

Limitations of Greed: Influence Maximization in Undirected Networks Re-visited*

Grant Schoenebeck
University of Michigan
Ann Arbor, MI
schoeneb@umich.edu

Biaoshuai Tao
University of Michigan
Ann Arbor, MI
bstao@umich.edu

Fang-Yi Yu
University of Michigan
Ann Arbor, MI
fayu@umich.edu

ABSTRACT

We consider the influence maximization problem (selecting k seeds in a network maximizing the expected total influence) on undirected graphs under the linear threshold model. On the one hand, we prove that the greedy algorithm always achieves a $(1 - (1 - 1/k)^k + \Omega(1/k^3))$ -approximation, showing that the greedy algorithm does slightly better on undirected graphs than the generic $(1 - (1 - 1/k)^k)$ bound which also applies to directed graphs. On the other hand, we show that substantial improvement on this bound is impossible by presenting an example where the greedy algorithm can obtain at most a $(1 - (1 - 1/k)^k + O(1/k^{0.2}))$ approximation.

This result stands in contrast to the previous work on the independent cascade model. Like the linear threshold model, the greedy algorithm obtains a $(1 - (1 - 1/k)^k)$ -approximation on directed graphs in the independent cascade model. However, Khanna and Lucier [24] showed that, in undirected graphs, the greedy algorithm performs substantially better: a $(1 - (1 - 1/k)^k + c)$ approximation for constant $c > 0$. Our results show that, surprisingly, no such improvement occurs in the linear threshold model.

Finally, we show that, under the linear threshold model, the approximation ratio $(1 - (1 - 1/k)^k)$ is tight if 1) the graph is directed or 2) the vertices are weighted. In other words, under either of these two settings, the greedy algorithm cannot achieve a $(1 - (1 - 1/k)^k + f(k))$ -approximation for any positive function $f(k)$. The result in setting 2) is again in a sharp contrast to Khanna and Lucier’s $(1 - (1 - 1/k)^k + c)$ -approximation result for the independent cascade model, where the $(1 - (1 - 1/k)^k + c)$ approximation guarantee can be extended to the setting where vertices are weighted.

We also discuss extensions to more generalized settings including those with edge-weighted graphs.

CCS CONCEPTS

• **Theory of computation** → **Social networks**; • **Mathematics of computing** → *Approximation algorithms*;

KEYWORDS

social network, influence maximization, linear threshold model, greedy algorithm

*A full version of this paper is available at <https://arxiv.org/abs/2002.11679>. Grant Schoenebeck, Biaoshuai Tao, and Fang-Yi Yu are pleased to acknowledge the support of National Science Foundation AitF #1535912 and CAREER #1452915.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand. © 2020 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

ACM Reference Format:

Grant Schoenebeck, Biaoshuai Tao, and Fang-Yi Yu. 2020. Limitations of Greed: Influence Maximization in Undirected Networks Re-visited. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020)*, Auckland, New Zealand, May 9–13, 2020, IFAAMAS, 9 pages.

1 INTRODUCTION

Viral marketing is an advertising strategy that gives the company’s product to a certain number of users (the seeds) for free such that the product can be promoted through a cascade process in which the product is recommended to these users’ friends, their friends’ friends, and so on. The *influence maximization problem* (INFMAX) is an optimization problem which asks which seeds one should give the product to; that is, given a graph, a *diffusion model* defining how each node is infected by its neighbors, and a limited budget k , how to pick k seeds such that the total number of infected vertices in this graph at the end of the cascade is maximized. For INFMAX, nearly all the known algorithms are based on a greedy algorithm which iteratively picks the seed that has the largest marginal influence. Some of them improve the running time of the original greedy algorithm by skipping vertices that are known to be suboptimal [18, 25], while the others improve the scalability of the greedy algorithm by using more scalable algorithms to approximate the expected total influence [4, 12, 30, 37, 38] or computing a score of the seeds that is closely related to the expected total influence [9–11, 15, 19, 21, 35]. Therefore, improving the approximation guarantee of the standard greedy algorithm improves the approximation guarantees of most INFMAX algorithms in the literature in one shot!

Two diffusion models that have been studied almost exclusively are *the linear threshold model* and *the independent cascade model*, which were proposed by Kempe et al. [22]. In the independent cascade model, a newly-infected vertex (or seed) u infects each of its not-yet-infected neighbors v with a fixed probability independently. In the linear threshold model for unweighted graphs¹, each non-seed vertex has a threshold sampled uniformly and independently from the interval $[0, 1]$, and becomes infected when the fraction of its infected neighbors exceeds this threshold.

Both models were shown to be *submodular* (see Theorem 2.4 for details) even in the case with directed graphs [22], which implies that the greedy algorithm achieves a $(1 - (1 - 1/k)^k)$ -approximation, or, a $(1 - 1/e)$ -approximation for any k . A natural and important question is, can we show that the greedy algorithm can perform

¹The linear threshold model can be defined for general weighted directed graphs. However, if the graph is undirected, the linear threshold model is normally defined with the edges unweighted. Since this paper mainly deals with undirected graphs, we will adopt the definition of the linear threshold model for unweighted graphs.

better than a $(1 - (1 - 1/k)^k)$ -approximation through a more careful analysis?

To answer this question, it is helpful to notice that INFMAX is a special case of the MAX-K-COVERAGE problem: given a collection of subsets of a set of elements and a positive integer k , find k subsets that cover maximum number of elements (see details in Sect. 2.2). For MAX-K-COVERAGE, it is well known that the greedy algorithm cannot overcome the $(1 - (1 - 1/k)^k)$ barrier: for any positive function $f(k)$ which may be infinitesimal, there exists a MAX-K-COVERAGE instance where the greedy algorithm cannot achieve $(1 - (1 - 1/k)^k + f(k))$ -approximation. Thus, to hope that the greedy algorithm can overcome this barrier for INFMAX, we need to find out what makes INFMAX more special and exploit those INFMAX features that are not in MAX-K-COVERAGE.

Unfortunately, INFMAX with the independent cascade model for general directed graphs is nothing more special than MAX-K-COVERAGE, as it can simulate any MAX-K-COVERAGE instance: set the probability that u infects v to be 1 for all edges (u, v) (i.e., a vertex will be infected if it contains an infected in-neighbor); use a vertex to represent a subset in the MAX-K-COVERAGE instance, and use a clique of size m to represent an element; create a directed edge from the vertex representing the subset to an arbitrary vertex in the clique representing the element if this subset contains this element. It is easy to see that this simulates a MAX-K-COVERAGE instance if m is sufficiently large. Therefore, the greedy algorithm cannot achieve a $(1 - (1 - 1/k)^k + f(k))$ -approximation for any positive function $f(k)$. This implies we must use properties beyond mere submodularity (a property shared by MAX-K-COVERAGE) to improve the algorithmic analysis.

Khanna and Lucier [24] showed that the $(1 - (1 - 1/k)^k)$ barrier can be overcome if we restrict the graphs to be undirected in the independent cascade model. They proved that the greedy algorithm for INFMAX with the independent cascade model for undirected graphs achieves a $(1 - (1 - 1/k)^k + c)$ -approximation for some constant $c > 0$ that does not even depend on k .² This means the greedy algorithm produces a $(1 - 1/e + c)$ -approximation for any k . Moreover, this result holds for the more general setting where 1) there is a prescribed set of vertices $V' \subseteq V$ as a part of input to the INFMAX instance such that the seeds can only be chosen among vertices in V' and 2) a positive weight is assigned to each vertex such that the objective is to maximize the total weight of infected vertices (instead of the total number of infected vertices). This result is remarkable, as many of the social networks in our daily life are undirected by their nature (for example, friendship, co-authorship, etc.). Knowing that the $(1 - (1 - 1/k)^k)$ barrier can be overcome for the independent cascade model, a natural question is, what is the story for the linear threshold model?

1.1 Our Results

We show that Khanna and Lucier’s result on the independent cascade model can only be partially extended to the linear threshold

model. Our first result is an example showing that the greedy algorithm can obtain at most a $(1 - (1 - 1/k)^k + O(1/k^{0.2}))$ -approximation for INFMAX on undirected graphs under the linear threshold model. This shows that, up to lower order terms, the approximation guarantee $1 - (1 - 1/k)^k$ is tight. In particular, no analogue of Khanna and Lucier’s $(1 - 1/e + c)$ result is possible if $c > 0$ is a constant.

For our second result, we prove that the greedy algorithm does achieve a $(1 - (1 - 1/k)^k + \Omega(1/k^3))$ -approximation under the same setting (the linear threshold model with undirected graphs). This indicates that the greedy algorithm can overcome the $(1 - (1 - 1/k)^k)$ barrier by a lower order term. In particular, the barrier is overcome for constant k . We remark that the additive term $\Omega(1/k^3)$ does not depend on the number of vertices/edges in the graph, so this improvement is not diminishing as the size of the graph grows.

Our results corresponding to the last two paragraphs in the abstract is deferred to the full version of this paper.

1.2 Related Work

The influence maximization problem was initially posed by Domingos and Richardson [13, 32]. Kempe et al. [22] showed the linear threshold model and the independent cascade model are submodular, so the greedy algorithm achieves a $(1 - (1 - 1/k)^k)$ -approximation. This result was later generalized to all diffusion models that are locally submodular [23, 28]. As mentioned earlier, for the independent cascade model with undirected graphs, Khanna and Lucier [24] showed that the greedy algorithm achieves a $(1 - (1 - 1/k)^k + c)$ -approximation for some constant $c > 0$.

On the hardness or inapproximability side, Kempe et al. [22] showed that INFMAX on both the linear threshold model and the independent cascade model is NP-hard. For the independent cascade model with directed graphs, Kempe et al. [22] showed a reduction from MAX-K-COVERAGE preserving the approximation factor. Since Feige [14] showed that MAX-K-COVERAGE is NP-hard to approximate within factor $(1 - (1 - 1/k)^k + \epsilon)$ for any constant $\epsilon > 0$, the same inapproximability factor holds for the independent cascade INFMAX. Therefore, up to lower order terms, the gap between the upper bound and the lower bound for the independent cascade (on directed graphs) INFMAX is closed. If undirected graphs are considered, Schoenebeck and Tao [35] showed that, for both the linear threshold model and the independent cascade model, INFMAX is NP-hard to approximate to within factor $(1 - \tau)$ for some constant $\tau > 0$.

If the diffusion model can be nonsubmodular, Kempe et al. [22] showed that INFMAX is NP-hard to approximate to within a factor of $N^{1-\epsilon}$ for any $\epsilon > 0$. Many works after this [5, 26, 33, 34, 39] showed that strong inapproximability results extend to even very specific nonsubmodular models.

INFMAX has also been studied in the adaptive setting, where the seeds are selected iteratively, and the seed-picker can observe the cascade of the previous seeds before choosing the next one [6, 17, 31]. Due to its iterative nature, the greedy algorithm can be easily generalized to an adaptive version [7, 20].

As mentioned in the introduction section, there was extensive work on designing implementations that are more efficient and scalable [4, 9, 10, 12, 15, 18, 19, 21, 25, 30, 37, 38]. These algorithms speedup the greedy algorithm by either disregarding those seed

²Khanna and Lucier [24] only claimed that the greedy algorithm achieves a $(1 - 1/e + c)$ -approximation. However, c being a constant implies that there exists k_0 such that $1 - (1 - 1/k)^k < 1 - 1/e + c/2$ for all $k \geq k_0$ (notice that $(1 - (1 - 1/k)^k)$ is decreasing and approaches to $1 - 1/e$); the greedy algorithm will then achieve a $(1 - (1 - 1/k)^k + c/2)$ -approximation for $k \geq k_0$.

candidates that are identified to be clearly suboptimal or finding smart ways to approximate the expected number of infected vertices. Arora et al. [2] benchmark most of the aforementioned variants of the greedy algorithms. We remark that there do exist INFMAX algorithms that are not based on greedy [1, 3, 16, 33, 34, 36], but they are typically for nonsubmodular diffusion models.

2 PRELIMINARIES

2.1 INFMAX with Linear Threshold Model

Throughout this paper, we use $G = (V, E)$ to represent the graph which may or may not be directed. We use S to denote the set of seeds, k to denote $|S|$. Let $\deg(v)$ be the degree of v when G is undirected and the *in-degree* of vertex v otherwise. For each $v \in V$, let $\Gamma(v) = \{u : (u, v) \in E\}$ be the set of (*in-*)neighbors of vertex v .

Definition 2.1. The *linear threshold model* LT_G is defined by a directed graph $G = (V, E)$. On input seed set $S \subseteq V$, $LT_G(S)$ outputs a set of infected vertices as follows:

- (1) Initially, only vertices in S are infected, and for each vertex v a *threshold* $\theta_v \in \mathbb{Z}^+$ is sampled uniformly at random from $\{1, 2, \dots, \deg(v)\}$ independently. If $\deg(v) = 0$, set $\theta_v = \infty$.
- (2) In each subsequent iteration, a vertex v becomes infected if v has at least θ_v infected in-neighbors.
- (3) After an iteration where there are no additional infected vertices, $LT_G(S)$ outputs the set of infected vertices.

In this paper, we mostly deal with undirected graphs. When we restrict our attention to undirected graphs, the undirected graph is viewed as a special directed graph with each undirected edge of the graph being viewed as two anti-parallel directed edges.

Previous work showed that the linear threshold model has *live-edge interpretation* as stated in the theorem below.

THEOREM 2.2 (CLAIM 2.6 IN [22]). *Let $\widehat{LT}_G(S) \subseteq V$ be the set of vertices that are reachable from S when each vertex v picks exactly one of its incoming edges uniformly at random to be included in the graph and vertices pick their incoming edges independently. Then $\widehat{LT}_G(S)$ and $LT_G(S)$ have the same distribution. Those picked edges are called “live edges”.*

The intuition of this interpretation is as follows: consider a not-yet-infected vertex v and a set of its infected in-neighbors $IN(v) \subseteq \Gamma(v)$. By the definition of the linear threshold model, v will be infected by vertices in $IN(v)$ with probability $|IN(v)|/\deg(v)$. On the other hand, the live edge coming into v will be from the set $IN(v)$ with probability $|IN(v)|/\deg(v)$.

Once again, when considering undirected graphs, those live edges in Theorem 2.2 are still directed. Whenever we mention a live edge in the remaining part of this paper, it should always be clear that this edge is directed.

Remark 1. Since each vertex can choose only one incoming edge as being live, if a vertex v is reachable from a vertex u after sampling all the live edges, then there exists a unique simple path consisting of live edges connecting u to v .

Remark 2. When considering the probability that a given vertex v will be infected by a given seed set S , we can consider a “reverse random walk without repetition” process. The random walk starts

at v , and it chooses one of its neighbors (in-neighbors for directed graphs) uniformly at random and moves to it. The random walk terminates when it reaches a vertex that has already been visited or when it reaches a seed. Each move in the reverse random walk is analogous to selecting one incoming live edge. Theorem 2.2 implies that the probability that this random walk reaches a seed is exactly the probability that v will be infected by seeds in S .

Given a set of vertices A and a vertex v , let $A \rightarrow v$ be the event that v is reachable from A after sampling live edges. Alternatively, this means that the reverse random walk from v described in Remark 2 reaches a vertex in A . If A is the set of seeds, then $\Pr(A \rightarrow v)$ is exactly the probability that v will be infected. Intuitively, $A \rightarrow v$ can be seen as the event that “ A infects v ”. We set $\Pr(A \rightarrow v) = 1$ if $v \in A$. In this paper, we mean $A \rightarrow v$ when we say v reversely walks to A or v is reachable from A . In particular, the reachability is in terms of the live edges, not the original edges.

Given a set of vertices A , a vertex v , and a set of vertices B , let $A \xrightarrow{B} v$ be the event that the reverse random walk from v reaches a vertex in A and the vertices on the live path from v to A , excluding v and the reached vertex in A , do not contain any vertex in B . By definition, $A \xrightarrow{B} v$ is the same as $A \rightarrow v$ if $B = \emptyset$, and $\Pr(A \xrightarrow{B} v) = 1$ for any B if $v \in A$.

Let $\sigma(S)$ be the *expected* total number of infected vertices due to the influence of S , $\sigma(S) = \mathbb{E}[|LT_G(S)|]$, where the expectation is taken over the samplings of thresholds of all vertices, or equivalently, over the choices of incoming live edges of all vertices. By the linearity of expectation, we have $\sigma(S) = \sum_{v \in V} \Pr(S \rightarrow v)$. It is known that computing $\sigma(S)$ or $\Pr(A \rightarrow v)$ for the linear threshold model is #P-hard [10].³ On the other hand, a simple Monte Carlo sampling can approximate $\sigma(S)$ arbitrarily close with probability arbitrarily close to 1. In this paper, we adopt the standard assumption $\sigma(\cdot)$ can be accessed by an oracle.

Definition 2.3. The INFMAX problem is an optimization problem which takes as inputs $G = (V, E)$ and a positive integer k , and outputs $\operatorname{argmax}_{S \subseteq V: |S|=k} \sigma(S)$, a seed set of size k that maximizes the expected number of infected vertices.

The *greedy algorithm* consists of k iterations; in each iteration i , it includes the seed s_i into the seed set S (i.e., $S \leftarrow S \cup \{s_i\}$) with the highest marginal increment to $\sigma(\cdot)$: $s_i \in \operatorname{argmax}_{s \in V \setminus S} (\sigma(S \cup \{s\}) - \sigma(S))$. Under the linear threshold model, the objective function $\sigma(\cdot)$ is monotone and *submodular* (see Theorem 2.4), which implies that the greedy algorithm achieves a $(1 - (1 - 1/k)^k)$ -approximation [22, 29]. Notice that this approximation ratio becomes $1 - 1/e$ when k tends to infinity, and $1 - (1 - 1/k)^k > 1 - 1/e$ for all positive k .

THEOREM 2.4 ([22]). *Consider INFMAX with the linear threshold model. For any two sets of vertices A, B with $A \subseteq B$ and any vertex $v \notin B$, we have $\sigma(A \cup \{v\}) - \sigma(A) \geq \sigma(B \cup \{v\}) - \sigma(B)$, and for any vertex $u \notin B \cup \{v\}$, $\Pr(A \cup \{v\} \rightarrow u) - \Pr(A \rightarrow u) \geq \Pr(B \cup \{v\} \rightarrow u) - \Pr(B \rightarrow u)$.*

Remark 2 straightforwardly implies the following lemma, which describes a negative correlation between the event that $\{u\}$ infects

³Computing $\sigma(S)$ and $\Pr(S \rightarrow v)$ are also #P-hard for the independent cascade model [8].

v and the event that u is infected by another seed set. Some other properties for the linear threshold are presented in Sect. 4.2. We introduce Lemma 2.5 in the preliminary section because this negative correlation property is a signature property that makes the linear threshold model quite different from the independent cascade model. In the independent cascade model, knowing the existence of certain connections between vertices only makes it more likely that another pair of vertices are connected. Intuitively, this is because, in the independent cascade model, each vertex does not “choose” one of its incoming edges, but rather, each incoming edge is included with a certain probability independently. In addition, Lemma 2.5 holds for directed graphs, while all the lemmas in Sect. 4.2 hold only for undirected graphs.

LEMMA 2.5. *For any three sets of vertices A, B_1, B_2 and any two different vertices u, v , $\Pr(A \xrightarrow{B_1} u) \geq \Pr(A \xrightarrow{B_1} u \mid \{u\} \xrightarrow{A \cup B_2} v)$.*

PROOF. Consider any simple path p from u to v . If $u \xrightarrow{A \cup B_2} v$ happens with all edges in p being live, then $\Pr(A \xrightarrow{B_1} u) \geq \Pr(A \xrightarrow{B_1} u \mid p \text{ is live})$. This is apparent by noticing Remark 2: if p is already live, then the reverse random walk starting from u should reach A without touching any vertices on p (if the random walk touches a vertex in p , it will follow the reverse direction of p and eventually go back to u), which obviously happens with less probability compared to the case without restricting that the random walk cannot touch vertices on p .

Noticing this, the remaining part of the proof is trivial:

$$\begin{aligned} \Pr\left(A \xrightarrow{B_1} u \mid u \xrightarrow{A \cup B_2} v\right) &= \sum_p \frac{\Pr(A \xrightarrow{B_1} u \mid p \text{ is live}) \Pr(p \text{ is live})}{\Pr(u \xrightarrow{A \cup B_2} v)} \\ &\leq \Pr(A \xrightarrow{B_1} u) \sum_p \frac{\Pr(p \text{ is live})}{\Pr(\{u\} \xrightarrow{A \cup B_2} v)} = \Pr\left(A \xrightarrow{B_1} u\right), \end{aligned}$$

where the summation is over all simple paths p connecting u to v without touching any vertices in $A \cup B_2$, and Remark 1 ensures that the events “ p is live” over all possible such p ’s form a partition of the event $u \xrightarrow{A \cup B_2} v$. \square

2.2 INFMAX—A Special Case of MAX-K-COVERAGE

In this section, we establish that linear threshold INFMAX is a special case of the well-studied MAX-K-COVERAGE problem, a folklore that is widely known in the INFMAX literature. This section also introduces some key intuitions that will be used throughout the paper. We will only discuss the linear threshold model for the purpose of this paper, although submodular INFMAX in general can also be viewed as a special case of MAX-K-COVERAGE.

Definition 2.6. The MAX-K-COVERAGE problem is an optimization problem which takes as input a universe of elements $U = \{e_1, \dots, e_N\}$, a collection of subsets $\mathcal{M} = \{S_1, \dots, S_M : S_i \subseteq U\}$ and an positive integer k , and outputs a collection of k subsets that maximizes the total number of covered elements: $\mathcal{S} \in$

$$\operatorname{argmax}_{\mathcal{S} \subseteq \mathcal{M}, |\mathcal{S}|=k} \left| \bigcup_{S \in \mathcal{S}} S \right|. \text{ Given } \mathcal{S} \subseteq \mathcal{M}, \text{ we denote } \operatorname{val}(\mathcal{S}) = \left| \bigcup_{S \in \mathcal{S}} S \right|.$$

It is well-known that the greedy algorithm (that iteratively selects a subset that maximizes the marginal increment of $\operatorname{val}(\cdot)$)

achieves a $(1 - (1 - 1/k)^k)$ -approximation for MAX-K-COVERAGE. On the other hand, this approximation guarantee is tight: for any positive function $f(k) > 0$ which may be infinitesimal, there exists a MAX-K-COVERAGE instance such that the greedy algorithm cannot achieve a $(1 - (1 - 1/k)^k + f(k))$ -approximation. We will review some properties of MAX-K-COVERAGE in Sect. 4.1 that will be used in our analysis for INFMAX.

INFMAX with the linear threshold model can be viewed as a special case of MAX-K-COVERAGE in that an instance of INFMAX can be transformed into an instance of MAX-K-COVERAGE. Given an instance of INFMAX ($G = (V, E), k$), let H be the set of all possible live-edge samplings. That is, H is the set of directed graphs on V that are subgraphs of G where each vertex has in-degree equal to 1. In particular, $|H| = \prod_{v \in V} \deg(v)$.⁴ We create an instance of MAX-K-COVERAGE by letting the universe of elements be $V \times H$, i.e., pairs of vertices and live-edge samplings, (v, g) , where $v \in V$ and $g \in H$. We then create a subset for each vertex $v \in V$. The subset corresponding to $v \in V$ contains (u, g) if u is reachable from v in g . We denote by $\Sigma(S) = \{(u, g) : u \text{ is reachable from } S \text{ under } g\}$ the set of “elements” that the “subsets” in S cover. Since $\sigma(S) = \sum_{v \in V} \Pr(S \rightarrow v) = \sum_{v \in V} \frac{|\{g : v \text{ is reachable from } S \text{ under } g\}|}{\prod_{w \in V} \deg(w)} = \frac{|\Sigma(S)|}{\prod_{w \in V} \deg(w)}$, $\sigma(S)$ equals to the total number of elements covered by “subsets” in S , divided by $|H|$. As a result, $\sigma(S)$ is proportional to the total number of covered elements if viewing S as a collection of subsets. This establishes that INFMAX is a special case of MAX-K-COVERAGE.

Having established the connection between INFMAX and MAX-K-COVERAGE, we take a closer look at the intersection, union and difference of two subsets. Let S_1, S_2 be two seed sets. $\Sigma(S_1) \cup \Sigma(S_2)$ contains all those (u, g) such that u is reachable from either S_1 or S_2 under g . Clearly, $\sigma(S_1 \cup S_2) = |\Sigma(S_1 \cup S_2)| / \prod_{v \in V} \deg(v) = |\Sigma(S_1) \cup \Sigma(S_2)| / \prod_{v \in V} \deg(v)$. The first equality holds by definition which holds for set intersection and set difference as well. The last equality, however, does not hold for set intersection and set difference.

$\Sigma(S_1) \cap \Sigma(S_2)$ contains all those (u, g) such that u is reachable from both S_1 and S_2 under g . We have $|\Sigma(S_1) \cap \Sigma(S_2)| / \prod_{v \in V} \deg(v) = \sum_{v \in V} \Pr((S_1 \rightarrow v) \wedge (S_2 \rightarrow v))$. For the special case where $S_1 = \{u_1\}$ and $S_2 = \{u_2\}$, by Remark 1, the event $(S_1 \rightarrow v) \wedge (S_2 \rightarrow v)$ can be partitioned into two disjoint events: 1) v reaches u_2 before u_1 in the reverse random walk, $(\{u_1\} \xrightarrow{\text{rev}} u_2) \wedge (\{u_2\} \xrightarrow{\text{rev}} v)$, and 2) v reaches u_1 before u_2 in the reverse random walk, $(\{u_2\} \xrightarrow{\text{rev}} u_1) \wedge (\{u_1\} \xrightarrow{\text{rev}} v)$. For general S_1, S_2 with $S_1 \cap S_2 = \emptyset$, the event $(S_1 \rightarrow v) \wedge (S_2 \rightarrow v)$ can be partitioned into two disjoint events depending on whether v reversely reaches S_1 or S_2 first.

Similarly, $\Sigma(S_1) \setminus \Sigma(S_2)$ contains all those (u, g) such that u is reachable from S_1 but not from S_2 under g , and we have $|\Sigma(S_1) \setminus \Sigma(S_2)| / |H| = \sum_{v \in V} \Pr((S_1 \rightarrow v) \wedge \neg(S_2 \rightarrow v))$.

3 UPPER BOUND

In this section, we show that the approximation guarantee for the greedy algorithm on INFMAX is at most $(1 - (1 - 1/k)^k + O(1/k^{0.2}))$ with the linear threshold model on undirected graphs. This shows

⁴Of course, vertices with in-degree 0 should be excluded from this product. Whenever we write this product, we always refer to the one excluding vertices with in-degree 0.

that the approximation guarantee $(1-1/e)$ cannot be asymptotically improved, even if undirected graphs are considered.

Before we prove our main theorem in this section, we need the following lemma characterizing the cascade of a single seed on a complete graph which is interesting on its own.

LEMMA 3.1. *Let G be a complete graph with n vertices, and let S be a set containing a single vertex. We have $\sigma(S) < 3\sqrt{n}$.*

The proof of Lemma 3.1 is in the full version of this paper. The intuition behind this lemma is simply the birthday paradox. Consider the reverse random walk starting from any particular vertex v with seed set $\{u\}$. At each step, the walk chooses a random vertex other than the current vertex. By the birthday paradox, the expected time for the walk to reach a previously visited vertex is $\Theta(\sqrt{n})$. The probability v is infected is the probability that the random walk reaches the seed $\{u\}$ before reaching a previously visited vertex. This is approximately $1 - (1 - 1/n)^{\sqrt{n}} \approx 1/\sqrt{n}$. Finally, by the linearity of expectation, the total number of infected vertices is about \sqrt{n} .

The remainder of this section proves the following theorem.

THEOREM 3.2. *Consider INFMAX on undirected graphs with the linear threshold model. There exists an instance where the greedy algorithm only achieves a $(1 - (1-1/k)^k + O(1/k^{0.2}))$ -approximation.*

The INFMAX instance mentioned in Theorem 3.2 is shown below.

Example 3.3. Given the number of seeds k , we construct the undirected graph $G = (V, E)$ with $k\lceil k^{1.2} \rceil + \lfloor (1 - \frac{100}{k^{0.2}})k^{1.8} \rfloor$ vertices as follows. Firstly, construct k cliques C_1, \dots, C_k of size $\lceil k^{1.2} \rceil$, and in each clique C_i label an arbitrary vertex u_i . Secondly, construct k vertices v_1, \dots, v_k . For each $i = 1, \dots, k$, create $\lceil k^{0.8}(1 - 1/k)^{i-1} \rceil - 1$ vertices and connect them to v_i . For each i , those $\lceil k^{0.8}(1 - 1/k)^{i-1} \rceil - 1$ vertices combined with v_i form a star of size $\lceil k^{0.8}(1 - 1/k)^{i-1} \rceil$, and we will use D_i to denote the i -th star. Thirdly, we continue creating ℓ of these kinds of stars $D_{k+1}, \dots, D_{k+\ell}$ centered at $v_{k+1}, \dots, v_{k+\ell}$ such that $|D_{k+1}| = \dots = |D_{k+\ell-1}| = \lceil k^{0.8}(1 - 1/k)^k \rceil$, $|D_{k+\ell}| \leq \lceil k^{0.8}(1 - 1/k)^k \rceil$, and $\sum_{i=1}^{k+\ell} |D_i| = \lfloor (1 - \frac{100}{k^{0.2}})k^{1.8} \rfloor$. In other words, we keep creating stars of the same size $\lceil k^{0.8}(1 - 1/k)^k \rceil$ until we reach the point where the total number of vertices in all those stars is $\lfloor (1 - \frac{100}{k^{0.2}})k^{1.8} \rfloor$ (we assume k is sufficiently large), where the last star created may be “partial” and have a size smaller than $\lceil k^{0.8}(1 - 1/k)^k \rceil$. Notice that $|D_1| \geq |D_2| \geq \dots \geq |D_k| \geq |D_{k+1}| = \dots = |D_{k+\ell-1}| \geq |D_{k+\ell}| = \Theta(k^{0.8})$.⁵ Finally, create $k \times (k + \ell)$ edges $\{(u_i, v_j) : i = 1, \dots, k; j = 1, \dots, k + \ell\}$.

Proof Sketch of Theorem 3.2. We want that the greedy algorithm picks the seeds v_1, \dots, v_k , while the optimal seeds are u_1, \dots, u_k . The purpose of constructing a clique C_i for each u_i is to simulate directed edges (u_i, v_j) (such that, as mentioned earlier, each u_i will be infected with $o(1)$ probability even if all of $v_1, \dots, v_{k+\ell}$ are infected, and the total number of infections among the cliques is negligible so that the “gadget” itself is not “heavy”). In the optimal seeding strategy, each v_i will be infected with probability $1 - o(1)$, as the number of edges connecting to the seeds u_1, \dots, u_k is k , which is significantly more than the number of edges inside D_i (which is at most $\lceil k^{0.8} \rceil$). Therefore, $\sigma(\{u_1, \dots, u_k\}) \approx \sum_{i=1}^{k+\ell} |D_i| =$

⁵These inequalities may not be strict. In fact, $|D_1|$ may be equal to $|D_2|$ as $k^{0.8} - k^{0.8}(1 - 1/k) = 1/k^{0.2} < 1$.

$\lfloor (1 - \frac{100}{k^{0.2}})k^{1.8} \rfloor$, which is slightly less than $k^{1.8}$. Moreover, each $\sigma(\{u_i\})$ is approximately $\frac{1}{k}$ of $\sigma(\{u_1, \dots, u_k\})$, which is slightly less than $k^{0.8}$.

The greedy algorithm would pick v_1 as the first seed, as $\sigma(v_1)$ is at least $\lceil k^{0.8} \rceil$ (by only accounting for the infected vertices in D_1) which is slightly larger than each $\sigma(\{u_i\})$. After picking v_1 as the first seed, the marginal increment of $\sigma(\cdot)$ by choosing each of u_1, \dots, u_k becomes approximately $\frac{1}{k} \sum_{i=2}^{k+\ell} |D_i| = \frac{1}{k}(-|D_1| + \sum_{i=1}^{k+\ell} |D_i|)$, which is slightly less than $\frac{1}{k}(-\lceil k^{0.8} \rceil + k^{1.8}) \approx |D_2|$. On the other hand, noticing that v_1 infects each of u_1, \dots, u_k as well as v_2 with probability $o(1)$, the marginal increment of $\sigma(\cdot)$ by choosing v_2 is approximately $|D_2|$, which is slightly larger than the marginal increment by choosing any u_i based on our calculation above. Thus, the greedy algorithm will continue to pick v_2 . In general, we have designed the sizes of D_1, D_2, \dots, D_k such that they are just large enough to make sure the greedy algorithm will pick v_1, v_2, \dots, v_k one by one.

Our construction of cliques C_1, \dots, C_k makes sure that each of u_1, \dots, u_k will be infected with $o(1)$ probability even if all of v_1, \dots, v_k are seeded. Therefore, $\sigma(\{v_1, \dots, v_k\}) \approx \sum_{i=1}^k |D_i| = \sum_{i=1}^k \lceil k^{0.8}(1 - 1/k)^{i-1} \rceil \leq k + \sum_{i=1}^k k^{0.8}(1 - 1/k)^{i-1} = k + k^{1.8}(1 - (1 - 1/k)^k)$. On the other hand, we have seen that $\sigma(\{u_1, \dots, u_k\})$ is just slightly less than $k^{1.8}$. To be more accurate, $\sigma(\{u_1, \dots, u_k\}) \approx (1 - \frac{100}{k^{0.2}})k^{1.8}$. Dividing $\sigma(\{v_1, \dots, v_k\})$ by $\sigma(\{u_1, \dots, u_k\})$ gives us the desired upper bound on the approximation ratio in Theorem 3.2. The numbers 0.2, 0.8, 1.2 on the exponent of k are optimized for getting the tightest bound while ensuring that the greedy algorithm still picks v_1, \dots, v_k .

See the full version of this paper for a rigorous proof.

4 LOWER BOUND

In this section, we prove that the greedy algorithm can obtain a $(1 - (1 - 1/k)^k + \Omega(1/k^3))$ -approximation, stated in Theorem 4.1. This indicates that the barrier $1 - (1 - 1/k)^k$ can be overcome if k is a constant. We have seen that INFMAX is a special case of MAX-K-COVERAGE in Sect. 2.2, and it is known that the greedy algorithm cannot overcome the barrier $1 - (1 - 1/k)^k$ in MAX-K-COVERAGE. Theorem 4.1 shows that INFMAX with the linear threshold model on undirected graphs has additional structure. To prove Theorem 4.1, we first review in Sect. 4.1 some properties of MAX-K-COVERAGE that are useful to our analysis, and then we prove Theorem 4.1 in Sect. 4.2 by exploiting some special properties of INFMAX that are not satisfied in MAX-K-COVERAGE.

THEOREM 4.1. *Consider INFMAX on undirected graphs with the linear threshold model. The greedy algorithm achieves a $(1 - (1 - 1/k)^k + \Omega(1/k^3))$ -approximation.*

4.1 Some Properties of MAX-K-COVERAGE

In this section, we list some of the properties of MAX-K-COVERAGE which will be used in proving Theorem 4.1. The proofs of the lemmas in this section are all standard, and are deferred to the appendix. For all the lemmas in this section, we are considering a MAX-K-COVERAGE instance (U, \mathcal{M}, k) , where $\mathcal{S} = \{S_1, \dots, S_k\}$ denotes the k subsets output by the greedy algorithm and $\mathcal{S}^* = \{S_1^*, \dots, S_k^*\}$ denotes the optimal solution.

LEMMA 4.2. *If $S_1 \in S^*$, then $\text{val}(S) \geq (1 - (1 - \frac{1}{k})^k + \frac{1}{4k^2}) \text{val}(S^*)$.*

LEMMA 4.3. *If $\frac{|S_1 \cap (\cup_{i=1}^k S_i^*)|}{\text{val}(S^*)} \notin [\frac{1}{k} - \varepsilon, \frac{1}{k} + \varepsilon]$ for some $\varepsilon > 0$ which may depend on k , then $\text{val}(S) \geq (1 - (1 - 1/k)^k + \varepsilon/4) \text{val}(S^*)$.*

LEMMA 4.4. *If $\sum_{i=1}^k |S_i^*| > (1 + \varepsilon) \text{val}(S^*)$ for some $\varepsilon > 0$ which may depend on k , then $\text{val}(S) \geq (1 - (1 - \frac{1}{k})^k + \frac{\varepsilon}{8k}) \text{val}(S^*)$.*

LEMMA 4.5. *If $|S_1 \setminus (\cup_{i=1}^k S_i^*)| > \varepsilon \text{val}(S^*)$ for some $\varepsilon > 0$ which may depend on k , then $\text{val}(S) \geq (1 - (1 - 1/k)^k + \varepsilon/16) \text{val}(S^*)$.*

LEMMA 4.6. *If there exists $S_i^* \in S^*$ such that $|S_i^*| < (\frac{1}{k} - \varepsilon) \text{val}(S^*)$ for some $\varepsilon > 0$ which may depend on k , then $\text{val}(S) \geq (1 - (1 - \frac{1}{k})^k + \frac{\varepsilon}{8k}) \text{val}(S^*)$.*

4.2 Proof of Theorem 4.1

We begin by proving some properties that are exclusively for INF-MAX.

LEMMA 4.7. *Given a subset of vertices $A \subseteq V$, a vertex $v \notin A$ and a neighbor $u \in \Gamma(v)$ of v , with probability at most $\frac{|A|}{|A|+1}$, there is a simple live path from a vertex in A to vertex v such that the last vertex in the path before reaching v is not u .*

PROOF. We consider all possible reverse random walks starting from v , and define a mapping from those walks that eventually reach A to those that do not. For each reverse random walk that reaches a vertex $a \in A$, $v \leftarrow w_1 \leftarrow \dots \leftarrow w_{\ell-1} \leftarrow w_\ell \leftarrow a$ (with $w_1, \dots, w_\ell \notin A$), we map it to the random walk $v \leftarrow w_1 \leftarrow \dots \leftarrow w_{\ell-1} \leftarrow w_\ell \leftarrow w_{\ell-1}$, i.e., the one with the last step moving back. Notice that the latter reverse random walk visits $w_{\ell-1}$ more than once, and thus will not reach A . Specifically, for those reverse random walks that reach A in one single step $v \leftarrow a$ (in the case v is adjacent to $a \in A$), we map it to the reverse random walk $v \leftarrow u$, which are excluded from the event that “there is a simple live path from a vertex in A to vertex v such that the last vertex in the path before reaching v is not u ” (if $v \leftarrow u$, then every path that reaches v should then reach u in the penultimate step).

It is easy to see that at most $|A|$ different reverse random walks that reach A can be mapped to a same random walk that does not reach A . In order to make different reverse random walks have the same image in the mapping, they must share the same path $v \leftarrow w_1 \leftarrow \dots \leftarrow w_\ell$ except for the last step. The last step, which moves to a vertex in A , can only have $|A|$ different choices. For the special reverse random walks that move to A in one step, there are at most $|A|$ of them, which are mapped to the random walk $v \leftarrow u$.

It is also easy to see that each random walk happens with the same probability as its image does. This is because w_ℓ chooses its incoming edges uniformly, so choosing a happens with the same chance as choosing w_ℓ . Specifically, v chooses its incoming edge (a, v) with the same probability as (u, v) .

Since we have defined a mapping that maps at most $|A|$ disjoint sub-events in the positive case to a sub-event in the negative case with the same probability, the lemma follows. \square

LEMMA 4.8. *Given a subset of vertices $A \subseteq V$ and two different vertices $u, v \notin A$, we have $\Pr(A \rightarrow u \mid \{u\} \xrightarrow{A} v) \leq \frac{|A|}{|A|+1}$.*

PROOF. Let w_1, \dots, w_t enumerate all the neighbors of u that are not in A . For each $i = 1, \dots, t$, let E_i be the event that the reverse random walk starting from v reaches u without touching A and its last step before reaching u is at w_i . Clearly, $\{E_1, \dots, E_t\}$ is a partition of $\{u\} \xrightarrow{A} v$. Conditioning on the event E_i , if $A \rightarrow u$ happens, the reverse random walk from u to A cannot touch w_i , since w_i has already chosen its incoming edge (u, w_i) in the case E_i happens. Therefore, by Lemma 2.5 and Lemma 4.7, $\Pr(A \rightarrow u \mid E_i) = \Pr(A \xrightarrow{\{w_i\}} u \mid E_i) \leq \Pr(A \xrightarrow{\{w_i\}} u) \leq \frac{|A|}{|A|+1}$.⁶ We have

$$\begin{aligned} \Pr(A \rightarrow u \mid \{u\} \xrightarrow{A} v) &= \frac{\sum_{i=1}^t \Pr(A \rightarrow u \mid E_i) \Pr(E_i)}{\Pr(\{u\} \xrightarrow{A} v)} \\ &\leq \frac{|A|}{|A|+1} \frac{\sum_{i=1}^t \Pr(E_i)}{\Pr(\{u\} \xrightarrow{A} v)} = \frac{|A|}{|A|+1}, \end{aligned}$$

which concludes this lemma. \square

Finally, we need the following lemma which is due to Lim et al. [27], while a more generalized version is proved by Schoenebeck and Tao [35].

LEMMA 4.9 (LIM ET AL. [27]). *For any $v \in V$, $\sigma(\{v\}) \leq \text{deg}(v) + 1$.*

Now we are ready to show Theorem 4.1. In the remaining part of this section, we use $S = \{v_1, \dots, v_k\}$ and $S^* = \{u_1, \dots, u_k\}$ to denote the seed sets output by the greedy algorithm and the optimal seed set respectively. Recall that INFMAX is a special case of MAX-K-COVERAGE (Sect. 2.2), and $v_1, \dots, v_k, u_1, \dots, u_k$ can be viewed as subsets in MAX-K-COVERAGE. Thus, the lemmas in Sect. 4.1 can be applied here.

First of all, if $v_1 \in S^*$, Lemma 4.2 implies Theorem 4.1 already. In particular, Lemma 4.2 implies that $|\Sigma(S)| \geq (1 - (1 - 1/k)^k + 1/4k^2) |\Sigma(S^*)|$ (refer to Sect. 2.2 for the definition of $\Sigma(\cdot)$), which implies $\sigma(S) \geq (1 - (1 - 1/k)^k + 1/4k^2) \sigma(S^*)$ by dividing $\prod_{w \in V} \text{deg}(w)$ on both side of the inequality. Therefore, we assume $v_1 \notin S^*$ from now on.

Next, we analyze the intersection between $\Sigma(\{v_1\})$ and $\Sigma(S^*)$. As an overview of the remaining part of our proof, suppose the barrier $1 - (1 - 1/k)^k$ cannot be overcome, Lemma 4.4 and Lemma 4.6 imply that $\Sigma(\{u_1\}), \dots, \Sigma(\{u_k\})$ must be almost disjoint and almost balanced, Lemma 4.3 implies that $\Sigma(\{v_1\})$ must intersect approximately $1/k$ fraction of $\Sigma(S^*)$, and Lemma 4.5 implies that $\Sigma(\{v_1\}) \setminus \Sigma(S^*)$ should not be large. We will prove that these conditions cannot be satisfied at the same time.

⁶Rigorously speaking, the statement of Lemma 2.5 does not directly imply $\Pr(A \xrightarrow{\{w_i\}} u \mid E_i) \leq \Pr(A \xrightarrow{\{w_i\}} u)$. However, the proof of Lemma 2.5 can be adapted to show this. Instead of summing over all simple paths p from u to v in the summation of the last inequality in the proof, we sum over all simple paths from u to v such that u first moves to w_i . The remaining part of the proof is the same. The idea here is that, the event v reversely walks to u is negatively correlated to the event that u reversely walks to A , as the latter walk cannot hit the vertices on the path $u \rightarrow v$ if there is already a path from u to v .

The intersection $\Sigma(\{v_1\}) \cap \Sigma(S^*)$ consists of all the tuples (w, g) such that w is reachable from both v_1 and S^* under the live-edge realization g . Consider the reverse random walk starting from w . There are three different disjoint cases: 1) w reaches v_1 first, and then reaches a vertex in S^* ; 2) w reaches a vertex in S^* , and then reaches v_1 ; 3) w visits more than one vertex in S^* , and then reaches v_1 . The three terms in the following equation, which are named C_1, C_2, C_3 , correspond to these three cases respectively.

$$\frac{|\Sigma(\{v_1\}) \cap \Sigma(S^*)|}{\prod_{w \in V} \deg(w)} = \sum_{w \in V} \Pr \left((S^* \rightarrow v_1) \wedge \left(\{v_1\} \xrightarrow{\mathcal{S}^*} w \right) \right) + (C_1)$$

$$\sum_{w \in V} \sum_{i=1}^k \Pr \left(\left(\{v_1\} \xrightarrow{\mathcal{S}^*} u_i \right) \wedge \left(\{u_i\} \xrightarrow{\mathcal{S}^*} w \right) \right) + (C_2)$$

$$\sum_{w \in V} \sum_{i \neq j} \Pr \left(\left(\{v_1\} \rightarrow u_j \right) \wedge \left(\{u_j\} \xrightarrow{\mathcal{S}^*} u_i \right) \wedge \left(\{u_i\} \xrightarrow{\mathcal{S}^*} w \right) \right) (C_3)$$

Notice that this decomposition assumes $v_1 \notin S^*$.

Firstly, we show that C_1 cannot be too large if the barrier $1 - (1 - 1/k)^k$ is not overcome. Intuitively, C_1 describes those w that first reversely reaches v_1 and then reversely reaches a vertex in S^* . Lemma 4.8 tells us that v_1 will reversely reach S^* with at most probability $k/(k+1)$ conditioning on w reversely reaching v_1 . This implies that, if w reversely reaches v_1 , v_1 will not reversely reach S^* with probability at least $1/(k+1)$, which is at least $1/k$ of the probability that v_1 reversely reaches S^* . Therefore, whenever we have a certain number of elements in $\Sigma(\{v_1\}) \cap \Sigma(S^*)$ that corresponds to C_1 , we have at least $1/k$ fraction of this number in $\Sigma(\{v_1\}) \setminus \Sigma(S^*)$. Lemma 4.5 implies that the $1 - (1 - 1/k)^k$ barrier can be overcome if $|\Sigma(\{v_1\}) \setminus \Sigma(S^*)|$ is large.

PROPOSITION 4.10. *If $C_1 > \frac{9}{10k} \cdot \sigma(S^*)$, then $\sigma(S) \geq (1 - (1 - \frac{1}{k})^k + \frac{1}{640k^2}) \cdot \sigma(S^*)$.*

PROOF. If $w = v_1$, $\{v_1\} \xrightarrow{\mathcal{S}^*} w$ happens automatically, and $\Pr(\{v_1\} \xrightarrow{\mathcal{S}^*} w) \wedge (S^* \rightarrow v_1) = \Pr(S^* \rightarrow v_1)$. Substituting this into C_1 , we have

$$\begin{aligned} C_1 &= \Pr(S^* \rightarrow v_1) + \sum_{w \in V \setminus \{v_1\}} \Pr \left((S^* \rightarrow v_1) \wedge \left(\{v_1\} \xrightarrow{\mathcal{S}^*} w \right) \right) \\ &\leq 1 + \sum_{w \in V \setminus \{v_1\}} \Pr \left(\{v_1\} \xrightarrow{\mathcal{S}^*} w \right) \cdot \Pr \left(S^* \rightarrow v_1 \mid \{v_1\} \xrightarrow{\mathcal{S}^*} w \right) \\ &\leq 1 + \sum_{w \in V \setminus \{v_1\}} \Pr \left(\{v_1\} \xrightarrow{\mathcal{S}^*} w \right) \cdot k \Pr \left(\neg(S^* \rightarrow v_1) \mid \{v_1\} \xrightarrow{\mathcal{S}^*} w \right) \\ &\hspace{15em} (\text{Lemma 4.8}) \\ &= 1 + k \sum_{w \in V \setminus \{v_1\}} \Pr \left(\left(\{v_1\} \xrightarrow{\mathcal{S}^*} w \right) \wedge \neg(S^* \rightarrow v_1) \right), \end{aligned}$$

where the penultimate step is due to Lemma 4.8 from which we have $\Pr(S^* \rightarrow v_1 \mid \{v_1\} \xrightarrow{\mathcal{S}^*} w) \leq \frac{k}{k+1}$, which implies $\Pr(\neg(S^* \rightarrow v_1) \mid \{v_1\} \xrightarrow{\mathcal{S}^*} w) \geq \frac{1}{k+1}$, which further implies $\Pr(S^* \rightarrow v_1 \mid \{v_1\} \xrightarrow{\mathcal{S}^*} w) \leq k \cdot \Pr(\neg(S^* \rightarrow v_1) \mid \{v_1\} \xrightarrow{\mathcal{S}^*} w)$.

Notice that $\sum_{w \in V \setminus \{v_1\}} \Pr(\{v_1\} \xrightarrow{\mathcal{S}^*} w) \wedge \neg(S^* \rightarrow v_1)$ describes those (w, g) such that w is reachable from v_1 but not S^* under realization g , which corresponds to elements in $\Sigma(\{v_1\}) \setminus \Sigma(S^*)$.

Therefore, $\frac{|\Sigma(\{v_1\}) \setminus \Sigma(S^*)|}{\prod_{w \in V} \deg(w)} \geq \sum_{w \in V \setminus \{v_1\}} \Pr(\{v_1\} \xrightarrow{\mathcal{S}^*} w) \wedge \neg(S^* \rightarrow v_1) \geq \frac{C_1 - 1}{k}$.

If $\sigma(S^*) \leq \frac{8}{7}k$, we can see that $\sigma(S) \geq k \geq \frac{7}{8}\sigma(S^*) > (1 - (1 - \frac{1}{k})^k + \frac{1}{640k^2})\sigma(S^*)$ and the proposition is already implied. Thus, we assume $\sigma(S^*) > \frac{8}{7}k$ from now on.

If we have $C_1 > \frac{9}{10k}\sigma(S^*)$ as given in the proposition statement, we have $C_1 - 1 > \frac{9}{10k}\sigma(S^*) - \frac{7}{8k}\sigma(S^*) = \frac{1}{40k}\sigma(S^*) = \frac{1}{40k} \frac{|\Sigma(S^*)|}{\prod_{w \in V} \deg(w)}$. Putting together,

$$\frac{|\Sigma(\{v_1\}) \setminus \Sigma(S^*)|}{\prod_{w \in V} \deg(w)} \geq \frac{C_1 - 1}{k} > \frac{1}{40k^2} \frac{|\Sigma(S^*)|}{\prod_{w \in V} \deg(w)},$$

which yields $|\Sigma(\{v_1\}) \setminus \Sigma(S^*)| > \frac{1}{40k^2} |\Sigma(S^*)|$. Lemma 4.5 implies $|\Sigma(S)| \geq (1 - (1 - \frac{1}{k})^k + \frac{1}{640k^2}) |\Sigma(S^*)|$, which further implies this proposition. \square

Secondly, we show that C_2 cannot be too large if the barrier $1 - (1 - 1/k)^k$ is not overcome. To show this, we first show that there exists $u_i \in S^*$ such that $\Pr(\{v_1\} \rightarrow u_i) \geq \frac{C_2}{\sigma(S^*)}$, and then show that this implies that $|\Sigma(\{v_1\}) \setminus \Sigma(S^*)|$ is large by accounting for v_1 's influence to u_i 's neighbors.

PROPOSITION 4.11. *If $C_2 > \frac{1}{100k} \cdot \sigma(S^*)$, then $\sigma(S) \geq (1 - (1 - \frac{1}{k})^k + \frac{1}{64000k^3})\sigma(S^*)$.*

PROOF. We give an outline of the proof first. Assume $u_1 \in \operatorname{argmax}_{u_i \in S^*} \Pr(\{v_1\} \xrightarrow{\mathcal{S}^*} u_i)$ without loss of generality. The proof is split into two steps.

- Step 1: We will show that $\sum_{w \in \Gamma(u_1) \setminus S^*} \Pr(\{v_1\} \xrightarrow{\mathcal{S}^*} w) = \Omega\left(\frac{1}{k^2}\right) \sigma(S^*)$ if we have $C_2 > \frac{1}{100k} \cdot \sigma(S^*)$ in the proposition statement. Notice that the summation consists of the neighbors of u_1 (that are not in S^*) that reversely reaches v_1 , which is a lower bound to $\sigma(v_1)$ (v_1 may infect more vertices than only the neighbors of u_1). To show this, we first find an upper bound of C_2 in terms of this summation: $\frac{C_2}{\sigma(S^*)} \leq \frac{1}{\deg(u_1)} \sum_{w \in \Gamma(u_1) \setminus S^*} \Pr(\{v_1\} \xrightarrow{\mathcal{S}^*} w)$. This will imply that $\sum_{w \in \Gamma(u_1) \setminus S^*} \Pr(\{v_1\} \xrightarrow{\mathcal{S}^*} w) = \Omega\left(\frac{1}{k^2}\right) \sigma(S^*)$ if assuming $C_2 > \frac{1}{100k} \cdot \sigma(S^*)$, because $\deg(u_1)$ is (approximately) an upper bound to $\sigma(\{u_1\})$ by Lemma 4.9, and $\sigma(\{u_1\})$ is approximately $\frac{1}{k}\sigma(S^*)$ (otherwise, the proposition holds directed by Lemma 4.6).
- Step 2: We will show that $\Pr(\neg(S^* \rightarrow v_1) \mid \{v_1\} \xrightarrow{\mathcal{S}^*} w) \geq \frac{1}{2(k+1)}$ for each $w \in \Gamma(u_1) \setminus S^*$. This says that, for each of u_1 's neighbor w , if it reversely reaches v_1 , it will not reach S^* with a reasonably high probability. Correspondingly, a reasonably large fraction of $\Sigma(\{v_1\})$ will not be in $\Sigma(S^*)$. By Lemma 4.5, this proposition is concluded.

Step 1. Based on the first vertex in S^* that w reversely reaches, we can decompose $\sigma(S^*)$ as $\sigma(S^*) = \sum_{w \in V} \sum_{i=1}^k \Pr\left(\{u_i\} \xrightarrow{\mathcal{S}} w\right)$.

Next, we have

$$\begin{aligned}
& C_2 / \sigma(S^*) \\
&= \frac{\sum_{w \in V} \sum_{i=1}^k \Pr\left(\{u_i\} \xrightarrow{\mathcal{S}} w\right) \Pr\left(\{v_1\} \xrightarrow{\mathcal{S}} u_i \mid \{u_i\} \xrightarrow{\mathcal{S}} w\right)}{\sum_{w \in V} \sum_{i=1}^k \Pr\left(\{u_i\} \xrightarrow{\mathcal{S}} w\right)} \\
&\leq \frac{\sum_{w \in V} \sum_{i=1}^k \Pr\left(\{u_i\} \xrightarrow{\mathcal{S}} w\right) \Pr\left(\{v_1\} \xrightarrow{\mathcal{S}} u_i\right)}{\sum_{w \in V} \sum_{i=1}^k \Pr\left(\{u_i\} \xrightarrow{\mathcal{S}} w\right)} \quad (\text{Lemma 2.5}) \\
&\leq \Pr\left(\{v_1\} \xrightarrow{\mathcal{S}} u_1\right) \cdot \frac{\sum_{w \in V} \sum_{i=1}^k \Pr\left(\{u_i\} \xrightarrow{\mathcal{S}} w\right)}{\sum_{w \in V} \sum_{i=1}^k \Pr\left(\{u_i\} \xrightarrow{\mathcal{S}} w\right)} \\
&= \Pr\left(\{v_1\} \xrightarrow{\mathcal{S}} u_1\right) \\
&= \frac{1}{\deg(u_1)} \sum_{w \in \Gamma(u_1) \setminus S^*} \Pr\left(\{v_1\} \xrightarrow{\mathcal{S}} w\right).
\end{aligned}$$

For the last step, v_1 needs to first connect to one of u_1 's neighbors before connecting to u_1 . Notice that these neighbors may include v_1 itself. In this special case $w = v_1 \in \Gamma(u_1) \setminus S^*$, we have $\Pr(\{v_1\} \xrightarrow{\mathcal{S}} w) = 1$ and u_1 chooses its incoming live edge to be (v_1, u_1) with probability $\frac{1}{\deg(u_1)}$, which is also a valid term in the summation above.

If $C_2 > \frac{1}{100k} \cdot \sigma(S^*)$ as suggested by the proposition statement, we have

$$\begin{aligned}
& \sum_{w \in \Gamma(u_1) \setminus S^*} \Pr\left(\{v_1\} \xrightarrow{\mathcal{S}} w\right) \geq \frac{\deg(u_1) C_2}{\sigma(S^*)} \\
&> \frac{\deg(u_1)}{100k} \geq \frac{\deg(u_1) + 1}{200k} \geq \frac{\sigma(\{u_1\})}{200k} \geq \frac{9\sigma(S^*)}{2000k^2},
\end{aligned}$$

where the penultimate step is due to Lemma 4.9 and the last step is based on the assumption $\sigma(\{u_1\}) \geq \frac{9}{10k} \sigma(S^*)$. Notice that we can assume this without loss of generality, as otherwise Lemma 4.6 implies that $|\Sigma(S)| \geq (1 - (1 - \frac{1}{k})^k + \frac{1}{80k^2}) |\Sigma(S^*)|$, which directly implies this proposition.

Step 2. If $w \neq v_1$, Lemma 4.8 implies that $\Pr(\neg(S^* \rightarrow v_1) \mid \{v_1\} \xrightarrow{\mathcal{S}} w) \geq \frac{1}{k+1} > \frac{1}{2(k+1)}$. If $w = v_1$, then u_1 and v_1 are adjacent. Notice that $\deg(v_1) \geq 2$, for otherwise $\sigma(\{u_1\}) > \sigma(\{v_1\})$ so v_1 cannot be the first seed picked by the greedy algorithm. Therefore, v_1 reversely reaches u_1 in one step with probability at most $\frac{1}{2}$. If v_1 reversely reaches a vertex in S^* such that the first step of the reverse random walk is not towards u_1 , Lemma 4.7 implies that the probability this happens is at most $\frac{k}{k+1}$. Putting together, for $w = v_1$, $\Pr(S^* \rightarrow v_1 \mid \{v_1\} \xrightarrow{\mathcal{S}} w) \leq \frac{1}{2} + \frac{1}{2} \cdot \frac{k}{k+1}$. Therefore, it is always true that $\Pr(\neg(S^* \rightarrow v_1) \mid \{v_1\} \xrightarrow{\mathcal{S}} w) \geq \frac{1}{2(k+1)}$.

Finally, we consider $\Sigma(\{v_1\}) \setminus \Sigma(S^*)$ by only accounting for those vertices in $\Gamma(u_1) \setminus S^*$.

$$\begin{aligned}
\frac{|\Sigma(\{v_1\}) \setminus \Sigma(S^*)|}{\prod_{w \in V} \deg(w)} &\geq \sum_{w \in \Gamma(u_1) \setminus S^*} \Pr\left(\left(\{v_1\} \xrightarrow{\mathcal{S}} w\right) \wedge \neg(S^* \rightarrow v_1)\right) \\
&\geq \sum_{w \in \Gamma(u_1) \setminus S^*} \frac{1}{2(k+1)} \Pr\left(\{v_1\} \xrightarrow{\mathcal{S}} w\right) \\
&> \frac{1}{2(k+1)} \cdot \frac{9\sigma(S^*)}{2000k^2} \quad (\text{result from Step 1}) \\
&> \frac{1}{4000k^3} \frac{|\Sigma(S^*)|}{\prod_{w \in V} \deg(w)}.
\end{aligned}$$

By Lemma 4.5, this implies $|\Sigma(S)| \geq (1 - (1 - \frac{1}{k})^k + \frac{1}{64000k^3}) |\Sigma(S^*)|$, which further implies this proposition. \square

Finally, we prove that C_3 cannot be too large if the greedy algorithm does not overcome the $1 - (1 - 1/k)^k$ barrier. Informally, this is because C_3 corresponds to a subset of the intersection among $\Sigma(\{u_1\}), \dots, \Sigma(\{u_k\})$, and Lemma 4.4 implies that it cannot be too large.

PROPOSITION 4.12. *If $C_3 > \frac{1}{k^2} \cdot \sigma(S^*)$, then $\sigma(S) \geq (1 - (1 - \frac{1}{k})^k + \frac{1}{8k^3}) \sigma(S^*)$.*

PROOF. Notice that $C_3 \prod_{w \in V} \deg(w)$ is at most the number of tuples (w, g) such that w is reachable from more than one vertex in S^* under g . It is easy to see that

$$C_3 \prod_{w \in V} \deg(w) \leq \left(\sum_{i=1}^k |\Sigma(\{u_i\})| \right) - |\Sigma(S^*)|$$

because: 1) each (w, g) such that w is reachable by more than one vertex in S^* under g is counted at most once by $C_3 \prod_{w \in V} \deg(w)$, exactly once by $\Sigma(S^*)$, and at least twice by $\sum_{i=1}^k \Sigma(\{u_i\})$, so the contribution of each such (w, g) to the right-hand side of the inequality is at least the contribution of it to the left-hand side; 2) each (w, g) such that w is reachable by exactly one vertex in S^* under g is not counted by $C_3 \prod_{w \in V} \deg(w)$ and is counted exactly once by both $\sum_{i=1}^k \Sigma(\{u_i\})$ and $\Sigma(S^*)$, so the contribution of such (w, g) is the same on both sides of the inequality; 3) each (w, g) such that g is not reachable from S^* contributes 0 to both sides of the inequality. Observing this inequality, if $C_3 > \frac{1}{k^2} \cdot \sigma(S^*)$, we have

$$\left(\sum_{i=1}^k |\Sigma(\{u_i\})| \right) - |\Sigma(S^*)| > \frac{1}{k^2} \sigma(S^*) \prod_{w \in V} \deg(w) = \frac{1}{k^2} |\Sigma(S^*)|.$$

Lemma 4.4 implies $|\Sigma(S)| \geq (1 - (1 - \frac{1}{k})^k + \frac{1}{8k^3}) |\Sigma(S^*)|$, which implies this proposition. \square

With Proposition 4.10, 4.11 and 4.12, if $\sigma(S) = (1 - (1 - 1/k)^k + o(1/k^3)) \sigma(S^*)$, it must be that $\frac{|\Sigma(\{v_1\}) \cap \Sigma(S^*)|}{\prod_{w \in V} \deg(w)} = C_1 + C_2 + C_3 \leq \left(\frac{1}{k^2} + \frac{9}{10k} + \frac{1}{100k} \right) \sigma(S^*) < \frac{92}{100k} \frac{|\Sigma(S^*)|}{\prod_{w \in V} \deg(w)}$. However, Lemma 4.3 then would have implied $\sigma(S) \geq (1 - (1 - \frac{1}{k})^k + \frac{8}{400k}) \sigma(S^*)$, which is a contradiction. This finishes proving Theorem 4.1.

REFERENCES

- [1] Rico Angell and Grant Schoenebeck. 2017. Don't be greedy: Leveraging community structure to find high quality seed sets for influence maximization. In *International Conference on Web and Internet Economics*. Springer, Indian Institute of Science, Bangalore, India, 16–29.
- [2] Akhil Arora, Sainyam Galhotra, and Sayan Ranu. 2017. Debunking the Myths of Influence Maximization. In *Proceedings of the 2017 ACM International Conference on Management of Data-SIGMOD'17*. ACM, Chicago, Illinois, USA, 651–666.
- [3] S Bharathi, D Kempe, and M Salek. 2007. Competitive influence maximization in social networks. In *WINE*. Springer, Indian Institute of Science, Bangalore, India, 306–311.
- [4] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. 2014. Maximizing social influence in nearly optimal time. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, Portland, Oregon, USA, 946–957.
- [5] Ning Chen. 2009. On the approximability of influence in social networks. *SIAM Journal on Discrete Mathematics* 23, 3 (2009), 1400–1415.
- [6] Wei Chen and Binghui Peng. 2019. On Adaptivity Gaps of Influence Maximization under the Independent Cascade Model with Full Adoption Feedback. *CoRR* abs/1907.01707 (2019). arXiv:1907.01707 <http://arxiv.org/abs/1907.01707>
- [7] Wei Chen, Binghui Peng, Grant Schoenebeck, and Biaoshuai Tao. 2019. Adaptive Greedy versus Non-adaptive Greedy for Influence Maximization. *arXiv preprint arXiv:1911.08164* (2019).
- [8] Wei Chen, Chi Wang, and Yajun Wang. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, Washington, DC, USA, 1029–1038.
- [9] Wei Chen, Yajun Wang, and Siyu Yang. 2009. Efficient influence maximization in social networks. In *ACM SIGKDD*. ACM, Paris, France, 199–208.
- [10] Wei Chen, Yifei Yuan, and Li Zhang. 2010. Scalable influence maximization in social networks under the linear threshold model. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, Sydney, Australia, 88–97.
- [11] Wei Chen, Yifei Yuan, and Li Zhang. 2010. Scalable Influence Maximization in Social Networks under the Linear Threshold Model. In *2010 IEEE International Conference on Data Mining*. IEEE, Sydney, Australia, 88–97.
- [12] Suqi Cheng, Huawei Shen, Junming Huang, Guoqing Zhang, and Xueqi Cheng. 2013. Staticgreedy: solving the scalability-accuracy dilemma in influence maximization. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. ACM, San Francisco, California, USA, 509–518.
- [13] Pedro Domingos and Matt Richardson. 2001. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, San Francisco, California, USA, 57–66.
- [14] Uriel Feige. 1998. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM (JACM)* 45, 4 (1998), 634–652.
- [15] Sainyam Galhotra, Akhil Arora, and Shourya Roy. 2016. Holistic influence maximization: Combining scalability and efficiency with opinion-aware models. In *Conference on Management of Data*. ACM, San Francisco, California, USA, 743–758.
- [16] Sharon Goldberg and Zhenming Liu. 2013. The diffusion of networking technologies. In *Proceedings of the twenty-fourth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, New Orleans, Louisiana, USA, 1577–1594.
- [17] Daniel Golovin and Andreas Krause. 2011. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of AI Research* 42 (2011), 427–486.
- [18] Amit Goyal, Wei Lu, and Laks VS Lakshmanan. 2011. Celf+: optimizing the greedy algorithm for influence maximization in social networks. In *Proceedings of the 20th international conference WWW*. ACM, Hyderabad, India, 47–48.
- [19] Amit Goyal, Wei Lu, and Laks VS Lakshmanan. 2011. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, Vancouver, Canada, 211–220.
- [20] Kai Han, Keke Huang, Xiaokui Xiao, Jing Tang, Aixin Sun, and Xueyan Tang. 2018. Efficient algorithms for adaptive influence maximization. *Proceedings of the VLDB Endowment* 11, 9 (2018), 1029–1040.
- [21] Kyomin Jung, Wooram Heo, and Wei Chen. 2012. IRIE: Scalable and robust influence maximization in social networks. In *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE, Brussels, Belgium, 918–923.
- [22] David Kempe, Jon M. Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *ACM SIGKDD*. ACM, Washington, D.C., USA, 137–146.
- [23] David Kempe, Jon M. Kleinberg, and Éva Tardos. 2005. Influential Nodes in a Diffusion Model for Social Networks. In *ICALP*. Springer, Lisboa, Portugal, 1127–1138.
- [24] Sanjeev Khanna and Brendan Lucier. 2014. Influence maximization in undirected networks. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, Portland, Oregon, USA, 1482–1496.
- [25] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne Van-Briesen, and Natalie Glance. 2007. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, San Jose, California, USA, 420–429.
- [26] Qiang Li, Wei Chen, Xiaoming Sun, and Jialin Zhang. 2017. Influence Maximization with ϵ -Almost Submodular Threshold Functions. In *NIPS*. Curran Associates, Long Beach, CA, USA, 3804–3814.
- [27] Yongwhan Lim, Asuman Ozdaglar, and Alexander Teytelboym. 2015. A simple model of cascades in networks. (2015).
- [28] Elchanan Mossel and Sébastien Roch. 2010. Submodularity of Influence in Social Networks: From Local to Global. *SIAM J. Comput.* 39, 6 (2010), 2176–2188.
- [29] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming* 14, 1 (1978), 265–294.
- [30] Naoto Ohsaka, Takuya Akiba, Yuichi Yoshida, and Ken-ichi Kawarabayashi. 2014. Fast and Accurate Influence Maximization on Large Networks with Pruned Monte-Carlo Simulations. In *AAAI*. AAAI, Quebec City, Quebec, Canada, 138–144.
- [31] Binghui Peng and Wei Chen. 2019. Adaptive influence maximization with myopic feedback. In *Advances in Neural Information Processing Systems*. Curran Associates, Vancouver, Canada, 5575–5584.
- [32] M. Richardson and P. Domingos. 2002. Mining knowledge-sharing sites for viral marketing. In *ACM SIGKDD*. ACM, Edmonton Alberta, Canada, 61–70.
- [33] Grant Schoenebeck and Biaoshuai Tao. 2017. Beyond Worst-Case (In)approximability of Nonsubmodular Influence Maximization. In *International Conference on Web and Internet Economics*. Springer, Bangalore, India, 368–382.
- [34] Grant Schoenebeck and Biaoshuai Tao. 2019. Beyond Worst-case (In)approximability of Nonsubmodular Influence Maximization. *ACM Trans. Comput. Theory* 11, 3, Article 12 (April 2019), 56 pages. <https://doi.org/10.1145/3313904>
- [35] Grant Schoenebeck and Biaoshuai Tao. 2019. Influence Maximization on Undirected Graphs: Towards Closing the $(1 - 1/e)$ Gap. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. ACM, Phoenix, Arizona, USA, 423–453.
- [36] Grant Schoenebeck, Biaoshuai Tao, and Fang-Yi Yu. 2019. Think Globally, Act Locally: On the Optimal Seeding for Nonsubmodular Influence Maximization. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019) (Leibniz International Proceedings in Informatics (LIPIcs))*, Dimitris Achlioptas and László A. Végh (Eds.), Vol. 145. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 39:1–39:20. <https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2019.39>
- [37] Youze Tang, Yanchen Shi, and Xiaokui Xiao. 2015. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, Melbourne Victoria, Australia, 1539–1554.
- [38] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence maximization: Near-optimal time complexity meets practical efficiency. In *SIGMOD international conference on Management of data*. ACM, Snowbird, Utah, USA, 75–86.
- [39] Yaron Singer Thibaut Horel. 2016. Maximization of Approximately Submodular Functions. In *NIPS*. Curran Associates, Barcelona, Spain, 3045–3053.