0}$, in which $b(u, v) = b(v, u)$ for the unordered pair $\{u, v\}$ of node u and v. Our goal is to find a set of initial users to maximize total profit related to both the number of the influenced nodes and the interaction between influenced nodes.

Since the randomness of propagation proess in IC model, consider a point in the cascade process when node u has just become active, and it attempts to activate its neighbor v, succeeding with probability $p_{u,v}$. We can view the outcome of this random event as being determined by flipping a coin of bias $p_{u,v}$. With all the coins flipped in advance, the edges in G for which the coin flip indicated a successful activation are declared to be live; the remaining edges are declared to be blocked(Kempe et al., 2003b). We use g to represent the outcome of this process which is called a live graph of G since it consists of all edges declared to be live. We denote as $g \sim D$, where D is the distribution of g. For any seed set S, denote by $I_g(S)$ the set of all active nodes at end of the cascade process in live graph g. It's cardinality is represented by $|I_g(S)|$.

Definition 11. *The total expected benefit would be defined as*

$$f(S) = \mathbb{E}_{g\sim D}[\alpha \cdot |I_g(S)| + \beta \cdot \sum_{\{u,v\}\subseteq I_g(S)} b(u,v)]$$
$$= \sum_g Prob[g] \cdot (\alpha \cdot |I_g(S)| + \beta \cdot \sum_{\{u,v\}\subseteq I_g(S)} b(u,v)) \tag{5.1}$$

The benefit consists of two parts, the first part denoted as $\alpha \cdot I_g(S)$ is related to the number of nodes that are finally activated, and the second part $\beta \cdot \sum_{\{u,v\}\subseteq I_g(S)} b(u,v)$ is related to the strength of the interaction between the active nodes. The parameters α , β are used to balance the weight of two parts of the profits, and $\{u,v\} \subseteq I(S)$ denotes the all unordered pair in the set $I(S)$. Note that for each unordered pair $\{u,v\}$, since $b(u,v) = b(v,u)$, we only compute once the benefit between them. The expectation is respected to g.

In this paper, we study the following problem.

Definition 12 (Interaction-aware Influence Maximization Problem, IAIM)**.** *Given a social network $G = (V,E)$, a propagation probability p_{uv} for each edge (u,v) under the IC model, a benefit function $b : V \times V \to \mathbb{R}_{\geq 0}$, and a positive integer k, find a set S of k seeds to maximize the expected profit through influence propagation:*

$$\max f(S) \tag{5.2}$$
$$s.t. |S| \leq k \tag{5.3}$$

5.3.2 Modularity of Objective Function

We say that $g(\cdot)$ is submodular if it satisfies a natural "diminishing returns" property: the marginal gain from adding an element to a set X is at least as high as the marginal gain from adding the same element to a superset of X. Formally, for every set X, Y such that $X \subseteq Y \subseteq V$ and every $e \in V \setminus Y$, it follows that

$$g(X \cup \{e\}) - g(X) \geq g(Y \cup \{e\}) - g(Y)$$

And it is monotone if $g(X) \leq g(Y)$ whenever $X \subseteq Y$.

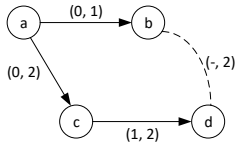


Figure 5.1. Counter example

Theorem 14. *$f(S)$ is neither submodular nor supermodular under IC model.*

Proof. We prove by the counter example shown in Fig.6.1. The first element in the tuple tied on each edge represents the propogation probability, and the second one denote the benefit between its two end nodes. For pairs $\{u, v\}$ between which there is no edge set $b(u, v) = 0$ except pair $\{b, d\}$. In Fig.6.1, $(0, 1)$ on edge (a, b) means propogation probability $p_{ab} = 0$ and $b(a, b) = 1$, then we have $f(\{a\}) = 1 + 0 = 1$, $f(\{a, b\}) = 2 + 1 = 3$, $f(\{a, d\}) = 2 + 0 = 2$ and $f(\{a, b, d\}) = 3 + 3 = 6$. Thus, $f(\{a, d\}) - f(\{a\}) < f(\{a, b, d\}) - f(\{a, b\})$, which implies $f(S)$ is not submodular. Also, we have $f(\{c\}) = 2 + 2 = 4$, $f(\{d, c\}) = 2 + 2 = 4$, $f(\{d\}) = 1$. Thus, $f(\{c\}) - f(\emptyset) > f(\{d, c\}) - f(\{c\})$ which implies $f(S)$ is not supermodular. □

5.3.3 Hardness Result

Theorem 15. *Interaction-aware influence maximization problem is NP-hard.*

Proof. We prove by showing a special case of the interaction-aware influence maximization problem is NP-hard, where $\beta = 0$, then it become the traditional influence maximization problem which is $NP - hard$. Note that a problem is NP-hard in a special case implies NP-hardness in general case. □

5.4 Sandwich Theory

Since the interaction-aware influence maximization problem is not submodular, the greedy strategy can't be directly applied to our problem to get a guaranteed approximate solution.

To solve this non-submodular problem, we propose the sandwich theory that for any set function there are a modular upper bound and a modular lower bound respectively(Wu et al., 2018). The sandwich theory is mainly based on the fact that any set function can be expressed as a difference between two submodular functions (Narasimhan and Bilmes, 2005b).

5.4.1 Preliminary

Before proposing our sandwich theory, let's first introduce a few important conclusions about the submodular function and the set function(Iyer and Bilmes, 2012b; Narasimhan and Bilmes, 2005b).

Lemma 7. *For any submodular set function $g(\cdot)$ on ground set V, we have the following two tight modular upper bounds that are tight at a given set Y (Iyer and Bilmes, 2012b):*

$$U^g_{Y,1}(X) \triangleq g(Y) - \sum_{j \in Y \setminus X} g(j \mid Y \setminus j) + \sum_{j \in X \setminus Y} g(j \mid \emptyset) \tag{5.4}$$

$$U^g_{Y,2}(X) \triangleq g(Y) - \sum_{j \in Y \setminus X} g(j \mid V \setminus j) + \sum_{j \in X \setminus Y} g(j \mid Y) \tag{5.5}$$

Lemma 8. *For any submodular set function $g(\cdot)$, a modular lower bound of $g(\cdot)$ is tight at a given set Y can be obtained as follows (Iyer and Bilmes, 2012b). Let σ be a permutation of V and define $P^\sigma_i = \{\sigma(1), \sigma(2), \ldots, \sigma(i)\}$ as σ's chain containing Y, in which $P^\sigma_0 = \emptyset$ and $P^\sigma_{|Y|} = Y$. Define*

$$L^g_{Y,\sigma}(\sigma(i)) = g(P^\sigma_i) - g(P^\sigma_{i-1}). \tag{5.6}$$

Then

$$L^g_{Y,\sigma}(X) \triangleq \sum_{v \in X} L^g_{Y,\sigma}(v) \tag{5.7}$$

is a tight lower bound of $g(X)$, i.e., $L^g_{Y,\sigma}(X) \leq g(X)$, $\forall X \subseteq V$, and $L^g_{Y,\sigma}(Y) = g(Y)$.

Lemma 9. *Every set function $f : 2^X \to R$ can be expressed as the difference of two monotone nondecreasing submodular functions f_1 and f_2, i.e $f = f_1 - f_2$, where X is a finite set (Narasimhan and Bilmes, 2005b).*

5.4.2 Sandwich Theory

Theorem 16. *For any set function $f : 2^X \to R$ and any set $Y \subset X$, there are two modular funtions $m_f^u : 2^X \to R$ and $m_f^l : 2^X \to R$ such that $m_f^u(X) \geq f(X) \geq m_f^l(X)$ and $m_f^u(Y) = f(Y) = m_f^l(Y)$.*

Proof. This theorem means that for any set function we can find a modular upper bound and modular lower bound which are exact at some given point. By lemma 9, there exist submodular function f_1 and f_2 such that $f = f_1 - f_2$. By lemmas 7 and 8, there exist modular functions $U_Y^{f_1}, L_Y^{f_1}$ such that $U_Y^{f_1}(X) \geq f_1(X) \geq L_Y^{f_1}(X)$, and $U_Y^{f_1}(Y) = f_1(Y) = L_Y^{f_1}(Y)$ for submodular function f_1. By the same reason, there exist modular functions $U_Y^{f_2}, L_Y^{f_2}$ such that $U_Y^{f_2}(X) \geq f_2(X) \geq L_Y^{f_2}(X)$, and $U_Y^{f_2}(Y) = f_2(Y) = L_Y^{f_2}(Y)$ for submodular function f_2. We set $m_f^u = U_Y^{f_1} - L_Y^{f_2}$, and $m_f^l = L_Y^{f_1} - U_Y^{f_2}$, then $m_f^u(X) \geq f(X) \geq m_f^l(X)$ and $m_f^u(Y) = m_f^l(Y) = f(Y) = f_1(Y) - f_2(Y)$. Note that both m_f^u and m_f^l are modolar, since the linear combination of modular functions is still modular. □

Theorem 17. *For any set function $f : 2^X \to R$ and any set $Y \subset X$, there are two submodular funtions $h_f^u : 2^X \to R$ and $h_f^l : 2^X \to R$ such that $h_f^u(X) \geq f(X) \geq h_f^l(X)$ and $h_f^u(Y) = f(Y) = h_f^l(Y)$.*

Proof. This theorem means that for any set function we can find a submodular upper bound and submodular lower bound which are exact at some given point. By the lemma 9, there exist submodular function f_1 and f_2 such that $f = f_1 - f_2$. By the lemmas 7 and 8, there exist modular functions $U_Y^{f_2}, L_Y^{f_2}$ such that $U_Y^{f_2}(X) \geq f_2(X) \geq L_Y^{f_2}(X)$, and $U_Y^{f_2}(Y) = f_2(Y) = L_Y^{f_2}(Y)$ for submodular function f_2. We set $h_f^u = f_1 - L_Y^{f_2}$, and $h_f^l = f_1 - U_Y^{f_2}$, then

$h_f^u(X) \geq f(X) \geq h_f^l(X)$ and $h_f^u(Y) = h_f^l(Y) = f(Y) = f_1(Y) - f_2(Y)$. Note that both h_f^u and h_f^l are submodular. □

Theorem 18. *For any set function $f : 2^X \to R$ and any set $Y \subset X$, there are two supermodular funtions $p_f^u : 2^X \to R$ and $p_f^l : 2^X \to R$ such that $p_f^u(X) \geq f(X) \geq p_f^l(X)$ and $p_f^u(Y) = f(Y) = p_f^l(Y)$.*

Proof. By the lemma 9, there exist submodular function f_1 and f_2 such that $f = f_1 - f_2$. By lemmas 7 and 8, there exist modular functions $U_Y^{f_1}, L_Y^{f_1}$ such that $U_Y^{f_1}(X) \geq f_1(X) \geq L_Y^{f_1}(X)$, and $U_Y^{f_1}(Y) = f_1(Y) = L_Y^{f_1}(Y)$ for submodular function f_1. We set $p_f^u = U_Y^{f_1} - f_2$, and $p_f^l = L_Y^{f_1} - f_2$, then $p_f^u(X) \geq f(X) \geq p_f^l(X)$ and $p_f^u(Y) = p_f^l(Y) = f(Y) = f_1(Y) - f_2(Y)$. □

5.4.3 DS decomposition

Since our sandwich theorem is based on the DS decomposition of a set function that expressing it as a difference between two submodular functions. Thus the key point is finding such a decomposition. However, it is unknown whether there exists a polynomial-time algorithm for finding such a pair of monotone nondecreasing submodular functions for every given set function. Moreover, the DS decomposition in this paper is nontrivial and two constructed monotone nondecreasing submodular functions are easily computable.

Give a seed set S and a live graph g, we define the $B_1(S)$ as benefit between activated users $I_g(S)$ and all users V, and define $B_2(S)$ as the benefit among all activated users $I_g(S)$ plus the benefit between the activated users $I_g(S)$ and the non-activated users $V \setminus I(S)$, which are formulated as follows:

$$B_1(S) = \sum_{u \in I_g(S)} \sum_{v \in V} b(u, v) \tag{5.8}$$

$$B_2(S) = \sum_{\{u,v\} \subseteq I_g(S)} b(u, v) + \sum_{u \in I_g(S)} \sum_{v \in V \setminus I_g(S)} b(u, v) \tag{5.9}$$

And given a seed set S, we define the following functions

$$f_1(S) = \mathbb{E}_{g\sim D}[\alpha \cdot |I_g(S)| + \beta \cdot B_1(S)] \tag{5.10}$$

$$f_2(S) = \mathbb{E}_{g\sim D}[\beta \cdot B_2(S)] \tag{5.11}$$

Then we have

$$\begin{aligned} f(S) &= \mathbb{E}_{g\sim D}[\alpha \cdot |I_g(S)| + \beta \cdot \sum_{\{u,v\}\subseteq I(S)} b(u,v)] \\ &= \mathbb{E}_{g\sim D}[\alpha \cdot |I_g(S)| + \beta \cdot (B_1(S) - B_2(S))] \\ &= \mathbb{E}_{g\sim D}[\alpha \cdot |I_g(S)| + \beta \cdot B_1(S)] - \mathbb{E}_{g\sim D}[\beta \cdot B_2(S)] \\ &= f_1(S) - f_2(S) \end{aligned} \tag{5.12}$$

Actually $f(S)$ is decomposed as a difference between function f_1 and f_2, now we prove both of them are submodular.

Lemma 10. *$B_1(S)$ is submodular and monotone under the IC model.*

Proof. According the definition of $B_1(S)$ shown in equation 5.8, we have

$$\begin{aligned} B_1(S) &= \sum_{u\in I_g(S)} \sum_{v\in V} b(u,v) \\ &= \sum_{\{u,v\}\subseteq I_g(S)} 2 \cdot b(u,v) + \sum_{u\in I(S)} \sum_{v\in V\setminus I_g(S)} b(u,v) \qquad (5.13) \\ &= \sum_{v\in I_g(S)} w(v) \qquad (5.14) \end{aligned}$$

where $I_g(S)$ denotes the set of all active nodes in a live graph g, and $w(v)$ is the weight of the node v which is defined as follows

$$w(v) = \sum_{u\in V} b(v,u) \tag{5.15}$$

It is actually the sum of benefit between v and the remaining nodes in V. Thus, we can see that the $B_1(S)$ is essentially a weighted version of influence spread. And the submodularity follows immediately(Kempe et al., 2003b). □

Since the profit function $b : V \times V \to \mathbb{R}_{\geq 0}$ is nonnegative which means the profit of each pair of nodes is non-negative. Thus the weight of every node is non-negative and the monotonicity of $B_1(S)$ follows immediately. For the submodularity, we need prove $B_1(M \cup \{v\}) - B_1(M) \geq B_1(N \cup \{v\}) - B_1(N)$, such that $M \subseteq N \subseteq V$ and $v \in V \setminus N$. The left side of inequality is the weight of nodes which can be activated by v but can not by M. The right side is the weight of nodes which can be activated by v but can not by N. We have $I_g(v) - I_g(M) \supseteq I_g(v) - I_g(N)$, since $M \subseteq N$ and $I_g(M) \subseteq I_g(N)$. And the submodularity follows immediately. □

Theorem 19. *$f_1(S)$ is submodular and monotone under the IC model.*

Proof. According to the definition of $f_1(S)$ shown in equation 5.10, we have

$$\begin{aligned} f_1(S) &= \mathbb{E}_{g\sim D}[\alpha \cdot |I_g(S)| + \beta \cdot B_1(S)] \\ &= \sum_g Prob[g] \cdot (\alpha \cdot |I_g(S)| + \beta \cdot B_1(S)) \end{aligned} \tag{5.16}$$

The first part $|I_g(S)|$ of the $f_1(S)$ is the traditional influence maximization problem which is submodular (Kempe et al., 2003b). Given $\alpha \geq 0$, $\beta \geq 0$, $Prob[g] \geq 0$ and $B_2(S)$ is submodular prove and monotone proved by lemma 10, $f_1(S)$ is submodular and monotone follows immediately since the fact that a non-negative linear combination of submodular functions is also submodular. □

Lemma 11. *$B_2(S)$ is submodular and monotone under the IC model.*

Proof. Let M, N to be any two seed sets such that $M \subseteq N \subseteq V$ and x to be any element such that $x \in V \setminus N$. According the definition of $B_2(S)$ shown in equation 5.9, we have Then

we have

$$
\begin{aligned}
&B_2(M \cup \{x\}) - B_2(M) \\
&= \sum_{\{u,v\} \subseteq I_g(x) \setminus I_g(M)} b(u,v) + \sum_{u \in I_g(x) \setminus I_g(M)} \sum_{v \in V \setminus I_g(M) \cup I_g(x)} b(u,v)
\end{aligned} \tag{5.17}
$$

Through the same analysis process, we can get

$$
\begin{aligned}
&B_2(N \cup \{x\}) - B_2(N) \\
&= \sum_{\{u,v\} \subseteq I_g(x) \setminus I_g(N)} b(u,v) + \sum_{u \in I_g(x) \setminus I_g(N)} \sum_{v \in V \setminus I_g(N) \cup I_g(x)} b(u,v)
\end{aligned} \tag{5.18}
$$

Comparing all terms on the right-hand sides of 5.17 and 5.18, since $M \subseteq N$,we have $I_g(M) \subseteq I_g(N)$. So $I_g(x) \setminus I_g(M) \supseteq I_g(x) \setminus I_g(N)$ and $V \setminus I_g(M) \cup I_g(x) \supseteq V \setminus I_g(N) \cup I_g(x)$ follows. Thus both the first item and second item of 5.17 are greater than the first item and second item of 5.18 respectively. Through above analysis, we obtain $B_2(M \cup \{x\}) - B_2(M) \geq B_1(N \cup \{x\}) - B_2(N)$. Therefore, $B_2(S)$ is submodular.

For monotonicity, we need prove $B_2(M) \leq B_2(N)$, which is non-decreasing. According to equation 5.9, we have

$$
\begin{aligned}
&B_2(M) \\
&= \sum_{\{u,v\} \subseteq I_g(M)} b(u,v) + \sum_{u \in I_g(M)} \sum_{v \in V \setminus I_g(M)} b(u,v) \\
&= \sum_{\{u,v\} \subseteq I_g(M)} b(u,v) + \sum_{u \in I_g(M)} \sum_{v \in I_g(N) \setminus I_g(M)} b(u,v) \\
&\qquad + \sum_{u \in I_g(M)} \sum_{v \in V \setminus I_g(N)} b(u,v)
\end{aligned} \tag{5.19}
$$

$$
\begin{aligned}
&B_2(N) \\
&= \sum_{\{u,v\} \subseteq I_g(N)} b(u,v) + \sum_{u \in I_g(N)} \sum_{v \in V \setminus I_g(N)} b(u,v)
\end{aligned} \tag{5.20}
$$

Since $I_g(M) \subseteq I_g(N)$, we have $\forall(i,j) \in \{(u,v) \mid u \in I_g(M), v \in I_g(N) \setminus I_g(M)\}$, $i \in I_g(N)$, $j \in I_g(N)$. Thus the sum of first two items of $B_2(M)$ is less than the first item of $B_1(N)$. By the same reason, we have the third item of $B_2(M)$ is less than the second item of $B_2(N)$. Through above analysis, the monotonicity of $B_2(S)$ follows immediately. □

Theorem 20. *$f_2(S)$ is submodular and monotone under the IC model.*

Proof. According the definition of $f_2(S)$ shown in equation 5.11, we have

$$\begin{aligned} f_2(S) &= \mathbb{E}_{g\sim D}[\beta \cdot B_2(S)] \\ &= \beta \cdot \sum_g Prob[g] \cdot B_2(S) \end{aligned} \tag{5.21}$$

Given $\beta \geq 0$, $Prob[g] \geq 0$ and $B_2(S)$ is submodular and monotone proved by lemma 11, $f_2(S)$ is submodular and monotone follows immediately since the fact that a non-negative linear combination of submodular functions is also submodular. □

5.5 Algorithms

According to the sandwich theorem and our DS decomposition, we designed a iterated sandwich algorithm. The algorithm named iterated modular sandwich algorithm is based on the modular upper and lower bounds of our objective function. Our algorithm are guaranteed to gain data dependent approximation solutions.

5.5.1 Iterated Sandwich Algorithm

For algorithm 1 named Iterated Modular Sandwich Algorithm, we iteratively find the optimal solutions for three functions: the modular upper bound function $m_t^u(X)$, the modular lower bound function $m_t^l(X)$ and the original objective function $f(X)$, and then choose the best solution from $f(X)$ as the input of next iteration.

Algorithm 4 Iterated Modular Sandwich Algorithm

initialize $\epsilon > 0$, an integer k, $t \leftarrow 0$, $S^t \leftarrow$ a random seeds of size k, $S_{\max} = S^0$
repeat
  choose a permutation σ^t whose chain contains S^t
  construct a modular upper bound $U^{f_1}_{S^t}(X)$ (and, $U^{f_2}_{S^t}(X)$) and a modular lower bound $L^{f_1}_{S^t,\sigma^t}(X)$ (and, $L^{f_2}_{S^t,\sigma^t}(X)$) for f_1 (and, f_2)
  $S^t_u \leftarrow \text{argmax}_X m^u_t(X) = U^{f_1}_{S^t}(X) - L^{f_2}_{S^t,\sigma^t}(X)$;
  $S^t_l \leftarrow \text{argmax}_X m^l_t(X) = L^{f_1}_{S^t,\sigma^t}(X) - U^{f_2}_{S^t}(X)$;
  $S_o \leftarrow \text{argmax}_X f(X)$;
  Let $S^{t+1} \leftarrow \text{argmax}_X(f(S^t_u), f(S^t_l), f(S_o))$
  if $f(S^{t+1}) \geq (1+\epsilon) f(S_{\max})$ **then**
    $S_{\max} \leftarrow S^{t+1}$
    $t \leftarrow t+1$
  end if
until *converged, i.e.,* $f(S^{t+1}) < (1+\epsilon) f(S_{\max})$
return $S_{\max}$

Algorithm 5 Iterated Submodular Sandwich Algorithm

initialize $\epsilon > 0$, an integer k, $t \leftarrow 0$, $S^t \leftarrow$ a random seeds of k; $S_{\max} = S^0$
decompose the f as the difference between two submodular functions,i.e., $f = f_1 - f_2$
repeat
  choose a permutation σ^t whose chain contains S^t;
  compute a modular upper bound $U^{f_2}_{S^t}(X)$ and a modular lower bound $L^{f_2}_{S^t,\sigma^t}(X)$ for f_2
  $S^t_u \leftarrow \text{argmax}_X h^u_t(X) = f_1(X) - L^{f_2}_{S^t,\sigma^t}(X)$;
  $S^t_l \leftarrow \text{argmax}_X h^l_t(X) = f_1(X) - U^{f_2}_{S^t}(X)$;
  $S_o \leftarrow \text{argmax}_X f(X)$;
  $S^{t+1} \leftarrow \text{argmax}_X(f(S^t_u), f(S^t_l), f(S_o))$
  if $f(S^{t+1}) \geq (1+\epsilon) f(S_{\max})$ **then**
    $S_{\max} \leftarrow S^{t+1}$
    $t \leftarrow t+1$
  end if
until *converged, i.e.,* $f(S^{t+1}) < (1+\epsilon) f(S_{\max})$
return S_{max}

For algorithm 2 named Iterated Submodular Sandwich Algorithm, we iteratively find the optimal solutions for three functions: the submodular upper bound function $h_t^u(X)$, the submodular lower bound function $h_t^l(X)$ and the original objective function $f(X)$, and then choose the best solution for $f(X)$ from them as the input of next iteration.

Based on the decomposition of f which is a difference of two submodular functions $f_1(X) - f_2(X)$, we obtain $m_t^u(X)$ and $m_t^l(X)$ in each iteration as follows: we derive the modular upper bound and modular lower bound for $f_1(X)$ and $f_2(X)$ with respect to current seed set respectively, which are denoted as $U_{S^t}^{f_1}(X)$, $L_{S^t}^{f_1}(X)$ and $U_{S^t}^{f_2}(X)$, $L_{S^t}^{f_2}(X)$; the modular upper bound function $m_f^u(X)$ for $f(X)$ is then the difference between $U_{S^t}^{f_1}(X)$ and $L_{S^t}^{f_2}(X)$, the modular lower bound function $m_t^l(X)$ for $f(X)$ is the difference between $L_{S^t}^{f_1}(X)$ and $U_{S^t}^{f_2}(X)$ as described in Theorem 16. Note that the solution S_o for the original function $f(X)$ can be calculated by a greedy algorithm. Since $m_t^u(X)$ and $m_t^l(X)$ are modular, S_u^t is simply the set of the first k elements that maximize $m_t^u(X)$, S_l^t is simply the set of the first k elements that maximize $m_t^l(X)$. At the end of each iteration, the algorithm updates S_{max} if the solution obtained in the current iteration is better. The algorithm iterates until converged, i.e. $S^t = S^{t-1}$.

We do not design an iterated supermodular sandwich algorithm which could be derived from Theorem 18 of sandwich theory since the existing algorithms to the supermodular maximization problem under the constraint of size k do not work efficiently in general.

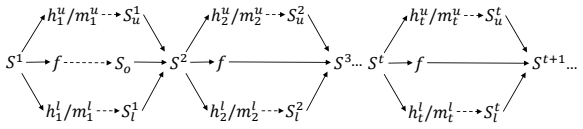


Figure 5.2. Iterated sandwich algorithms flow

The main idea of our algorithms is to find the upper bound function and lower bound function based on the current seed set, then solve three functions: the upper bound function, the lower bound function, and the objective functions. then we choose the best solution from

those three solutions, this best solution is then the seed set for generation of upper and lower bound functions in next iteration as shown in Fig 5.2. The procedure iterates until converged.

5.5.2 Analysis

We say a set S is local optimum solution of a submodular function f, if for any $T \subseteq S$ or $T \supseteq S$, we have $f(S) \geq f(T)$. Similarly, we say set S is a $(1+\epsilon)$-approximate local optimum solution of a submodular function f, if for any $e \in V$, we have $(1+\epsilon)f(S) \geq f(S \cup \{e\})$ and $(1+\epsilon)f(S) \geq f(S \setminus \{e\})$. Consider iteration t, let $F_t(X) = U^{f_1}_{S^t}(X) - L^{f_2}_{S^t,\sigma^t}(X)$ for any $X \subseteq V$, we firstly give a notation *approximation coefficient* $\eta_t = \max_{X \subseteq V} \frac{F_t(X)}{f(X)}$, which is denoted as how the approximate extent of the replace function $F_t(X)$ to the original function $f(X)$. Let $\eta = \max_t \eta_t$. Now we can bound the value of set returned by the Iterative Modular Sandwich algorithm by the following theorem.

Theorem 21. *Let $S_{\max}$ be the returned set by Algorithm 4, then we have $S_{\max}$ either is a $(1+\epsilon)$-approximate local maximum solution by justly checking $O(n)$ permutations, or is $\frac{1}{\eta(1+\epsilon)}$-approximation solution for the interaction-aware influence maximization problem.*

Proof. For the case that the return set $S_{\max}$ is derived by the modular lower bound and set t as the terminal iteration of Algorithm 4, then we have

$$\begin{aligned}
(1+\epsilon)f(S^t) &\geq f(S^{t+1}) = f_1(S^{t+1}) - f_2(S^{t+1}) \\
&\geq \frac{1}{\eta} \cdot (U^{f_1}_{S^t}(S^{t+1}) - L^{f_2}_{S^t,\sigma^t}(S^{t+1})) \\
&\geq \frac{1}{\eta} \cdot (U^{f_1}_{S^t}(OPT) - L^{f_2}_{S^{t+1},\sigma^t}(OPT)) \\
&\geq \frac{1}{\eta} \cdot (f_1(OPT) - f_2(OPT)) \\
&= \frac{1}{\eta} \cdot f(OPT)
\end{aligned}$$

The first inequality is derived by the line 9 of Algorithm 4, the second inequality follows the definition of approximation coefficient, the third inequality is derived by the optimality of S^{t+1} according to $F(\cdot)$ at iteration $t+1$, and last inequality is obtained by the construction of the upper and lower bounds. Thus we have

$$f(S^t) \geq \frac{1}{\eta(1+\epsilon)} f(OPT).$$

The rest of our proof is to show that the set $S_{\max}$ is obtained from the lower bound is a ϵ-approximate local maximum solution. We follow the ideas presented by Iyer and Bilmes (Iyer and Bilmes, 2012b), who consider a general DS-decomposition minimization by an iterative modular approximation algorithm. For one subcase, we need to show under the terminate iteration t, if we add an element j, our local solution S^t will not increase an enough amount. By the construction of bounds, and optimality condition, we have

$$\begin{aligned}
(1+\epsilon) f(S^t) &\geq f(S^{t+1}) = f_1(S^{t+1}) - f_2(S^{t+1}) \\
&\geq L^{f_1}_{S^t,\sigma^t}(S^{t+1}) - U^{f_2}_{S^t}(S^{t+1}) \\
&\geq L^{f_1}_{S^t,\sigma^t}(S^t \cup \{j\}) - U^{f_2}_{S^t}(S^t \cup \{j\}) \\
&= f_1(S^t \cup \{j\}) - f_2(S^t \cup \{j\}) \\
&= f(S^t \cup \{j\}).
\end{aligned}$$

Similarly, we can lower bound $(1+\epsilon) f(S^t) \geq f(S^t \setminus \{j\})$ under the subcase that of deleting a element j. By the definition of ϵ-approximate local maximum, we know S^t is the ϵ-approximate local maximum solution. □

Theorem 22. *The Algorithm 1 terminates at most $O(1/\epsilon \log(OPT/f(S^0)))$ steps and the total time complexity is bounded by $O(C/\epsilon \log(OPT/f(S^0)))$, where C is the upper bound of time of computing optimal solution of modular function.*

Proof. Follows from the repeat process of Algorithm 1, we have $f(S^{i+1}) \geq (1+\epsilon)f(S^i)$ for any iteration $i(< t)$. It is easy to check out that the number of steps of the repeat process is at most $\log_{1+\epsilon} \frac{f(S^t)}{f(S^0)} (\leq O(1/\epsilon \log(OPT/f(S^0))))$. Assume the time of computing the optimal solution of modular function is at most C, with a multiplicative factor C, we can bound the total complexity of Algorithm 1. □

The above claims also hold for the Iterative Submodular Sandwich algorithm by losing a constant factor in approximation. The details are presented by the following theorem.

Theorem 23. *For any given $\epsilon > 0$, and set $S_{\max}$ as the returned set by Algorithm 5, then we have $S_{\max}$ either is a $(1 + (1/e - \epsilon)/(1 - 1/e))$-approximate local maximum solution by checking $O(n)$ permutations, or is $\frac{1-1/e}{\eta(1+\epsilon)}$-approximation solution for the interaction-aware influence maximization problem.*

Proof. For any given iteration t, the Iterative Submodular Sandwich algorithm performs a solution set S^t by greedy, then there is $(1 - 1/e)$ factor loss in approximation. As the proofs of Theorem 21, we easily conclude the two cases and obtain the according to lower bounds present by the theorem. □

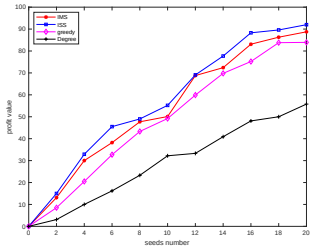
Theorem 24. *The total time complexity of Algorithm 2 is bounded by $O(D/\epsilon \log(OPT/f(S^0)))$, where D is the upper bound of time of computing greedy solution of submodular function.*

Proof. Similarly, assume D is an upper bound of time of computing greedy solution of submodular function, we can conclude that the total time complexity of Algorithm 2 is at most $O(D/\epsilon \log(OPT/f(S^0)))$. □

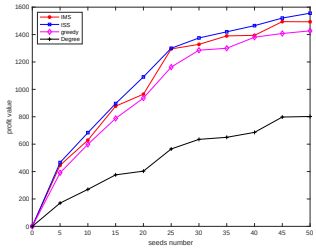
5.6 Experiment

5.6.1 Settings

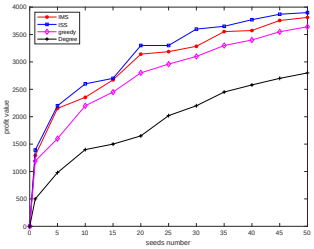
We use four social networks in our experiments. All datasets are publicly available. Email, DBLP can be obtained from SNAP website (htpp://snap.stanford.edu), while Facebook



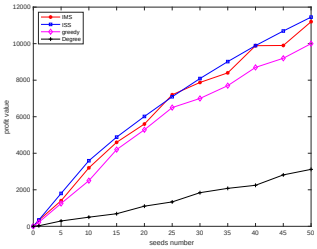
(a) Email



(a) Facebook



(a) Douban



(a) DBLP

Figure 5.3. The relationship between profit and seed set size

Table 5.1. The statistics of the data sets

Network	Vertices	Edges
Email	1005	25571
Facebook	63731	817035
Douban	154908	327162
DBLP	317080	1049866

and Douban can be obtained from KONECT website (http://konect.uni-koblenz.de). The details of datasets are shown in Table 5.1. The propagation probability for IC model is set to $\frac{1}{degree(v)}$ as widely used in other literature(Tang et al., 2014b; Chen et al., 2010b), and the profit between nodes is proportional to propagation probability on corresponding edges. The intuition is that there may be more interactions between u and v if u is more likely to activate v. To balance the two parts of profits, we use two settings. In the first case we uniformly set $\alpha = \beta = 1$. In the second case, we set $\alpha = 1$, $\beta = 1.3$. For comparison, we use greedy algorithm and HighDegree algorithm (Tang et al., 2017) as baselines.

5.6.2 Effectiveness

The results of profit computed by our proposed algorithms on four data sets are shown in Fig.6.2 respectively. As the number of selected seeds increases, the performance of iterated sandwich algorithms denoted as IMS(Iterated Modular Sandwich Algorithm) and ISS(Iterated Submodular Sandwich Algorithm) are always superior to the baseline algorithms. The Degree algorithm's performance is the worst since it does not consider the network structure. The reason that IMS/ISS performs better than greedy is our proposed iterated sandwich algorithm always choose the best solution from the solutions of upper bound function, lower bound function and the objective function, this selection guarantees the solution is at least as good as the greedy solution of the objective function. Apparently, the iterated sandwich algorithm is also superior to a sandwich algorithm proposed in (Wang et al., 2017b), since the returned solution is the best out of all the solutions calculated by sandwich algorithm in many iterations. They perform quite well on both small-scale and large-scale networks and demonstrate good scalability.

CHAPTER 6

ROBUST PROFIT MAXIMIZATION WITH DOUBLE SANDWICH ALGORITHMS IN SOCIAL NETWORKS[1]

Authors – Chuangen Gao, Shuyang Gu, Ruiqi Yang,

Hongwei Du, Smita Ghosh, and Hua Wang

The Computer Science Department, EC 31

The University of Texas at Dallas

800 West Campbell Road

Richardson, Texas 75080-3021

[1]©2019 IEEE. Reprinted, with permission, from Chuangen Gao, Shuyang Gu,Ruiqi Yang, Hongwei Du, Smita Ghosh, and Hua Wang, Robust Profit Maximization with Double Sandwich Algorithms in Social Networks, 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 2019, pp. 1539-1548, July 2019

6.1 Introduction

With the proliferation of online social networks, considerable research on viral marketing has been done, which shows that the "word-of-mouth" effect plays a very important role in spreading innovations and ideas in social networks. The influence maximization problem with applications in viral marketing is one of the most important topics. In this problem, a few individuals are provided with free products, hoping that the product will be recursively recommended by each individual to his friends to create a large cascade of further adoptions. Kempe et al. (Kempe et al., 2003b) first formulate it as a discrete optimization problem on independent cascade (IC) model and linear threshold (LT) model. After that, a large body of works has been performed on viral marketing to maximize the benefits associated with the number of active nodes.

Although the existing works on influence maximization have obtained plentiful results in many ways, these works are based on the assumption that the number of influenced users determines the profit of a product. However, the number of influenced users may not reflect the success of a promotion campaign, some types of products' profit rely on the interactions among the influenced users. The revenue model of online games is an example, the game revenue depends on the participation and interaction of players.

The interactions among players contribute to game profit in several ways. First, the interactive users play games in an online manner, which will attract more in-game advertising. In-game advertising allows advertisers to pay to have their name or products featured in games (Willis et al., 2014), in 2017, $109 billion was spent on in-game advertising. Second, the virtual goods transactions in games depend on players' interactions (Lehdonvirta, 2009). In 2009, games played on social networks such as Facebook primarily derive revenue from the sale of virtual goods, which brought in $1 billion. Therefore, in this paper we aim to maximize game profit which depends on the interactions among influenced users in a broad sense.

The uncertainty of the influence probabilities on edges is another issue we want to tackle in this paper. Due to inherent data limitation, the exact values of the edge probabilities could not be obtained through learning method, what we can get is a probability range on each edge, in other words, it might be an interval in which the true probability lies in for each edge in the social network graph (He and Kempe, 2015). This motivates us to study the problem of finding a seed set that maximizes the profit under the worst-case propagation probability condition. Specifically, we propose the *Robust Profit Maximization Problem* (RPMP), which maximizes the worst-case ratio between the total profit of the chosen seed set and the profit with optimal seed set, given the uncertainty of the propagation probability.

The main idea of our double sandwich algorithm to solve the robust profit maximization problem is as follows, it first solves the profit maximization problem on the minimum and maximum parameter vectors respectively, then we choose one solution that has higher total profit on the minimum parameter vector as the output seed set. The profit maximization problem on the minimum/maximum parameter vector is non-submodular, to solve these two non-submodular problems, our sandwich algorithm finds submodular lower and upper bounds of the objective function and solves them with greedy algorithm, then choose the solution with maximal total profit. We show that the double sandwich algorithm has a solution-dependent bound on its performance, which means the robust ratio is bounded by a selected seed set. In order to further improve the robustness of our algorithm, we study sampling methods to tighten the parameter space of propagation probability. We provide theoretical results on the uniform sample complexity for achieving a given robust ratio of the output seed set. Then we extend the double sandwich algorithm with the uniform sampling method to solve RPMP, the experimental results show that it significantly improves the robustness of the profit maximization problem.

The contributions of this paper are summarized as follows.

- We introduce the profit maximization problem (PMP), analyze its complexity and prove it is non-submudular, and further define a robust profit maximization problem (RPMP) given the propagation probabilities along social relations are uncertain and prove it is NP-hard.
- We find the upper and lower bounds of PMP and prove both of them are submolular. Based on the two submodular bounds we design the sandwich algorithm with an accelerated greedy method.
- For our major problem RPMP, we design a double sandwich algorithm using the sandwich algorithm for PMP as its subroutine and get a data-dependent approximate solution.
- To further improve the robust ratio of RPMP, we study the uniform sampling method and combine it into our double sandwich algorithm and give its theoretical guarantee.
- Through real data sets, we verify the effectiveness of our proposed algorithms.

The rest of the paper is organized as follows. Sec. 6.2 is devoted to the related work. The robust profit maximization problem and profit maximization problem are proposed in Sec. 6.3. The strategy for profit maximization problem is provided in Sec. 6.4. The strategy for robust profit maximization problem is presented in Sec. 6.5, where we propose the double sandwich algorithm and improve it with uniform sampling. The experiments are presented in Sec. 6.6.

6.2 Related work

Kempe et al. (Kempe et al., 2003b) formulate the influence maximization problem under IC and LT models and provide a greedy algorithm with an approximation ratio. Since then, considerable work (Chen et al., 2010, 2016b; Lu et al., 2015b; Tang et al., 2014b, 2015;

Nguyen et al., 2016,?; Tang et al., 2017, 2014b; Tong et al., 2017) has been devoted to extending existing models to study influence maximization and its variants. Recently, Wang et al. (Wang et al., 2017b) study the activity maximization problem to maximize the sum of activity strengths among the influenced users. Different from their work, we study the profit maximization problem, in which the profit is associated with the activity between any two influenced users in social networks. The motivation for this problem is the activity between two users without social relation may also contribute to the overall profit.

The robustness in the influence maximization problem has come into notice in recent research. In (Jung et al., 2012), an algorithm that integrates influence ranking (IR) and influence estimation (IE) methods for influence maximization in independent cascade (IC) model is proposed, which is demonstrated robust on different structural properties. The issue of uncertainty of propagation probability on social relations was addressed by Chen *et al.* (Chen et al., 2016a) and He *et al.* (He and Kempe, 2015) in the influence maximization problem. In (He and Kempe, 2015), an influence difference maximization problem is proposed to maximize the additive difference between two influence spreads of the same seed set using different parameter values. The problem proposed in (Chen et al., 2016a) aims to find a seed set that maximizes the worst-case ratio between the objective function of the chosen seed set and the objective function of the optimal solution, given the uncertainty of the propagation probability. The main idea is similar to our paper.

Nonetheless, our work is different from it in two aspects. First, the robust ratio is defined differently. Their robust influence maximization problem is to find the best possible seed set for influence maximization purpose while considering the adverse effect of the uncertainty. Our work is to find the best seed set that maximizes the overall profit among influenced users. In a word, the maximization objective function is influence spread in their work, while our maximization objective is the total benefit related to user-to-user interaction. Second, influence spread, the maximization objective function in their work is monotone

submodular, while our total profit function is neither submodular nor supermodular. To the best of our knowledge, this paper is the first study on robust nonsubmodular optimization problem in social networks.

6.3 Problem Formulation

In this section, we first define the robust profit maximization problem in IC model formally to address the uncertainty of the propagation probability of the edge. As a strategy, we then formulate profit maximization problem in social networks and prove it is neither submodular nor supermodular by counter examples.

6.3.1 Robust Profit Maximization Problem

In IC model, every initial user who is selected as the seed starts to propagate the influence to her neighbors according to propagation probability on the edge. The process consists of discrete steps. In each step, each node which was newly activated at the last step would try to influence its out-neighbors. An active node has only one chance to influence its out-neighbors. The process ends when no node is activated in the current step.

In this paper, we use the directed graph $G = (V, E)$ to represent a social network, where V is the set of users and E is the set of social relations between users. Each edge $(u, v) \in E$ is assigned with a probability p_{uv} so that when u is active, v is activated by u with probability p_{uv}. And the benefit related to the interaction between nodes is represented by a nonnegative function $b : V \times V \to \mathbb{R}_{\geq 0}$, in which $b(u, v) = b(v, u)$ for the unordered pair $\{u, v\}$ of node u and v. Note that for each $\{u, v\}$, we only compute once the benefit between them, i.e., $b(u, v)$ or $b(v, u)$ instead of $b(u, v) + b(v, u)$.

Since the randomness of the propagation process in IC model, consider a point in the cascade process when node u has just become active, and it attempts to activate its neighbor v, succeeding with probability p_{uv}. We can view the outcome of this random event as being

determined by flipping a coin of bias p_{uv}. With all the coins flipped in advance, the edges in G for which the coin flip indicated a successful activation are declared to be live; the remaining edges are declared to be blocked(Kempe et al., 2003b). We use g to represent the outcome of this process which is called a live graph of G since it consists of all edges declared to be live. We denote as $g \sim D$, where D is the distribution of g. For any seed set S, denote by $I_g(S)$ the set of all active nodes at the end of the cascade process in live graph g. Its cardinality is represented by $|I_g(S)|$.

The total expected benefit would be defined as

$$\begin{aligned} f(S) &= \mathbb{E}_{g\sim D}[\sum_{\{u,v\}\subseteq I_g(S)} b(u,v)] \\ &= \sum_g Prob[g] \cdot \sum_{\{u,v\}\subseteq I_g(S)} b(u,v) \end{aligned} \tag{6.1}$$

The benefit $\sum_{\{u,v\}\subseteq I_g(S)} b(u,v)$ is related to the strength of interaction between the active nodes. The $\{u,v\} \subseteq I(S)$ denotes the all unordered pair in the set $I(S)$. Note that for each unordered pair $\{u,v\}$, since $b(u,v) = b(v,u)$, we only compute once the benefit between them. The expectation is respected to g.

Due to inherent data limitation, the exact values of the edge probabilities could not be obtained through the learning method, what we can get is a probability range on each edge. Suppose for every edge e, we are given an interval $[l_e, r_e]$ $(0 \leq l_e \leq r_e \leq 1)$ indicating the range of the propagation probability, and the exact probability $p_e \in [l_e, r_e]$ of this edge is unknown. Denote $\Theta = \times_{e\in E}[l_e, r_e]$ as the parameter space of network G, and $\theta = (p_e)_{e\in E}$ as the latent parameter vector. Specifically, let $\theta^-(\Theta) = (l_e)_{e\in E}$ and $\theta^+(\Theta) = (r_e)_{e\in E}$ as the minimum and maximum parameter vectors, respectively, and when the context is clear, we would only use θ^- and θ^+. For a seed set $S \subset V$ and $|S| = k$, define its robust ratio under parameter space Θ as

$$g(\Theta, S) = \min_{\theta\in\Theta} \frac{f_\theta(S)}{f_\theta(S^*_\theta)} \tag{6.2}$$

Given Θ and solution S, the robust ratio $g(\Theta, S)$ characterizes the *worst-case* ratio of game profit of S and the underlying optimal one, when the true probability vector θ is unknown (except knowing that $\theta \in \Theta$). Then, the Robust Profit Maximization Problem is defined as follows.

Definition 13 (Robust Profit Maximization Problem, RPMP)**.** *Given a social network $G = (V, E)$, a propagation probability interval $[l_e, r_e]$ for each edge $e \in E$ under the IC model, a benefit function $b : V \times V \to \mathbb{R}_{\geq 0}$, and a positive integer k, find a set S of k seeds to maximize the robust ratio:*

$$S^*_\Theta = \underset{S \subset V, |S|=k}{\operatorname{argmax}} g(\Theta, S) = \underset{S \subset V, |S|=k}{\operatorname{argmax}} \min_{\theta \in \Theta} \frac{f_\theta(S)}{f_\theta(S^*_\theta)} \tag{6.3}$$

6.3.2 Profit Maximization Problem

To process the robust profit maximization problem, we first introduce the following problem.

Definition 14 (Profit Maximization Problem, PMP)**.** *Given a social network $G = (V, E)$, a propagation probability p_{uv} for each edge (u, v) under the IC model, a benefit function $b : V \times V \to \mathbb{R}_{\geq 0}$, and and a positive integer k, find a set S of k seeds to maximize the expected profit through influence propagation:*

$$\max f(S) \tag{6.4}$$

$$s.t. |S| \leq k \tag{6.5}$$

We say that $g(\cdot)$ is submodular if it satisfies a natural "diminishing returns" property: the marginal gain from adding an element to a set X is at least as high as the marginal gain from adding the same element to a superset of X. Formally, for every set X, Y such that $X \subseteq Y \subseteq V$ and every $e \in V \setminus Y$, it follows that

$$g(X \cup \{e\}) - g(X) \geq g(Y \cup \{e\}) - g(Y)$$

And it is monotone if $g(X) \leq g(Y)$ whenever $X \subseteq Y$.

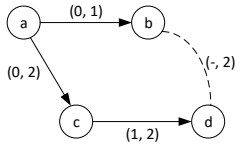


Figure 6.1. Counter example

Theorem 25. *$f(S)$ is neither submodular nor supermodular under IC model.*

Proof. We prove by the counter example shown in Fig.6.1. The first element in the tuple tied on each edge represents the propogation probability, and the second one denotes the benefit between its two end nodes. For pairs $\{u, v\}$ between which there is no edge set $b(u, v) = 0$ except pair $\{b, d\}$. In Fig.6.1, $(0, 1)$ on edge (a, b) means propogation probability $p_{ab} = 0$ and $b(a, b) = 1$, then we have $f(\{a\}) = 0$, $f(\{a, b\}) = 1$, $f(\{a, d\}) = 0$ and $f(\{a, b, d\}) = 3$. Thus, $f(\{a, d\}) - f(\{a\}) < f(\{a, b, d\}) - f(\{a, b\})$, which implies $f(S)$ is not submodular. Also, we have $f(\{b, d\}) = 2$, $f(\{b\}) = 0$, $f(\{b, c, d\}) = 4$, $f(\{b, c\}) = 4$ Thus, $f(\{b, d\}) - f(\{b\}) > f(\{b, c, d\}) - f(\{b, c\})$ which implies $f(S)$ is not supermodular. □

Theorem 26. *Profit maximization problem is NP-hard.*

Proof. Now we prove by reducing from the set cover problem, which is NP-complete (Alon et al., 2003b). Given a ground set $U = \{u_1, u_2, \ldots, u_n\}$ and a collection of sets $\{S_1, S_2, \ldots, S_m\}$ whose union equals the ground set, the set cover problem is to decide if there exist k sets in S so that the union equals U. Given an instance of the set cover problem, we construct a corresponding graph with $m + 2n$ nodes as follows. For each set S_i we create one node p_i, and for each element u_j we create two nodes q_j and q'_j. If the S_i contains the element u_j, then we create two edges (p_i, q_j) and (p_i, q'_j). Note that each edge is live which means the probability is 1. Now we design the benefit function over pairs of nodes. For the pairs $\{q_j, q'_j\}$, the benefit equals to 1, and the other pairs equal to 0. Then the set cover problem is equivalent to deciding if there is a set S of k nodes such that the benefit of S equals to n. The theorem follows immediately. □

Note that the activity maximization problem proposed by Wang *et al.* (Wang et al., 2017b) is similar to our problem. But they only consider the activity between active nodes which are connected by edges, we consider the benefits among all active nodes, regardless of whether there is an edge connecting them. For example, in online games, regardless of whether they are friends in the social society, that is, they are connected in the social network by edges, they can play games together and generate corresponding profit for the game company, which means it generates profit whether they are friends or not. To this end, our profit maximization problem can be viewed as a significant extension of the activity maximization problem.

6.4 Strategy for Profit Maximization Problem

Since the profit maximization problem is not submodular, the greedy algorithm cannot be directly applied to it to get a guaranteed approximate solution. To solve this non-submodular problem, we use the sandwich algorithm that gains a data-dependent solution. The sandwich algorithm is mainly based on submodular upper bound and submodular lower bound of the original problem. In this section, we propose an upper and a lower bound of the profit maximization problem PMP, and we prove that both of them are submodular. Thus, the greedy algorithm can achieve a $1-1/e$ approximate solution for the bounds. Our idea for the non-submodular problem mainly follows the paper(Wang et al., 2017b), but we make some improvements and propose the sandwich strategy based on an accelerated greedy algorithm.

6.4.1 Upper bound

Given the seed set S, we define the upper bound $U(S)$ of $f(S)$ as the profit between nodes both of which are active plus half of the profit between nodes which are active and nodes which are not active. The former is exactly the profit we want to compute. And it can be defined as

$$
\begin{aligned}
& U(S) \\
=& \mathbb{E}_{g\sim D}[\sum_{\{u,v\}\subseteq I_g(S)} b(u,v) + \frac{1}{2}\sum_{u\in I_g(S), v\in V\setminus I_g(S)} b(u,v)] \\
=& \mathbb{E}_{g\sim D}[\sum_{v\in I_g(S)} w(v)] \\
=& \sum_g Prob[g]\cdot \sum_{v\in I_g(S)} w(v) \qquad (6.6)
\end{aligned}
$$

where $I(S)$ denotes the set of all active nodes, when S is the seed set and

$$w(v) = \frac{1}{2}\sum_{u\in V} b(v,u) \qquad (6.7)$$

Thus, we can see that the upper bound is essentially a weighted version of influence spread, where the weight of node v is $\frac{1}{2}\sum_{u\in V} b(v,u)$, which equals to half of the sum of profit between v and the other nodes in V.

Theorem 27. *$U(S)$ is monotone and submodular.*

Proof. We just need to prove $U(S)$ is monotone and submodular in a live graph, since an non-negative linear combination of submodular functions is also submodular. Since the profit function $b : V \times V \to \mathbb{R}_{\geq 0}$ is nonnegative which means the profit of each pair of nodes is non-negative. Thus the weight of every node is non-negative and the monotonicity of $U_f(S)$ follows immediately. For the submodularity, we need to prove $U(A\cup\{v\}) - U(A) \geq U(B\cup\{v\}) - U(B)$, such that $A \subseteq B \subseteq V$ and $v \in V \setminus B$. The left side of inequality is the weight of nodes which can be activated by v but can not by A in live graph g. The right side is the weight of nodes which can be activated by v but can not by B. We have $I_g(v) - I_g(A) \supseteq I_g(v) - I_g(B)$, since $A \subseteq B$. And the submodularity follows immediately. □

6.4.2 Lower bound

For the lower bound $L(S)$ of $f(S)$, the major idea is that we only consider the profit between nodes which are activated by the same seed node. Accordingly, the lower bound can be defined as

$$\begin{aligned} & L(S) \\ =& \mathbb{E}_{g\sim D}[\sum_{x\in S}\sum_{\{u,v\}\subseteq I_g(\{x\})} b(u,v)] \\ =& \sum_{g} Prob[g]\cdot\sum_{x\in S}\sum_{\{u,v\}\subseteq I_g(\{x\})} b(u,v) \end{aligned} \tag{6.8}$$

where $I(\{x\})$ are nodes activated by node x. It is easy to see that $L(S) \leq f(S)$ for any $S \subseteq V$ since we ignore the profit of pairs of nodes which are activated by different seeds.

Theorem 28. *$L(S)$ is monotone and submodular.*

Proof. We just need to prove $L(S)$ is monotone and submodular in a live graph, since a non-negative linear combination of submodular functions is also submodular. As the profit function $b : V \times V \to \mathbb{R}_{\geq 0}$ is nonnegative which means the profit of each pair of nodes is non-negative, the monotonicity of $L(S)$ follows immediately. For the submodularity, we need prove $L(A \cup \{e\}) - L(A) \geq L(B \cup \{e\}) - L(B)$, such that $A \subseteq B \subseteq V$ and $e \in V \setminus B$. We have $I_g(e) - \cup_{x\in A} I_g(x) \supseteq I_g(e) - \cup_{x\in B} I_g(x)$, since $A \subseteq B$. Thus the submodularity follows immediately. □

6.4.3 Sandwich Strategy

In this section, we design a sandwich strategy based on the submodular upper and lower bounds of PMP which can get a data-dependent approximation solution. To address the upper and lower bounds we propose an accelerated greedy algorithm with lazy evaluation according to submodularity of them.

Algorithm 6 Accelerated Greedy Algorithm, AG

Input: Seeds number k; $G(V, E, P)$, profit function f.
initialize $S \leftarrow \emptyset$;
for $v \in V$ **do**
  $Q_v \leftarrow +\infty$;
end for
for $i = 1$ to k **do**
  $\delta_{max} \leftarrow -\infty$; $v_{max} \leftarrow Null$;
  while $\delta_{max} < Q.max()$ **do**
    $v \leftarrow Max(Q)$; $\delta \leftarrow \triangle(v|S)$; $Q_v = \delta$;
    if $\delta > \delta_{max}$ **then**
      $\delta_{max} \leftarrow \delta$; $v_{max} \leftarrow v$;
    end if
  end while
  $S \leftarrow S \cup v_{max}$;
  Delete v_{max} in Q;
end for
return S;

Lazy Evaluation

As we know, when greedy algorithm selects a seed in the round i, it computes $\Delta(v|S_i)$ for all $v \in V$ and $0 \leq i \leq k$, unless $v \in S$, where S_i is the selected seeds after round i and k is constraint on number of seeds. The key insight is that $i \mapsto \Delta(v|S_i)$ is nonincreasing for all $v \in V$, because of the submodularity of the objective. Specifically speaking, in the round i of selceting a seed, if we know $\Delta(v'|S_j) \leq \Delta(v|S_i)$ for some items v' and v and $j \leq i$, then $\Delta(v'|S_i) \leq \Delta(v|S_i)$ follows, since $\Delta(v'|S_j) \geq \Delta(v'|S_i)$ based on the submodularity. So there is no need to compute $\Delta(v'|S_i)$ in the round i.

Accelerated Greedy Algorithm

The pseudocode of the accelerated greedy algorithm with lazy evaluation is given in Algorithm 1, AG for short. Its core idea is to choose the seed with the largest expected marginal benefit at each step. S is used to store the seeds selected. The expected marginal gain of each node is stored in the list Q. According to the strategy of lazy evaluation, the while loop

Algorithm 7 Sandwich Algorithm, SA

1: $S_U \leftarrow$ a greedy solution to upper bound $U(\cdot)$ by calling AG;
2: $S_L \leftarrow$ a greedy solution to lower bound $L(\cdot)$ by calling AG;
3: $S_f \leftarrow$ a greedy solution to the original problem $f(\cdot)$;
4: $S^{SA} = argmax_{S \in \{S_U, S_L, S_f\}} f(S)$;
5: **return** S^{SA};

computes expected marginal benefit $\Delta(v|S)$ for node v in decreasing order of upper bounds known on them, until it finds an item whose value is at least as great as the upper bounds of all other nodes. Note the $Q.max()$ returns max value in the marginal gain list Q, and $Max(Q)$ returns the node which has the maximum marginal gain. When a seed with the greatest marginal benefit is found,we will delete node v_{max} as shown in line 15 since their marginal gain will be 0 in the next round. The greedy policy will return the seeds.

Sandwich Algorithm

By adopting the sandwich strategy which is first proposed by (Lu et al., 2015b), and using the proving technique introduced by (Wang et al., 2017b), we design an sandwich approximation policy based on the submodular upper and lower bound for profit maximization problem, which is given in Algorithm 2. The basic idea of the sandwich strategy is that we find greedy solutions to the upper bound, the lower bound, and original function respectively. Then we choose the best greedy strategy for the original problem as the solution. Specifically, the sandwich approximation strategy works as follows. First, for the upper bound and lower bound we find a greedy solution by calling Algorithm 1 respectively as shown in line 1 and 2. Since both the upper and lower bound are submodular, the accelerated greedy algorithm could obtain an approximation solution with $1 - \frac{1}{e}$. Note that for original problem $f(S)$ the accelerated greedy algorithm cannot be used to solve it and get an approximation solution since it is not submodular. Actually any useful algorithm can be used to solve it, we resort to the normal greedy algorithm instead of accelerated greedy algorithm in our sandwich

algorithm. Then, we choose the best solution among the three solutions for the original problem as the solution of the sandwich algorithm as shown in line 4.

Approximation Ratio of Sandwich Algorithm

Theorem 29. *Let S^{SA} be the solution returned by Algorithm 2, then we have*

$$f(S^{SA}) \geq \max\left\{\frac{f(S_U)}{U(S_U)}, \frac{L(S_L^*)}{f(S^*)}\right\} \cdot (1 - \frac{1}{e}) \cdot f(S^*)$$

where S_f^*, S_U^*, S_L^* is the optimal policy for $f(\cdot)$, $U_f(\cdot)$, $L_f(\cdot)$ respectively.

Proof. We have

$$\begin{aligned}
&f(S_U) \\
&= \frac{f(S_U)}{U(S_U)} U(S_U) \geq \frac{f(S_U)}{U(S_U)} \cdot (1 - \frac{1}{e}) \cdot U(S_U^*) \\
&\geq \frac{f(S_U)}{U(S_U)} \cdot (1 - \frac{1}{e}) \cdot U(S^*) \geq \frac{f(S_U)}{U(S_U)} (1 - \frac{1}{e}) \cdot f(S^*)
\end{aligned}$$

$$\begin{aligned}
&f(S_L) \\
&\geq L(S_L) \geq (1 - \frac{1}{e}) \cdot L(S_L^*) \geq \frac{L(S_L^*)}{f(S^*)} \cdot (1 - \frac{1}{e}) \cdot f(S^*)
\end{aligned}$$

Let $S^{SA} = \text{argmax}_{S \in \{S_f, S_U, S_L\}} f(S)$, then

$$f(S_f^{SA}) \geq \max\left\{\frac{f(S_U)}{U(S_U)}, \frac{L(S_L^*)}{f(S^*)}\right\} (1 - \frac{1}{e}) f(S^*).$$

□

Implementation Issues

To implement the proposed sandwich algorithm, the key step is to calculate the expected marginal gain $\triangle(v|S)$ for the upper bound, lower bound, and original function. Unfortunately, all of them are ♯P-hard. Due to space constraints, we could not show proof. Although

they are difficult to calculate, we can employ the Monte Carlo simulation to obtain an accurate estimation. By the Hoeffding's Inequality, the error of the estimation can be infinitely small when a sufficient number of simulations are performed. Or we can use the technique of reverse set (Wang et al., 2017b) which will not be repeated here.

6.5 Strategy for Robust profit maximization problem

We deal with Robust Profit Maximization Problem in two situations. The first case is to solve this problem in the given parameter space Θ. The double sandwich algorithm is designed and its main idea is to apply the sandwich algorithm to solve the profit maximization problem under the parameter settings θ^- and θ^+ respectively. So we call it the double sandwich algorithm. In the second case, we shorten the width of the confidence interval by sampling, within which the true propagation probability of each edge falls. We combine the sampling technology of the new parameter space with the double sandwich and design a double sandwich algorithm with sampling.

We first prove the complexity of RPMP as follows.

6.5.1 Complexity

Consider the problem of RPMP, parameter space $\Theta = \times_{e\in E}[l_e, r_e]$ is given, and we don't know ture propagation probability of each edge, all we know is $\theta \in \Theta$.

Theorem 30. *Robust profit maximization problem is NP-hard.*

Proof. When Θ is a single vector which means $l_e = r_e, \forall e \in E$, the Robust Profit Maximization Problem is trivially reduced to the Profit Maximization Problem, which is NP-hard according to Theorem 26. Therefore, the theorem follows immediately. □

Algorithm 8 Double Sandwich Algorithm, DS

1: **Input**: Seeds number k; $G(V, E)$, profit function f, parameter space $\Theta = \times_{e\in E}[l_e, r_e]$
2: $S^{SA}_{\theta^-} \leftarrow$ a sandwich solution to f_{θ^-} by calling SA;
3: $S^{SA}_{\theta^+} \leftarrow$ a sandwich solution to f_{θ^+} by calling SA;
4: $S^{DS}_{\Theta} \leftarrow \arg\max_{S\in\{S^{SA}_{\theta^-}, S^{SA}_{\theta^+}\}} f_{\theta^-}(S)$;
5: **return** S^{DS}_{Θ};

6.5.2 Double Sandwich Algorithm

For the first case that we are not allowed to make new samples on the edges to improve the input interval, we utilize the sandwich algorithm as the subroutine to solve profit maximization problem shown and propose the Double Sandwich algorithm that achieves reasonably large robust ratio.

Given parameter space $\Theta = \times_{e\in E}[l_e, r_e]$ with the minimum and maximum parameter vectors $\theta^- = (l_e)_{e\in E}$ and $\theta^+ = (r_e)_{e\in E}$, our Double Sandwich Algorithm is described in Algorithm 8. First, we solve the Profit Maximization Problem under the setting of the minimum parameter vector $\theta^- = (l_e)_{e\in E}$, which is non-submodular as we demonstrated in the Theorem 25. We find its upper and lower bounds both of which are submodular as described in section 6.4, then we call sandwich algorithm to get a solution$S^{SA}_{\theta^-}$ for $f_{\theta^-}(\cdot)$ as shown in line 2. By the same way, under the setting of maximum parameter vector $\theta^+ = (r_e)_{e\in E}$, we get a solution $S^{SA}_{\theta^+}$ for $f_{\theta^+}(\cdot)$ by call a sandwich algorithm as shown in line 3. Then, we output the best seed set S^{DS}_{Θ} for the minimum parameter vector θ^- such that

$$S^{DS}_{\Theta} = \arg\max_{S\in\{S^{SA}_{\theta^-}, S^{SA}_{\theta^+}\}} f_{\theta^-}(S) \tag{6.9}$$

To evaluate the performance of this output, we first define the gap ratio $\alpha(\Theta) \in [0, 1]$ of the input parameter space to be

$$\alpha(\Theta) := \frac{f_{\theta^-}(S^{DS}_{\Theta})}{f_{\theta^+}(S^{SA}_{\theta^+})} \tag{6.10}$$

Then, the Double Sandwich Algorithm achieve the following result:

Theorem 31. *Given a graph $G(V, E)$, parameter space Θ and budget limit k, Double Sandwich algorithm gain a seed set S_Θ^{DS} of size k such that*

$$g(\Theta, S_\Theta^{DS}) \geq \alpha(\Theta) \cdot \beta(\theta^+) \cdot (1 - \frac{1}{e}),$$

where

$$\alpha(\Theta) := \frac{f_{\theta^-}(S_\Theta^{DS})}{f_{\theta^+}(S_{\theta^+}^{SA})}$$

$$\beta(\theta^+) = \max\left\{\frac{f_{\theta^+}(S_{U_{\theta^+}})}{U_{\theta^+}(S_{U_{\theta^+}})}, \frac{L_{\theta^+}(S^*_{L_{\theta^+}})}{f_{\theta^+}(S^*_{\theta^+})}\right\}.$$

Proof. For any seed set S, the robust ration $g(\Theta, S) = \min_{\theta\in\Theta} \frac{f_\theta(S)}{f_\theta(S^*_\theta)}$ by definition. Obviously, it is a fact that $f_\theta(S)$ is monotone on θ for any fixed seed set S. From the definition of optimal solutions and the sandwich algorithm, we can get $f_\theta(S^*_\theta) \leq f_{\theta^+}(S^*_\theta) \leq f_{\theta^+}(S^*_{\theta^+})$. By the approximaiton ratio of sandwich algorithm shown in Therom 29, we have

$$\begin{aligned} f_{\theta^+}(S^*_{\theta^+}) &\leq \frac{f_{\theta^+}(S^{SA}_{\theta^+})}{\max\left\{\frac{f_{\theta^+}(S_{U_{\theta^+}})}{U_{\theta^+}(S_{U_{\theta^+}})}, \frac{L_{\theta^+}(S^*_{L_{\theta^+}})}{f_{\theta^+}(S^*_{\theta^+})}\right\} \cdot (1 - \frac{1}{e})} \\ &\leq \frac{f_{\theta^+}(S^{SA}_{\theta^+})}{\beta(\theta^+) \cdot (1 - \frac{1}{e})} \end{aligned}$$

Moreover, it can be implied that

$$\begin{aligned} g(\Theta, S) &= \min_{\theta\in\Theta} \frac{f_\theta(S)}{f_\theta(S^*_\theta)} \\ &\geq \min_{\theta\in\Theta} \frac{f_\theta(S)}{f_{\theta^+}(S^*_{\theta^+})} \\ &\geq \min_{\theta\in\Theta} \frac{f_\theta(S)}{f_{\theta^+}(S^{SA}_{\theta^+})} \cdot \beta(\theta^+) \cdot (1 - \frac{1}{e}) \\ &\geq \frac{f_{\theta^-}(S)}{f_{\theta^+}(S^{SA}_{\theta^+})} \cdot \beta(\theta^+) \cdot (1 - \frac{1}{e}) \end{aligned}$$

Use seed set S_{Θ}^{DS} gained from Double Sandwich Algorithm, and it follows immediately that

$$\begin{aligned} g(\Theta, S_{\Theta}^{DS}) &\geq \frac{f_{\theta^-}(S_{\Theta}^{DS})}{f_{\theta^+}(S_{\theta^+}^{SA})} \cdot \beta(\theta^+) \cdot (1 - \frac{1}{e}) \\ &\geq \alpha(\Theta) \cdot \beta(\theta^+)(1 - \frac{1}{e}) \end{aligned}$$

□

Note that we have obtained a bound that depends on the solution of double sandwich algorithm, named data-dependent bound. Specifically, the first parameter $\alpha(\Theta)$ is related to the solution of double sandwich algorithm, the second parameter β is only related to the solution of sandwich algorithm for the profit maximizaiton problem under the θ^+. Actually, $\beta(\theta^+)(1 - \frac{1}{e})$ is the approximation ratio of the sandwich algorithm for the PMP f_{θ^+} when the parameter vector is θ^+.

6.5.3 Double Sandwich Algorithm with Uniform Sampling

The worst ratio is affected by the confidence interval in which the true propagation probability of each edge falls. The larger interval is, the greater the uncertainty is, and the corresponding robust ratio may not be guaranteed by a satisfied bound. A natural question is how we can further improve this worst-case ratio by sampling techniques based on a given probability space? This is the second case we are going to study. Our idea is to sample the propagation probability of each edge to improve the accuracy of the estimate, since according to Chernoff's inequality, the more samples are taken, the shorter the confidence interval becomes within which true propagation probability falls. Thus we can shorten the width of the interval by sampling.

How to connect the width of confidence interval with the profit gained by influence spread is a crucial issue. We apply the properties of additive confidence interval to this issue, and incorporate into Double Sandwich Algorithm with theoretical justification. Based on this idea, we design a sampling-based double sandwich algorithm as shown in algorithm 9. First,

Algorithm 9 Double Sandwich With Uniform Sampling, UDS

1: **Input**: Seeds number k; $G(V, E)$, profit function b, (ϵ, γ)
2: **Output**: Parameter space Θ_{out}, seed set S_{out}
3: **for** $e \in E$ **do**
4: Sample e for t times, and observe $x_e^1, \cdots, x_e^t$
5: $p_e \leftarrow \frac{1}{t}\sum_{i=1}^{t} x_e^i$, and set δ_e according to Theorem 32
6: $r_e \leftarrow \min\{1, p_e + \delta_e\}$, $l_e \leftarrow \max\{0, p_e - \delta_e\}$
7: **end for**
8: $\Theta_{out} \leftarrow \times_{e\in E}[l_e, r_e]$
9: $S_{out} \leftarrow$ a double sandwich solution by calling DS with parameter space Θ_{out};
10: **return** (Θ_{out}, S_{out});

the algorithm samples for each edge and generates a new parameter space as shown in line 3-7, then takes this parameter space as input and calls the double sandwich algorithm shown in line 9. Our Double Sandwich Algorithm with Uniform Sampling has theoretical bound which is presented in Theorem 32. In sampling for improving RPMP, the goal is to design a sampling and maximization algorithm that outputs a parameter space Θ_{out} and S_{out} such that with high probability the robust ratio of S in Θ is large.

To prove our theorem, we first establish the relationship between the propagation probability interval of the edge and the profit of active nodes, as shown by the following lemma.

Lemma 12. *Given any graph $G = (V, E, \Theta, b)$, where $\Theta = \times_{e\in E}[\ell_e, r_e]$, b is the profit function and $B = \sum_{\{u,v\}\in V} b(u, v)$, $m = |V|$. Let S_Θ^{DS} be the returned set by Algorithm by additive sampling, i.e., there exists $\delta > 0$ such that $r_e - \ell_e \leq \delta$ for any $e \in E$, then the difference of profit under parameter θ^+ and θ^- is upper bounded by*

$$f_{\theta^+}(S_\Theta^{DS}) - f_{\theta^-}(S_\Theta^{DS}) \leq mB \cdot \|\theta^+ - \theta^-\|_\infty.$$

Proof. For any given seed subset $S \subseteq V$ and parameter space Θ, our utility function $f_\theta(S) = \sum_g \text{Prob}_\theta[g] \sum_{\{u,v\}\subseteq I_g(S)} b(u, v)$, where g is a random subgraph generated by setting each edge $e \in E$ with probability p_e. For simplicity we denote $g = (E(g), V(g))$ and $\delta = \|\theta^+ - \theta^-\|_\infty$. We firstly show a claim that for any random subgraph g and seed set S, we have $f_{\theta^+}(S) -$

$f_{\theta^-}(S) \leq \delta \cdot B$. Since for any fixed $e_0 \in E(g)$, we update θ^+ to θ^1 by setting

$$\theta^1 = \begin{cases} \theta_e^+ - \delta, & \text{if } e = e_0 \\ \theta_e^+, & \text{o.w.} \end{cases}.$$

Then the difference of $f_{\theta^+}(S)$ and $f_{\theta^1}(S)$ is upper bounded by

$$f_{\theta^+}(S) - f_{\theta^1}(S) \leq \delta \cdot B. \tag{6.11}$$

Repeat the above updates, until we obtain the first iteration t such that

$$\theta^t = \begin{cases} \theta_e^+ - \delta, & \text{if } e \in E(g) \\ \theta_e^+, & \text{o.w.} \end{cases}.$$

Inspired by inequality 6.11, we have

$$\begin{aligned} f_{\theta^+}(S) - f_{\theta^-}(S) &\leq f_{\theta^+}(S) - f_{\theta^t}(S) \\ &\leq t\delta B \\ &\leq m\delta B, \end{aligned}$$

where the first inequality follows from the definition of θ^t, the second inequality is obtained by the process of update, and the last inequality is derived as $t \leq |E|$. Thus, we have

$$f_{\theta^+}(S_\Theta^{DS}) - f_{\theta^-}(S_\Theta^{DS}) \leq m\delta B = mB \cdot \|\theta^+ - \theta^-\|_\infty.$$

□

Now we show our theoretical bound for Double Sandwich with Uniform Sampling as follows.

Theorem 32. *Given a graph $G = (V, E, \Theta, b)$, integer k, and parameters $\epsilon, \gamma > 0$ and $b_{min} = \min_{\{u,v\} \subseteq V} b(u, v)$. Let $\delta = \frac{\epsilon b_{min}}{2mB}$ and $t = \frac{2m^2B^2 \ln \frac{2m}{\gamma}}{\epsilon^2 b_{min}^2}$, then under additive form uninform sampling, Algorithm returns a pair (Θ_{out}, S_{out}) such that*

$$g(\Theta_{out}, S_{out}) \geq (1 - \epsilon) \cdot \beta(\theta_{out}^+) \cdot (1 - \frac{1}{e}),$$

with a probability $\text{Prob}[\theta \in \Theta_{out}] \geq 1-\gamma$, *where*

$$\beta(\theta^+_{out}) = \max\left\{ \frac{f_{\theta^+_{out}}(S_{U_{\theta^+_{out}}})}{U_{\theta^+_{out}}(S_{U_{\theta^+_{out}}})}, \frac{L_{\theta^+_{out}}(S^*_{L_{\theta^+_{out}}})}{f_{\theta^+_{out}}(S^*_{\theta^+_{out}})} \right\}.$$

Proof. To prove this theorem we have the following

$$\begin{aligned}
& g(\Theta_{out}, S_{out}) \\
& \geq \beta(\theta^+_{out}) \cdot (1-\frac{1}{e}) \cdot \alpha(\Theta_{out}) \\
& = \beta(\theta^+_{out}) \cdot (1-\frac{1}{e}) \cdot \frac{f_{\theta^-_{out}}(S^{DS}_{\Theta_{out}})}{f_{\theta^+_{out}}(S^{SA}_{\theta^+_{out}})} \\
& \geq \beta(\theta^+_{out}) \cdot (1-\frac{1}{e}) \cdot \frac{f_{\theta^-_{out}}(S^{SA}_{\theta^+_{out}})}{f_{\theta^+_{out}}(S^{SA}_{\theta^+_{out}})} \\
& = \beta(\theta^+_{out}) \cdot (1-\frac{1}{e}) \cdot \left(1 - \frac{f_{\theta^+_{out}}(S^{SA}_{\theta^+_{out}}) - f_{\theta^-_{out}}(S^{SA}_{\theta^+_{out}})}{f_{\theta^+_{out}}(S^{SA}_{\theta^+_{out}})}\right).
\end{aligned} \tag{6.12}$$

The first inequality is obtained by Theorem 31, the second inequality follows from the monotonicity. By the construction of parameter space $\Theta_{out} = \times_{e\in E}[\ell_e, r_e]$, for any $e \in E, \delta > 0$, let $\ell_e = \frac{1}{t}\sum_{i=1}^{t} x_e^i - \delta$ and $r_e = \frac{1}{t}\sum_{i=1}^{t} x_e^i + \delta$ as the lower bound and the upper bound of probability p_e, respectively. Then we have $\max_{e\in E} \|\theta^+_{out} - \theta^-_{out}\|_\infty = r_e - \ell_e = 2\cdot\delta$. Follows from the Lemma 12, we have

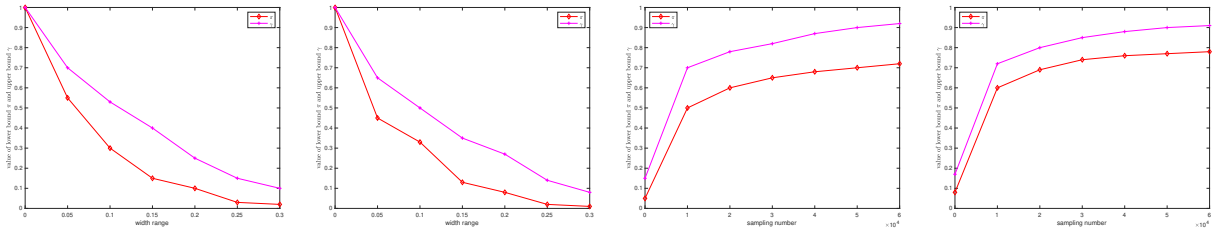
$$f_{\theta^+_{out}}(S^{SA}_{\theta^+_{out}}) - f_{\theta^-_{out}}(S^{SA}_{\theta^+_{out}}) \leq 2\delta m B.$$

Thus we have

$$
\begin{aligned}
& g(\Theta_{out}, S_{out}) \\
& \geq \beta(\theta^+_{out}) \cdot (1 - \frac{1}{e}) \cdot \left(1 - \frac{f_{\theta^+_{out}}(S^{SA}_{\theta^+_{out}}) - f_{\theta^-_{out}}(S^{SA}_{\theta^+_{out}})}{f_{\theta^+_{out}}(S^{SA}_{\theta^+_{out}})}\right) \\
& \geq \beta(\theta^+_{out}) \cdot (1 - \frac{1}{e}) \cdot \left(1 - \frac{2\delta mB}{f_{\theta^+_{out}}(S^{SA}_{\theta^+_{out}})}\right) \\
& \geq \beta(\theta^+_{out}) \cdot (1 - \frac{1}{e}) \cdot \left(1 - \frac{2\delta mB}{b_{min}}\right) \\
& = \beta(\theta^+_{out}) \cdot (1 - \frac{1}{e}) \cdot \left(1 - \frac{\epsilon b_{min}}{2mB} \cdot \frac{2mB}{b_{min}}\right) \\
& = \beta(\theta^+_{out}) \cdot (1 - \frac{1}{e}) \cdot (1 - \epsilon),
\end{aligned}
\tag{6.13}
$$

where the equality is obtained by setting $\delta = \frac{\epsilon b_{min}}{2mB}$. The rest of our proof is to show that the inequality 6.13 holds with high probability $\text{Prob}[\theta \in \Theta_{out}] \geq 1 - \gamma$ by choosing the proper sampling times t. For any given $e \in E$, our Algorithm samples t times, there are t random variables which are denotes as $x^1_e, ..., x^t_e$ with $x^i_e \in [\ell_e, r_e]$ for any e. Let $\bar{p}_e = \frac{x^1_e + \cdots + x^t_e}{t}$ as average of theses variables. By the additive form of Chernoff-Hoeffding Inequality, we have

$$
\begin{aligned}
\text{Prob}[|\bar{p}_e - p_e| > \delta] &\leq 2exp\left(\frac{-2\delta^2 t^2}{\sum_{i=1}^{t}(r_e - \ell_e)^2}\right) \\
&\leq 2exp\left(-2t\delta^2\right) \\
&= 2exp\left(-2\delta^2 \cdot \frac{\ln \frac{2m}{\gamma}}{2\delta^2}\right) \\
&= \frac{\gamma}{m}.
\end{aligned}
$$



(a) Email (b) Facebook (c) Email (d) Facebook

Figure 6.2. The experiment results

The equality follows by setting $t = \frac{\ln \frac{2m}{\gamma}}{2\delta^2} = \frac{2m^2B^2 \ln \frac{2m}{\gamma}}{\epsilon^2 b_{min}^2}$. Thus we have

$$\begin{aligned}
\text{Prob}\left[\theta \in \Theta_{out}\right] &= \text{Prob}\left[\forall e \in E, |\bar{p}_e - p_e| \leq \frac{\epsilon b_{min}}{2mB}\right] \\
&\geq 1 - \sum_{i=1}^{m} \text{Prob}\left[|\bar{p}_{e_i} - p_{e_i}| > \frac{\epsilon b_{min}}{2mB}\right] \\
&= 1 - m \cdot \text{Prob}\left[|\bar{p}_e - p_e| > \frac{\epsilon b_{min}}{2mB}\right] \\
&\geq 1 - \gamma.
\end{aligned}$$

□

An intuitive feeling is that when the number of samples is sufficient, the bound of the double sandwich algorithm with sampling becomes an approximate guarantee for the sandwich algorithm under parameter settings θ^+. This is also easy to understand, because when the number of samples is sufficient, we get the real parameter vector. The double sandwich algorithm becomes a sandwich algorithm.

6.6 Experiment

6.6.1 Settings

In the experimental part, we mainly study the effectiveness of the algorithm in two situations. The first case is that we verify the double sandwich algorithm for a given probability space. The second case is to verify the sample-based double sandwich algorithm. Actually, given a

seed set S, it is hard to calculate the robust ratio $g(\Theta, S)$, since it contains solving the profit maximization problem which is NP hard. In order to facilitate the experimental comparison, we use two indicators following the idea of Chen et al. (Chen et al., 2016b). One is about the upper bound of the solution, the other is about the lower bound of the solution. The lower bound shown in Theorem 31 is,

$$\alpha(\Theta) \cdot \beta(\theta^+) \cdot (1 - \frac{1}{e}),$$

where

$$\beta(\theta^+) = \max \left\{ \frac{f_{\theta^+}(S_{U_{\theta^+}})}{U_{\theta^+}(S_{U_{\theta^+}})}, \frac{L_{\theta^+}(S^*_{L_{\theta^+}})}{f_{\theta^+}(S^*_{\theta^+})} \right\},$$

Note that it can not be estimated since $\beta(\theta^+)$ contains optimal solution corresponding to L_{θ^+} and f_{θ^+} which are hard to find, when we gain the solution of Double Sandwich algorithm. But a good idea is that we replace the $\beta(\theta^+)$ with its lower bound $\frac{f_{\theta^+}(S_{U_{\theta^+}})}{U_{\theta^+}(S_{U_{\theta^+}})}$. Thus we get our new lower bound which is able to estimated as follows:

$$\begin{aligned}
\pi(\Theta) &= \alpha(\Theta) \cdot \frac{f_{\theta^+}(S_{U_{\theta^+}})}{U_{\theta^+}(S_{U_{\theta^+}})} \cdot (1 - \frac{1}{e}) \\
&= \frac{f_{\theta^-}(S^{DS}_{\Theta})}{f_{\theta^+}(S^{SA}_{\theta^+})} \cdot \frac{f_{\theta^+}(S_{U_{\theta^+}})}{U_{\theta^+}(S_{U_{\theta^+}})} \cdot (1 - \frac{1}{e})
\end{aligned}$$

And it is easy to obtain an upper bound $\gamma(\Theta, \theta)$ of the robust ratio $g(\Theta, S^{DS}_{\Theta})$ by

$$\begin{aligned}
g(\Theta, S^{DS}_{\Theta}) &= \min_{\theta \in \Theta} \frac{f_\theta(S^{DS}_{\Theta})}{f_\theta(S^*_\theta)} \\
&\leq \frac{f_\theta(S^{DS}_{\Theta})}{f_\theta(S^*_\theta)} \leq \frac{f_\theta(S^{DS}_{\Theta})}{f_\theta(S^{SA}_\theta)} = \gamma(\Theta).
\end{aligned}$$

We use two social networks in our experiments. All datasets are publicly available. Email can be obtained from SNAP website (htpp://snap.stanford.edu), while Facebook can be obtained from KONECT website (http://konect.uni-koblenz.de). The ground-truth propagation probability for IC model is set to $\frac{1}{degree(v)}$ as widely used in other literatures, and the

profit between nodes is proportional to propagation probability on corresponding edges. In the first case, given interval width w, we set $l_e = \min\{p_e - w/2, 0\}$, $r_e = \max\{p_e + w/2, 1\}$, $\forall e \in E$, where p_e is the ground-truth probability of e. Then we calculate the upper bound $\gamma(\Theta)$ and lower bound $\pi(\Theta)$.

We implement all the algorithms in Python and the experiments run on a workstation with an Intel Xeon 4.0GHz CPU and 64GB memory.

6.6.2 Effectiveness and Analysis

For the first case when the parameter space is given, the results of bounds about robust ratio computed by our proposed algorithms on two data sets are shown in Fig.6.2 (a) and (b) respectively. We will figure out the relationship between bounds and the width w. First, we observe that as the parameter space Θ becomes wider, the values of both π and γ become smaller, which matches our intuition that larger uncertainty results in worse robustness. The overall trends of π and γ suggest that the robust ratio may be sensitive to the uncertainty of the parameter space. Furthermore, when the interval width is less than 0.1, the two boundaries fall faster. When the interval width is greater than 0.1, the two boundary change slower as the interval width increases. When the interval span reaches a certain value, the robust ratio bounds will become very poor, which is consistent with the definition of robustness.

For the second case in which we use sampling technology, the results on all two networks are shown in Fig.6.2 (c) and (d) respectively. First, as the number of samples increases, both bounds π and γ become larger. It is reasonable because as the number of sampling becomes larger, the corresponding confidence interval becomes shorter. In addition, the number of sampling has a greater impact on the two bounds in the early period. As the number of samples reaches a certain level, the effect will become smaller. It can be seen that the width of the interval has a great influence on the robustness. For each pair of π and γ in two

graphs, the trends are consistent. All these consistencies suggests that gap ratio could be used as an indicator for the robustness of double sandwich algorithm.

CHAPTER 7

CONCLUSION

In this dissertation we study three influence optimization problems in social networks. They are active friending, interaction-aware influence maximization, robust profit maximization.

The first problem studied is active friending, in our work the active friending problems in both IC model and LT model on the general social graph and convert the original problem into an equivalent submodular cost submodular knapsack problem observing that only nodes forming paths between the source and the target contributes to the acceptance probability, we give a general algorithm ICSK to solve it under both models, which guarantees to obtain $1-e^{-1}$ solution with size constraint restrained by a factor depending on the curvature of the submodular constraint function. In addition, we give a greedy algorithm to the problem. The experimental results on real data set validate the effectiveness of the proposed algorithms.

For the interaction-aware influence maximization problem, we take the interaction among users into consideration and prove the problem is NP-hard and the objective function is non-submodular. To solve this non-submodular optimization problem, we propose the sandwich theory based on decomposing the original function into the difference between two submodular functions and design two iterated sandwich algorithms which iteratively finds a better solution from the solution of a modular/submodular upper bound function, the solution of a modular/submodular lower bound function as well as the solution of the original objective function. Experiment results validate the effectiveness of our approach.

For the robust profit maximization problem, we address the issue of uncertainty of the influence probability. Since the objective function of total profit is non-submodular, we design a double sandwich algorithm which solves non-submodular problem by means of finding the submodular upper and lower bounds of the original function. In order to further improve the robustness of the algorithm, we study the uniform sampling method and enhance the double sandwich algorithm.

REFERENCES

Alon, N., B. Awerbuch, and Y. Azar (2003a). The online set cover problem. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pp. 100–105. ACM.

Alon, N., B. Awerbuch, and Y. Azar (2003b). The online set cover problem. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, New York, NY, USA, pp. 100–105. ACM.

Bai, W. and J. A. Bilmes (2018). Greed is still good: Maximizing monotone submodular+ supermodular functions. *arXiv preprint arXiv:1801.07413*.

Bharathi, S., D. Kempe, and M. Salek (2007). Competitive influence maximization in social networks. In *International Workshop on Web and Internet Economics*, pp. 306–311. Springer.

Bi, Y., W. Wu, A. Wang, and L. Fan (2013). Community expansion model based on charged system theory. In *International Computing and Combinatorics Conference*, pp. 780–790. Springer.

Bi, Y., W. Wu, and L. Wang (2013). Community expansion in social network. In *International Conference on Database Systems for Advanced Applications*, pp. 41–55. Springer.

Bi, Y., W. Wu, Y. Zhu, L. Fan, and A. Wang (2014). A nature-inspired influence propagation model for the community expansion problem. *Journal of Combinatorial Optimization 28*(3), 513–528.

Chen, H., W. Xu, X. Zhai, Y. Bi, A. Wang, and D.-Z. Du (2014). How could a boy influence a girl? In *2014 10th International Conference on Mobile Ad-hoc and Sensor Networks*, pp. 279–287. IEEE.

Chen, S., J. Fan, G. Li, J. Feng, K.-l. Tan, and J. Tang (2015). Online topic-aware influence maximization. *Proceedings of the VLDB Endowment 8*(6), 666–677.

Chen, W., A. Collins, R. Cummings, T. Ke, Z. Liu, D. Rincon, X. Sun, Y. Wang, W. Wei, and Y. Yuan (2011). Influence maximization in social networks when negative opinions may emerge and propagate. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pp. 379–390. SIAM.

Chen, W., T. Lin, Z. Tan, M. Zhao, and X. Zhou (2016a). Robust influence maximization. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 795–804. ACM.

Chen, W., T. Lin, Z. Tan, M. Zhao, and X. Zhou (2016b). Robust influence maximization. In *International Conference on Knowledge Discovery and Data Mining*, KDD '16, New York, NY, USA, pp. 795–804. ACM.

Chen, W., C. Wang, and Y. Wang (2010a). Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1029–1038. ACM.

Chen, W., C. Wang, and Y. Wang (2010b). Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '10, New York, NY, USA, pp. 1029–1038. ACM.

Chen, W., Y. Wang, and S. Yang (2009). Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 199–208. ACM.

Chen, W., Y. Yuan, and L. Zhang (2010, Dec). Scalable influence maximization in social networks under the linear threshold model. In *2010 IEEE International Conference on Data Mining*, pp. 88–97.

Clauset, A., C. Moore, and M. E. Newman (2008). Hierarchical structure and the prediction of missing links in networks. *Nature 453*(7191), 98.

Conforti, M. and G. Cornuéjols (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.

Domingos, P. and M. Richardson (2001a). Mining the network value of customers. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pp. 57–66. ACM.

Domingos, P. and M. Richardson (2001b). 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.

Du, D.-Z., K.-I. Ko, and X. Hu (2011). *Design and analysis of approximation algorithms*, Volume 62. Springer Science & Business Media.

Feige, U., D. Peleg, and G. Kortsarz (2001). The dense k-subgraph problem. *Algorithmica 29*(3), 410–421.

Fox, J., M. Gilbert, and W. Y. Tang (2018). Player experiences in a massively multiplayer online game: A diary study of performance, motivation, and social interaction. *New Media & Society*, 1461444818767102.

Goyal, A., W. Lu, and L. V. Lakshmanan (2011a). Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference companion on World wide web*, pp. 47–48. ACM.

Goyal, A., W. Lu, and L. V. Lakshmanan (2011b). Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pp. 211–220. IEEE.

Han, M., J. Li, Z. Cai, and Q. Han (2016). Privacy reserved influence maximization in gps-enabled cyber-physical and online social networks. In *Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)(BDCloud-SocialCom-SustainCom), 2016 IEEE International Conferences on*, pp. 284–292. IEEE.

He, X. and D. Kempe (2015). Stability of influence maximization. *arXiv preprint arXiv:1501.04579*.

Iyer, R. and J. Bilmes (2012a). Algorithms for approximate minimization of the difference between submodular functions, with applications. *arXiv preprint arXiv:1207.0560*.

Iyer, R. and J. Bilmes (2012b). Algorithms for approximate minimization of the difference between submodular functions, with applications. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, UAI'12, Arlington, Virginia, United States, pp. 407–417. AUAI Press.

Iyer, R. K. and J. A. Bilmes (2013). Submodular optimization with submodular cover and submodular knapsack constraints. In *Advances in Neural Information Processing Systems*, pp. 2436–2444.

Iyer, R. K., S. Jegelka, and J. A. Bilmes (2013). Curvature and optimal algorithms for learning and minimizing submodular functions. In *Advances in Neural Information Processing Systems*, pp. 2742–2750.

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

Jung, K., W. Heo, and W. Chen (2012). Irie: Scalable and robust influence maximization in social networks. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pp. 918–923. IEEE.

Kashima, H. and N. Abe (2006). A parameterized probabilistic model of network evolution for supervised link prediction. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pp. 340–349. IEEE.

Kempe, D., J. Kleinberg, and É. Tardos (2003a). Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137–146. ACM.

Kempe, D., J. Kleinberg, and E. Tardos (2003b). Maximizing the spread of influence through a social network. In *International Conference on Knowledge Discovery and Data Mining*, KDD '03, pp. 137–146. ACM.

Kunegis, J. and A. Lommatzsch (2009). Learning spectral graph transformations for link prediction. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 561–568. ACM.

Kwon, J. and S. Kim (2010). Friend recommendation method using physical and social context. *International journal of Computer science and network security 10*(11), 116–120.

Lehdonvirta, V. (2009). Virtual item sales as a revenue model: identifying attributes that drive purchase decisions. *Electronic commerce research 9*(1-2), 97–113.

Leskovec, J., A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance (2007). Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 420–429. ACM.

Leung, C. W.-k., E.-P. Lim, D. Lo, and J. Weng (2010). Mining interesting link formation rules in social networks. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pp. 209–218. ACM.

Li, Y., D. Zhang, and K.-L. Tan (2015). Real-time targeted influence maximization for online advertisements. *Proceedings of the VLDB Endowment 8*(10), 1070–1081.

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.

Lu, W., W. Chen, and L. V. Lakshmanan (2015a). From competition to complementarity: comparative influence diffusion and maximization. *Proceedings of the VLDB Endowment 9*(2), 60–71.

Lu, W., W. Chen, and L. V. S. Lakshmanan (2015b, October). From competition to complementarity: Comparative influence diffusion and maximization. *Proc. VLDB Endow. 9*(2), 60–71.

Lu, Z., Z. Zhang, and W. Wu (2017). Solution of bharathi–kempe–salek conjecture for influence maximization on arborescence. *Journal of Combinatorial Optimization 33*(2), 803–808.

Mokken, R. J. (1979). Cliques, clubs and clans. *Quality and quantity 13*(2), 161–173.

Narasimhan, M. and J. Bilmes (2005a). A submodular-supermodular procedure with applications to discriminative structure learning. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, pp. 404–412. AUAI Press.

Narasimhan, M. and J. Bilmes (2005b). A submodular-supermodular procedure with applications to discriminative structure learning. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*, UAI'05, Arlington, Virginia, United States, pp. 404–412. AUAI Press.

Narasimhan, M. and J. A. Bilmes (2012). A submodular-supermodular procedure with applications to discriminative structure learning. *arXiv preprint arXiv:1207.1404*.

Nemhauser, G. L., L. A. Wolsey, and M. L. Fisher (1978). An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming 14*(1), 265–294.

Nguyen, H. T., T. N. Dinh, and M. T. Thai (2016, April). Cost-aware targeted viral marketing in billion-scale networks. In *The 35th Annual IEEE International Conference on Computer Communications*, INFOCOM 2016, pp. 1–9.

Nguyen, H. T., M. T. Thai, and T. N. Dinh (2016). Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *International Conference on Management of Data*, SIGMOD '16, pp. 695–710. ACM.

Richardson, M. and P. Domingos (2002a). Mining knowledge-sharing sites for viral marketing. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, pp. 61–70. ACM.

Richardson, M. and P. Domingos (2002b). 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.

Rodriguez, M. G. and B. Schölkopf (2012). Influence maximization in continuous time diffusion networks. *arXiv preprint arXiv:1205.1682*.

Shang, Y. and B. W. Wah (1998). A discrete lagrangian-based global-search method for solving satisfiability problems. *Journal of global optimization 12*(1), 61–99.

Shen, C.-Y., D.-N. Yang, W.-C. Lee, and M.-S. Chen (2015). Maximizing friend-making likelihood for social activity organization. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 3–15. Springer.

Shuai, H.-H., D.-N. Yang, P. S. Yu, and M.-S. Chen (2013). Willingness optimization for social group activity. *Proceedings of the VLDB Endowment 7*(4), 253–264.

Silva, N. B., R. Tsang, G. D. Cavalcanti, and J. Tsang (2010). A graph-based friend recommendation system using genetic algorithm. In *IEEE Congress on Evolutionary Computation*, pp. 1–7. IEEE.

Surian, D., N. Liu, D. Lo, H. Tong, E.-P. Lim, and C. Faloutsos (2011). Recommending people in developers' collaboration network. In *Reverse Engineering (WCRE), 2011 18th Working Conference on*, pp. 379–388. IEEE.

Sviridenko, M. (2004). A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters 32*(1), 41–43.

Tang, J., X. Tang, and J. Yuan (2017). Towards profit maximization for online social network providers. *CoRR abs/1712.08963*.

Tang, Y., Y. Shi, and X. Xiao (2015). Influence maximization in near-linear time: A martingale approach. In *International Conference on Management of Data*, SIGMOD '15, pp. 1539–1554. ACM.

Tang, Y., X. Xiao, and Y. Shi (2014a). 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.

Tang, Y., X. Xiao, and Y. Shi (2014b). Influence maximization: Near-optimal time complexity meets practical efficiency. In *International Conference on Management of Data*, SIGMOD '14, pp. 75–86. ACM.

Thai, M. T. and P. M. P. M. Pardalos (2012). *Handbook of optimization in complex networks*. Springer US.

Thai, M. T. and P. M. Pardalos (2012). *Handbook of optimization in complex networks*. Springer US.

Tong, A., D.-Z. Du, and W. Wu (2018). On misinformation containment in online social networks. In *Advances in Neural Information Processing Systems*, pp. 339–349.

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.

Vondrák, J. (2010). Submodularity and curvature: The optimal algorithm (combinatorial optimization and discrete algorithms).

Wan, P.-J., D.-Z. Du, P. Pardalos, and W. Wu (2010). Greedy approximations for minimum submodular cover with submodular cost. *Computational Optimization and Applications 45*(2), 463–474.

Wang, A., W. Wu, and L. Cui (2016). On bharathi–kempe–salek conjecture for influence maximization on arborescence. *Journal of Combinatorial Optimization 31*(4), 1678–1684.

Wang, Z., J. Liao, Q. Cao, H. Qi, and Z. Wang (2015). Friendbook: a semantic-based friend recommendation system for social networks. *IEEE transactions on mobile computing 14*(3), 538–551.

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

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

Wasserman, S. and K. Faust (1994). *Social network analysis: Methods and applications*, Volume 8. Cambridge university press.

Willis, D., D. Godse, and G. Freedman (2014, September 30). Online video game advertising system and method supporting multiplayer ads. US Patent 8,849,701.

Wolsey, L. A. (1982). An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica 2*(4), 385–393.

Wu, W., D.-Z. Du, et al. (2018). An approximation algorithm for active friending in online social networks. *arXiv preprint arXiv:1811.00643*.

Wu, W.-L., Z. Zhang, and D.-Z. Du (2018). Set function optimization. *Journal of the Operations Research Society of China*, 1–11.

Xie, X. (2010). Potential friend recommendation in online social network. In *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, pp. 831–835. IEEE Computer Society.

Yang, D.-N., Y.-L. Chen, W.-C. Lee, and M.-S. Chen (2011). On social-temporal group query with acquaintance constraint. *Proceedings of the VLDB Endowment 4*(6), 397–408.

Yang, D.-N., H.-J. Hung, W.-C. Lee, and W. Chen (2013). Maximizing acceptance probability for active friending in online social networks. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 713–721. ACM.

Yang, D.-N., C.-Y. Shen, W.-C. Lee, and M.-S. Chen (2012). On socio-spatial group query for location-based social networks. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 949–957. ACM.

Yuan, J., W. Wu, Y. Li, and D. Du (2017). Active friending in online social networks. In *Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies*, pp. 139–148. ACM.

Zhu, Q., H. Hu, C. Xu, J. Xu, and W.-C. Lee (2014). Geo-social group queries with minimum acquaintance constraint. *arXiv preprint arXiv:1406.7367*.

BIOGRAPHICAL SKETCH

Shuyang Gu attended Qinhuangdao First High School for high school study, in Hebei, China. She spent four years learning Electrical Engineering at North China Electric Power University and obtained a Bachelor of Engineering. She joined the Computer Science Department at The University of Texas at Dallas as a master's student in 2012 and then became a PhD student in 2014. Her PhD study is under the supervision of Dr. Weili Wu. She obtained a PhD in Computer Science in 2020. Her research interests include social network analysis, data science.

CURRICULUM VITAE

Shuyang Gu

March 15, 2020

Contact Information:

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

Email: shuyang.gu@utdallas.edu

Educational History:

BS, Electrical Engineering, North China Electric Power University, 2003
MS, Computer Science, The University of Texas at Dallas, 2020
PhD, Computer Science, The University of Texas at Dallas, 2020

Influence Optimization Problems in Social Networks
PhD Dissertation
Computer Science Department, The University of Texas at Dallas
Advisors: Dr.Weili Wu

Employment History:

Electrical Engineer, State Grid Corporation of China, Jan 2006 – August 2012
Electrical Engineer, Beijing Electric Power Company, July 2003 – Jan 2006

Professional Recognitions and Honors:

Outstanding Thesis Award, North China Electric Power University, 2003
Excellent Academic Performance Fellowship, North China Electric Power University, 2000-2002
Freshman Award for Excellent Performance in National College Admission Exam, North China Electric Power University, 1999

Professional Memberships:

Institute of Electrical and Electronics Engineers (IEEE), 2014–present