
Influence Maximization with ε -Almost Submodular Threshold Functions

Qiang Li[†], Wei Chen[‡], Xiaoming Sun^{*†}, Jialin Zhang^{*†}

^{*}CAS Key Lab of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences

[†]University of Chinese Academy of Sciences

[‡]Microsoft Research

{liqiang01, sunxiaoming, zhangjialin}@ict.ac.cn
weic@microsoft.com

Abstract

Influence maximization is the problem of selecting k nodes in a social network to maximize their influence spread. The problem has been extensively studied but most works focus on the submodular influence diffusion models. In this paper, motivated by empirical evidences, we explore influence maximization in the non-submodular regime. In particular, we study the general threshold model in which a fraction of nodes have non-submodular threshold functions, but their threshold functions are closely upper- and lower-bounded by some submodular functions (we call them ε -almost submodular). We first show a strong hardness result: there is no $1/n^{\gamma/c}$ approximation for influence maximization (unless $P = NP$) for all networks with up to n^γ ε -almost submodular nodes, where γ is in $(0,1)$ and c is a parameter depending on ε . This indicates that influence maximization is still hard to approximate even though threshold functions are close to submodular. We then provide $(1 - \varepsilon)^\ell(1 - 1/e)$ approximation algorithms when the number of ε -almost submodular nodes is ℓ . Finally, we conduct experiments on a number of real-world datasets, and the results demonstrate that our approximation algorithms outperform other baseline algorithms.

1 Introduction

Influence maximization, proposed by Kempe, Kleinberg, and Tardos [1], considers the problem of selecting k seed nodes in a social network that maximizes the spread of influence under pre-defined diffusion model. This problem has many applications including viral marketing [2, 3], media advertising [4] and rumors spreading [5] etc., and many aspects of the problem has been extensively studied.

Most existing algorithms for influence maximization, typically under the independent cascade (IC) model and the linear threshold (LT) model [1], utilize the submodularity of the influence spread as a set function on the set of seed nodes, because it permits a $(1 - 1/e)$ -approximation solution by the greedy scheme [1, 6, 7], following the foundational work on submodular function maximization [29]. One important result concerning submodularity in the influence model is by Mossel and Roch [8], who prove that in the general threshold model, the global influence spread function is submodular when all local threshold functions at all nodes are submodular. This result implies that “local” submodularity ensures the submodularity of “global” influence spread.

Although influence maximization under submodular diffusion models is dominant in the research literature, in real networks, non-submodularity of influence spread function has been observed. Backstrom et al. [9] study the communities of two networks LiveJournal and DBLP and draw pictures

of the impulse that a person joins a community against the number of his friends already in this community. The curve is concave overall, except that a drop is observed in first two nodes. Yang et al. [10] track emotion contagion under *Flickr* and find that the probability that an individual becomes happy is superlinear to the number of his happy friends with higher PageRank scores. These are all instances of non-submodular influence spread functions.

Influence maximization under many non-submodular diffusion models are proved to be hard to approximate. For example, in the diffusions of rumors, innovations, or riot behaviors, the individual in a social network is activated only when the number of her neighbors already adopting the behavior exceeds her threshold. It has been shown that the influence maximization problem based on this fixed threshold model cannot be approximated within a ratio of $n^{1-\varepsilon}$ for any $\varepsilon > 0$ [1]. Meanwhile Chen [11] proves that the seed minimization problem, to activate the whole network with minimum size of seed set, is also inapproximable, in particular, within a ratio of $O(2^{\log^{1-\varepsilon} n})$.

In this paper we give the first attempt on the influence maximization under the non-submodular diffusion models. We study the general threshold model in which a fraction of nodes have non-submodular threshold functions, but their threshold functions are closely upper- and lower-bounded by some submodular functions (we call them ε -almost submodular). Such a model bears conceptual similarity to the empirical finding in [9, 10]: both studies show that the influence curve is only slightly non-concave, and Yang et al. [10] further shows that different roles have different curves — some are submodular while others are not, and ordinary users usually have behaviors close to submodular while opinion leaders may not. We first show a strong hardness result: there is no $1/n^{\frac{\gamma}{\varepsilon}}$ approximation for influence maximization (unless $P = NP$) for all networks with up to n^γ ε -almost submodular nodes, where γ is in $(0, 1)$ and c is a parameter depending on ε . On the other hand, we propose constant approximation algorithms for networks where the number of ε -almost submodular nodes is a small constant. The positive results imply that non-submodular problem can be partly solved as long as there are only a few non-submodular nodes and the threshold function is not far away from submodularity. Finally, we conduct experiments on real datasets to empirically verify our algorithms. Empirical results on real datasets show that our approximation algorithms outperform other baseline algorithms.

Related Work. Influence maximization has been well studied over the past years [12, 6, 7, 13, 14]. In particular, Leskovec et al. [6] propose a *lazy-forward* optimization that avoids unnecessary computation of expected size. Chen et al. [7, 13] propose scalable heuristic algorithms that handle network of million edges. Based on the technique of *Reverse Reachable Set*, Borgs et al. [15] reduce the running time of greedy algorithms to near-linear under the IC model [1]. Tang et al. [16] implement the near-linear algorithm and process *Twitter* network with million edges. Subsequently, Tang et al. [17] and Nguyen et al. [18] further improve the efficiency of algorithms. These works all utilize the submodularity to accelerate approximation algorithms.

Seed minimization, as the dual problem of influence maximization, is to find a small seed set such that expected influence coverage exceeds a desired threshold. Chen [11] provide some strong negative results on seed minimization problem under fixed threshold model, which is a special case of general threshold model where its threshold function has breaking points. Goyal et al. [19] propose a greedy algorithm with a bicriteria approximation. Recently, Zhang et al. [20] study the probabilistic variant of seed minimization problem.

Due to the limitation of independent cascade and linear threshold model, *general threshold model* has been proposed [1, 8]. Not much is known about the general threshold model, other than it is NP-hard to approximate [1]. One special case which receives many attention is k -complex contagion where a node becomes active if at least k of its neighbours have been activated [21, 22, 23]. Gao et al. [24] make one step further of k -complex contagion model by considering the threshold comes from a probability distribution.

Optimization of non-submodular function is another interesting direction. Du et al. [25] introduce two techniques — restricted submodularity and shifted submodularity — to analyze greedy approximation of non-submodular functions. Recently, Horel et al. [26] study the problem of maximizing a set function that is very close to submodular. They assume that function values can be obtained from an oracle and focused on its query complexity. In our study, the local threshold functions are close to submodular and our target is to study its effect on the global influence spread function, which is the result of complex cascading behavior derived from the local threshold functions.

2 Preliminaries

For a set function $f : 2^V \rightarrow \mathbb{R}$, we say that it is *monotone* if $f(S) \leq f(T)$ for all $S \subseteq T \subseteq V$; we say that it is *submodular* if $f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$, for all $S \subseteq T \subseteq V$ and $v \in V \setminus T$. For a directed graph $G = (V, E)$, we use $N^{in}(v)$ to denote the in-neighbors of v in G . We now formally define the general threshold model used in the paper.

Definition 1 (General Threshold Model [1]). *In the general threshold model, for a social graph $G = (V, E)$, every node $v \in V$ has a threshold function $f_v : 2^{N^{in}(v)} \rightarrow [0, 1]$. The function $f_v(\cdot)$ should be monotone and $f_v(\emptyset) = 0$. Initially at time 0, each node $v \in V$ is in the inactive state and chooses θ_v uniformly at random from the interval $[0, 1]$. A seed set S_0 is also selected, and their states are set to be active. Afterwards, the influence propagates in discrete time steps. At time step $t \geq 1$, node v becomes active if $f_v(S_{t-1} \cap N^{in}(v)) \geq \theta_v$, where S_{t-1} is the set of active nodes by time step $t - 1$. The process ends when no new node becomes active in a step.*

General threshold model is one of the most important models in the influence maximization problem. Usually we focus on two properties of threshold function – submodularity and supermodularity. Submodularity can be understood as diminishing marginal returns when adding more nodes to the seed set. In contrast, supermodularity means increasing marginal returns. Given a seed set S , let $\sigma(S)$ denote the expected number of activated nodes after the process of influence propagation terminates, and we call $\sigma(S)$ the *influence spread* of S .

Submodularity is the key property that guarantees the performance of greedy algorithms [8]. In this paper, we would like to study the influence maximization with nearly submodular threshold function – ε -almost submodular function, or in short ε -AS.

Definition 2 (ε -Almost Submodular (ε -AS)). *A set function $f : 2^V \rightarrow \mathbb{R}$ is ε -almost submodular if there exists a submodular function f^{sub} defined on 2^V and for any subset $S \subseteq V$, $f^{sub}(S) \geq f(S) \geq (1 - \varepsilon)f^{sub}(S)$. Here ε is a small positive number.*

The definition of ε -almost submodular here is equivalent to "Approximate submodularity" defined in [26]. For an ε -almost submodular threshold function f_v , define its upper and lower submodular bound as \bar{f}_v and \underline{f}_v . Hence by definition, we have $\underline{f}_v = (1 - \varepsilon)\bar{f}_v$. Given the definition of ε -almost submodular function, we then model the *almost submodular graph*. In this paper, we consider the influence maximization problem based on this kind of graphs.

Definition 3 ((γ, ε) -Almost Submodular Graph). *Given fixed parameters $\gamma, \varepsilon \in [0, 1]$, we say that a graph with n ($n = |V|$) nodes is a (γ, ε) -Almost Submodular Graph (under the general threshold model), if there are at most n^γ nodes in the graph with ε -almost submodular threshold functions while other nodes have submodular threshold functions.*

Definition 4 (ε -ASIM). *Given a graph containing ε -almost submodular nodes and an input k , Influence Maximization problem on graph with ε -Almost Submodular nodes (ε -ASIM) is the problem to find k seed nodes such that the influence spread invoked by the k nodes is maximized.*

3 Inapproximability of ε -ASIM

In this section we show that it is in general hard to approximate the influence maximization problem even if there are only sublinear number of nodes with ε -almost submodular threshold functions. The main reason is that even a small number of nodes with ε -almost submodular threshold functions $f_v(\cdot)$ would cause the global influence spread function far from submodularity, making the maximization problem very difficult. The theorem below shows our hardness result.

Theorem 1. *For any small $\varepsilon > 0$ and any $\gamma \in (0, 1)$, there is no $1/n^{\frac{2}{\varepsilon}}$ -approximation influence maximization algorithm for all (γ, ε) -almost submodular graphs where $c = 3 + 3/\log \frac{2}{2-\varepsilon}$, unless $P=NP$.*

We first construct a probabilistic-AND gate gadget by amplifying the non-submodularity through a binary tree. Then we prove the lower bound of approximation ratio by the reduction from set cover problem. Due to page limits, we only sketch the main technique. The full proof can be found in the supplementary material.

Here we construct a basic gadget with input s_1, s_2 and output t (see Figure 1a). We assume that node t has two in-neighbours s_1, s_2 and the threshold function $g(\cdot)$ of t is ε -almost submodular: $g(S) = |S|/2$, when $|S| = 0$ or 2 ; $\frac{1-\varepsilon}{2}$, when $|S| = 1$.

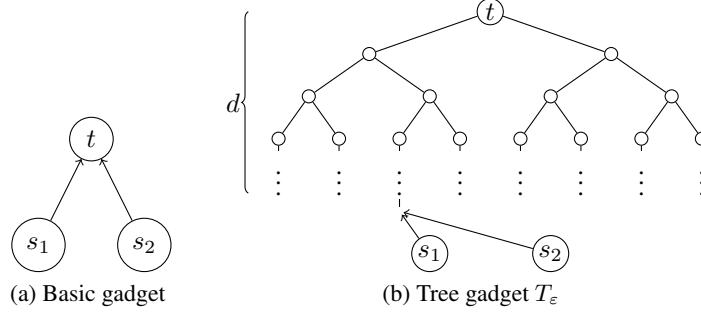


Figure 1: Diagrams of gadgets

Let $P_a(v)$ be the activation probability of node v in this case. This simple gadget is obviously far away from the AND gate, and our next step is to construct a more complex gadget with input node s_1, s_2 . We hope that the output node t is active only when both s_1, s_2 are active, and if only one of s_1 and s_2 is active, the probability that node t becomes active is close to 0. We call it a probabilistic-AND gate.

The main idea is to amplify the gap between submodularity and non-submodularity by binary tree (figure 1b). In this gadget T_ε with a complete binary tree, node t is the root of a full binary tree and each node holds a directed edge to its parent. For each leaf node v in the tree, both s_1, s_2 hold the directed edges towards it. The threshold function for each node in the tree is $g(\cdot)$ defined above while ε is the index of gadget T_ε . The depth of the tree is parameter d which will be determined later. We use v_i to denote a node of depth i (t is in depth 1). It is obviously that $P_a(t) = 1$ if both s_1, s_2 are activated, and $P_a(t) = 0$ if neither s_1 or s_2 is activated. Thus, we would like to prove, in case when only one of s_1, s_2 is activated, the activation probability becomes smaller for inner nodes in the tree.

Lemma 2. For gadget T_ε with depth d , the probability of activating output node t is less than $(\frac{2-\varepsilon}{2})^d$ when only one of s_1, s_2 is activated.

Proof. In this case, for leaf node v_d , we have $P_a(v_d) = \frac{1-\varepsilon}{2}$. Apparently, the probability of becoming activated for nodes with depth d are independent with each other. Given a basic gadget, if each of the two children nodes is activated with an independent probability p , then the parent node will be activated with probability

$$p^2 \times g(2) + 2p(1-p) \times g(1) + (1-p)^2 \times g(0) = p^2 + 2p(1-p) \frac{1-\varepsilon}{2} = p(1 - \varepsilon(1-p)).$$

So we have $P_a(v_i) \leq P_a(v_{i+1})(1 - \varepsilon(1 - P_a(v_{i+1})))$. Since $P_a(v_d) = \frac{1-\varepsilon}{2} < 1/2$, and $P_a(v_i) \leq P_a(v_{i+1})$ from above, we have $p_a(v_i) < 1/2$ for all i , and thus we can rewrite the recurrence as $P_a(v_i) \leq P_a(v_{i+1})(1 - \varepsilon/2)$. Hence for the gadget with depth d , the probability that node t becomes activated is $P_a(t) = P_a(v_1) \leq \frac{1-\varepsilon}{2} (\frac{2-\varepsilon}{2})^{d-1} < (\frac{2-\varepsilon}{2})^d$. \square

Lemma 2 shows that gadget T_ε is indeed a *probabilistic-AND gate* with two input nodes, and the probability that t is activated when only one of s_1 and s_2 is activated approaches 0 exponentially fast with the depth d . We say a gadget T_ε works well if output node t stay inactive when only one of the input nodes is activated.

By the similar method we construct multi-input-AND gates based on 2-input-AND gates. Finally, we show that if the influence maximization problem can be approximated beyond the ratio shown above, we can solve the set cover problem in polynomial time. The main idea is as follows. For any set cover instance, we will put all elements to be the input of our multi-input-probabilistic-AND gate, and connect the output with a large number of additional nodes. Thus, if k sets can cover all elements, all of those addition nodes will be activated, on contrast, if at least one of the elements cannot be covered, almost all of the additional nodes will remain inactive.

4 Approximation Algorithms

In the previous section, we show that influence maximization is hard to approximate when the number of ε -almost submodular nodes is sublinear but still a non-constant number. In this section, we discuss the situation where only small number of nodes hold ε -almost submodular threshold functions. We firstly provide a greedy algorithm for small number of non-submodular nodes which may not be ε -almost submodular, then, we restrict to the case of ε -almost submodular nodes.

4.1 Approximation Algorithm with Small Number of Non-submodular Nodes

In the case of ℓ ($\ell < k$) non-submodular nodes, we provide an approximate algorithm as follows. We first add these non-submodular nodes into the seed set, and then generate the rest of the seed set by the classical greedy algorithm. The proof of Theorem 3 can be found in the supplementary material.

Theorem 3. *Given a graph of n nodes where all nodes have submodular threshold functions except $\ell < k$ nodes, for influence maximization of k seeds with greedy scheme we can obtain a $(1 - e^{-\frac{k-\ell}{k}})$ -approximation ratio.*

4.2 Approximation Algorithm of ε -ASIM

In this section, we consider the case when all non-submodular nodes have ε -almost submodular threshold functions, and provide an approximation algorithm that allows more than k ε -almost submodular nodes, with the approximation ratio close to $1 - 1/e$ when ε is small. The main idea is based on the mapping between probability spaces.

Given a graph containing nodes with ε -almost submodular threshold functions, we simply set each node's threshold function to its submodular lower bound and then run classical greedy algorithm \mathcal{A} on this graph (Algorithm 1). Algorithm 1 takes the lower bounds of ε -almost submodular threshold functions as input parameters. The following theorem analyzes the performance of Algorithm 1.

Algorithm 1 Galg-L algorithm for Influence Maximization

Input: $G = (V, E)$, \mathcal{A} , $\{f_v\}$, $\{\underline{f}_v\}$, k

Output: Seed set S

- 1: set $S = \emptyset$
 - 2: replace each nodes v 's threshold function f_v with \underline{f}_v
 - 3: run algorithm \mathcal{A} on G with $\{\underline{f}_v\}$ and obtain S
 - 4: **return** S
-

Theorem 4. *Given a graph $G = (V, E)$, under the general threshold model, assuming that ℓ nodes have ε -almost submodular threshold functions and the other nodes have submodular threshold functions. Then the greedy algorithm **Galg-L** has approximation ratio of $(1 - \frac{1}{e})(1 - \varepsilon)^\ell$.*

Proof. Let V_e be the set of nodes with ε -almost submodular threshold functions. Without loss of generality, we assume $V_e = \{v_1, v_2, \dots, v_\ell\}$. Now consider two general threshold models $\overline{M}, \underline{M}$ with different threshold functions. Both models hold threshold functions $\{f_v\}$ for $v \in V - V_e$. For node v in V_e , $\overline{M}, \underline{M}$ hold $\{\overline{f}_v\}$ and $\{\underline{f}_v\}$ respectively.

In any threshold model, after we sample each node's threshold θ_v , the diffusion process becomes deterministic. A graph with threshold functions $\{f_v\}$ and sampled thresholds $\{\theta_v\}$ is called a *possible world* of the threshold model, which is similar to the live-edge graph in the independent cascade model. An instance of threshold model's possible world can be written as $\{\theta_{v_1}, \theta_{v_2}, \dots, \theta_{v_n}; f_{v_1}, f_{v_2}, \dots, f_{v_n}\}$. Here we build a one-to-one mapping from all \underline{M} 's possible worlds with $\theta_v \leq 1 - \varepsilon$ ($v \in V_e$) to all \overline{M} 's possible worlds:

$$\{\theta_{v_1}, \dots, \theta_{v_n}; f_{v_1}, \dots, f_{v_n}\} \leftrightarrow \left\{ \frac{\theta_{v_1}}{1-\varepsilon}, \dots, \frac{\theta_{v_\ell}}{1-\varepsilon}, \theta_{v_{\ell+1}}, \dots, \theta_{v_n}; \frac{f_{v_1}}{1-\varepsilon}, \dots, \frac{f_{v_\ell}}{1-\varepsilon}, f_{v_{\ell+1}}, \dots, f_{v_n} \right\}.$$

The above corresponding relation shows this one-to-one mapping between \underline{M} and \overline{M} . For any instance of \underline{M} 's possible world with $\theta_v \leq 1 - \varepsilon$ ($v \in V_e$), we amplify the threshold of node v in V_e to $\frac{\theta_v}{1-\varepsilon}$. At the same time, we amplify the corresponding threshold function by a factor of $\frac{1}{1-\varepsilon}$.

Obviously, this amplification process will not effect the influence process under this possible world, because for each $v \in V_e$, both its threshold value and the its threshold function are amplified by the same factor $1/(1 - \varepsilon)$. Furthermore, the amplified possible world is an instance of \overline{M} .

Expected influence can be computed by $\sigma(S) = \int_{\vec{\theta} \in [0,1]^n} \mathcal{D}(\vec{\theta}; \vec{f}, S) d_{\vec{\theta}}$, where $\mathcal{D}(\vec{\theta}; \vec{f}, S)$ is the deterministic influence size of seed set S under possible world $\{\vec{\theta}; \vec{f}\}$. We refer $\overline{M}, \underline{M}$'s expected influence size functions as $\overline{\sigma}, \underline{\sigma}$. We define $\vec{\theta} \in [0, 1]^n$ as the vector of n nodes threshold, and $\vec{\theta}_e \in [0, 1]^\ell, \vec{\theta}' \in [0, 1]^{n-\ell}$ are the threshold vectors of V_e and $V - V_e$. Besides, the threshold functions of V_e and $V - V_e$ will be represented as \vec{f}_e, \vec{f}' . A possible world is symbolized as $\{\vec{\theta}_e, \vec{\theta}'; \vec{f}_e, \vec{f}'\}$. For any seed set S , we have

$$\begin{aligned} \underline{\sigma}(S) &= \int_{\vec{\theta} \in [0,1]^n} \mathcal{D}(\vec{\theta}; \vec{f}, S) d_{\vec{\theta}} \\ &\geq \int_{\vec{\theta}_e \in [0,1-\varepsilon]^\ell} \int_{\vec{\theta}' \in [0,1]^{n-\ell}} \mathcal{D}((\vec{\theta}_e, \vec{\theta}'); \vec{f}, S) d_{\vec{\theta}_e} d_{\vec{\theta}'} \\ &= (1 - \varepsilon)^\ell \int_{\frac{\vec{\theta}_e}{1-\varepsilon} \in [0,1]^\ell} \int_{\vec{\theta}' \in [0,1]^{n-\ell}} \mathcal{D}((\vec{\theta}_e, \vec{\theta}'); \vec{f}, S) d_{\frac{\vec{\theta}_e}{1-\varepsilon}} d_{\vec{\theta}'} \\ &= (1 - \varepsilon)^\ell \int_{\frac{\vec{\theta}_e}{1-\varepsilon} \in [0,1]^\ell} \int_{\vec{\theta}' \in [0,1]^{n-\ell}} \mathcal{D}((\frac{\vec{\theta}_e}{1-\varepsilon}, \vec{\theta}'); (\frac{\vec{f}_e}{1-\varepsilon}, \vec{f}'), S) d_{\frac{\vec{\theta}_e}{1-\varepsilon}} d_{\vec{\theta}'} \\ &= (1 - \varepsilon)^\ell \int_{\vec{\theta} \in [0,1]^n} \mathcal{D}(\vec{\theta}; (\frac{\vec{f}_e}{1-\varepsilon}, \vec{f}'), S) d_{\vec{\theta}} \\ &= (1 - \varepsilon)^\ell \overline{\sigma}(S). \end{aligned}$$

The third equality utilizes our one-to-one mapping, in particular $\mathcal{D}((\vec{\theta}_e, \vec{\theta}'); \vec{f}, S) = \mathcal{D}((\frac{\vec{\theta}_e}{1-\varepsilon}, \vec{\theta}'); (\frac{\vec{f}_e}{1-\varepsilon}, \vec{f}'), S)$ for $\frac{\vec{\theta}_e}{1-\varepsilon} \in [0, 1]^\ell$, because they follow the same deterministic propagation process. Hence given a seed set S , the respective influence sizes in model $\overline{M}, \underline{M}$ satisfy the relation $\underline{\sigma}(S) \geq (1 - \varepsilon)^\ell \overline{\sigma}(S)$.

Let σ be the expected influence size function of the original model, and assume that the optimal solution for $\overline{\sigma}, \sigma, \underline{\sigma}$ are $\overline{S}^*, S^*, \underline{S}^*$ respectively. Apparently, $\overline{\sigma}(\overline{S}^*) \geq \sigma(S^*)$ since for every node v , $\overline{f}_v \geq f_v$. According to the previous analysis, we have $\sigma(\underline{S}^*) \geq \underline{\sigma}(\overline{S}^*) \geq (1 - \varepsilon)^\ell \overline{\sigma}(\overline{S}^*)$. Hence for output S^A of the greedy algorithm for optimizing $\underline{\sigma}$, we have approximation ratio

$$\sigma(S^A) \geq \underline{\sigma}(S^A) \geq (1 - \frac{1}{e}) \underline{\sigma}(\underline{S}^*) \geq (1 - \frac{1}{e})(1 - \varepsilon)^\ell \overline{\sigma}(\overline{S}^*) \geq (1 - \frac{1}{e})(1 - \varepsilon)^\ell \sigma(S^*).$$

The theorem holds. \square

If we replace threshold functions by their upper bound and run the greedy algorithm, we obtain **Galg-U**. With similar analysis, **Galg-U** also holds approximation ratio of $(1 - \frac{1}{e})(1 - \varepsilon)^\ell$ on graphs with ℓ ε -almost submodular nodes. The novel technique used to prove approximation ratio is similar to the *sandwich approximation* in [27]. But their approximation ratio relies on instance-dependent influence sizes, while we utilize mapping of probabilistic space to provide instance-independent approximation ratio.

5 Experiments

In addition to the theoretical analysis, we are curious about the performance of greedy algorithms **Galg-U**, **Galg-L** on real networks with non-submodular nodes. Our experiments run on a machine with two 2.4GHz Intel(R) Xeon(R) E5-2620 CPUs, 4 processors (24 cores), 128GB memory and 64bit Ubuntu 14.04.1. All algorithms tested in this paper are written in C++ and compiled with g++ 4.8.4. Some algorithms are implemented with multi-thread to decrease the running time.

5.1 Experiment setup

Datasets. We conduct experiments on three real networks. The first network is NetHEPT, an academic collaboration network extracted from "High Energy Physics - Theory" section of arXiv (<http://www.arXiv.org>) used by many works [7, 13, 14, 18, 19]. NetHEPT is an undirected network with 15233 nodes and 31376 edges, each node represents an author and each edge represents that two authors collaborated on a paper. The second one is Flixster, an American movie rating social site. Each node represents a user, and directed edge (u, v) means v rated the same movie shortly after u

did. We select topic 3 with 29357 nodes and 174939 directed edges here. The last one is the DBLP dataset, which is a larger collaboration network mined from the computer science bibliography site DBLP with 654628 nodes and 1990259 undirected edges [13]. We process its edges in the same way as the NetHEPT dataset.

Propagation Models. We adapt general threshold model in this paper. Our Galg-U, Galg-L are designed on submodular upper and lower bounds, respectively. Since directly applying greedy scheme on graphs with submodular threshold function is time-consuming, we assign the submodular threshold function and submodular upper bound of ε -AS function as linear function here: $f_v(S) = |S|/d(v)$, where $d(v)$ is the in-degree of v . This makes the corresponding model an instance of the linear-threshold model, and thus the greedy algorithm can be accelerated with Reverse Reachable Set (RRset) technique [16].

We construct two different ε -almost submodular threshold functions in this paper: (1) a power function $\frac{|S|^\beta}{d(v)^\beta}$ with β satisfying $\frac{1}{d(v)^\beta} = \frac{1}{d(v)}(1 - \varepsilon)$; (2) $f_v(S) = \frac{|S|}{d(v)}(1 - \varepsilon)$ for $|S| \leq 2$ and $|S|/d(v)$ otherwise. The former ε -almost submodular function is a supermodular function. The supermodular phenomenon has been observed in Flickr [10]. The second ε -almost submodular function is just dropping down the original threshold function for the first several nodes, which is consistent with the phenomenon observed in LiveJournal [9]. We call them ε -AS-1 and ε -AS-2 functions respectively.

Algorithms. We test our approximation Algorithm 1 and other baseline algorithms on the graphs with ε -almost submodular nodes.

- **TIM-U, TIM-L:** Tang et al. [16] proposed a greedy algorithm TIM^+ accelerated with Reverse Reachable Set (RRset). The running time of TIM^+ is $O(k(m+n)\log n)$ on graphs with n nodes and m edges. RRset can be sampled in live-edge graph of IC model, and with some extension we can sample RRset under Triggering model [1]. LT model also belongs to Triggering model, but General Threshold model with non-submodular threshold functions does not fall into the category of Triggering model. Thus TIM^+ can not be directly applied on original graphs with non-submodular nodes. In our experiments, we choose ε -AS-1 and ε -AS-2 thresholds to ensure that TIM^+ can run with their upper or lower bound. We then run Algorithm 1 with TIM^+ as input. Algorithm Galg-L based on TIM^+ can be written in short as TIM-L. By using the upper bound we obtain TIM-U.
- **Greedy:** We can still apply the naive greedy scheme on graph with ε -almost submodular nodes and generate results without theoretical guarantee. The naive greedy algorithm is time consuming, with running time is $O(k(m+n)n)$.
- **High-degree:** High-degree outputs seed set according to the decreasing order of the out-degree.
- **PageRank:** PageRank is widely used to discover nodes with high influence. The insight of PageRank is that important nodes point to important nodes. In this paper, The transition probability on edge $e = (u, v)$ is $1/d(u)$. We set restart probability as 0.15 and use the power method to compute the PageRank values. Finally PageRank outputs nodes with top PageRank values.
- **Random:** Random simply selects seeds randomly from node set.

Experiment methods. The datasets provide the structure of network, and we first assume each node holds linear threshold function as described above. In each experiment, we randomly sample some nodes with in-degree greater than 2, and assign those nodes with our ε -almost submodular functions, ε -AS-1 or ε -AS-2. Since the naive greedy algorithm is quite time-consuming, we just run it on NetHEPT.

5.2 Experiment results

Results on NetHEPT. Our first set of experiments focuses on the NetHEPT dataset with the aim of comparing TIM-U, TIM-L and Greedy. TIM-U, TIM-L have theoretical guarantee, but the approximation ratio is low when the graph contains a considerable number of ε -AS nodes. Figure 2 shows the influence size of each method, varying from 1 to 100 seeds. Figure 2a and 2b are results conducted on constructed graph with ε -AS-1 nodes. Observe that TIM-U, TIM-L slightly outperform Greedy in all cases. Compared with results of 3000 ε -AS nodes, influence of output seeds drops obviously in graph with 10000 ε -AS nodes. But the ratio that TIM-U, TIM-L exceed PageRank

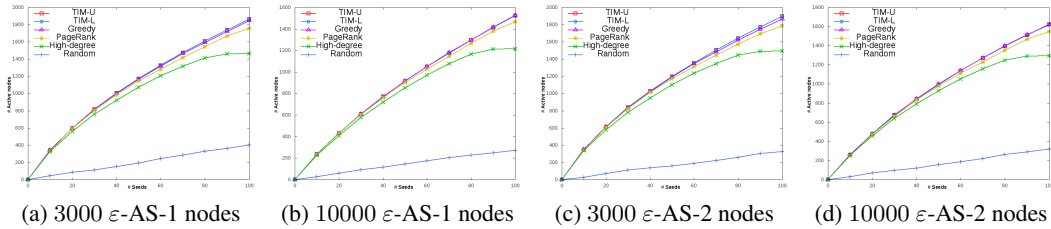


Figure 2: Results of IM on NetHEPT with $\varepsilon = 0.2$

increases with rising fraction of ε -AS nodes. In particular, ε -AS-1 is indeed supermodular, TIM-U, TIM-L beats Greedy even when many nodes have supermodular threshold functions.

We remark that TIM-U, TIM-L and Greedy outperform other baseline algorithms significantly. When $k = 100$, TIM-U is 6.1% better compared with PageRank and 27.2% better compared with High-degree. When conducted with ε -AS-2 function, Figure 2c and 2d report that TIM-U, TIM-L and Greedy still perform extremely well. Influence size conducted on graphs with ε -AS-2 function is better than those with ε -AS-1 function. This is what we expect: supermodular function is harder to handle among the class of ε -almost submodular functions.

Another thing to notice is that TIM-U, TIM-L can output seeds on NetHEPT within seconds, while it takes weeks to run the naive greedy algorithm. With RRsets technique, TIM⁺ dramatically reduces the running time. The ε -almost submodular functions selected here ensure that TIM⁺ can be invoked. Since TIM-U, TIM-L match the performance of Greedy while TIM-U, TIM-L are scalable, we do not run Greedy in the following larger datasets.

Results on Flixster. Figure 3 shows the results of experiments conducted on Flixster with

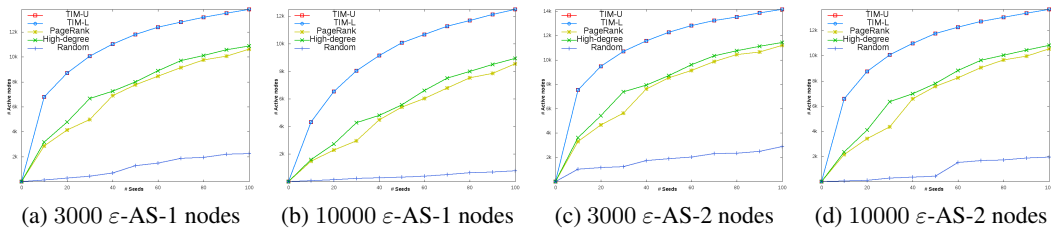


Figure 3: Results of IM on Flixster with $\varepsilon = 0.2$

$\varepsilon = 0.2$. We further evaluate algorithms by Flixster with $\varepsilon = 0.4$ (see Figure 4). Observe that TIM-U, TIM-L outperform other heuristic algorithms in all cases. Compared with PageRank, 30%, 46.3%, 26%, 29.7% improvement are observed in the four experiments in Figure 3. TIM-U performs closely to TIM-L consistently. The improvement is larger than that in NetHEPT. The extra improvement might due to more complex network structure. The average degree is 5.95 in Flixster, compared to 2.05 in NetHEPT. In dense network, nodes may be activated by multiple influence chains, which makes influence propagates further from seeds. Baseline algorithms only pay attention to the structure of the network, hence they are defeated by TIM-U, TIM-L that focus on influence spread. The more ε -AS nodes in network, the more improvement is obtained.

When we set ε as 0.4, Figure 4 shows that TIM-U is 37.6%, 74.2%, 28%, 35.6% better than PageRank respectively. Notice that the gap between the performances of TIM-U and PageRank increases as ε increases. In Flixster dataset, we observe that TIM-U, TIM-L hold greater advantage in case of larger number of ε -AS nodes and larger ε .

Results on DBLP. For DBLP dataset, the results are shown in Figure 5. TIM-U and TIM-L are still the best algorithms according to performance. But PageRank and High-degree also performs well, just about 2.6% behind TIM-U and TIM-L. DBLP network has many nodes with large degree, which correspond to those active scientists. Once such active authors are activated, the influence will increase significantly. This may partly explain why TIM-U, TIM-L perform similarly to PageRank.

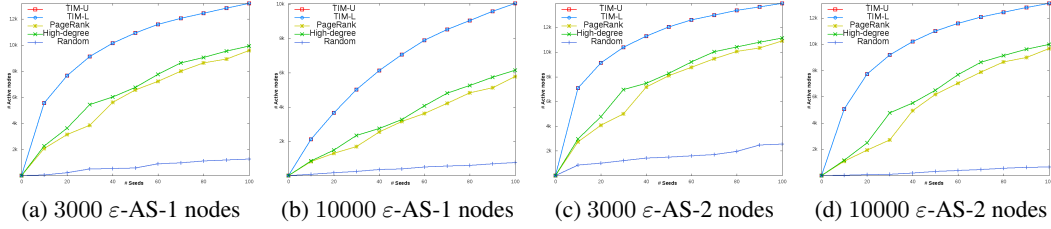


Figure 4: Results of IM on Flixster with $\varepsilon = 0.4$

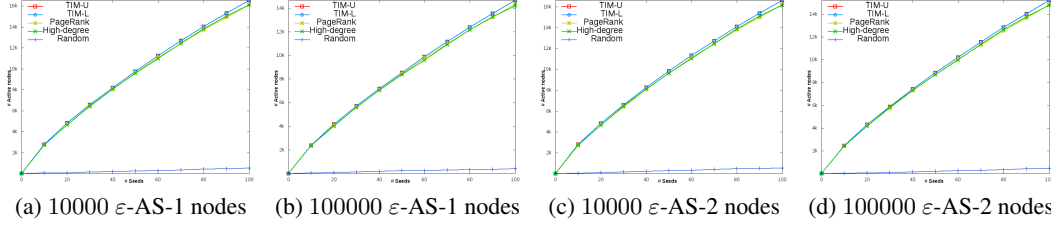


Figure 5: Results of IM on DBLP with $\varepsilon = 0.2$

6 Conclusion and Future Work

In this paper, we study the influence maximization problem on propagation models with non-submodular threshold functions, which are different from most of existing studies where the threshold functions and the influence spread function are both submodular. We investigate the problem by studying a special case — the ε -almost submodular threshold function. We first show that influence maximization problem is still hard to approximate even when the number of ε -almost submodular nodes is sub-linear. Next we provide a greedy algorithm based on submodular lower bounds of threshold function to handle the graph with small number of ε -almost submodular nodes and show its theoretical guarantee. We further conduct experiments on real networks and compare our algorithms with other baselines to evaluate our algorithms in practice. Experimental results show that our algorithms not only have good theoretical guarantees on graph with small number of ε -almost submodular nodes, they also perform well on graph with a fairly large fraction of ε -almost submodular nodes.

Our study mainly focuses on handling ε -almost submodular threshold functions. One future direction is to investigate models with arbitrary non-submodular threshold functions. Another issue is that the greedy algorithms we propose are slow when the submodular upper bound or lower bound of threshold function do not correspond to the Triggering model. It remains open whether we could utilize RRset or other techniques to accelerate our algorithms under this circumstance. How to accelerate the naive greedy process with arbitrary submodular threshold functions is another interesting direction.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China Grant 61433014, 61502449, 61602440, the 973 Program of China Grants No. 2016YFB1000201.

References

- [1] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD*, pages 137–146. ACM, 2003.
- [2] Mani Subramani and Balaji Rajagopalan. Knowledge-sharing and influence in online social networks via viral marketing. *Communications of The ACM*, 46(12):300–307, 2003.
- [3] Wei Chen, Fu Li, Tian Lin, and Aviad Rubinfeld. Combining traditional marketing and viral marketing with amphibious influence maximization. In *ACM Conference on Economics and Computation*, 2015.
- [4] Cigdem Aslay, Wei Lu, Francesco Bonchi, Amit Goyal, and Laks V S Lakshmanan. Viral marketing meets social advertising: ad allocation with minimum regret. In *Proceedings of The VLDB Endowment*, pages 814–825, 2015.
- [5] Biao Wang, Ge Chen, Luoyi Fu, Li Song, Xinbing Wang, and Xue Liu. Drimux: Dynamic rumor influence minimization with user experience in social networks. In *AAAI’16*, pages 791–797, 2016.
- [6] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne M Vanbriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *ACM Knowledge Discovery and Data Mining*, pages 420–429, 2007.
- [7] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD*. ACM, 2009.
- [8] Elchanan Mossel and Sebastien Roch. On the submodularity of influence in social networks. In *STOC’07*, pages 128–134, 2007.
- [9] Lars Backstrom, Dan Huttenlocher, Jon Kleinberg, and Xiangyang Lan. Group formation in large social networks: membership, growth, and evolution. In *KDD’06*, pages 44–54. ACM, 2006.
- [10] Yang Yang, Jia Jia, Boya Wu, and Jie Tang. Social role-aware emotion contagion in image social networks. In *AAAI*, pages 65–71, 2016.
- [11] Ning Chen. On the approximability of influence in social networks. In *SODA’08*, 2008.
- [12] Shishir Bharathi, David Kempe, and Mahyar Salek. Competitive influence maximization in social networks. In *International Workshop on Web and Internet Economics*, pages 306–311. Springer, 2007.
- [13] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD’10*, 2010.
- [14] Amit Goyal, Wei Lu, and Laks V. S. Lakshmanan. SIMPATH: An Efficient Algorithm for Influence Maximization under the Linear Threshold Model. In *ICDM’11*, pages 211–220, 2011.
- [15] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. Maximizing social influence in nearly optimal time. In *SODA’14*, pages 946–957. ACM-SIAM, 2014.
- [16] Youze Tang, Xiaokui Xiao, and Yanchen Shi. Influence maximization: near-optimal time complexity meets practical efficiency. In *SIGMOD’14*, 2014.
- [17] Youze Tang, Yanchen Shi, and Xiaokui Xiao. Influence maximization in near-linear time: A martingale approach. In *SIGMOD’15*, pages 1539–1554. ACM, 2015.
- [18] H. T. Nguyen, M. T. Thai, and T. N. Dinh. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *SIGMOD’16*, pages 695–710. ACM, 2016.
- [19] Amit Goyal, Francesco Bonchi, Laks VS Lakshmanan, and Suresh Venkatasubramanian. On minimizing budget and time in influence propagation over social networks. *Social Network Analysis and Mining*, pages 1–14, 2012.

- [20] Peng Zhang, Wei Chen, Xiaoming Sun, Yajun Wang, and Jialin Zhang. Minimizing seed set selection with probabilistic coverage guarantee in a social network. In *KDD'14*, pages 1306–1315, 2014.
- [21] Golnaz Ghasemiefteh, Roozbeh Ebrahimi, and Jie Gao. Complex contagion and the weakness of long ties in social networks: revisited. In *ACM Conference on Electronic Commerce*, 2013.
- [22] Roozbeh Ebrahimi, Jie Gao, Golnaz Ghasemiefteh, and Grant Schoenebeck. Complex contagions in kleinberg’s small world model. In *ITCS'15*, 2015.
- [23] Wei Chen, Qiang Li, Xiaoming Sun, and Jialin Zhang. The routing of complex contagion in kleinberg’s small-world networks. In *International Computing and Combinatorics Conference*, pages 307–318, 2016.
- [24] Jie Gao, Golnaz Ghasemiefteh, Grant Schoenebeck, and Fang-Yi Yu. General threshold model for social cascades: Analysis and simulations. In *ACM Conference on Economics and Computation*, 2016.
- [25] Ding-Zhu Du, Ronald L Graham, Panos M Pardalos, Peng-Jun Wan, Weili Wu, and Wenbo Zhao. Analysis of greedy approximations with nonsubmodular potential functions. In *SODA'08*, pages 167–175, 2008.
- [26] Thibaut Horel and Yaron Singer. Maximization of approximately submodular functions. In *NIPS'16*, pages 3045–3053, 2016.
- [27] Wei Lu, Wei Chen, and Laks VS Lakshmanan. From competition to complementarity: comparative influence diffusion and maximization. *Proceedings of the VLDB Endowment*, 9(2):60–71, 2015.
- [28] Nicola Barbieri, Francesco Bonchi, and Giuseppe Manco. Topic-aware social influence propagation models. In *ICDM'12*, 2012.
- [29] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming*, 14:265–294, 1978.

Supplementary Material

A Missing Proof of Theorem 1

Proof. We have described the construction of gadget T_ε , we can further construct n -input trees T_ε^n with gadget T_ε — a multi input AND gate. Given n input nodes s_1, s_2, \dots, s_n , we can use s_0 and s_1 as input nodes of T_ε and set output node as s_{12} . And then combine s_{12} and s_3 with gadget T_ε to obtain output s_{123} . Finally we obtain ultimate output node $s_{12\dots n}$. With high probability $s_{12\dots n}$ will not be activated if not all input nodes s_1, s_2, \dots, s_n are activated. We calculate the probability that T_ε^n works well. Since each T_ε is a binary tree with depth d , according to Lemma 2 each T_ε breaks the AND gate rule with probability at most $(\frac{2-\varepsilon}{2})^d$. With Union Bound, we can say that T_ε^n is an AND gate with probability at least $1 - n(\frac{2-\varepsilon}{2})^d$. Besides, a T_ε^n have $n \times (2^d - 1) = n2^d - n$ nodes.

We consider a variant of set cover problems and show that if influence maximization problem can be approximated beyond the ratio shown above, then we can solve the set cover problem. Let e_1, e_2, \dots, e_n denote the nodes corresponding to the n elements. Let s_1, s_2, \dots, s_m denote the nodes corresponding to the m sets. We can assume that $m < n$. When $m \geq n$ we could add m dummy nodes that contained by all m sets, the solution of the newly created graph is the same with original graph. Note that in this reduction, n is the number of elements in the Set Cover instance. We use N to denote the size of the graph constructed from the Set Cover instance.

Figure 6 shows the graph structure for the reduction. Each set node s_i holds a directed edge towards the element nodes it covers and each element node e_j has threshold function with $f_{e_j}(1) = 1$, which means that e_j becomes activated when at least one of the nodes corresponding to sets covering it is activated. We add n^α copies of T_ε^n trees, each of which has e_1, e_2, \dots, e_n as the input nodes, and x_i as the output node. Moreover, for a large positive constant β , we add n^β children $c_1, c_2, \dots, c_{n^\beta}$ for each node x_k . Each child node c_l also has threshold function with $f_{c_l}(1) = 1$. That is, these c_l nodes have submodular threshold functions. For the constructed graph, only nodes in T_ε^n (nodes painted red) holds ε -almost submodular threshold function. We can adjust the ratio of ε -almost submodular nodes with the assignment of α, β . Intuitively, to make the approximation ratio to be proved in the hardness result at a low value of $1/N^a$ for some $a > 0$, we want to increase α to create a larger gap between the existence and non-existence of a set cover. In particular, the fraction of nodes that could be activated by k seeds in the case of no set cover should be less than the approximation ratio $1/N^a$. However, increasing α would generate more ε -almost submodular nodes. To keep the fraction of ε -almost submodular nodes small, we need to increase β . But increasing β would increase the total size of the graph N , decreasing the approximation ratio $1/N^a$. Therefore, we need to find the right balance between α and β to achieve what we claim in Theorem 1. The following is the detailed analysis of setting parameters α and β and the depth of trees T_ε^n .

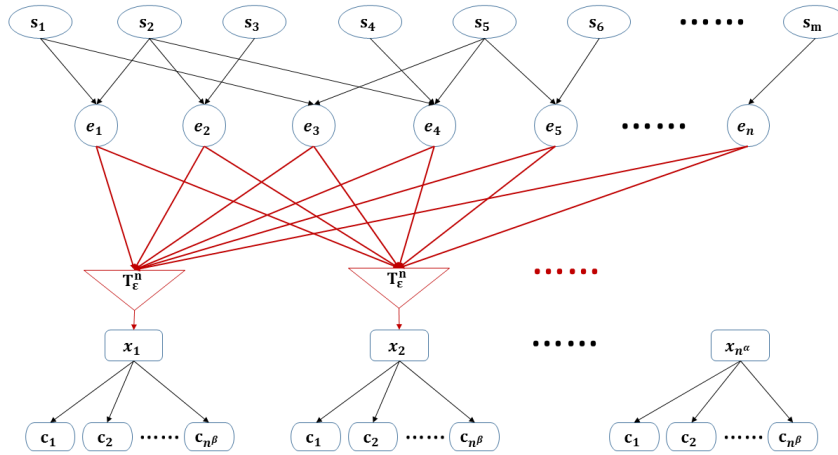


Figure 6: Reduction structure from a Set Cover instance to an ε -ASIM instance

Here we calculate the depth of T_ε in order to make sure that the $n^\alpha T_\varepsilon^n$ are all AND gate with high probability. Again we apply Union Bound to calculate the probability upper bound of at least one T_ε^n fails — $n^{1+\alpha}(\frac{2-\varepsilon}{2})^d$. We set $d = (1 + \alpha + \lambda) \log n / \log \frac{2}{2-\varepsilon}$, so all the T_ε^n observe the rule of AND gate with probability at least $1 - n^{-\lambda}$.

If we can find k sets that cover all the n elements, then we just select the k nodes corresponding to the k sets as seed nodes. Then the seed nodes will activate nodes e_1, e_2, \dots, e_n and then all nodes in gadgets T_ε^n will become activated. Finally, all the child nodes of $x_1, x_2, \dots, x_{n^\alpha}$ become activated, and totally $k + n + n^{\alpha+1}(2^d - 1) + n^{\alpha\beta} \geq n^{\alpha\beta}$ nodes become activated in total, while the graph consists of $N = m + n + n^{\alpha+1}(2^d - 1) + n^{\alpha\beta}$ nodes, nearly all nodes are activated. On the other hand, if there is no solution of size k for this set cover problem, we can not activate all element nodes e_j . During the influence spread of a given target nodes, with probability at least $1 - n^{-\lambda}$ none of $x_1, x_2, \dots, x_{n^\alpha}$ will become activated. Under this circumstance, we can select nodes labeled with x as seeds or just try to activate more nodes e_j . Totally we can just activate at most $k + n + kn^\beta$ nodes if we choose x_k as seeds. Otherwise we can assume that at most $n - 1$ elements are covered and all gadget nodes are activated. In this case $k + n - 1 + n^{\alpha+1}(2^d - 1)$ nodes will become active if all probabilistic AND gate work well. On the other hand if these T_ε^n fails with probability at most $n^{-\lambda}$, we just assume that all the nodes will become activated eventually. We can set $n^\beta = n^{\alpha-1} \cdot 2^d \cdot n^\delta$ for $\delta > 0$ to ensure that the fraction of gadget nodes and dummy nodes is slightly less than $n^{\frac{1}{\alpha-1}}$. When n is large enough the influence size we obtain that influence size is less than $kn^\beta \leq n^{\beta+1}$ or $n^{\alpha+1}2^d$ w.h.p. when Set Cover is not solved. For any influence maximization algorithm, there exists a graph of N nodes where at most $n^{\alpha+1}(2^d - 1)$ nodes have ε -almost submodular threshold function, for any result obtained, with probability at least $1 - n^{-\lambda}$, the influence size is less than $1/n^{\alpha-1}$ of $\sigma(S^*)$, unless we can solve Set Cover problems within polynomial time. Here $N \geq n^{\alpha+\beta}$, $d = (1 + \alpha + \lambda) \log n / \log \frac{2}{2-\varepsilon}$, $\beta = (\alpha + 1) + (1 + \alpha + \lambda) / \log \frac{2}{2-\varepsilon} + \delta$. We can substitute the parameters into the conclusion, and obtain that $\forall \alpha > 1, \lambda > 0, \delta > 0$, there exist $b = 1 / \log \frac{2}{2-\varepsilon}$, $\varphi = \frac{\min\{\alpha-1, \lambda\}}{2\alpha+\delta-1+b(1+\alpha+\lambda)}$, and $\gamma = \frac{\alpha+1+b(1+\alpha+\lambda)}{2\alpha+\delta-1+b(1+\alpha+\lambda)}$, for any influence maximization algorithm based on (γ, ε) -almost submodular graphs, there exists a graph instance such that the approximation ratio cannot be higher than $N^{-\varphi}$, unless P=NP. Notice that $\varphi \geq \frac{\gamma}{3+3b}$ when we set $\alpha = \lambda + 1$ and $\lambda \geq 1$, and γ ranges in $(0, 1)$, hence the theorem follows. \square

B Missing Proof of Theorem 3

The result follows the general framework on submodular functions laid out in the original work [29]. For convenience, we reproduce the proof specific to our Theorem 3 below.

Proof. Assuming that S^* is the optimal set that maximizes the expectation of influenced nodes, V_e are the set of ℓ nodes with non-submodular threshold function. let $S_{V_e}^* = S^* \cup V_e$. Apparently, $\sigma(S_{V_e}^*) \geq \sigma(S^*)$ since $\sigma(\cdot)$ is monotone. Since nodes in V_e have non-submodular threshold functions, so we can not directly apply greedy scheme to find the seed set. But we can first add V_e to the seed set and therefore the influence function $\sigma(S \cup V_e)$ is submodular when $S \subseteq V - V_e$. In this case, we obtain a greedy solution $S_{V_e, k}^g$ with adding extra k nodes to original seed set V_e with greedy scheme.

We assume that $S_{V_e}^* = V_e \cup \{s_1, s_2, \dots, s_{k'}\}$, $k' \leq k$, $S_i = \{s_1, s_2, \dots, s_i\}$, where $s_i \notin V_e$. Indeed, according to the proof of approximation ratio of $1 - 1/e$, for $i \leq k$, we have

$$\begin{aligned} & \sigma(S_{V_e}^*) \\ & \leq \sigma(S_{V_e}^* \cup S_{V_e, i}^g) \\ & = \sigma(S_{V_e, i}^g) + \sum_{j=1}^{k'} (\sigma(S_{V_e, i}^g \cup S_j) - \sigma(S_{V_e, i}^g \cup S_{j-1})) \\ & \leq \sigma(S_{V_e, i}^g) + \sum_{j=1}^{k'} (\sigma(S_{V_e, i}^g \cup \{s_j\}) - \sigma(S_{V_e, i}^g)) \\ & \leq \sigma(S_{V_e, i}^g) + k \cdot (\sigma(S_{V_e, i+1}^g) - \sigma(S_{V_e, i}^g)) \end{aligned}$$

The first line follows from monotonicity of σ , third line follows from the submodularity of σ , fourth line holds because $S_{V_e, i+1}^g$ is built greedily from $S_{V_e, i}^g$ in order to maximize marginal benefit. Hence

$$\begin{aligned} \sigma(S_{V_e}^*) - \sigma(S_{V_e, i}^g) & \leq k \cdot (\sigma(S_{V_e, i+1}^g) - \sigma(S_{V_e, i}^g)) \\ \sigma(S_{V_e}^*) - \sigma(S_{V_e, i+1}^g) & \leq (1 - \frac{1}{k})(\sigma(S_{V_e}^*) - \sigma(S_{V_e, i}^g)) \end{aligned}$$

When we apply greedy scheme for $k - \ell$ steps,

$$\begin{aligned}\sigma(S_{V_e}^*) - \sigma(S_{V_e, k-\ell}^g) &\leq (1 - \frac{1}{k})^{k-\ell} (\sigma(S_{V_e}^*) - \sigma(V_e)) \leq (1 - \frac{1}{k})^{k-\ell} \sigma(S_{V_e}^*) \\ \sigma(S_{V_e, k-\ell}^g) &\geq (1 - (1 - \frac{1}{k})^{k-\ell}) \sigma(S_{V_e}^*) \geq (1 - e^{-\frac{k-\ell}{k}}) \sigma(S^*)\end{aligned}$$

Up till now we obtain that $\sigma(S_{V_e, k-\ell}^g) \geq (1 - e^{-\frac{k-\ell}{k}}) \sigma(S^*)$. Hence we can first add all nodes without submodular threshold function to initial seed set and then apply greedy scheme to obtain a solution with theoretical approximation ratio. The theorem follows. \square