Computational Challenges in Adaptive Influence Maximization

Debashmita Poddar

A Proposal Submitted as Partial Fulfilment of the Requirements for the PhD Degree in Computer Science

Supervisors:
Prof. Gianlorenzo D'Angelo
Dr. Cosimo Vinci
Computer Science Department
Gran Sasso Science Institute
L'Aquila, Italy
27 Septemper 2019

Abstract

This proposal studies the *Influence Maximization* problem: given a budget k, the goal is to select a subset of k nodes in a network which can influence the maximum number of nodes participating in the network, according to some model for influence diffusion. There are two variations of the problem, namely *non-adaptive* and *adaptive*. In the non-adaptive version, the subset of all the nodes that are considered to be influential is selected before the beginning of the diffusion process. The adaptive version allows the selection of the influential nodes to be made one at a time, depending on the feedback received on the performance of the previously selected nodes.

However, most of the issues are studied in the non-adaptive setting, and nearly all the real-world networks are unpredictable. The proposal explores the variations of the *Adaptive Influence Maximization* problem and its connection to the marketing strategies using different diffusion models.

In this proposal, we discuss the *Adaptivity Gaps*, which give a measure of the efficiency of an optimal adaptive policy over an optimal non-adaptive one. Different feedback models and their generalized versions have been inquired. A list of open problems for the adaptive version of the Influence Maximization problem has been provided, which sketches the future works that need to be investigated. Finally, several applications can be seen as an Adaptive Influence Maximization problem and maybe designed according to some specifications.

.

Contents

1 Introduction

Most companies have resorted to social media for advertising their products. Social media plays a crucial role in spreading new ideas or innovations through its fundamental network structure consisting of individuals and their relationship with other individuals. The *Influence Maximization* problem states that a company, in order to maximize its profit, selects a set of users, also called *influencers*, who can promote the product in a market. In particular, the company has a specific budget k allocated for the influencers, who will be selected initially and given the product for free or on a discounted price.

However, since the marketing world is uncertain, sometimes it is feasible to choose one influencer at a time, and observe the status of the marketed product. Based on the feedback received from the initial batch of products sold, the company can decide to select a different influencer for marketing the product. The adaptive nature of this strategy gives birth to a new field of research called *Adaptive Influence Maximization* [1]. The main aim of Adaptive Influence Maximization remains the same as the original Influence Maximization problem; however, it has an induced flexibility to cater the marketing needs. The adaptive nature of the problem is more realistic, and it also gives more power to companies.

Various kinds of diffusion models that can be used as a prototype for solving the Influence Maximization problem are explained. A background of Influence Maximization problem is provided along with a greedy approximation algorithm achieving 63% of the optimal value since the problem is shown to be NP-hard [2]. This proposal details on the Adaptive Influence Maximization problem, and also elucidates on the different kinds of feedback models. The scope of this proposal is, (i) analyzing the different models in the adaptive setting, and (ii) finding a tight ratio between how efficient the adaptivity is when compared to non-adaptive policies. A lot of significance on the practicality of the adaptive models is provided by finding a middle ground between efficiency and effectiveness. A number of open problems are listed. Finally, the application section provides the different scenarios where Adaptive Influence maximization can be used to design the problem.

1.1 Social Graphs

Social networking sites have seen a significant boom in the past decade. Indeed, more and more people are taking part in the services offered by these sites. Some of the major networking sites with over a hundred million active users include *Facebook, Twitter, Instagram*, etc. A *social network* is a collection of *entities* which form *relationships* with other entities participating in that network.

Social Network Analysis (SNA) has emerged due to the increase of information propagation in the networking sites. SNA is the study of social networks using *collaboration graphs* in which the entities are modelled by the nodes of a graph and the relationships between the entities are modelled by the edges. Formally, a *Social Graph* $G = (V, E)$ is a directed graph, where V is the set of nodes representing the entities of the network, and E is the set of edges representing the relationship existing between those entities. SNA is the field of study between the intersections of *Graph Theory, Networking* and *Sociology*.

1.2 Viral Marketing

Marketing is the promotion of a product by a company to its customer base. Marketing takes into account the *word of mouth* effect, i.e., a customer who has used the products, can recommend it to her friends. *Viral Marketing (VM)* supports the same theory, and uses social networks as a platform for recommendations. The method follows the principle of contiguous viruses: if the flu infects one person, any person who comes in contact with that person gets infected by the flu with high probability.

In social networks, the company distributes some free or discounted products to specific individuals who are capable of marketing it. These individuals may recommend it to their neighbours, after using it, who in turn might recommend it to their neighbours, and so on, following an iterative diffusion process. The direct analogy to this is *Influencer Marketing*. Influencers are those particular individuals in the network, who have a significant number of followers in a specific niche, and can affect the decisions of the people following her. Companies who want to launch new products in the market generally give them to some influencers, particularly to those who are closer to the specific customer base for selling the considered products.

2 Literature Review

Milgram, in his classic paper, *The Small World Problem*, introduced for the first time the intuition behind Viral Marketing, and he brings the notion of transitive dependency among people in a social network. He states that when a person, denoted as Bob, is connected by a series of links to another person, denoted as Alice, there are at most five intermediate persons between Bob and Alice. Krackhardt et al. [3] states that a marketer can leverage her sales profit by knowing the topology of the underlying social network. They also introduced the concept of distributing free products to the influencers.

Domingos and Richardson [4] introduce some models about VM, and raise the following algorithmic question: in a vast connected network, if a seller markets a product, how should she choose a set of impactful buyers/nodes who can influence as many nodes as possible to buy that product? Indeed, the number of such buyers is directly proportional to the seller's profit. Kempe et al. [2], in their seminal work, presented the VM problem as a discrete optimization problem, called Influence Maximization problem, discussed in details in the upcoming sections.

2.1 Problem Definition

We consider the following dynamics, consisting in multiple time steps. Given a social graph $G = (V, E)$ as an input instance, we have a set $S \subseteq V$ of *active nodes*, called the *seed set*, which is responsible for the spread of information at the first time step. Furthermore, we have an underlying *diffusion model* M, determining how the set of active nodes, at each time step, influence some inactive nodes, that become active at the next time step. The *influence function* $\sigma_{G,M} : 2^V \to \mathbb{R}_{\geq 0}$ associates to each possible initial seed set S, the expected value $\sigma_{G,M}(S)$ of nodes which are active at the end of the process.

The *Influence Maximization (IM) Problem* can be formally defined as follows: given a budget of k users, a social graph G, and a diffusion model M, the goal is to select a seed set S^* of k users from V in order to maximize the influence function, i.e., a set S^* such that

$$\sigma_{G,M}(S^*) = \underset{S \subseteq V : |S| \leq k}{\arg\max}\, \sigma_{G,M}(S). \tag{1}$$

Initially, all the nodes present in the seed set S are active. As the diffusion of information propagates through the nodes, an inactive node v will be activated by its active neighbours eventually, according to the considered diffusion model M. A node $u \in V$ is said to be the *neighbour* of v if there is an edge $(u, v) \in E$. The diffusion process comes to a halt when there are no more activations. One classic example of influence function is the expected number of influenced nodes.

The main objective of the IM problem is finding a seed set of at most k nodes that maximizes the value of $\sigma_{G,M}$ at the end of the process. However, maximizing $\sigma_{G,M}$ is often a computationally hard problem. To overcome this issue, one can resort to an *r-approximation algorithm*, that computes in polynomial time an approximatively optimal seed set S with an *approximation factor* of $r \leq 1$, i.e., a seed set S such that $\sigma_{G,M}(S) \geq r \cdot \sigma_{G,M}(S^*)$. Most of the approximation algorithms considered for the IM problem require that function $\sigma_{G,M}$ verify two desirable properties: *monotonicity* and *submodularity*. Monotonicity ensures that adding nodes to the seed set will not reduce the value of $\sigma_{G,M}$, whereas submodularity ensures that the marginal gain of adding a node to the seed set does not increase as the size of the seed set increases. In particular, $\sigma_{G,M}$ is *monotone* if, for any $S \subseteq V$ and for any $x \in V \setminus S$, we get $\sigma_{G,M}(S) \leq \sigma_{G,M}(S \cup \{x\})$. Furthermore, $\sigma_{G,M}$ is *submodular* if, for any $S, P \subseteq V$ such that $S \subseteq P$ and for any $x \in V \setminus P$, we get $\sigma_{G,M}(P \cup \{x\}) - \sigma_{G,M}(P) \leq \sigma_{G,M}(S \cup \{x\}) - \sigma_{G,M}(S)$. In subsection 2.3, we will formally describe a greedy approximation algorithm that exploits these properties to guarantee a good approximation ratio.

2.2 Diffusion Models

The design of *Diffusion Models* is an integral aspect of the IM problem. There are two kinds of diffusion processes: *progressive* and *non-progressive*. In the progressive model, once a node is activated, deactivation is impossible. In contrast, in the non-progressive model, each node, once enabled, can also be disabled. Even-Dar et al. [5] studied the IM problem by exploiting the *Voter Model* [6], which is an example of a non-progressive case. In this proposal, we restrict our focus on the progressive models.

Models can also be *time-unbounded* and *time-bounded*. Time-unbounded models are those models where termination occurs when no other activation is possible. Whereas, the time-bounded models are those models where they having a separate time parameter for the completion of an ongoing diffusion process. Most of the previous works mainly focus on time-unbounded models. Some examples of time-unbounded models are the *Independent Cascade,* *Linear Threshold* and *Triggering* models, which will be explained better in the following sections.

2.2.1 Independent Cascade (IC) Model

Goldenberg et al. [7, 8] used the concept of *Cellular Automata* as a framework to model the diffusion of information in the network. Kempe et al. [2] named it as the *Independent Cascade Model*. Initially, the diffusion model starts with a set of active nodes S_0 such that $S_0 \leq k$. Each edge $e = (u, v) \in E$, has an influence probability $p_{u,v}$ attached to it, so that node u will activate node v with a probability $p_{u,v}$, independently from what happens to the other nodes. With the above parameters as input to the system, the diffusion process unravels in different time steps. At time step t, if node u is active, it is given a single opportunity to activate its neighbour v with $p_{u,v}$. If node v becomes active at time step t, it will further attempt to activate its neighbours at time step $t+1$ according to the influence probability. The process iteratively continue and comes to a halt when there are no more activations.

We point out that the influence probability is a random event, and the game of flipping a coin with a bias of $p_{u,v}$, associated with head determines it. Before the start of the diffusion process, the biased coin is flipped for every edge in the graph G. By looking at the influence probabilities; one can conclude whether a node v will be active after the termination of the cascade process or not.

Kempe et al. [2] shows that finding the set of influential nodes maximizing the influence function in the IC model is NP-hard, and they use a reduction to the *Maximum Coverage* problem. A particular model of IC is that of *Weighted Cascade*. In such a model, each influence probability is defined as $p_{u,v} = 1/d_{in}(v)$, where $d_{in}(v)$ is the in-degree of v. In this process, each node is equally influenced by each of its neighbours.

Other models assume that the influence probabilities are not known, thus learning them is an interesting challenge. Saito et al. [9] predict the influence probabilities by using the *Expectation Maximization* algorithm to compute the diffusion probabilities of all the edges in G, repeatedly, to maximize the objective function. Goyal et al. [10] developed a learning algorithm which takes as input a social graph and the *Action Log*, which reports the actions performed by all the nodes.

2.2.2 Linear Threshold (LT) Model

Granovetter [11] introduced the *Linear Threshold model*, where he proposes a diffusion model in which each node v becomes active if the weighted sum of its active neighbours reaches a certain threshold. More formally, the model takes as input a graph G, the seed set S_0, and an edge weight $W_{u,v}$ for any edge $(u, v) \in E$. Before the diffusion starts, a *threshold value* θ_v is picked uniformly at random in the interval $[0, 1]$ by each node $v \in V$, independently from the other nodes.

At each time step t a node v will get activated if the summation of edge weights of its active neighbours reaches the threshold value θ_v, i.e.,

$$\sum_{(u,v)\in E:u \text{ is active}} W_{u,v} \geq \theta_u \tag{2}$$

The process proceeds in several time steps and terminates when no more activations are possible. Kempe et al. [2], showed that finding the set S_0 maximizing the influence function in the LT model is also NP-hard, and they use a reduction to the *Vertex Cover* problem.

A *Generalized Threshold Model* can be designed by removing the linearity from the LT model. The model follows the same structure as the LT. However, v now relies upon an arbitrary *monotone threshold function* f_v which maps the subsets of v's neighbours into the interval $[0, 1]$, with $f_v(\emptyset) = 0$. Let N represent the set of neighbours of v that are active at time step $t - 1$. Node v becomes active at time step t if and only if $f_v(N) \geq \theta_v$. The LT model is a special case of the Generalized Threshold model. The threshold function of the LT model takes the following form:

$$f_v(N) = \min\left(1, \sum_{u\in N} W_{u,v}\right). \tag{3}$$

The General Threshold model is *submodular* if each function f_v is submodular. If the General Threshold model is not submodular, its analysis may be intractable.

2.2.3 Triggering Model

The IC and the LT are exceptional cases of the *Triggering Model* introduced by Kempe et al. [2]. In this model, instead of having a threshold value, a node v has a quiescent subset T_v, called the *triggering set*, which affects the state of v. Initially, the process starts with a seed set S_0, and each node v chooses a triggering set T_v independently from each other, according to some distribution over the subsets of its neighbours. An inactive node v can be activated at time step t, if it has an active neighbour u in T_v at time step $t-1$.

Another way to think of this process is by considering the *live edge technique*. Edges of a graph $G = (V, E)$ can be either live or blocked. If a node u is present in the triggering set T_v of v, the edge $(u, v) \in E$ will be declared as live and claimed to be blocked otherwise. So, it can be stated that S is the set of nodes v, such that v is reachable from the seed set S_0 by traversing a path made entirely out of live edges. The influence function σ is shown to be submodular in the Triggering Model.

2.2.4 Continuous-Time Models

The influence spread occurs in discrete time steps in the diffusion models stated in the previous subsections. However, marketing or campaigns can often be time-constrained, and the models following the discrete time may not be able to capture the real life scenario. In such cases, a model which can work in continuous time will be useful.

Rodriguez et al. [12] studied the *Continuous-Time model* for a *static unknown network*. The problem is defined as follows: given an edge $(u, v) \in E$, the probability that u activates a node v within a certain time t_v, conditioned by the fact that $t_u < t_v$ is the activation time of u, has the form of $p(t_v|t_u; \alpha_{u,v})$, where $p(t_v|t_u; \alpha_{u,v})$ only depends on $(t_v - t_u)$ and a parameter $\alpha_{u,v}$, called *transmission rate*. For instance, if we consider the exponential distribution as the underlying probability distribution for our model, we have that $p(t_v|t_u; \alpha_{u,v}) = \alpha_{u,v} \cdot \exp(-\alpha_{u,v}(t_v - t_u))$. The diffusion terminates when the spread reaches the end of a given observation window $T > 0$. The goal is to find a seed set maximizing the expected number of active nodes at time T.

Xie et al. [13], designed a continuous-time model for *dynamic networks*. In particular, they created a model named *Dynadiffuse* based on the *Continuous Time Markov Chain*. The model captures the dynamic nature of the network by observing the change of *propagation rates* of the nodes, that decrease exponentially over time.

2.3 Non-Adaptive Influence Maximization

The IM problem as described by Kempe et al. [2] is an one-shot task: select the seed set S_0, and with only one propagation pass through the entire network, S_0 activates a handful number of nodes, with the expectation of maximizing the number of active nodes. A simple greedy framework is used by most of the IM algorithms, and more generally, in submodular function maximization. Algorithm 1 shows a greedy approximation algorithm, denoted as *Non-adaptive greedy algorithm*, that has been used by Kempe et al. [2] to find an approximately optimal seed set S_0.

Algorithm 1: Non-Adaptive Greedy (G, k, σ)

```
Input: G: Social Graph; k: budget; σ(.): Influence Function
Output: S_0: Seed Set
S_0 ← ∅
while |S_0| < k do
    u* ← arg max_{u∈V\S_0}(σ(S_0 ∪ {u}) − σ(S_0))
    S_0 ← S_0 ∪ {u*}
end
return S_0
```

By using the fact that σ is a monotone submodular function with $\sigma(\emptyset) = 0$, the results of Nemhauser and Wolsey [14] on submodular functions approximation imply that Algorithm 1 will always produce a solution guaranteeing at least $1 - [(k-1)/k]^k \geq 1 - 1/e$ times the optimal value, where $e \simeq 2.71$ is the *Nepero number*. At each step of the algorithm, we are required to compute the value $\sigma(S_o \cup \{u\})$ for any node $u \in V \setminus \{u\}$ to find the local optimal influence function, but this computation is hard [15]. Kempe et al. [2] overcame this problem by resorting to a polynomial sampling method to approximate the influence function with an arbitrary small $\epsilon > 0$, so that, the final approximation ratio becomes $1 - 1/e - \epsilon$.

2.4 Adaptive Influence Maximization

The adaptive version of the IM problem proclaims that instead of selecting the nodes for the seed set all at once, before the beginning of the process, the seeds are chosen sequentially one at a time. The decision of which seed to select, depends on the feedback obtained from the propagation result of the previously chosen seeds. This version of the IM is flexible, since the advertiser can now observe the market scenario, and make his decisions based on the results acquired from his previous actions.

2.4.1 Adaptive Maximization.

The *Adaptive Maximization (AM)* setting has been considered for the first time by Golovin and Krause [1], and has been later applied to the context of Influence Maximization, to get its adaptive version. The general problem of AM is defined as follows. Consider a graph $G = (V, E)$ and a set of states $O = \{alive, dead\}$. Each edge $e \in E$ is in one of the possible states, so that the states of the edges can be represented using a function $\phi : E \mapsto O$, called the *realization*. Hence, we can state that $\phi(e)$ is the state of e under realization ϕ. A random variable Φ taking values from all the possible realizations, according to a given probability distribution, is called *random realization*. For a given realization ϕ, let $p(\phi)$ denote the probability that $\Phi = \phi$.

We probe each $e \in E$ sequentially and observe its state $\Phi(e)$. The observations made after each pick can be represented as a *partial realization* function ψ, over some subset of E. The domain of ψ, i.e., the set of edges observed in ψ is represented by $dom(\psi)$. A partial realization function is said to be consistent with a realization ϕ if they are equal in $dom(\psi)$ and is written as $\phi \sim \psi$. When ψ and ψ' are both consistent with some ϕ, and $dom(\psi) \subseteq dom(\psi')$, ψ is called the *subrealization* of ψ'.

The node selection for the next round is strategized by an *adaptive policy* π, which is a function from a set of partial realizations to E. When ψ is not present in the domain of π, the termination of the policy occurs upon observations of ψ. The domain of π, denote by $dom(\pi)$, should be closed under subrealizations. The function we want to maximize takes the form of $f : 2^E \times O^E \to \mathbb{R}_{\geq 0}$, and takes as input the realization ϕ, and the set of edges $E(\pi, \phi)$ selected by the considered adaptive policy π under realization ϕ. Unfortunately, we do not know what is the final realization ϕ. The AIM overcomes this problem, and its goal is to find a policy π^* that maximizes the expected value $\mathbb{E}_\Phi(f(E(\pi, \Phi), \Phi))$ subject to $E(\pi, \Phi) \leq k$, where k, denoted as *budget*, is the maximum number of edges that a policy can select.

Another important term coined by Golovin and Krause [1] is the *Adaptivity Gap (AG)*. The AG is the measure of how well an adaptive policy can perform when compared to a *non-adaptive policy*, that is a policy in which all the edges should be selected at beginning, without observing the realization.

More formally, the AG on an input instance is defined as

$$AG = \frac{\arg\max_{\text{adaptive policy } \pi} \mathbb{E}_\Phi(f(E(\pi, \Phi), \Phi))}{\arg\max_{F \subseteq E : |F| \leq k} \mathbb{E}_\Phi(f(F, \Phi))}. \tag{4}$$

2.4.2 Adaptivity and Influence Maximization.

Now, we adapt the previous framework of adaptivity maximization to that of influence maximization, to get the *Adaptive Influence Maximization (AIM) Problem*. In particular, we consider the adaptive version of the IC model. We recall that the input instance of the IC model is a directed graph $G = (V, E)$, a budget $k \geq 1$, and an activation probability $p_{u,v}$ for any edge $(u, v) \in E$. As in the model for general adaptive maximization, a realization ϕ determines if each edge is active or non-active, and the probability distribution of an arbitrary realization ϕ is implicitly given by the fact that each edge (u, v) is active with probability $p_{u,v}$, independently from the other edges. Similarly as in the previous model of adaptive maximization, an adaptive policy π is a function associating the *network state* with a new node that should be added to the seed set, where the network state is what a policy observes at each time. The function we want to maximize in expectation can be defined as $f : 2^V \times O^E \to \mathbb{R}_{\geq 0}$, where $f(U, \phi)$ is the number of nodes that are reached by the spread of information, given that $U \subseteq V$ is the seed set and ϕ is the realization. The AIM problem asks to find a policy that finds a seed-set maximizing the expected value of f (according to the probability distribution of the realizations), where the cardinality of the seed set must be at most k.

2.5 Feedback Models of AIM

Depending on the type of network state that a policy can observe at each time, we have two general models of AIM: the *edge-feedback model*, in which the network state is a set of active edges, and the *node-feedback model*, in which the network state is a set of active nodes.

In this section, we discuss some particular models of AIM, that can be either node-feedback, or edge-feedback. In particular we consider the *Full-Adoption Feedback model*, the *Myopic Feedback model*, and the *General Feedback model*.

2.5.1 Full-Adoption Feedback Model

The *Full-Adoption Feedback model* in an edge-feedback model selects a node and returns the entire cascade as feedback for the next round. Each node is selected one by one, and the selection needs to be done at the beginning of each round. Once the budget k is reached, the process terminates and returns the seed set. In this model, a realization ϕ can be alternatively seen as a function $\phi : V \to 2^E$, associating to node $u \in V$, the set $\phi(u) \subseteq E$ of edges which are traversed by the spread generated by u. The random realization Φ and a partial realization ψ are defined starting from the new definition of realization, according to the model of adaptive maximization.

To solve the AIM problem in the Full-Adoption Feedback model, in particular the edge-feedback version, Golovin and Krause [1] designed a generalization of the non-adaptive greedy algorithm, shown in Algorithm 2.

Algorithm 2: Adaptive Greedy (G,k,p,f)

Input: G: social graph; k: budget; p: probability distribution of random realization Φ; f: utility function

Output: S_0: seed set

$S_0 \leftarrow \emptyset$; $\psi \leftarrow \emptyset$

while $|S_0| < k$ **do**

  $u^* \leftarrow \arg\max_{u \in V \setminus S_0} \mathbb{E}_\Phi \left[f(S_0 \cup \{u\}, \Phi) - f(S_0, \Phi) | \Phi \sim \psi \right]$

  $S_0 \leftarrow S_0 \cup \{u^*\}$

  Observe $\Phi(u^*)$

  $\psi \leftarrow \psi \cup (u^*, \Phi(u^*))$

end

return S_0

Golovin and Krause [1] showed that the Full Adoption Feedback Model can guarantee a good approximation ratio by applying the concepts of monotonicity and submodularity to the adaptive framework. Given a function $f : 2^V \times O^E \mapsto \mathbb{R}_{\geq 0}$ w.r.t. a random realization Φ distributed according to a probability p, the *conditional expected marginal gain* $\Delta(u|\psi, S_0)$ is defined as

$$\Delta(u|S_0) := \mathbb{E}_\Phi[f(S_0 \cup \{u\}, \Phi) - f(S_0, \Phi)|\Phi \sim \psi], \tag{5}$$

for any seed set $S_0 \subseteq V$, and node $u \in V \setminus S_0$, and where ψ is the partial realization determined after selecting the nodes in seed set S_0. A function f is *adaptive monotone* if the conditional expected marginal gain is non-negative.

i.e., if $\Delta(u|\psi) \geq 0$. A function f is *adaptive submodular* if the conditional expected marginal gain of any node does not increase when more nodes are added to the set and observed. i.e., for any $S_0, S'_0 \subseteq V$ such that $S'_0 \subseteq S_0$, and for any $u \in V \setminus S_0$, we get $\Delta(u|S'_0) \geq \Delta(u|S_0)$.

Initially, in Algorithm 2, the set of seed nodes S_0 and the set of partial realizations ψ are an empty set. The greedy algorithm greedily selects nodes one by one from the graph G until the budget k has been reached. For each node $u \in V$, the conditional expected marginal gain is determined. The node with the largest estimate for the gain is added to S_0, and the random realization of the selected node u is observed. The set of partial realizations is updated by adding the edges that have been activated by u. Golovin and Krause [1] proved that for the adaptive IM problem, the greedy policy π^G determined by Algorithm 2 gets at least $\left(1 - \frac{1}{e}\right)$ of the optimal policy, when f is monotone and submodular.

Together with the problem of finding an optimal adaptive policy, another interesting direction has been determining the Adaptivity Gap for the Full Adoption Feedback Model. Chen and Peng [16] derive the upper and lower bounds on the AG when graphs are *in-arborescences* or *out-arborescences*. An arborescence is a directed rooted tree. An in-arborescence (resp. out-arborescense) is an arborescence when the edges are directed from the leaves to the root (resp. from the root to the leaves).

Since the information propagates from the leaves to the root in an in-arborescence graph, the boundary of active nodes shrinks. The leaves are the set of discovered nodes, and the boundary forms the subset of the set of nodes reachable from the leaves. Using this properties, the AG has been shown to be between $\frac{e}{(e-1)}$ and $\frac{2e}{(e-1)}$. In the case of an out-arborescence graph, the main observation has been that the predecessors of each node in the graph form a directed line. In order to activate a certain node u, all its predecessors must be activated one at a time. Using this property, the AG has been shown to be between $\frac{e}{(e-1)}$ and 2.

2.5.2 Myopic Feedback Model

The main drawback of the Full Adoption feedback model is the delay in the process. Indeed, one should select a node one by one, observe its complete diffusion, return it as a feedback for the selection of the next seed node, and continue this over the k steps. To overcome this potential set back, we could select a seed node u and check which of its neighbours are getting influenced by it, and return the set of nodes influenced by u as a feedback for the next round. Since only the active neighbours of the selected seed are returned as feedback, the model is called *Myopic*. The Myopic model has been mentioned by Golovin and Krause [1], and in a reprised version of the same paper, the authors claimed that the objective function f in the Myopic feedback model is not adaptive submodular. The submodularity condition is violated since the seed node selection process is not made after a diffusion terminates. Instead, the diffusion of a certain round ceases after the current node activates its neighbours.

Peng and Chen [17] confirmed a conjecture of Golovin and Krause [1], which states that the adaptive greedy algorithm with myopic feedback is a constant approximation of the adaptive optimal solution. They obtain an upper and lower bound for the AG of the IC model with myopic feedback, $[\frac{e}{e-1}, 4]$. They also proved that the adaptive and the non-adaptive greedy algorithm has an approximation ratio of $\frac{1}{4}(1 - \frac{1}{e})$ under the myopic model. The authors conclude by stating that the Myopic feedback model is not beneficial since the approximation ratios for both the adaptive and non-adaptive version are the same. Also, the solution is $\frac{e^2+1}{(e+1)^2} \approx 60\%$ of the optimal value, which is less than the results obtained by Kempe et al. [2]. In simple words, the performance is not commendable since the feedback received is not enough to determine the selection of the best possible node in the next round.

2.5.3 Partial and General Feedback Model

Yuan and Tang [18] have expressed the trade-off between performance and delay between the Full Adoption and the Myopic feedback model. They developed the *Partial Feedback* model, which however is not adaptive submodular, due to the same reasons as for the Myopic Model. The Partial Feedback model can capture a more realistic scenario of the marketing world. The model consider a general *α-greedy policy*, depending on a *control parameter* $\alpha \in [0, 1]$ given as input. At each round r, a policy selects a new seed node v, and the cascade proceeds via *time slots* in which the depth of the cascade increases by one; we can equivalently denote each time slot with the depth d of the cascade generated by v. At each round r and time slot d, the policy observes the state of all the edges involved in the cascade, and the round ends when the following condition is verified:

$$\alpha \leq \frac{\sum_{v \in V} p_v(S_r, \psi_{r,d})}{|V \setminus O_{r,d}|}, \tag{6}$$

where $\psi_{r,d}$ represent the observations, $O_{r,d}$ represents the set of nodes whose activation probabilities are zero at round r and at time slot $d \geq 0$, and $\sum_{v \in V} p_v(S_0, \psi_{r,d})$ is the expected number of activated nodes, under the activation probability of $p_v(S_r, \psi_r)$, with S_r being the seed set. After this, a new round starts, and the previous process iteratively continues until the seed set contains k nodes. Informally, each round terminates when the expected probability that a node becomes active is higher than α. Observe that, the partial model becomes a full adoption feedback model when $\alpha = 1$, and a non-adaptive model when $\alpha = 0$.

Tong and Wang [19] introduced the *General Feedback* model. Such model is defined as the partial model, except for the fact that each round terminates after exactly d time slots, where $d \geq 0$ is an integer given as input. Observe that for $d = 1$ we get the myopic feedback model, and for $d = \infty$ we get the full adoption feedback model. Since the model is not adaptive submodular, one cannot immediately guarantee that the greedy strategy is a good approximation algorithm. To overcome this problem, the authors introduced a new metric called the *regret ratio*. The regret ratio R of a diffusion process is the maximum ratio between (i) the best marginal profit obtained for a given seed set and a given round, when a new seed node is added to S at the beginning of round r, and (ii) the best marginal profit achieved when the new seed node is added to S at the end of round r, i.e., after d time slots. The regret ratio has been used to show that the greedy algorithm guarantees an approximation factor of $1 - e^{-1/R}$.

2.6 Multi-Round Influence Maximization

Sun et al. [20] captures the marketing scenario, where the advertiser considers multiple rounds to market a product. In particular, instead of considering one round for the diffusion process, the advertiser conducts the diffusion process for T rounds. The advertiser has a budget k for each round, i.e., a seed set S_t of at most k nodes can be selected at each round. An independent diffusion process is carried out in each round t, where a subset of nodes is activated, starting from the initial seed set S_t. The total influence spread over the T rounds is calculated as the union of the sets of nodes activated over all the T rounds. The goal is to find the seed sets that maximize the expected total influence spread. The authors work on both the adaptive and non-adaptive version and design three greedy algorithms.

In the non-adaptive version, the advertiser needs to find all the T seed sets before the start of the diffusion processes. The authors consider two different algorithms: *Cross-Round* and *Within-Round*. The within-round has an approximation ratio of $1 - e^{-(1-\frac{1}{e})} \approx 0.46$. However, the cross-round has a better approximation factor of $1/2 - \varepsilon$, but has a higher running time.

In the adaptive version of the Multi Round Influence Maximization, in each round, the advertiser needs to select the seed set for that round and observe its influence spread. Once the budget k is reached for a specific round r, the next round $r+1$ receives the influence spreads of the previous r rounds as feedback. In the next round, based on the observations of the propagation made by the previous rounds, the seed set for that round is selected. The authors designed a greedy approximation algorithm called *AdaGreedy*, and show that the algorithm guarantees a $1 - e^{-(1-\frac{1}{e})} - \varepsilon$ approximation ratio.

3 Open Problems

This section depicts several open problems and challenges in the domain of Adaptive Influence Maximization.

3.1 Settling Down the Adaptivity Gaps

The Adaptivity Gap has been introduced to measure by how much an adaptive policy outperforms a non-adaptive one. We have that, as the AG is high, an adaptive policy can be much better than the non-adaptive one. In contrast, if the AG is small, an adaptive policy does not perform better than the non-adaptive policy. There are two different approaches to analyze the Adaptivity Gaps for the AIM:

- When considering general *Stochastic Submodular Maximization* problems, a property called *Multilinear Extension* has been exploited by Asadpour et al. [21]. Such property has been used to the AIM problem, too, since this is a particular case of Stochastic Submodular Maximization problems.

- For *Stochastic Probing* problems, a *random walk non-adaptive* policy on decision trees has been considered by Gupta et al. [22]. The stochastic probing problem has a set of elements with random costs attached to it. The problem requires us to probe an element to find the cost, and decide whether it should be added to the seed set S or not. The goal is to maximize the total cost of selections. The AIM problem can be modelled in terms of Stochastic Probing Problems, thus the techniques depicted above has been applied to AIM, too.

Despite the previous approaches which has been successfully applied to get the upper and lower bound on the AGs, such bounds are not tight. Thus, determining the exact value (or tightening the upper and lower bounds) of the AG of different families of graphs is an interesting open problem.

3.2 Using different Diffusion Models to design the AIM problem

Most of the works on AIM have considered the Independent Cascade model, only. A natural extension would be to look into the other diffusion models, such as the Linear Threshold and Continuous-Time Models. The Continuous-Time Model captures the real-world scenario better than the IC and LT. An interesting direction would be to find the Adaptivity Gaps for the continuous time scenario. Also, studying general graphs under the Full-Adoption feedback model would be an intriguing issue.

In the past, research has mainly been conducted on the edge-feedback models because it gathers more information about the diffusion. However, in practice, it is tough to figure out which neighbour u has influenced a specific neighbour v. For example, when a person shares a review about a product, it is easily conceived that the node has been influenced, but extremely hard to determine which neighbour has influenced that person to buy the product. So, in certain cases, it is better to consider and analyze the node-feedback model.

The above challenge brings us to a particular version of the IM problem known as the *Competitive IM*. Bharathi et al. [23] were the firsts to propose such a model, where the main objective of a node is to maximize its influence, by decreasing the influence of other active nodes. In light of the Viral Marketing scene, it can be viewed as multiple companies trying to market their products at the same time. Designing an adaptive policy for this competitive version can be equally challenging.

3.3 Adopting a Partial Feedback Model with further Generalizations

The General Feedback successfully captures the different Viral Marketing strategies. It can turn into a Full Adoption, or a Myopic feedback model by adjusting the number of rounds for the cascading process. Travelling through the same path, we can further generalize the General feedback model for modelling several real life applications. Some of the examples listed below gives us an overview of the problems that we can address.

- In a social network, sometimes, due to privacy settings, it becomes difficult to observe the states of all the edges. We know that to have information about the states of edges, it needs to be probed. However, given the privacy setting, one can only examine at most $p \in \mathbb{Z}^+$ edges at a time. So, it becomes essential to select those edges that needs more attention. The issue can be handled by making some modifications in the seeding process.

- Instead of observations being made round by round, a more practical approach would be to make the observations based on some significant event. For example, an advertiser can start a new seeding process once its previous seeds have reached the targeted influence spread: this will, in turn, save a significant amount of computational time.

- The trade-off between the quality of observations and the quality of seed nodes being activated is another crucial issue. Suppose we are given a set of observable edges, denoted by $E^* \subseteq E$ and a feedback model. The problem requires us to make the observations on E^*, and select the seed for the next phase. Since our observations are limited, sometimes, the seed nodes, which has a higher influence may be distant or unreachable from E^*.

The other two exciting directions would be:

- Define the regret ratios for the batch mode, where instead of selecting one seed per round, one can choose a set of nodes. [24]

- People in real life not only gets influenced by a single person, but a group of which they are a part of can also influence them. [25]
  It will be an interesting aspect to incorporate AIM in such situations and analyze the problem.

4 Applications

The AIM problem is not only restricted to the Viral Marketing framework but also applies to other real-world issues, and serves as a motivating factor to conduct research in this field. Some of the prominent areas listed below have a similar structure as the VM, and might be expounded using AIM.

4.1 Network Monitoring

Leskovec et al. [26] explained about the problem of placement of sensors in specific locations to monitor the environment. The objective here is to place the minimum number of sensors to monitor the most crucial points in the network, where there can be an outbreak of an event. In such a scenario, it might happen that some of the sensors can start malfunctioning and might not be able to give proper data. The malfunctioning of the sensors can only be known via probing or monitoring those sensors. Due to the uncertain nature of the functionality of the sensors, it is recommended to design the system in an adaptive setting.

4.2 Political Campaigns

These class of problems are the most studied ones in the aspects of Game Theory and Network Analysis. The aim is to maximize the influence made by a particular candidate on the population. The problem can be made adaptive by selecting a batch of people who are responsible for conducting campaigns for a specific candidate. Based on the observations made on the performance of those selected group of people and the impact made on the targeted audience, the candidate might make further decisions on how to proceed with his campaign.

4.3 Rumour Control

It is quite observable that social networks can also serve as a breeding ground of malicious contents and misinformation. Budak et al. [27] studied the phenomenon of the misinformation cascade and how to limit such bad influential nodes in the network. However, due to privacy issues and missing information, it becomes intractable to find the nodes. This issue can be examined via Section 3.3, where the observations are made based on some triggering events.

4.4 Limiting an Epidemic

The IM problem can be designed to control an *epidemic*. If a certain number of people are detected to have shown symptoms of a disease, those people can be vaccinated and kept in isolation to prevent the spread. An adaptive version would be to vaccinate the initial number of people affected, and wait until another batch of people starts showing the same symptoms. The issue can be designed as observations made on the basis of some events, and the epidemic can be kept under check.

References

[1] Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42(2005):427–486, 2011.

[2] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11(4):105–147, 2015.

[3] Dawn Iacobucci and David Krackhardt. Structural Leverage in Marketing, 2014.

[4] Pedro Domingos and Matt Richardson. Mining the network value of customers. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 57–66, 2001.

[5] Eyal Even-Dar and Asaf Shapira. A Note on Maximizing the Spread of Influence. *Proceedings of the 3rd international conference on Internet and network economics, WINE'07*, pages 281–286, 2007.

[6] Peter Clifford and Aidan Sudbury. A model for spatial conflict. *Biometrika*, 60(3):581–588, 1973.

[7] Goldenberg J., Libai B., and Muller E. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth. *Marketing Letters*, 12(3):211–223, 2001.

[8] Jacob Goldenberg, Barak Libai, and Eitan Muller. Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata. *Academy of Marketing Science Review*, 2001(9):1, 2001.

[9] Kazumi Saito, Ryohei Nakano, and Masahiro Kimura. Prediction of information diffusion probabilities for independent cascade model. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5179 LNAI(PART 3):67–75, 2008.

[10] Amit Goyal, Francesco Bonchi, and Laks V.S. Lakshmanan. Learning influence probabilities in social networks. *WSDM 2010 - Proceedings of*

the 3rd ACM International Conference on Web Search and Data Mining, pages 241–250, 2010.

[11] Mark Granovetter. Threshold Models of Collective Behavior. *American Journal of Sociology*, 83(6):1420–1443, 1978.

[12] Manuel Gomez-Rodriguez, David Balduzzi, and Bernhard Schölkopf. Uncovering the temporal dynamics of diffusion networks. *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, pages 561–568, 2011.

[13] Miao Xie, Qiusong Yang, Qing Wang, Gao Cong, and Gerard De Melo. DynaDiffuse: A dynamic diffusion model for continuous time constrained influence maximization. *Proceedings of the National Conference on Artificial Intelligence*, 1(March):346–352, 2015.

[14] G.L. Nemhauser and L.A. Wolsey. An Analysis of Approximations for Maximizing Submodular Set Functions-I. *Mathematical Programming*, 14:265–294, 1978.

[15] Wei Chen, Yifei Yuan, and Li Zhang. Scalable influence maximization in social networks under the linear threshold model. *Proceedings - IEEE International Conference on Data Mining, ICDM*, pages 88–97, 2010.

[16] Wei Chen and Binghui Peng. On Adaptivity Gaps of Influence Maximization under the Independent Cascade Model with Full Adoption Feedback. *30th International Symposium on Algorithms and Computation, ISAAC 2019*, (61433014), 2019.

[17] Binghui Peng and Wei Chen. Adaptive Influence Maximization with Myopic Feedback. *Conference on Neural Information Processing Systems, NIPS*, 2019.

[18] Jing Yuan and Shaojie Tang. No time to observe: Adaptive influence maximization with partial feedback. *IJCAI International Joint Conference on Artificial Intelligence*, pages 3908–3914, 2017.

[19] Guangmo Tong and Ruiqi Wang. Adaptive Influence Maximization under General Feedback Models. 2019.

[20] Lichao Sun, Weiran Huang, Philip S. Yu, and Wei Chen. Multi-round influence maximization. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2249–2258, 2018.

[21] Arash Asadpour, Hamid Nazerzadeh, and Amin Saberi. Stochastic submodular maximization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5385 LNCS:477–489, 2008.

[22] Anupam Gupta, Viswanath Nagarajan, and Sahil Singla. Algorithms and adaptivity gaps for stochastic probing. *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, 3:1731–1747, 2016.

[23] Shishir Bharathi, David Kempe, and Mahyar Salek. Competitive influence maximization in social networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4858 LNCS:306–311, 2007.

[24] Yuxin Chen and Andreas Krause. Near-optimal batch mode active learning and adaptive submodular optimization. *30th International Conference on Machine Learning, ICML 2013*, 28(PART 1):160–168, 2013.

[25] Jie Tang, Sen Wu, and Jimeng Sun. Confluence: Conformity influence in large social networks. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Part F1288:347–355, 2013.

[26] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne Vanbriesen, and Natalie Glance. Cost-effective outbreak detection in networks. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 420–429, 2007.

[27] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. Limiting the spread of misinformation in social networks. *Proceedings of the 20th International Conference on World Wide Web, WWW 2011*, pages 665–674, 2011.