Chapter 1

Recent Advances in Information Diffusion and Influence Maximization of Complex Social Networks

Huiyuan Zhang

University of Florida

Subhankar Mishra

University of Florida

My T. Thai

University of Florida

CONTENTS

1.1 Abstract

Nowadays, social influence is ubiquitous in everyday life, online social networks have become a focal point for research in science. Formal mathematical models for the analysis of spread of social influence have emerged as a major topic of interest in diverse areas such as sociology, economics and computer science. Empirical studies of diffusion on social networks date back to the 1940s. Later on, theoretical propagation models were introduced in late 1970s. Then, motivated by the design of marketing strategy, along with the problem of influence maximization has been formally defied, the field of studying social influence has received lots of research interests. In particular, the rapid growth of online social networks such as Facebook, Twitter and Google+ has intensified interests in this field, and the past decade has seen a burgeoning network literature from computer community.

In this chapter, our goal is to provide readers with a comprehensive review of this burgeoning literature. We begin with an overview of widely used theoretical diffusion models, in which three families of diffusion models: threshold models, cascading models and epidemic models are introduced. Our subsequent discussion mainly focuses on the recent algorithmic study and analytical results of the influence maximization problem. We end with a discussion of some open problems and challenges.

1.2 Introduction

Nowadays, the development of Internet have revolutionized the way we communicate with each other. Communication helps us better share knowledge, ideas and beliefs, thus influencing people behaviors. The study of information diffusion and social influence have been attracted scientists from sociology and economy can be tracked back to the early 60's. In the recent decades, the rapid growth of Online Social Networks (OSNs) such as Facebook, Twitter and Google+ provide a nice platform for information diffusion and fast information exchange among their users. In addition, the massive data obtained from millions of users and more than a billion social ties in those giant networks has greatly facilitated analytical works about user behavior, and even a large scale algorithmic study scientists from computer science have being engaged in this popular field.

Diffusion, according to Roger's definition [49], is the process by which an innovation is communicated through certain channels over time among the members of a social system. Three important elements: individual member, mutual interactions and communication channels are introduced from this definition, which are set as the basis for future analytical framework.

Later on, various diffusion models have been proposed to study the contagion properties in a vast area such as widespread adoption in viral marketing [16, 47, 37], information propagation on blogs [33, 35] and infectious diseases transmissions in epidemiology [15, 3].

One of the goals in studying social influence is the problem of *Influence Maximization*, which arises from the context of widespread adoption in viral marketing.

This problem is firstly proposed by Kempe et al [27], then rapidly becoming a hot topic in social network field. The influence maximization problem is formally described as follows: given a social network represented by a(n) directed/undirected graph with nodes as users, edges are corresponding to social ties, edge weights are capturing influence probabilities, and a budget k, which is a integer; the goal is to find a seed set of k users such that by targeting these, the expected influence spread (defined as the expected number of influenced users) is maximized. Here, the expected influence spread of a seed set depends on the influence diffusion process which is captured by diffusion models.

Therefore, in this book chapter, we start with providing an overview of diffusion models that have been extensively used in studying social influence. In general, all existing diffusion models can be categorized into three classes: threshold models [26, 27, 28, 43, 48, 6], cascading models [21, 22, 9, 8] and epidemic models [29, 36]. Figure 1.1 provides an overview of those models. For each model, we give detailed description diffusion process, activation condition as well as its own properties and applications. With the framework in place, we move on to the algorithmic results of the influence maximization problem.

We are now interested in choosing an influential set to target in the context of above models. Kempe et al. [27] prove the influence maximization problem is NP-hard under both of the Linear Threshold model and Independent Cascading model, and give a simple greedy algorithm with approximation ratio of $1-1/e$. However, the nature greedy algorithm suffers from the severe scalability problem. Therefore, considerable work has been done to improve it. In the second half of this book chapter, we demonstrate recent algorithmic study such as CELF [34], CELF++ [24], Simpath [25] and LDAG [11] algorithms, which can obtain high scalability for influence maximization problem.

Outline In this chapter, we survey the recent advances in theoretical propagation models of online social networks, as well as the algorithms for the Influence Maximization problem. In section 1.4, we give an overview of existing diffusion models, which can be categorized into three main classes: threshold models, cascade models and epidemic models. Upon each kind of diffusion model, we also provide some interesting extensions. And with this framework, we move forward to the next section 1.5, in which we survey various approaches for the influence maximization problem with high scalability. In the last section. we conclude the chapter with some applications of social influence and information diffusion.

1.3 Social Influence And Influence Maximization

Social influence, as defined by Rashotte [46], is the change in an individual's thoughts, feelings, attitudes, and behaviors that results from interaction from other people or group. Social influence takes many forms and can been seen everywhere in OSNs. In the field of data mining and big data analysis, many applications such as viral marketing, recommendation systems and information diffusion are involved with social influence.

Influence maximization (IM) is one of the fundamental problems in studying social influence. For the reason that people are likely to be affected by decisions of their friends and colleagues, some researchers and marketers have investigated into social influence and the word-of-mouth effect in promoting new products and making profitable marketing strategies. Suppose that with the knowledge individual's preference and their influence on each other, and we would like to promote a new product that will be adopted by a large amount of users in this network. The strategy of viral marketing is to select a small number of influential members within this network at the beginning, and then by convincing them to adopt the new product and utilizing the social influence effect – users advertise and recommend the product to their friends, we can trigger a widespread of adoptions. Henceforth, the influence maximization problem has arisen: which key individuals should we target as the promising seeds in order to maximize the spread of influence?

In [17, 47], the influence maximization problem was studied in a probabilistic model of interaction, selection of the most influential seeds were based on individual's overall effect on the network. In other works [27, 28, 34, 10, 54], many researchers take this seeding selection as a problem in discrete optimization. Formally, the influence maximization problem is defined as follows:

Definition 1.1 (**Influence Maximization**) Given a budget k and a social network, which is represented as a directed graph $G = (V,E)$, where users are represented as nodes and edges indicate their relationships, the goal is to select a seed set of k users such that by initially targeting them, the expected influence spread (in terms of expected number of adopted users) can be maximized.

The expected influence spread is related to the propagation process of the influence, which is captured by the diffusion models. In section 1.4, an overview of theoretical diffusion models is provided, and for most of the models we introduced, the optimal solution for the influence maximization problem is shown to be NP-hard. A well-known greedy (1-1/e) approximation algorithm is extensively used for approximating the optimal solution of the original problem and its extensions under different models. However, the approximation algorithm requires that the influence function hold two basic properties:

Definition 1.2 (**Monotonicity**) A set function f is monotone if $f(S) \leq f(T)$ such that $S \subset T \subset U$;

Definition 1.3 (**Submodularity**) A set function f is submodular if it satisfies

$$f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T) \tag{1.1}$$

for all elements $v \in S$ and $S \subset T$.

1.4 Information Diffusion Models

Influence diffusion is the process that information propagates through certain intermediaries over time among the individuals of a social network. Empirical studies of diffusion in social networks began in the middle of the 20th century and Granovetter [26] was the first to introduce a formal mathematical model. Currently, there are a variety of diffusion models arising from the economics and sociology communities. The most popular models are *Linear Threshold* model and *Independent Cascading* model, which are widely used in studying the social influence problems. Besides those two well-known models, there are many variations and extensions models to reflect more complicated real-world situations. For example, in addition to the expected number of adopted users, [56] considered the expected total opinions of adopted users, which is more meaningful. [9] proposed a new model, named IC-N model, which took into account the negativity bias during the propagation process. In this section, we survey the recent literature on theoretical models of influence diffusion.

Social network is a kind of social structure, which is consist of social actors such as individual users or organizations and a complex set of relationship between each two of them. Formally, a social network is represented as a graph $G = (V, E)$, which can be either a directed or undirected graph according to its real application and network property. In graph G, each vertex $v \in V$ represents an individual user. In a directed graph, an edge $(u, v) \in E$ represents u has an influence on v; in an undirected graph, an edge (u, v) represents mutual influence between u and v. Particularly, an undirected graph can be viewed as a directed graph by treating each edge as a bidirectional edge with the same influence on both direction. In addition, let $N(v)$ denote v's neighbors in an undirected graph, let $N^{in}(v)$ and N^{out} denote the sets of incoming neighbors (or in-neighbors) and outgoing neighbors (or out-neighbors), respectively.

1.4.1 Threshold Models

In this subsection, we give an overview of the concept of threshold models and show how these models characterize collective behaviors. In mathematical or statistical modeling, a *threshold model* is any model where a threshold value, or set of threshold values, is used to distinguish ranges of values where the behavior predicted by the model varies in some important way.

In threshold models, someone first breaks the silence of the network because that activity provides the individual utility. It is the distribution of individual thresholds, defined as the number of other people who must be doing the activity before a given individual joins in, that determines whether or not others would follow this activity. The threshold models were firstly proposed by Mark Granovetter [26] to model collective behavior, which aimed at treating binary decisions problems, such as diffusion of innovations, spreading rumors and diseases, voting and so on. He used the threshold model to explain the riot, residential segregation, and the spiral of silence. In the spirit of Granovetters threshold model, the ”threshold” is ”the number or proportion of others who must make one decision before a given actor does so”. It

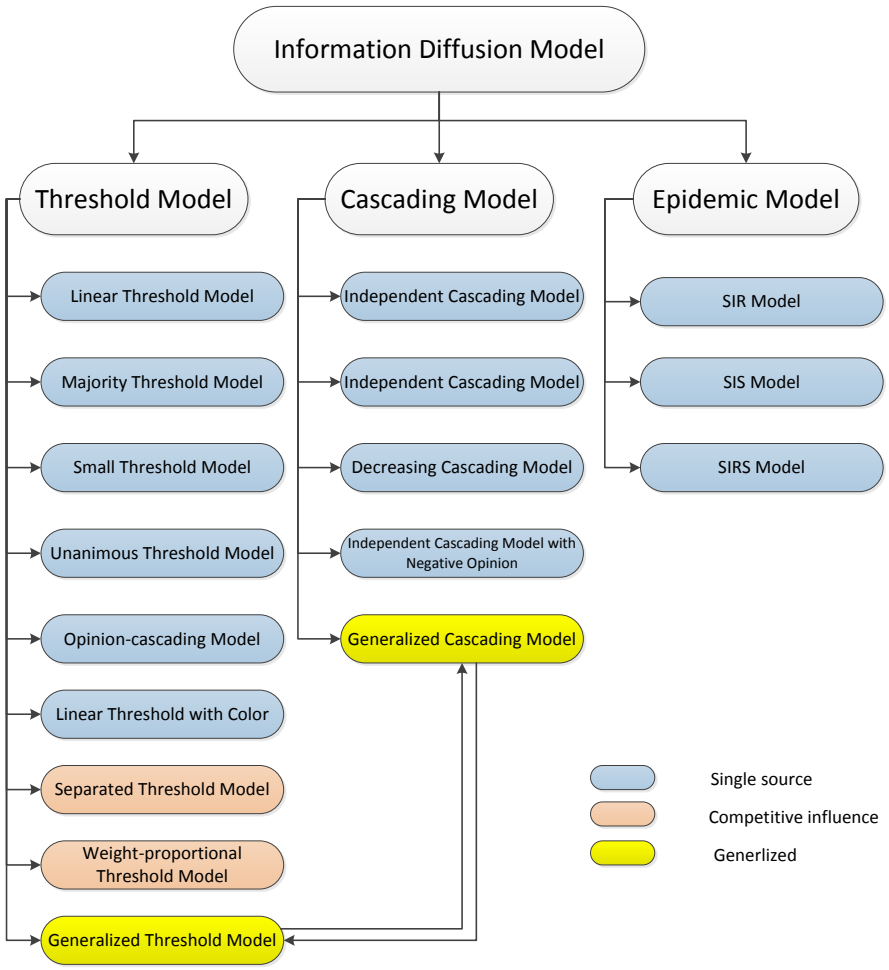

Figure 1.1: An overview of information diffusion models

is necessary to emphasize the determinants of threshold. A threshold is different for individuals, and it may be influenced by many factors: social economic status, education, age, personality, etc. Further, Granovetter relates "threshold" with utility that one gets from participating collective behavior or not. By using the utility function, each individual will calculate his cost and benefit from undertaking an action. And situation may change the cost and benefit of the behavior, so threshold is situation-specific. The distribution of the thresholds determines the outcome of the aggregate behavior (for example, public opinion). In other words, this threshold represents the number of other agents in the population or local neighborhood following that particular activity. Each agent has a threshold that, when exceeded, leads the agent to adopt an activity.

In his model, each edge (which represents a connection) (v,u) is associated with a weight $w_{v,u}$, and each node v has a threshold θ_v such that if the fraction of v's neighbors which are active exceeds v's threshold, then v will become an active. Granovetter claims that minor perturbations in the standard deviation of a distribution produce massive discontinuous changes in the number of people acting, from 6% to nearly 100% of the whole group. The reason is that in threshold models, the intrinsic utility of the behavior to an individual may be more important in determining that individual's behavior than social influence. However, even a limited amount of social influence may have a strong effect on the collective outcome [26].

Threshold models are especially useful in a structural analysis of collective action, an approach that most rational theorists have avoided. Sudden changes in the level of production of a particular public good does not necessarily reflect similar changes in the overall preferences of the actors. What really matters is the distribution of thresholds and the social connections through which members could have chances to learn about the others.

1.4.1.1 Linear Threshold Model

Linear Threshold (LT) model is the one that has been extensively used in studying diffusion models among the generalizations of threshold models. In this model, each node v has a threshold θ_v, and for every $u \in N(v)$, (u,v) has a nonnegative weight $w_{u,v}$ such that $\sum_{u \in N(v)} w_{u,v} \leq 1$. Given the thresholds and an initial set of active nodes, the process unfolds deterministically in discrete steps. At time t, an inactive node v becomes active if

$$\sum_{u \in N^a(v)} w_{u,v} \geq \theta_v$$

where $N^a(v)$ denotes the set of active neighbors of v. Every activated node remains active, and the process terminates if no more activations are possible. The threshold in this model is related to a linear constraint of edge weight, and hence get the name for the model. It is important to note that given the thresholds in advance, the diffusion process is deterministic, but we can still inject the randomness by randomizing the individual threshold. For example, the thresholds selected by Kempe

et al. [27, 28] are uniformly at random from the interval [0,1], which also intend to model the lack of knowledge of their values.

Given the influence function $\sigma(\cdot)$, Kempe et al. [27] prove that:

Theorem 1.1
For an arbitrary instance of the Linear Threshold Model, the objective influence function $\sigma(\cdot)$ is submodular.

Theorem 1.2
The influence maximization problem is NP-hard under the Linear Threshold model.

Granovetter and Schelling's approach is based on the use of node-specific thresholds [26, 51], there is another class of approaches hard-wires all thresholds at a known and fixed value. This kind of model is often used in treating binary decision problems such as voting, virus propagation network and so on. In this model, let $d(v)$ denote the degree of a node $v \in V$, and threshold value $\theta_v \in \mathcal{N}$, where $\theta_v \in [1, d(v)]$. This definition is adopted by the following three models.

1.4.1.2 The Majority Threshold Model

The Majority Threshold (MT) model is one of the most important and well studied model, in which each vertex $v \in V$ becomes active if the majority of its neighbours are active, that is the threshold $\theta_v = \frac{1}{2}d(v)$. This model has many applications in voting systems, distributing computing and so on [44, 45]. Chen [43] shows that with the majority thresholds setting, the influence maximization problem shares the same hardness of approximation ratio as the general one. Chen [43] also provides the following inapproximability result of the majority thresholds model.

Theorem 1.3
Assume the Influence Maximization problem with arbitrary thresholds can not be approximated within the ratio of $\sigma(n)$, for some polynomial time computable function $\sigma(n)$. Then the problem with majority thresholds can not be approximated within the ratio of $O(\sigma(n))$.

1.4.1.3 The Small Threshold Model

The other interesting case is the Small Threshold (ST) model, in which all thresholds are small constant [48]. Intuitively, when the threshold $\theta_v = 1$, the influence maximization problem can be easily solved by selecting an arbitrary node in each connected component. However, Chen [43] shows that the hardness of approximation result continues to hold when each vertex's threshold $\theta_v = 2$. In addition, Dreyer [18] proves that if the threshold of any vertex is θ_v for any $\theta_v \geq 3$, the problem is NP-hard as well.

Theorem 1.4
Assume the Influence Maximization problem with arbitrary thresholds can not be approximated within the ratio of $\sigma(n)$, for some polynomial time computable function $\sigma(n)$. Then the problem can not be approximated within the ratio of $O(\sigma(n))$ when all thresholds are at most 2.

1.4.1.4 The Unanimous Threshold Model

In the Unanimous Threshold (UT) model, the threshold for each vertex is $\theta_v = d(v)$, which is equal to its degree. With this setting, the UT model is the most influence-resistant model among all the threshold models. This model is usually used in studying complex network security and vulnerability. For example, in an ideal virus-resistant network, when the computer virus is spreading, a vertex can be affected if all of its neighbours have been infected. For this special case, the influence maximization problem is equivalent to the Vertex Cover problem. Thus, it admits a approximation algorithm with ratio 2 and is NP-hard as well [43].

Theorem 1.5
If all thresholds in a graph are unanimous, the Influence Maximization problem is NP-hard.

Other Extensions

The threshold models can be further generalized in a very natural way by replacing the activation function with an arbitrary function in relation with the set of a vertex's activated neighbours. For example, Bhagat et al. [4] propose a Linear Threshold with Color (LT-C) model that factors in user's experience with a product, in which they adapt the LT model by adding three more status of users activities and defining an objective function that explicitly captures product adoption, not the influence. Banerjee et al. [2] further extend the LT model to handle a more complicated case, in which each node is allowed to switch back and forth between active and inactive regarding each cascade. This model is shown to be a rapidly mixing Markov chain and the corresponding steady state distribution is used to estimate highly likely cascade adopted in the network. Furthermore, consider that users now are engaged in many different social networks, information can be diffused across multiple networks simultaneously, [42, 52] adapt the LT model to deal with IM problem under multiple networks.

1.4.2 Cascading Model

Inspired by the work on interacting particle systems [19, 38] and probability theory, dynamic cascade models are considered for the diffusion process. In the context of marketing, Goldenberg et al. [21, 22] firstly studied the cascade models. In the

cascade models, the dynamics is captured in a step-by-step fashion,: at time t, when a node v first becomes active, it has a single chance of influencing each previously inactive neighbour u at time $t+1$. And it successfully turns u to be activated with a probability $p_{v,u}$. In addition, if multiple neighbours of u become active at time t, their attempts to activate u are sequenced in an arbitrary order. If one of them say w succeeds in time t, then u becomes active in time $t+1$; however, whether w succeeds or not, it cannot make any more attempts in the following time steps. Similar to the threshold models, the process terminates until there are no more activations.

1.4.2.1 Independent Cascading Model

To better describe the cascading models, one thing we need to specify is that the probability for a newly activated node v to successfully make an attempt to activate its currently inactive neighbours u. The simplest case is Independent Cascading (IC) model, in which the probability is a constant $p_u(v)$, independent of the history of the diffusion process thus far. In addition to that, to better defined the model, we also need to introduce the *order-independence* here. Let S denote the set of nodes that have already attempted and failed to activate u, and the probability for v to successfully active u is denoted by $p_u(v|S)$. Let $v_1, v_2, ...v_k$, and $v'_1, v'_2, ...v'_k$ be two different permutations of S, and $T_i = \{v_1, v_2, ...v_i\}$, $T'_i = \{v'_1, v'_2, ...v'_i\}$. The order-independence indicates that the order of attempts made by each node in S does not affect the probability for u to be active in the end, which is

$$\Pi_{i=1}^{k}(1 - p_u(v_i|S \cup T_i)) = \Pi_{i=1}^{k}(1 - p_u(v'_i|S \cup T'_i))$$

where $S \cap T = \emptyset$.

1.4.2.2 Decreasing Cascading Model

Compared with the IC model, the Decreasing Cascading (DC) model [28] is more general and practical. (We adopt all the definitions in the IC model here) The DC model naturally incorporates a restriction that the function $p_u(v|S)$ is non-decreasing in S, which indicates that $p_u(v|S) \leq p_u(v|T)$, where $S \subset T$. This better reflects the information saturation problem in the real-world: the probability of a successful activation of a node u decreases if more people have already made the attempts. The DC model contains the IC model as a special case.

1.4.2.3 Independent Cascading Model with Negative Opinion

In [9], Chen proposed the Independent Cascading Model with Negative Opinion (IC-N) which incorporates the negative opinions into the propagation process. The IC-N model associates a new parameter q called the *quality factor* which models the natural behavior of users adopting negative opinions due to defects of the product/service. In this model, each activated can be either positive or negative, and with probability q, each newly active node become positive and with probability $1-q$, it becomes negative. In addition, when a node u is negatively activated, it becomes negative with

probability 1 and remain negative in the following rounds. This reflects the negativity bias and dominance phenomenon in social psychology [50].

Generalized Threshold and Cascade Models

We have thus far introduced two families of widely studied propagation models, before heading to the next kind of diffusion model, we want to introduce a more general and broader framework that generalize the classic LT model and IC model in this subsection. In particular, under such setting, Kempe et al. [27] prove that the general cascade model and general threshold model are equivalent. And because of this equivalence, we can unify these two different views of diffusion in social networks.

- **Generalized threshold model.** In the general threshold model, each node v has a threshold θ_v, and associates with a function f_v that maps the set of its neighbours $N(v)$ to the range [0,1] and subject to the condition $f_v(\emptyset) = 0$. This function could be an arbitrary monotone function. The dynamic of diffusion process follows the LT model. But a node v becomes active at time t if and only if $f_v(N^a(v)) \geq \theta_v$, where $N^a(v)$ is the subset of active neighbours of v at time $t-1$. It is easy to see that the generalized threshold model contain the LT model as a special case, in which the threshold function is subject to $f_v = \sum_{u \in N^a(v)} w_{u,v}$, and $\sum_{u \in N(v)} w_{u,v} \leq 1$.
- **Generalized cascade model.** Compared with the specific cascade models, we generalize the cascade model by allowing the probability that u successfully activates its neighbour v to depend on the other active neighbours of v that have tried. Thus, we change the activation probability $P_{u,v}$ to an incremental function $p_v(u,S) \in [0,1]$, where u and S are two disjoint subsets of $N(v)$. In each discrete time stamp, when a newly activated node u attempts to activate a currently inactive node v, it succeeds with probability $p_v(u,S)$, where S denotes the set of nodes that have already made their attempts. The IC model can be viewed as a special case of the generalized cascade model, in which $p_v(u,S)$ is set to a constant $p_{u,v}$. Furthermore, the *order-independence* which has been introduced in the IC model is also adopted here.

Next, we show that if the threshold function $theta_v$ is chosen independently and uniformly at random, then those two generalized models are equivalent as shown by the following conversion.

Let f_v be a threshold function of general threshold model, and S be the set of nodes that have already tried to activate v. Then in order to define an equivalent cascade model, we need to know the probability of additional node u can activate v if all the nodes in S have failed. Once the node in S failed, node v's threshold θ_v should be in the range $(f_v(S),1]$. Therefore, with the constraint that it should be uniformly distributed, the probability that a neighbour $u \notin S$ successfully activate v is

$$p_v(u,S) = \frac{f_v(S \cup \{u\}) - f_v(S)}{1 - f_v(S)}$$

where nodes in S failed to activate v. It is easy to see that the generalized cascade model can be converted to the generalized threshold model with this function.

On the other side, let v be a node in the cascade model, with its neighbour set denoted by $S = \{u_1, u_2, ...u_k\}$. All the nodes in S have tried to activate v in an order T and let us assume $T = \{u_1, u_2, ...u_k\}$, and $S_i = \{u_1, u_2, ...u_i\}$, then the probability that v hasn't been influenced is $\prod_{i=1}^{k}(1 - p_v(u_i, S_{i-1}))$. According to the order-independence, this value is not affected by the order of u_i, but only depends on the set S only, thus we can obtain that

$$f_v(S) = 1 - \prod_{i=1}^{k}(1 - p_v(u_i, S_{i-1}))$$

In this way, the threshold model can be shown be to equivalent to cascade model.

1.4.3 Epidemic Model

The epidemic has had a major impact on the life and politics of the country. Modeling the infectious diseases became a matter of general interest in the 19th century. An epidemic model describes the transmission of contagious disease through individuals. In the recent century, it has been widely used to model computer virus infections and information propagations such as news and rumors.

1.4.3.1 SIR Model

The SIR (Susceptible-Infectious-Recovered) model first proposed by Kermack and McKendrick [29]. In this model, it considers a fixed population which divided into three distinct classes: Susceptible (S),Infectious (I), and Recovered (R). The individual goes through consecutive states:

$$S \rightarrow I \rightarrow R$$

And the dynamics of the model cascades in such a way: given a fixed population at a particular time t, there exists three groups of people, $S(t)$ represents the number of people who are susceptible to the contagion, $I(t)$ represents the number of people who have been infected and are capable of infecting those who are susceptible; $R(t)$ is the number of people who have been infected and recovered, which means they are immune to be infected again in the future. Using the contact rate β from S to I, and $1/\gamma$ the average infectious period, Kermack and McKendrick [30] derived the following equations:

$$\frac{dS}{dt} = -\beta SI$$
$$\frac{dI}{dt} = \beta SI - \gamma I$$
$$\frac{dR}{dt} = \gamma I$$

And the critical parameter $R_0 = \beta S_0/\gamma$ is called the basic reproduction number. We can see that $R = 1$ is the critical value;$R < 1$ implies no epidemic and $R > 1$ that an epidemic is possible.

In this model, several assumptions are made in the formulation of the equations. First of all, each individual is considered as having the same probability of contracting the disease with a rate of β, which is also the infection rate of the disease. Therefore, an infected individual can transmit the disease with βN other susceptible people per unit time, and the fraction of contacts by an infected with a susceptible is S/N. In addition, given the rate of new infections as $\beta N(S/N)I = \beta SI$ [7], the number of newly infected people per unit time is $\beta N(S/N)$. Secondly, consider the population leaving the susceptible group is equal to the number of newly infected people, we can get the second and third equations above. Specifically, a number equals to the fraction of infective people who are leaving the this class per unit time to enter the removed group. These processes which occur simultaneously are known as the *Law of Mass Action* [12], which is a widely accepted idea that the rate of contact between two groups in a population is proportional to the size of each of the groups concerned.

1.4.3.2 SIS Model

The SIS model consider a fixed population with only two compartments Susceptible S(t) and infected I(t), thus the flow of this model may be considered as follows:

$$S \rightarrow I \rightarrow R$$

The SIS an be easily derived from the SIR model by simply considering that the individuals recover with no immunity to the disease, that is, individuals are immediately susceptible once they have recovered.

Thus Removing the equation representing the recovered population from the SIR model and adding those removed from the infected population into the susceptible population, we can get the following differential equations:

$$\frac{dS}{dt} = -\beta SI + \gamma I$$
$$\frac{dI}{dt} = \beta SI - \gamma I$$

1.4.3.3 SIRS Model

The SIRS model is an extension of the SIR model. An individual can go through consecutive states:

$$S \to I \to R$$

The difference between this model and the SIR model is that, it allows the individual of recovered group to leave and rejoin the susceptible group. Thus, we can get the following equations:

$$\begin{aligned} \frac{dS}{dt} &= -\beta SI + fR \\ \frac{dI}{dt} &= \beta SI - \gamma I \\ \frac{dR}{dt} &= \gamma I - fR \end{aligned}$$

where f is the average loss of immunity rate of recovered individuals.

1.4.4 Competitive Influence Diffusion Models

All of the above models have primarily focused on diffusion of single cascade, but when multiple innovations are competing within a social network, things become different yet interesting. Carnes et al. in [8] consider the problem faced by a company that would like to spread out its new product into market while a competing product is already being introduced. There are two assumptions: first, the consumers use only one of the two products and influence their friends in their decision of which product to use; second, the follower has a fixed budget available that can be used to target a subset of consumers. In [8], they propose two models for describing how two technologies simultaneously diffuse over a given network.

1.4.4.1 Distance-based Model

The first model, *a distance-based model*, is related to competitive facility location [20] on a network. In this model, the location of a node in the network is important, as well as the connectivity of a node. The central idea is that a consumer will be more likely to mimic the behavior of an early adopter if their distance in the social network is relatively small. It is pointed out in [8] that the expected number of nodes which adopt A will be denoted by

$$\rho(I_A|I_B) = \mathbf{E}[\sum_{u\in V} \frac{\nu_u(I_A, d_u(I,E_a))}{\nu_u(I_A, d_u(I,E_a)) + \nu_u(I_B, d_u(I,E_a))}]$$

where the expectation is over the set of active edges. I_A and I_B are the initial sets of adopters of A and B respectively, and I is their union set. $d_u(I,E_a)$ denotes the shortest

distance from u to I along the edges in E_a. After fixing I_B and trying to determine I_A so as to maximize the expected number of nodes that adopt technology A would be:

$$max\{\rho(I_A|I_B) : I_A \subseteq (V - I_B), |I_A| = k\}$$

The following theorem gives an approximation bound for this equation.

Theorem 1.6
For any given I_B with $|V - I_B| \geq k$, the Hill Climbing Algorithm gives a $(1 - 1/e - \varepsilon)$-approximation algorithm for the above result.

1.4.4.2 Wave Propagation Model

The second model, *wave propagation model*, regards the propagation as happening in discrete steps. In step d, all nodes that are at distance at most $d - 1$ from some node in the initial sets have adopted technology A or B, and all nodes for which the closest initial node is farther than $d - 1$ do not have a technology yet. Similar to the distance-base model, it gives out the solution:

$$max\{\pi(I_A|I_B) : I_A \subseteq (V - I_B), |I_A| = k\}$$

where

$$\pi(I_A|I_B) = \mathbf{E}[\sum_{v \in V} P(v|I_A, I_B, E_a)]$$

And the authors provide another theorem that gives the same approximation ratio as above:

Theorem 1.7
For any given I_B with $|V - I_B| \geq k$, the Hill Climbing Algorithm gives a $(1 - 1/e - \varepsilon)$-approximation algorithm for the above result.

Consequently by computational experiments, the authors point out although it is NP-hard to select the most influential subset to target, it is possible to give an efficient algorithm that is within 63% of optimal. Lastly, using the distance-based model with edge probabilities equal to 1, these problems can also be seen in the context of competitive facility location [1, 14] on a network.

1.4.4.3 Weight-proportional Threshold Model

Consider the real world scenarios where different kinds of innovations or products are competing with each other, competitive threshold models are suggested by Borodin et al. in [6]. Under the competitive setting, the goal is to maximize the spread of one cascade in the presence of one or more competitors.

In order to describe the process, we use the following notation for the next two models.

Definition 1.4 In discrete time stamp t, let Φ^t denote the set of active nodes, in particular, let Φ_A^t and Φ_B^t be the sets of A-active and B-active nodes in time stamp t respectively.

Given two different seeds S_A and S_B at the beginning, in each time stamp, every inactive node v changes its status according to the incoming influence from its currently active neighbours as follows: v becomes active when $\sum_{u\in\Phi^t} w_{u,v} \geq \theta_v$ is satisfied; in addition, v becomes a A-active node with probability

$$Pr[v \in \Phi_A^t | v \in \Phi^t \setminus \Phi^{t-1}] = \frac{\sum_{u\in\Phi_A^t} w_{u,v}}{\sum_{u\in\Phi^t} w_{u,v}}$$

It adopts cascade B, otherwise.

The problem maximizing the spread of cascade A can be easily reduced to original influence maximization problem by setting $S_B = \varnothing$. Thereby, this problem is also NP-hard, as proved in [27].

Intuitively, by adding one more node to the initial set S_A, the spread of cascade A could be expended. However, the influence function $\sigma(\cdot)$ is neither monotone nor submodular under the Weight-proportional Threshold (WT) model, as shown by a count example in [6].

1.4.4.4 Separated Threshold Model

In previous model, a node v changes its status from inactive to active whenever the influence from all of its currently active neighbours exceeds its threshold θ_v. However, nodes may not have the same threshold towards each competitor, and the influence strength between each pairs of nodes could be different regarding each cascade. Formally, each node v has two thresholds θ_v^A, θ_v^B, and each edge (u,v) is associated with two weights $w_{u,v}^A, w_{u,v}^B$ corresponding to cascades A and B, respectively. And both of weights satisfy the constraints as the LT model. In time stamp t, every inactive node v will be A-active when $\sum_{u\in N^a(v)\cap\Phi_A^{t-1}} w_{u,v}^A \geq \theta_v^A$, and will be B-active when $\sum_{u\in N^a(v)\cap\Phi_B^{t-1}} w_{u,v}^B \geq \theta_v^B$. If both thresholds are exceeded during the same stamp t, then v adopts a cascade uniformly at random.

However, unlike previous model, the probability that cascade A will be adopted by a node cannot be increased by adding additional B-activated node. Therefore, under the Separated Threshold (SepT) model, the influence function $\sigma(\cdot)$ is monotone, but not submodular, as proved in [6] by a counting example.

Summary

In this section ,we provide an overview of diffusion models that have been extensively used in studying social influence: threshold models [26, 27, 28, 43, 48, 6], cascading models [21, 22, 9, 8] and epidemic models [29, 36]. Table 1.1 summarizes the the activation condition, model properties and applications of each model. And with this framework in place, we move on to the next section which focuses on the algorithmic results of the influence maximization problem.

1.5 Influence Maximization and Approximation Algorithms

1.5.1 Influence Maximization

A social network is the graph of relationships and interactions within a group of individuals that plays a fundamental role as a medium for the spread of information, ideas, and influence among its members. *Influence Maximization(IM)* is the problem of choosing the most potential of individuals in a network to spread out information in order to trigger the widespread adoption of a product. Domingos and Richardson [17] model the problem as a Markov random field. Kempe et al. [27, 28] assume a fixed marketing budget sufficient to target k individuals and study the problem of finding the optimal k individuals in the network to target. This problem has applications in viral marketing, where a company may wish to spread the rumor of a new product via the most influential individuals in popular social networks. With online social networking sites such as Facebook, LinkedIn, Myspace, etc. attracting hundreds of millions of people, online social networks are also viewed as important platforms for effective viral marketing practice. This further motivates the research community to conduct extensive studies on various aspects of the influence maximization problem.

1.5.2 Approximation Algorithm

We are now in a position to choose a good initial set of nodes to target in the context of the above models. Based on the basic models we introduced above, in this section, we introduce the hardness of influence maximization problems on above models, and prove the influence maximization problem with budget k under both of LT and IC models is NP-hard.

In addition, the influence function $f(\cdot)$ is submodular and monotone increasing. Exploiting these properties, Kempe et al. [27] present a simple greedy algorithm that approximates the problem with the ratio of $1-1/e-\varepsilon$ for any $\varepsilon > 0$. However, the running time of worst-case of the naive greedy algorithm is $O(n^2(m+n))$, which is prohibitive for large-scale networks. Thus, considerable work has been done to improve it. In this section, we demonstrate recent algorithmic study such as CELF[34],

Table 1.1: Models Listing and Comparisons

Name	*Activation Condition*	*Application*	*Property*	*Reference*
LT	$\sum_{u\in N^a(v)} w_{u,v} \geq \theta_v$	Collective behavior, spreading rumors and diseases	The objective $\sigma(\cdot)$ is submodular, and IM is NP-hard	[30]
MT	$\theta_v = \frac{1}{2}d(v)$	Voting system, distributed computing	IM is NP-hard	[47, 50, 4]
ST	$\sum_{u\in N^a(v)} w_{u,v} \geq \theta_v$ where θ_v is a small constant		$\theta_v = 1$, select an arbitrary node in each connected component; $\theta_v \geq 2$, IM is NP-hard	[20, 51]
UT	$\sum_{u\in N^a(v)} w_{u,v} \geq \theta_v$ where $\theta_v = d(v)$	Network security and vulnerability	IM is NP-hard, 2-approximation algorithm	[47]
WT	$Pr[v \in \Phi_A^t \| v \in \Phi^t \setminus \Phi^{t-1}] = \frac{\sum_{u\in\Phi_A^t} w_{u,v}}{\sum_{u\in\Phi^t} w_{u,v}}$	Deal with two competitive influence	IM is NP-hard. $\sigma(\cdot)$ is neither monotone nor submodular	[30]
SepT	$i-active: \sum_{u\in N^a(v)\cap\Phi_A^{t-1}} w^i_{u,v} \geq \theta^i_v$	Network with competitive sources	IM is NP-hard. $\sigma(\cdot)$ is monotone, but not submodular	[30, 6]
LT-C	$\theta_v = \frac{\sum_{u\in N^a(v)} w_{u,v}(r_{u,i}-r_{min})}{r_{max}-r_{min}}$	Distinguish product adoption from influenced users	NP-hard, $\sigma(\cdot)$ is monotone and submodular	[4]
OC	$\sum_{u\in N^a(v)} \geq \theta_v$	Incorporate user opinions	IM is NP, $\sigma(\cdot)$ is neither monotone nor submodular	[56]
IC	$\Pi_{i=1}^k(1-p_u(v_i\|S\cup T_i)) = \Pi_{i=1}^k(1-p_u(v_i'\|S\cup T_i'))$	Collective behavior, promote new products	The objective $\sigma(\cdot)$ is submodular, and IM is NP-hard	[27]
DC	$p_u(v\|S) \leq p_u(v\|T)$	Collective behavior, spreading information	IC is a special case of DC, and the objective $\sigma(\cdot)$ is submodular, and IM is NP-hard	[28]
IC-N	$\Pi_{i=1}^k(1-p_u(v_i\|S\cup T_i)) = \Pi_{i=1}^k(1-p_u(v_i'\|S\cup T_i'))$	Incorporate negative opinions	With probability q, each newly active node become positive and with probability $1-q$	[9]
SIR		Transmission of contagious disease	$\frac{dS}{dt} = -\beta SI, \frac{dI}{dt} = \beta SI - \gamma I, \frac{dR}{dt} = \gamma I$	[30, 29]

CELF++ [24], Simpath [25] and LDAG [11] algorithms, which can obtain high scalability for influence maximization problems.

1.5.2.1 Greedy Algorithm

Algorithm 1: Greedy Algorithm

Input: G, k, f
Output: Seed set S
initialize $S \leftarrow \emptyset$;
while $|S| \leq k$ **do**
  select $u \leftarrow argmax_{w \in V \setminus S}(f(S \cup \{w\}) - f(S))$;
  $S \leftarrow S \cup \{u\}$;
end
return S ;

Following the definition in Section 1.3.1.1, we now provide the definitions and notations as follows. An influence graph is a weighted graph $G = (V, E, w)$ with a weight function w, where V is a set of n nodes and $E \subseteq V \times V$ is a set of m directed edges. And the weight function $w: V \times V \rightarrow [0,1]$ holds that $w(u,v) = 0$ if and only if $(u,v) \notin E$, and $\sum_{u \in N(v)} w(u,v) = 0$ where $N(v)$ means that u is the neighbor of v. In the LT model, when given a seed set $S \subseteq V$, influence cascades in graph G in discrete steps. At time t, each inactive node v becomes active if the weighted number of its activated in neighbors reaches its threshold, i.e. $\sum_{u \in N^a(v)} w_{u,v} \geq \theta_v$, where $N^a(v)$ denotes the set of active neighbors of v. The process stops at a step t when the seed set becomes empty. Each activated node remains active, and the process terminates if no more activation is possible.

The influence maximization problem under the linear threshold model is, when given the influence graph G and an integer k, finding a seed set S of size k such that its influence spread $\sigma_L(S)$ is the maximum where we call $\sigma_L(S)$ the *influence spread* of seed set S [10]. It is shown in [27] that finding the optimal solution is NP-hard, but because σ_L is monotone and submodular, a greedy algorithm has a constant approximation ratio. A generic greedy algorithm for any set function f is shown as Algorithm 1.

Algorithm 1 simply executes in k rounds, and in each round a new entry that gives the largest marginal gain in f will be selected. It is shown in [41] that for any monotone and submodular set function f with $f(\emptyset) = 0$, the greedy algorithm has an approximation ratio $f(S)/f(S*) \geq 1 - 1/e$, where S is the output of the greedy algorithm and $S*$ is the optimal solution. However, the generic greedy algorithm requires the evaluation of $f(S)$. In the context of influence maximization, the exact computation of $\sigma_L(S)$ was left as an open problem in [27] and was later proved that the exact computation of $\sigma_L(S)$ is #P-hard in [10].

The running time of worst-case of this naive greedy algorithm is $O(n^2(m+n))$, which is prohibitive for large-scale networks. Thus, considerable work has been done to improve it. We will introduce them in the following several subsections.

1.5.2.2 CELF Selection Algorithm

Relatively little work has been done on improving the quadratic nature of the greedy algorithm. The most notable work is [34], where submodularity is exploited to develop an efficient algorithm called Cost-Effective Lazy Forward (CELF) selection algorithm, based on a lazy-forward optimization in selecting seeds. The idea is that marginal gain of a node in the current iteration cannot be better than its marginal gain in the previous iterations. CELF maintains a table $< u, \Delta_u(S) >$ sorted on $\Delta_u(S)$ in decreasing order, where S is the current seed set and $\Delta_u(S)$ is the marginal gain of u w.r.t S. The $\Delta_u(S)$ here corresponds to $\sigma_L(S)$ in the previous sub section. $\sigma_L(S)$ is re-evaluated only for the top node at each step and the table is resorted when only it is necessary. If a node remains at the top, it will be picked as the next seed. In real implementation, a heap Q is employed to represent the priority of each node and maintain the sorted table information.

In [34], the authors empirically shows that CELF dramatically improves the efficiency of the greedy algorithm. Algorithm 2 shows the skeleton of CELF algorithm. In the algorithm, $\sigma_m(S)$ denotes the expected influence spread of seed set S under the propagation model m (like IC or LT). This m could be omitted if there is no confusion in the context. As clearly explained in [23], the optimization works as follows. Maintain a heap Q with nodes corresponding to users in the network G.

The node of Q corresponding to user u stores a tuple of the form $< u.mg, u.round >$ where $u.mg = \sigma_m(S \cup \{u\}) - \sigma_m(S)$ represents the marginal gain of u w.r.t. the current seed set S while $u.round$ is the iteration number when $u.mg$ was last updated. In the first iteration, marginal gains of each node is computed and added to Q in decreasing order of marginal gains (The first *for* loop). Later, in each iteration, look at the top node u in Q and see if its marginal gain was last computed in the current iteration (using the *round* attribute). If yes, then, due to submodularity, u must be the node that provides maximum marginal gain in the current iteration, hence, it is picked as the next seed. Otherwise, recompute the marginal gain of u, update its round flag and reinsert into Q such that the order of marginal gains is maintained. This process is realized in the *while* loop in the algorithm.

It is easy to see that this optimization avoids the recomputation of marginal gains of all the nodes in any iteration, except the first one. Therefore, from the experimental results, the CELF optimization leads to a 700 times speedup in the greedy algorithm shown in [34].

1.5.2.3 CELF++ Algorithm

In [24], Goyal et al. introduce CELF++ that further optimized CELF by exploiting submodularity. Algorithm 3 describes the CELF++ algorithm. The setup is similar to CELF: $\sigma(S)$ is used to denote the spread of seed set S. A heap Q with nodes corresponding to users in the network G.

Algorithm 2: Greedy Algorithm optimized with CELF

Input: G, k, σ_m
Output: Seed set S

initialize $S \leftarrow \emptyset, Q \leftarrow \emptyset$;
for *each* $u \in V$ **do**
  $u.mg = \sigma_m(\{u\})$;
  $u.round = 0$;
  Add u to Q in decreasing order of mg.
end
while $|S| \leq k$ **do**
  $u \leftarrow$ root element in Q ;
  if $u.round == |S|$ **then**
    $S \leftarrow S \cup \{u\}$;
    $Q \leftarrow Q - \{u\}$;
  end
  else
    $u.mg = \sigma_m(S \cup \{u\}) - \sigma_m(S)$;
    $u.round = |S|$;
    Reinsert u into Q and heapify.
  end
end
return S ;

The improvement is that instead of tuple of two attributes, they offer that the node of Q corresponding to user u stores a tuple of the form $< u.mg1, u.prev\_best, u.mg2, u.flag >$. Here $u.mg1 = \Delta_u(S)$, the marginal gain of u w.r.t. the current seed set S; $u.prev\_best$ is the node that has the maximum marginal gain among all the users examined in the current iteration, before user u; $u.mag2 = \Delta_u(S \cup \{prev\_best\})$, and $u.flag$ is the iteration number when $u.mg1$ was last updated.

The central idea is that if the node picked in the last iteration is still at the root of the heap, they don't need to recompute the marginal gains. This does save a lot of computations. It is important to note that in addition to computing $\Delta_u(S)$, it is not necessary to compute $\Delta_u(S \cup \{prev\_best\})$ from scratch. In other words, the algorithm can be implemented in an efficient manner such that both $\Delta_u(S)$ and $\Delta_u(S \cup \{prev\_best\})$ are evaluated simultaneously in a single iteration of Monte Carlo simulation. In that sense, the extra overhead is relatively insignificant compared to the huge run time gains they can achieve, as shown in the experimental results [24], leading to an improvement of CELF by 17-61%.

Algorithm 3 uses the variable S to denote the current seed set, *last_seed* to track the id of last seed user picked by the algorithm, and *cur_best* to track the user having the maximum marginal gain w.r.t. S over all users examined in the current iteration. The algorithm starts by building the heap Q initially. Then, it continues to select

Algorithm 3: Greedy algorithm optimized with CELF++

Input: G, k, σ_m
Output: Seed set S

initialize $S \leftarrow \emptyset, Q \leftarrow \emptyset, last\_seed \leftarrow NULL, cur\_best \leftarrow NULL$;
for *each* $u \in V$ **do**
  $u.mg1 \leftarrow \sigma(\{u\})$;
  $u.prev\_best \leftarrow cur\_best$;
  $u.mg2 \leftarrow \Delta_u\{cur\_best\}$;
  $u.flag \leftarrow 0$;
  $Q \leftarrow Q \cup \{u\}$;
  Update *cur_best* based on $u.mg1$;
end
while $|S| \leq k$ **do**
  $u \leftarrow$ root element in Q ;
  if $u.flag == |S|$ **then**
    $S \leftarrow S \cup \{u\}$;
    $Q \leftarrow Q - \{u\}$;
    $last\_seed \leftarrow u$;
    $cur\_best \leftarrow NULL$;
    Continue ;
  end
  else if $u.prev\_best == last\_seed$ ***and*** $u.flag == |S| - 1$ **then**
    $u.mg1 \leftarrow u.mg2$;
  end
  else
    $u.mg1 \leftarrow \Delta_u(S)$;
    $u.prev\_best \leftarrow cur\_best$;
    $u.mg2 \leftarrow \Delta_u(S \cup \{cur\_best\})$;
  end
  $u.flag = |S|$;
  Update *cur_best* ;
  Heapify Q ;
end
return S ;

seeds until the budget k is reached. The optimization of CELF++ comes from where they update $u.mg1$ without recomputing the marginal gain. Clearly, this can be done since $u.mg2$ has already been computed efficiently w.r.t. the last seed node picked. If none of the above cases applies, they recompute the marginal gain of u. From the experiments carried out in [24] one can note that although CELF++ maintains a larger data structure to store the look-ahead marginal gains of each node, the increase of the memory consumption is insignificant while the optimization on performance w.r.t. time is increased from CELF by 17-61%.

1.5.2.4 SPM and SP1M

The *Shortest-Path Model (SPM)* and *SP1 Model (SP1M)* were developed by Kimura et al. in [31]. These two models are special cases of the IC (independent cascade) model. In SPM, each node v has the chance to become active only at step $t = d(A, v)$. In other words, each node is activated only through the shortest paths from an initial active set. Namely, SPM is a special type of the ICM where only the most efficient information spread can occur. And SP1M, which slightly generalize SPM, instead considers the top-2 shortest paths from u to v.

The idea is that the majority of the influence flows through shortest paths. For these models, the influence $\sigma(A)$ of each target set A can be exactly and efficiently computed, and the provable performance guarantee for the natural greedy algorithm can be obtained. In [31], the approximation ratio is guaranteed as $\sigma(B_k) \geq (1 - 1/e)\sigma(A_k^*)$.

The experimental results show that SP1M outer-performs SPM. However, a critical issue with this approach is that it ignores the influence probabilities among users. Only considering the shortest paths are not enough.

1.5.2.5 Maximum Influence Paths

From the above contribution in SPM and SP1M, Chen et al. [10] extended this idea by considering Maximum Influence Paths (MIP) instead of shortest paths. A maximum influence path between a pair of nodes (u, v) is the path with the maximum propagation probability from u to v. The main idea of this heuristic scheme is to use local arborescence structures of each node to approximate the influence propagation.

The maximum influence paths between every pair of nodes in the network can be computed by the Dijkstra shortest-path algorithm. Then we ignore the MIPs with probability smaller than a influence threshold θ, this can help us effectively restrict influence to a local region. Next, we union the MIPs beginning or ending at each node into a arborescence structures, which represent the local influence regions of each node. When considering the influence propagation through these local arborescences, the diffusion model refers to the Maximum Influence Arborescence (MIA) model [10].

It is shown in [10] that the influence spread in the MIA model is submodular (i.e. having a diminishing marginal return property), and thus the simple greedy algorithm that selects one node in each round with the maximum marginal influence spread can

guarantee an influence spread within $(1-1/e)$ of the optimal solution in the MIA model, while any higher ratio approximation is NP-hard.

The complete greedy algorithm for the basic MIA model is presented in Algorithm 4. Before the process was introduced, the authors in [10] defined several methods. The maximum influence in-arborescence of a node $v \in V$ is defined as $MIIA(v,\theta) = \cup_{u \in V, pp(MIP_G(u,v)) \geq \theta} MIP_G(u,v)$. And the maximum influence out-arborescence $MIOA(v,\theta) = \cup_{u \in V, pp(MIP_G(v,u)) \geq \theta} MIP_G(v,u)$. Further, let the activation probability of any node u in $MIIA(v,\theta)$, denoted as $ap(u,S,MIIA(v,\theta))$, be the probability that u is activated when the seed set is S and influence is propagated in $MIIA(v,\theta)$. Due to the limit of pages, we would not discuss these methods, while one can easily find the definitions and details in [10].

The whole MIA algorithm works as follows. First, it evaluates the incremental influence spread $IncInf(u)$ for any node u when the current seed set is empty. The evaluation is described using the linear coefficients $\alpha(v,u)$. Second, the algorithm updates the incremental influences whenever a new seed is selected. Suppose u is selected as the new seed in an iteration, the influence of u in the MIA model only reaches nodes in $MIOA(u,\theta)$. Thus the incremental influence spread $IncInf(w)$ for some w needs to be updated if and only if w is in $MIIA(v,\theta)$ for some $v \in MIOA(u,\theta)$. This means that the update process is relatively local to u. The update is done by first subtracting $\alpha(v,w) \cdot (1 - ap(w,S,MIIA(v,\theta)))$ before adding u into the seed set, and then adding u into the seed set outside the loop. Recompute the $ap(w,S,MIIA(v,\theta))$ and $\alpha(v,w)$ under the new seed set, and add $\alpha(v,w) \cdot (1 - ap(w,S,MIIA(v,\theta)))$ into $IncInf(w)$.

The authors later proposed an extension model *prefix excluding MIA (PMIA)*. Intuitively, in the PMIA model, the seeds have an order. For any given seed s, its maximum influence paths to other nodes should avoid all seeds in the prefix before s. The major technical difference is the definition of the maximum influence in(out)-arborescence for the PMIA model, especially if one would like to design an efficient greedy algorithm in the framework of Algorithm 4. From experiments all four real networks with different scales, the authors argue that their algorithms are scalable and the running time is efficient. However, these heuristics would not perform well on high influence graphs, as pointed by [23], that is, when the influence probabilities through links are large.

Wang et al. [54] proposed an alternative approach. The focus on their study was on IC model. They argue that most of the diffusion happens only in small communities, even though the overall networks are huge. Taking this as an intuition, they first split the network in communities, and then using a greedy dynamic programming algorithm to select seed nodes. To compute the marginal gain of a prospective seed node, they restrict the influence spread to the community to which the node belongs.

1.5.2.6 SIMPATH

SIMPATH, proposed by Goyal et al. in [25], is an efficient and effective algorithm for influence maximization problem under the linear threshold model. According to the experiments in [25], SIMPATH consistently outperforms the state of the art w.r.t.

Algorithm 4: Greedy algorithm optimized with MIA

Input: G, k, θ
Output: Seed set S

initialize $S \leftarrow \emptyset$, $IncInf(v) \leftarrow 0$for each node $v \in V$;
for *each node* $v \in V$ **do**
  compute $MIIA(v,\theta)$ and $MIOA(v,\theta)$;
  set $ap(u,S,MIIA(v,\theta)) = 0, \forall u \in MIIA(v,\theta)$;
  compute $\alpha(v,u), \forall u \in MIIA(v,\theta)$;
  for *each node* $u \in MIIA(v,\theta)$ **do**
    $IncInf(u) += \alpha(v,u) \cdot (1 - ap(u,S,MIIA(v,\theta)))$;
  end
end
while $|S| \leq k$ **do**
  pick $u = argmax_{v \in V \setminus S}\{IncInf(v)\}$;
  /* update incremental influence spreads */ ;
  for $v \in MIOA(u,\theta) \setminus S$ **do**
    /* subtract previous incremental influence */ ;
    for $w \in MIIA(v,\theta) \setminus S$ **do**
      $IncInf(w) -= \alpha(v,w) \cdot (1 - ap(w,S,MIIA(v,\theta)))$;
    end
  end
  $S = S \cup \{u\}$;
  for $v \in MIOA(u,\theta)$ **do**
    compute $ap(w,S,MIIA(v,\theta)), \forall w \in MIIA(v,\theta)$;
    compute $\alpha(v,w), \forall w \in MIIA(v,\theta)$;
    for $w \in MIIA(v,\theta) \setminus S$ **do**
      $IncInf(w) += \alpha(v,w) \cdot (1 - ap(w,S,MIIA(v,\theta)))$;
    end
  end
end
return S ;

running time, memory consumption and the quality of the seed set chosen, measured in terms of expected influence spread achieved.

Algorithm 5: SIMPATH

Input: $G = (V, E, b), k, \delta, l$
Output: Seed set S

Find the vertex cover C of input graph G. ;
for *each* $u \in C$ **do**
    $U \leftarrow (V - C) \cap N^{in}(u)$;
    Compute $\sigma(u)$ and $\sigma^{V-v}(u), \forall v \in U$ in a single call to the $SIMPATH - SPREAD(u, \delta, U)$;
    Add u to CELF queue. ;
end
for *each* $v \in V - C$ **do**
    Compute $\sigma(v)$;
    Add v to CELF queue ;
end
$S \leftarrow \emptyset$, $spd \leftarrow 0$;
while $|S| \leq k$ **do**
    $U \leftarrow$top-l nodes in CELF queue Compute $\sigma^{V-x}(S)$, $\forall x \in U$, in a single call to the $SIMPATH - SPREAD(u, \delta, U)$;
    for *each* $x \in U$ **do**
        if *x is previously examined in the current iteration* **then**
            $S \leftarrow S + x$;
            Update spd ;
            Remove x from CELF queue, break out of the loop;
        end
        Call $BACKTRACK(x, \delta, V - S, \emptyset)$ to compute $\sigma^{V-S}(x)$. ;
        Compute $\sigma(S + x)$. ;
        Compute marginal gain of u as $\sigma(S + x) - spd$. ;
        Re-insert u in CELF queue such that its order is maintained. ;
    end
end
return S ;

SIMPATH builds on the CELF optimization that iteratively selects seeds in a lazy forward manner. However, instead of using expensive MC simulations to estimate the spread, it is shown in [25] that under the LT model, the spread can be computed by enumerating the simple paths starting from the seed nodes. It is known that the problem of enumerating simple paths is #P-hard [53]. However, the majority of the influence flows within a small neighborhood, since probabilities of paths

diminish rapidly as they get longer. Thus, the spread can be computed accurately by enumerating paths within a small neighborhood. In addition to the Simpath-Spread algorithm used by SIMPATH, two other optimizations to reduce the number of spread estimation calls in SIMPATH. The first one, Vertex Cover Optimization, addresses a key weakness of the simple greedy algorithm: The spread of a node can be computed directly using the spread of its out-neighbors. Thus, in the first iteration, a vertex cover of the graph is constructed and the spread only for these nodes using the spread estimation procedure is obtained. The spread of the rest of the nodes is derived from this. This significantly reduces the running time of the first iteration. Second, they observe that as the size of the seed set grows in subsequent iterations, the spread estimation process slows down considerably. They provide the optimization called Look Ahead Optimization which addresses this issue and keeps the running time of subsequent iterations small. These three inventions are quite helpful for speeding up the SIMPATH algorithm, one can find details about these in [25], and we will not discuss about them but rather present the complete algorithm in Algorithm 5.

The whole algorithm is presented in Algorithm 5. First, the algorithm find a vertex cover C, then for every node $u \in C$, its spread is computed on required subgraphs needed for the optimization. This is done in a single call to $SIMPATH - SPREAD$. Next, for the nodes that are not in the vertex cover, the spread is computed. The CELF queue is built accordingly, sorted in the decreasing order of marginal gains. Next, by using Look Ahead Optimization, the algorithm selects the seed set in a lazy forward fashion. The spread of the seed set S is maintained using the variable spd. At a time, they take a batch of top-l nodes, call it U, from the CELF queue. In a single call to $SIMPATH - SPREAD$, the spread of S is computed on required subgraphs needed for the optimization. For a node $x \in U$, if it is processed before in the same iteration, then it is added in the seed set as it implies that x has the maximum marginal gain w.r.t. S. Recall that the CELF queue is maintained in decreasing order of the marginal gains and thus, no other node can have a larger marginal gain [23]. Ifx is not seen before, its marginal gain needs to be recomputed, then CELF queue is updated accordingly.

1.5.2.7 VirAds

In recent studies, researchers have discovered that the propagation in a social network often fades quickly within only few hops from the sources, counteracting the assumption on the self-perpetuating of influence considered in some literature. Dinh et al. [13] investigated the cost-effective massive, and fast propagation (CFM) problem and proposed an algorithm, VirAds, to minimize the seeding cost and to tackle the problem on large-scale networks.

This scalable algorithm is shown as Algorithm 6, where r_v is the round in which v is activated, $n_v^{(e)}$ represents the number of new active edges after adding v into the seeding and $n_v^{(a)}$ refers to the number of extra active neighbors v needs in order to activate v. Besides, $r_v^{(i)}$ is the number of activated neighbors of v up to round i where $i = 1...d$. Generally, VirAds algorithm favors the vertex which can activate the most number of edges. This could distinguish between good and bad seeds. In early stages,

Algorithm 6: VirAds - Viral Advertising in OSNs

Input: $G = (V,E), 0 \leq \rho \leq 1, d \in \mathbb{N}^+$
Output: A small d-seeding

$n_v^e \leftarrow d(v), n_v^a \leftarrow \rho \cdot d(v), r_v \leftarrow d+1, v \in V$;
$r_v^i = 0, i = 0..d, P \leftarrow \emptyset$;
while *there exist inactive vertices* **do**
  while $u \neq argmax_{v \notin P}\{n_v^e + n_v^a\}$ **do**
    $u \leftarrow argmax_{v \notin P}\{n_v^e + n_v^a\}$;
    Recompute n_v^e as the number of new active edges after adding u. ;
  end
  $P \leftarrow P \cup \{u\}$;
  Initialize a queue: $Q \leftarrow \{(u, r_v)\}$;
  $r_u \leftarrow 0$;
  for *each* $x \in N(u)$ **do**
    $n_x^{(a)} \leftarrow max\{n_x^{(a)}\}$;
  end
  while $Q \neq \emptyset$ **do**
    $(t, \tilde{r}_t) \leftarrow Q.pop()$;
    for *each* $w \in N(t)$ **do**
      for *each* $i = r_t \rightarrow min\{\tilde{r}_t - 1, r_w - 2\}$ **do**
        $r_w^{(i)} = r_w^{(i)} + 1$;
        if $(r_w^{(i)} \geq \rho \cdot d_w) \wedge (r_w \geq d) \wedge (i+1 < d)$ **then**
          for *each* $x \in N(w)$ **do**
            $n_x^{(a)} \leftarrow max\{n_x^{(a)} - 1, 0\}$;
          end
          $r_w = i+1$;
          if $w \notin Q$ **then**
            $Q.push((w, r_w))$;
          end
        end
      end
    end
  end
end
return P ;

the algorithm behaves similar to the degree-based heuristics that favors vertices with high degree. However, after a certain number of vertices have been selected, VirAds will make the selection based on the information within d-hop neighbor around the considered vertices, which is different from degree-based heuristic that considers only one-hop neighborhoodship.

Given those measures, VirAds selects in each step the vertex u with the highest $effectiveness$ which is defined as $n_u^{(e)} + n_u^{(a)}$. After that, the algorithm needs to update the measures for all the remaining vertices.

It is introduced in [13] that the cost-effective, massive and fast propagation problem (CFM) can be easily shown to be NP-hard by a reduction from the set cover problem. It is also proved that there is unlikely an approximation algorithm with factor less than $O(logn)$. However, if we assume the network is power-law, their algorithm is an approximation algorithm for this problem with a constant factor.

1.6 Conclusion

Social networks are graphs of individuals and their relationships [5], such as friendships, collaborations, or advice seeking relationships. With the increasing popularity of social networks services, more and more people communicate with each other through such networks. This survey mainly conveys a framework for studying the information diffusion problems and their approximations as well as optimizations. It provides with the readers a number of interesting models, and wise algorithms on social networks. However, these techniques and models only form the foundation and the basis for further research, there are many open questions that need to be uncovered.

As we have went through, novel and interesting questions thrown out by the initial work from Domingos and Richardson [17, 47], inspires Kempe et al. [28, 27], Mossel and Roch [39] and many others to develop a solid theoretical foundation of literature resources on the influence maximization problem. The main challenge now is to find solutions that are applicable in real viral marketing environment. Working towards various models and algorithms, with the comprehensive experiments, researchers are trying to find a way that could really gives the satisfying result without requiring too much data load or making unrealistic independence assumptions. In order to achieve this goal and to determine the real applicability of the existing approaches, more wise designs, and empirical studies are needed, and the test of the approximation techniques are also required.

The more recent work of Leskovec et al. [55] gives us insight in modeling the diffusion through implicit networks, in which the underlying network structure is unknown, all the predicting of activation and influence spread is focusing on a global view. Furthermore, in [40], Myers et al. propose a new model which take into account the external influence from outside of the network. Inspired by those works, for future works, it would be interesting to relax the assumption of uniform influence inside of the network to seek better strategy to maximize the influence. Furthermore,

in contrast to the influence maximization problem, for misinformation or computer viruses spreading in the networks, how to efficiently prevent the audience from getting infected is also very attractive to us. Formulating and solving those problems with more practical model and efficient algorithms is a fascinating challenge with great potential.

References

[1] H. K. Ahn, S. W. Cheng, O. Cheong, M. Golin, and R. Oostrum. Competitive facility location: the voronoi game. *Theoretical Computer Science*, 310(1-3):457–467, 2004.

[2] A. Banerjee, N. Pathak, and J. Srivastava. A generalized linear threshold model for multiple cascades. In *In ICDM*, pages 965–970, 2010.

[3] N. Berger, C. Borgs, J. T. Chayes, and A. Saberi. On the spread of viruses on the internet. In *In Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithm (SODA)*, 2005.

[4] S. Bhagat, A. Goyal, and L. V. Lakshmanan. Maximizing product adoption in social networks. In *Proceedings of the fifth ACM international conference on Web search and data mining, WSDM*, pages 603–612, 2012.

[5] S. Bharathi, D. Kempe, and M. Salek. Competitive influence maximization in social networks. In *In WINE*, pages 306–311, 2007.

[6] A. Borodin, Y. Filmus, and J. Oren. Threshold models for competitive influence in social networks. In *Proceedings of the 6th international conference on Internet and network economics*, WINE'10, 2010.

[7] F. Brauer and C. Castillo-Chvez. Mathematical models in population biology and epidemiology. *Springer*, 2001.

[8] T. Carnes, R. Nagarajan, S. M. Wild, and A. V. Zuylen. Maximizing influence in a competitive social network: a follower's perspective. In *Proceedings of the ninth international conference on Electronic commerce*, pages 351–360. ACM, 2007.

[9] W. Chen, A. Collins, R. Cummings, T. Ke, Z. Liu, D. Rincon, X. Sun, Y. Wang, W. Wei, and Y. Yuan. Influence maximization in social networks when negative opinions may emerge and propagate. In *In Proceedings of the 11th SIAM International Conference on Data Mining*, 2011.

[10] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *16th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'10*, pages 1029–1038, New York, NY, USA, 2010.

[11] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. In *2010 IEEE International Conference on Data Mining, ICDM'10*, pages 88–97, Washington, DC, USA, 2010.

[12] D. J. Daley and J. Gani. Epidemic modeling: An introduction. *NY: Cambridge University Press*, 2005.

[13] T. N. Dinh, D. T. Nguyen, and M. T. Thai. Cheap, easy and massively effective viral marketing in social networks: Truth or fiction? In *ACM Conference on Hypertext and Social Media (Hypertext)*, 2012.

[14] G. Dobson and U. S. Karmarkar. Competitive location on a network. *European Journal of Operational Research*, 35(4):565–574, 1987.

[15] P. S. Dodds and D. J. Watts. Universal behavior in a generalized model of contagion. *Physical Review Letters*, 92, 2004.

[16] P. Domingos. Mining social networks for viral marketing. In *IEEE Intelligent Systems*, 2005.

[17] P. Domingos and M. Richardson. Mining the network value of customers. In *Seventh ACM SIGKDD international conference on Knowledge discovery and data mining, KDD 01*, pages 57–66, New York, NY, USA, 2001.

[18] P. A. Dreyer. Applications and variations of domination in graphs. *Ph.D. Thesis, Rutgers University*, 2000.

[19] R. Durrett. Lecture notes on particle systems and percolation. *Wadsworth Publishing*, 1988.

[20] H. Eiselt and G. Laporte. Competitive spatial models. *European Journal of Operational Research*, 39:231–242, 1989.

[21] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 3(211-223), 2001.

[22] J. Goldenberg, B. Libai, and E. Muller. Using complex systems analysis to advance marketing theory development. *Academy of Marketing Science Review*, 2001.

[23] A. Goyal. *Social Influence and its Applications*. PhD thesis, University of British Columbia, 2005-2013.

[24] A. Goyal, W. Lu, and L. V. S. Lakshmanan. Celf++: optimizing the greedy algorithm for influence maximization in social networks. In *20th international conference companion on World wide web, WWW'11*, New York, NY, USA,, 2011.

[25] A. Goyal, W. Lu, and L. V. S. Lakshmanan. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *2011 IEEE 11th International Conference on Data Mining, ICDM'11*, pages 211–220, Washington, DC, USA, 2011.

[26] M. S. Granovetter. Threshold models of collective behavior. *The American Journal of Sociology*, 83(6):1420–1443, 1978.

[27] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *Ninth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD 03*, pages 137–146, New York, NY, USA, 2003.

[28] D. Kempe, J. Kleinberg, and E. Tardos. Influential nodes in a diffusion model for social networks. In *32nd international conference on Automata, Languages and Programming, ICALP05*, pages 137–146, Berlin, Heidelberg, 2005. Springer-Verlag.

[29] M. Kermack. Contributions to the mathematical theory of epidemics. In *Royal Society of Edinburgh. Section A. Mathematics*, volume 115, 1972.

[30] W. Kermack and A. McKendrick. A contribution to the mathematical theory of epidemics. *In Royal Society of London*, 115:700–721, 1927.

[31] M. Kimura and K. Saito. Approximate solutions for the influence maximization problem in a social network. *Knowledge-Based Intelligent Information and Engineering Systems*, 4252(Lecture Notes in Computer Science):937–944, 2006.

[32] J. Leskovec, L. Adamic, and B. Huberman. The dynamics of viral marketing. In *In Proceedings of the Seventh ACM Conference on Electronic Commerce (EC)*, 2006.

[33] J. Leskovec, L. A. Adamic, and B. A. Huberman. The dynamics of viral marketing. In *Proceedings of the 7th ACM Conference on Electronic Commerce*, EC '06, pages 228–237. ACM, 2006.

[34] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *In Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 420–429, 2007.

[35] J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs. In *In SIAM International Conference on Data Mining (SDM)*, 2006.

[36] J. Leskovec, M. Mcglohon, C. Faloutsos, N. Glance, and M. Hurst. Cascading behavior in large blog graphs. In *In SDM*, 2007.

[37] J. Leskovec, A. Singh, and J. Kleinberg. Patterns of influence in a recommendation network. In *In Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 2006.

[38] T. M. Liggett. Interacting particle systems. *Springer*, 1985.

[39] E. Mossel and S. Roch. On the submodularity of influence in social networks. In *In Proceedings of the 39th ACM Symposium on Theory of Computing (STOC)*, 2007.

[40] S. Myers, C. Zhu, and J. Leskovec. Information diffusion and external influence in networks. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2012.

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

[42] D. T. Nguyen, S. Das, and M. T. Thai. Influence maximization in multiple online social networks. In *IEEE Globecome 2013*, 2013. (Accepted).

[43] C. Ning. On the approximability of influence in social networks. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, 2008.

[44] D. Peleg. Local majority voting, small coalitions and controlling monopolies in graphs: A review. In *in Proceedings of the 3rd Colloquium on Structural Information and Communication Complexity*, pages 170–179, 1996.

[45] D. Peleg. Size bounds for dynamic monopolies. *Discrete Applied Mathematics*, 86:263–273, 1998.

[46] L. Rashotte. Social influence. *In the blackwell encyclopedia of sociology*, IX:4426–4429, 2006.

[47] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *In Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2002.

[48] F. S. Roberts. Graph-theoretical problems arising from defending against bioterrorism and controllingthe spread of fires. In *DIMACS/DIMATIA/Renyi Combinatorial Challenges Conference*, 2006.

[49] E. M. Rogers. Diffusion of innovations. *New York: Free Press*, 1962.

[50] P. Rozin and E. B. Royzman. Negativity bias, negativity dominance and contagion. *Personality and Social Psychology Review*, 5(4):296–320, 2001.

[51] T. Schelling. Micromotives and macrobehavior. *Norton*, 1978.

[52] Y. Shen, H. Zhang, and M. T. Thai. Interest-matching information propagation in multiple online social networks. In *CIKM'12*, 2012.

[53] L. G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(3):410–421, 1979.

[54] Y. Wang, G. Cong, G. Song, and K. Xie. Community-based greedy algorithm for mining top-k influential nodes in mobile social networks. In *16th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD'10*, pages 1039–1048, New York, NY, USA, 2010.

[55] J. Yang and J. Leskovec. Modeling information diffusion in networks. In *IEEE International Conference On Data Mining (ICDM)*, 2010.

[56] H. Zhang, T. N. Dinh, and M. T. Thai. Maximizing the spread of positive influence in online social networks. In *In ICDCS*, pages 317–326, 2013.