How to Influence People with Partial Incentives*

Erik D. Demaine
MIT
edemaine@mit.edu

MohammadTaghi Hajiaghayi
University of Maryland
hajiagha@cs.umd.edu

Hamid Mahini
University of Maryland
hamid.mahini@gmail.com

David Malec
University of Maryland
dmalec@umiacs.umd.edu

S. Raghavan
University of Maryland
raghavan@umd.edu

Anshul Sawant
University of Maryland
asawant@cs.umd.edu

Morteza Zadimoghadam
MIT
morteza@csail.mit.edu

ABSTRACT

We study the power of fractional allocations of resources to maximize our influence in a network. This work extends in a natural way the well-studied model by Kleinberg, Kempe, and Tardos (2003), where a designer selects a (small) seed set of nodes in a social network to influence directly, this influence cascades when other nodes reach certain thresholds of neighbor influence, and the goal is to maximize the final number of influenced nodes. Despite extensive study from both practical and theoretical viewpoints, this model limits the designer to a binary choice for each node, with no chance to apply intermediate levels of influence. This model captures some settings precisely, such as exposure to an idea or pathogen, but it fails to capture very relevant concerns in others, for example, a manufacturer promoting a new product by distributing five "20% off" coupons instead of giving away a single free product.

While fractional versions of problems tend to be easier to solve than integral versions, for influence maximization, we show that the two versions have essentially the same computational complexity. On the other hand, the two versions can have vastly different solutions: the added flexibility of fractional allocation can lead to vastly improved influence. Our main theoretical contribution is to show how to adapt the major positive results from the integral case to the fractional case. Specifically, Mossel and Roch (2006) used the submodularity of influence to obtain their integral results; we introduce a new notion of *continuous submodularity*, and use this to obtain matching fractional results. We conclude that similar algorithms and heuristics from the integral case apply to the fractional case. In practice, we find that the fractional model performs substantially better than the integral model, according to simulations on real-world social network data.

*A full version of this paper is available at `http://www.cs.umd.edu/~hmahini/fracinf.pdf`

WWW 2014, Seoul, Korea

1. INTRODUCTION

The ideas we are exposed to and the choices we make are significantly influenced by our social context. It has long been studied how our social network (i.e., who we interact with) impacts the choices we make, and how ideas and behaviors can spread through social networks [3, 10, 21, 22]. With websites such as Facebook and Google+ devoted to the forming and maintaining of social networks, this effect becomes ever more evident. Individuals are linked together more explicitly and measurable, making it both easier and more important to understand how social networks affect the behaviors and actions that spread through a society.

A key problem in this area is to understand how such a behavioral cascade can start. For example, if a company wants to introduce a new product but has a limited promotional budget, it becomes critical to understand how to target their promotional efforts in order to generate awareness among as many people as possible. A well-studied model for this is the Influence Maximization problem, introduced by Kleinberg, Kempe, and Tardos [15]. The problem asks to find a small set of individuals to influence, such that this influence will cascade and grow through the social network to the maximum extent possible. For example, if a company wants to introduce a new piece of software, and believes that friends of users are likely to become users themselves, how should they allocate free copies of their software in order to maximize the size of their eventual user base?

Since the introduction of the Influence Maximization problem [15], there has been a great deal of interest and follow-up work in the model. While Kempe et al. [15] give a greedy algorithm for approximating the Influence Maximization problem, it requires costly simulation at every step; thus, while their solution provides a good benchmark, a key area of research has been on finding practical, fast algorithms that themselves provide good approximations to the greedy algorithm [4, 5, 6, 7, 16]. The practical, applied nature of the motivating settings means that even small gains in performance (either runtime or approximation factor) are critical, especially on large, real-world instances.

We believe that the standard formulation of the Influence Maximization problem, however, misses a critical aspect of practical applications. In particular, it forces a binary choice upon the optimizer, forcing a choice of either zero or complete influence on each individual, with no options in between. While this is reasonable for some settings — e.g., exposure to an idea or pathogen – it is far less reasonable for other settings of practical importance. For example, a company promoting a new product may find that giving away ten free copies is far less effective than offering a discount of ten percent to a hundred people. We propose a *fractional* version of the problem where the optimizer has the freedom to split influence across individuals as they see fit.

To make this concrete, consider the following problem an optimizer might face. Say that an optimizer feels there is some small, well-connected group whose adoption of their product is critical to success, but only has enough promotion budget remaining to influence one third of the group directly. In the original version of Influence Maximization, the optimizer is forced to decide which third of the group to focus on. We believe it is more natural to assume they have the flexibility to try applying uniform influence to the group, say offering everyone a discount of one third on the price of their product, or in fact any combination of these two approaches. While our results are preliminary, we feel that our proposed model addresses some very real concerns with practical applications of Influence Maximization, and offers many opportunities for important future research.

1.1 Our Results and Techniques

This work aims to understand how our proposed fractional version of the Influence Maximization problem differs from the integral version proposed by Kleinberg, Kempe, and Tardos [15]. We consider this question from both a theoretical and an empirical perspective. On the theoretical side, we show that, unlike many problems, the fractional version appears to retain essentially the same computational hardness as the integral version. The problems are very different, however: we give examples where the objective values for the fractional and integral versions differ significantly. Nonetheless, we are able to carry over the main positive results to the fractional setting. On the empirical side, we simulate the main algorithms and heuristics on real-world social network data, and find that the computed solutions are substantially more efficient in the fractional setting.

Our main theoretical result shows that the positive results of Mossel and Roch [18] extend to our proposed fractional model. Their result states that, in the integral case, when influence between agents is submodular, so too is the objective function in Influence Maximization. We show that, for a continuous analog of submodularity,[1] the same results holds for our fractional case. First we consider a discretized version of the fractional Influence Maximization Problem, where each vertex can be assigned a weight that is a multiple of some discretization parameter $\varepsilon = \frac{1}{N}$. Then we consider the final influenced set by choosing a weighted seed set S, where the weight of each element is a multiple of ε. We show that the fractional Influence Maximization objective is a submodular function of S for any $N \geq 1$ (Theorem 2). We further extend this result to the continuous case (Theorem 3). Note that this result does not follow simply by relating the fractional objective function to the integral objective and interpolating, or other similar methods; instead, we need to use a nontrivial reduction to the generalization of the influence maximization problem given by Mossel and Roch [18]. Not only does this result show that our problem admits a greedy solution with good approximation guarantee, it furthermore gives us hope that we can readily adapt the large body of work on efficient heuristics for the integral case to our problem and achieve good results.

In addition to showing the submodularity of the objective persists from the integral case to the fractional case, we show that the hardness of the integral case persists as well. In the case of fixed thresholds, we show that all of the hardness results of Kempe et al. [15] extend readily to the fractional case. Specifically, we show that, for the fractional version of linear influence model, even finding an $n^{1-\varepsilon}$-approximation is NP-hard. First we prove NP-hardness of the problem by a reduction from the independent set problem (Theorem 6), and then we strengthen the result to prove inapproximability (Corollary 7). In addition, when thresholds are assumed to be independent and uniformly distributed in $[0, 1]$, we show that it is NP-hard to achieve better than a $(1 - 1/e)$-approximation in the Triggering model introduced by Kempe et al. [15]. This holds even for the simple case where triggering sets are deterministic and have constant sizes, and shows that even for this simple case the greedy approximation is tight, just as in the integral case. An important aspect of all of these reductions is that they use very simple directed acyclic graphs (DAGs), with only two layers of vertices.

Our last set of results focus on the special case where the network is a DAG. Here, we focus on the linear influence model with uniform thresholds. In this case, we see that we can easily compute the expected influence from any single node via dynamic programming; this closely resembles a previous result for the integral case [7]. In the fractional case, this gives us a sort of linearity result. Namely, if we are careful to avoid interference between the influences we place on nodes, we can conclude that the objective is essentially linear in the seed set. While the conditions on this theorem seem strong at first glance, it has a very powerful implication: all of the hardness results we presented involved choosing optimal seed sets from among the sources in a DAG, and this theorem says that with uniform thresholds the greedy algorithm finds the *optimal* such seed set.

1.2 Related Work

Economics, sociology, and political science have all studied and modeled behaviors arising from information and influence cascades in social networks. Some of the earliest models were proposed by Granovetter [10] and Schelling [21]. Since then, many such models have been studied and proposed in the literature [3, 20, 22].

The advent of social networking platforms such as Facebook, Twitter, and Flickr has provided researchers with unprecedented data about social interactions, albeit in a virtual setting. The question of monetizing this data is critically important for the entities that provide these platforms and the entities that want to leverage this data to engineer effective marketing campaigns. These two factors have generated huge interest in algorithmic aspects of these systems.

A question of central importance is to recognize "important individuals" in a social network. Domingos and Richardson [8, 19] were the first to propose heuristics for selection of customers on a network for marketing. This work focuses on evaluating customers based on their intrinsic and network value. The network value is assumed to be generated by a customer influencing other customers in her social network to buy the product. In a seminal paper, Kempe et al. [15] give an approximation algorithm for selection of influential nodes under the linear threshold (LT) model. Mossel and Roch [17] generalized the results of Kempe et al. [15] to cases where the activation functions are monotone and submodular. Gunnec and Raghavan [12] were the first to discuss fractional incentives (they refer to these as partial incentives/inducements) in the context of a product design problem. They consider a fractional version of the target set selection problem (i.e., fixed thresholds, fractional incentives, a linear influence model, with the goal of minimizing the fractional incentives paid out so that all nodes in the graph are influenced). They provide an integer programming

[1] Note that our notion of continuous submodularity is neither of the two most common continuous extensions of submodularity, namely the multilinear and Lovász extensions.

model, and show that when the neighbors of a node have equal influence on it, the problem is polynomially solvable via a greedy algorithm [11, 12, 13].

Some recent work has directly tackled the question of revenue maximization in social networks by leveraging differential pricing to monetize positive externalities arising due to adoption of product by neighbors of a customer [1, 2, 9, 14]. Other work has focused on finding faster algorithms for the target set selection problem [5, 6, 7, 16]. A very recent theoretical result in this direction is an $O(\frac{(m+n)\log n}{\varepsilon^3})$ algorithm giving an approximation guarantee of $1-\frac{1}{e}-\varepsilon$ [4]. While Leskovec et al. [16] do not compare their algorithm directly with the greedy algorithm of Kempe et al. [15], the heuristics in other papers [5, 6, 7] approach the performance of the greedy algorithm quite closely. For example, in [6], the proposed heuristic achieves an influence spread of approximately 95% of the influence spread achieved by the greedy algorithm. An interesting fact on the flip side is that none of the heuristics beat the greedy algorithm (which itself is a heuristic) for even a single data set.

2. MODEL

Integral Influence Model. We begin by describing the model used for propagation of influence in social networks used by Mossel and Roch [18]; it captures the model of Kempe et al. [15] as a special case. While the latter described the spread of influence in terms of an explicit network, the former leaves the underlying social network implicit. In this model, the social network is given by a vertex set V and an explicit description of how vertices influence each other. For each vertex $v \in V$, we are given a function $f_v : 2^V \to [0,1]$ specifying the amount of influence each subset $S \subseteq V$ exerts on v. We denote the set of all influence functions by $\mathcal{F} = \{f_v\}_{v\in V}$.

Given a social network specified by $(V, \mathcal{F})$, we want to understand how influence propagates in this network. The spread of influence is modeled by a process that runs in discrete stages. In addition to the influence function f_v, each vertex v has a threshold $\theta_v \in [0,1]$ representing how resistant it is to being influenced. If, at a given stage, the currently activated set of vertices is $S \subseteq V$, then any unactivated $v \in V \setminus S$ becomes activated in the next stage if and only if $f_v(S) \geq \theta_v$. Our goal is to understand how much influence different sets of vertices exert on the social network as a whole under this process; we can measure this by running this process to completion starting with from a particular *seed set*, and seeing how large the final activated set is. In some settings, we may value activating certain (sets of) vertices more highly, and to capture this we define a weight function $w : 2^V \to \mathbb{R}_+$ on subsets of V. We now define the value of a seed set S as follows. For an initially activated set S_0, let $S_1^\Theta, S_2^\Theta, \ldots, S_n^\Theta$ be the activated sets after $1, 2, \ldots, n = |V|$ stages of our spreading process, when $\Theta = (\theta_v)_{v\in V}$ is our vector of thresholds. Our goal is understanding the value of $w(S_n^\Theta)$ when we set $S_0 = S$. Note this depends strongly on Θ; the exact values of thresholds can have a significant impact on the final activated set. If the vector Θ can be arbitrary, finding the best seed set – or even any nontrivial approximation of it – becomes NP-Hard (see Section 5 for discussion and proofs of this). Thus, we follow the lead of Kempe et al. [15] and assume that each threshold is independently distributed as $\theta_v \sim \mathcal{U}[0,1]$. Then, our goal in this problem is understanding the structure of the function $\sigma : 2^V \to \mathbb{R}_+$ given by

$$\sigma(S) = \mathop{\mathrm{E}}_{\Theta}[\ w(S_n^\Theta) \mid S_0 = S\ ],$$

with the goal of finding seed sets S maximizing $\sigma(S)$.

Fractional Influence Model. A major shortcoming of the model described above is that it isolates the effects of influence directly applied by the optimizer from those of influence cascading from other individuals in the network. In particular, note that every individual in the social network is either explicitly activated by the optimizer (and influence from their neighbors has no effect), or is activated by influence from other agents with no (direct) involvement from the optimizer. This separation is artificial, however, and in practical settings a clever optimizer could try to take advantage of known influences between the individuals they try to affect. For example, if an optimizer is already planning to activate some set S of individuals, it should require notably less effort to ensure activation of any individual who is heavily influenced by the set S.

We propose the following modification of the previously described influence model in order to capture this phenomenon. Rather than selecting a set S of nodes to activate, the optimizer specifies a vector $\mathbf{x} \in [0,1]^n$ indexed by V, where x_v indicates the amount of direct influence we apply to v. We assume that this direct influence is additive with influence from other vertices in the network, in the sense that if the current activated set is S in a stage of our process, v becomes activated in the next stage if and only if $f_v(S) + x_v \geq \theta_v$. Here, we assume that no vertices are initially activated, that is $S_0 = \emptyset$. Note, however, that even without contributions from other nodes, our directly-applied influence can cause activations. Notably, it is easy to see that

$$S_1^\Theta = \{v \in V : x_v \geq \theta_v\}.$$

We point out, however, that our process is not simply a matter of selecting an initial activated set at random with marginal probabilities $\mathbf{x}$. The influence x_v we apply to v not only has a chance to activate it at the outself, but also makes it easier for influence from other vertices to activate it in all future stages of the process. Lastly, we observe that this model captures the originally discussed model as a special case, since selecting sets to initially activate corresponds exactly with choosing $\mathbf{x} \in \{0,1\}^n$, just with a single-round delay in the process. This motivates us to term the original model as the integral influence model, and this new model as the fractional influence model. As before, we want to understand the structure of the expected value of the final influenced set as a function of how we apply influence to nodes in a graph. We extend our function to $\sigma : [0,1]^n \to \mathbb{R}_+$ by

$$\sigma(\mathbf{x}) = \mathop{\mathrm{E}}_{\Theta}[\ w(S_n^\Theta) \mid \text{we apply direct influences } \mathbf{x}\ ].$$

As before, we want to both understand the structure of σ and be able to find (approximately) optimal inputs $\mathbf{x}$.

Gap Between Integral and Fractional. A natural question when presented with a new model is whether it provides any additional power over the previous one. Here, we answer that question in the affirmative for the fractional extension of the Influence Maximization model. In particular, we present two examples here that show that fractional influence can allow a designer to achieve strictly better results than integral influence for a particular budget. The first example shows that with fixed thresholds, this gap is linear (perhaps unsurprisingly, given the hardness of the problem under hard thresholds). The second example, however, shows that even with independent uniform thresholds an optimizer with the power to apply fractional influence can see an improvement of up to a factor of $(1 - 1/e)$.

EXAMPLE 1. *The following example shows that when thresholds are fixed, the optimal objective values in the fractional and integral cases can differ by as much as a factor of n, where n is* *the number of vertices in the graph. The instance we consider is a DAG consisting of a single, directed path of n vertices. Each edge in the path has weight $1/(n+1)$, and every vertex on the path has threshold $2/(n+1)$. Note that since thresholds are strictly greater than edge weights, and every vertex, being on a simple path, has in degree at most one, it is impossible for a vertex to be activated without some direct influence being applied to it.*

Consider our problem on the above graph with budget 1. In the integral case, we cannot activate more than a single vertex – as previously observed, no vertex can be activated without direct application of influence, and with a budget of 1 we can only affect one vertex directly. On the other hand, in the fractional case the following strategy guarantees that all vertices are activated. Apply $2/(n+1)$ influence to the earliest vertex, and $1/(n+1)$ influence to the remaining $(n-1)$ vertices. Now, this is sufficient to activate the earliest vertex directly; furthermore, every other vertex has sufficient direct influence that it will activate as long as the vertex before it on the path does. Thus, a simple induction proves the claim, and we can see that the optimal integral and fractional solutions differ in objective value by a factor of n.

EXAMPLE 2. *Consider solving our problem on a directed graph consisting of a single (one-directional) cycle with n vertices. Assume that every edge has weight $1-K/n$, and that thresholds on nodes are drawn from $\mathcal{U}[0,1]$. We consider the optimal integral and fractional influence to apply.*

In the fractional case, consider applying influence of exactly K/n to every node. Note that for any node, the amount of influence we apply directly plus the weight on its sole incoming edge sum to 1. Thus, any time a node's predecessor on the cycle becomes activated, the node will become activated as well. Inductively, we can then see that any time at least one node is activated in the cycle, every node will eventually become activated. This means that the expected number of activated nodes under this strategy is precisely

$$\begin{aligned} &n \cdot \Pr[\textit{At least one node activates}] \\ &= n(1 - \Pr[\textit{Every node's threshold is above } K/n]) \\ &= n(1-(1-K/n)^n). \end{aligned}$$

In the integral case, however, we cannot spread our influence as evenly. Note that each node we activate has some chance to activate the nodes following it in the cycle; however, any cascade must stop once we reach the next node we directly activated. If we have an interval of length ℓ between directly activated nodes (including the initial node we activate directly in the length), we can see that the expected number of nodes activated in the interval is

$$\begin{aligned} &\sum_{i=1}^{\ell} \Pr[\textit{Node } i \textit{ in the interval is activated}] \\ &= \sum_{i=1}^{\ell} \Pr[\textit{Nodes } 2,3,\ldots,i \textit{ have thresholds below } 1-K/n] \\ &= \sum_{i=1}^{\ell} (1-K/n)^{i-1} = \frac{1-(1-K/n)^{\ell}}{K/n}. \end{aligned}$$

While this tells us the expected value for a single interval, we want to know the expected value summed over all intervals. Observing from the sum form that the benefit of adding another node to an interval is strictly decreasing in the length of the interval, we can see that we should always make the lengths of the intervals as close to equal as possible. Noting that the lengths of the intervals always sum to n, then, we can see that the total number of nodes activated in expectation is bounded by

$$K\frac{1-(1-K/n)^{n/K}}{K/n} = n(1-(1-K/n))^{n/K}.$$

Note, however, that if we choose $K \approx \ln n$, we get that

$$\frac{1-(1-K/n)^{n/K}}{1-(1-K/n)^n} \approx 1 - 1/e.$$

3. REDUCTION

In this section, we extend the submodularity results of Mossel and Roch [18] for the integral version of Influence Maximization to the fractional version. At a high level, our approach revolves around reducing a fractional instance to an integral one, such that evolution of the process and objective values are preserved. Thus, before presenting our extension, we begin by stating the main result of [18]. Before stating the theorems, however, we give definitions for the function functions properties each requires. Finally, we note that our main result of the section (Theorem 2) considers a *discretization* of the input space; at the end of this section we show that such discretization cannot affect our objective value by too much.

We begin by giving definitions for the following properties of set functions. Given a set N and a function $f : 2^N \to \mathbb{R}$, we say that:

- f is *normalized* if $f(\emptyset) = 0$;
- f is *monotone* if $f(S) \le f(T)$ for any $S \subseteq T \subseteq N$; and
- f is *submodular* if $f(S \cup \{x\}) - f(S) \ge f(T \cup \{x\}) - f(T)$ for any $S \subseteq T \subseteq N$ and $x \in N \setminus T$.

We say that a collection of functions satisfies the above properties if every function in the collection does. With the above definitions in hand, we are now ready to state the following result of Mossel and Roch.

THEOREM 1. *(Restatement of [18, Theorem 1.6]) Let $\mathcal{I} = (V, \mathcal{F}, w)$ be an instance of integral Influence Maximization. If both w and $\mathcal{F}$ are normalized, monotone, and submodular, then σ is as well.*

We want to extend Theorem 1 to the fractional influence model. We proceed by showing that for arbitrarily fine discretizations of $[0,1]$, any instance of our problem considered on the discretized space can be reduced to an instance of the original problem. Fix $N \in \mathbb{Z}_+$, and let $\delta = 1/N > 0$ be our discretization parameter. Let $\Delta = \{0, \delta, 2\delta, \ldots, 1\}$. We consider the fractional objective function σ restricted to the domain Δ^n. Lastly, let δ_v be the vector with δ in the component corresponding to v, and 0 in all other components. We extend the relevant set function properties to this discretized space as follows:

- we say σ is *normalized* if $\sigma(\mathbf{0}) = 0$;
- we say σ is *monotone* if $\mathbf{x} \le \mathbf{y}$ implies $\sigma(\mathbf{x}) \le \sigma(\mathbf{y})$; and
- we say σ is *submodular* if for any $\mathbf{x} \le \mathbf{y}$, and any $v \in V$, either $y_v = 1$ or $\sigma(\mathbf{x} + \delta_v) - \sigma(\mathbf{x}) \ge \sigma(\mathbf{y} + \delta_v) - \sigma(\mathbf{y})$,

where all comparisons and additions between vectors above are componentwise. We get the following extension of Theorem 1.

THEOREM 2. *Let $\mathcal{I} = (V, \mathcal{F}, w)$ be an instance of fractional Influence Maximization. For any discretization Δ^n of $[0,1]^n$ (as defined above), if both w and $\mathcal{F}$ are normalized, monotone, and submodular, then σ is normalized, monotone, and submodular on Δ^n.*

PROOF. We prove this by reducing and instance of the (discretized) fractional problem for $\mathcal{I}$ to an instance of the integral influence problem and then applying Theorem 1. We begin by modifying $\mathcal{I}$ to produce a new instance $\hat{\mathcal{I}} = (\hat{V}, \hat{\mathcal{F}}, \hat{w})$. Then, we show that $\hat{\mathcal{F}}$ and $\hat{w}$ will retain the properties of normalization, monotonicity, and submodularity. Lastly, we show a mapping from (discretized) fractional activations for $\mathcal{I}$ to integral activations for $\hat{\mathcal{I}}$ such that objective values are preserved, and our desired fractional set function properties for σ correspond exactly to their integral counterparts for the objective function $\hat{\sigma}$ for $\hat{\mathcal{I}}$. The result then follows immediately from Theorem 1.

We begin by constructing the instance $\hat{\mathcal{I}}$. The key idea is that we can simulate fractional activation with integral activation by adding a set of dummy activator nodes for each original node; each activator node applies an incremental amount of pressure on its associated original node. Then, for each original node we just need to add the influence from activator node to that from other original nodes, and truncate the sum to one. Fortunately, both of the aforementioned operations preserve the desired properties. Lastly, in order to avoid the activator nodes interfering with objective values, we simply need to give them weight zero. With this intuition in mind, we now define $\hat{\mathcal{I}} = (\hat{V}, \hat{\mathcal{F}}, \hat{w})$ formally.

First, we construct $\hat{V}$. For each node $v \in V$, create a set $A_v = \{v^1, v^2, \ldots, v^{1/\delta}\}$ of activator nodes for v. Our node set in the new instance is

$$\hat{V} = V \cup \left(\bigcup_{v\in V} A_v\right).$$

We now proceed to define the functions $\hat{f}_{\hat{v}}$ for each $\hat{v} \in \hat{V}$. If $\hat{v}$ is an activator node for some $v \in V$, we simply set $\hat{f}_{\hat{v}} \equiv 0$; otherwise, $\hat{v} \in V$ and we set

$$\hat{f}_{\hat{v}}(S) = \min\left(f_{\hat{v}}(S \cap V) + \delta|S \cap A_{\hat{v}}|, 1\right)$$

for each $S \subseteq \hat{V}$. Lastly, we set

$$\hat{w}(S) = w(S \cap V)$$

for all $S \subseteq \hat{V}$. Together, these make up our modified instance $\hat{\mathcal{I}}$.

We now show that since w and $\mathcal{F}$ are normalized, monotone, and submodular, $\hat{w}$ and $\hat{\mathcal{F}}$ will be as well. We begin with $\hat{w}$, since it is the simpler of the two. Now, $\hat{w}$ is clearly normalized since $\hat{w}(\emptyset) = w(\emptyset)$. Similarly, for any $S \subseteq T \subseteq \hat{V}$, we have that $S \cap V \subseteq T \cap V$, and so

$$\hat{w}(S) = w(S \cap V) \leq w(T \cap V) = \hat{w}(T),$$

by the submodularity of w. Lastly, let $\hat{u} \in \hat{V} \setminus T$. If $\hat{u} \in V$,

$$\begin{aligned} \hat{w}(S \cup \{\hat{u}\}) - \hat{w}(S) &= w((S \cap V) \cup \{\hat{u}\}) - \hat{w}(S \cap V) \\ &\geq w((T \cap V) \cup \{\hat{u}\}) - \hat{w}(T \cap V) \\ &= \hat{w}(T \cup \{\hat{u}\}) - \hat{w}(T), \end{aligned}$$

since w is submodular. On the other hand, if $\hat{u} \notin V$, we immediately get that

$$\hat{w}(S \cup \{\hat{u}\}) - \hat{w}(S) = 0 = \hat{w}(T \cup \{\hat{u}\}) - \hat{w}(T).$$

Thus, we can see that $\hat{w}$ is normalized, monotone, and submodular.

Next, we show that $\hat{\mathcal{F}}$ is normalized, monotone, and submodular. For $\hat{v} \in \hat{V} \setminus V$, it follows trivially since $\hat{\mathcal{F}}$ is identically 0. In the case that $\hat{V} \in V$, it is less immediate, and we consider each of the properties below.

- $\hat{f}_{\hat{v}}$ normalized. This follows by computing that

$$\begin{aligned} \hat{f}_{\hat{v}}(\emptyset) &= \min\left(f_{\hat{v}}(V \cap \emptyset) + \delta|A_{\hat{v}} \cap \emptyset|, 1\right) \\ &= \min\left(f_{\hat{v}}(\emptyset) + \delta|\emptyset|, 1\right) = 0, \end{aligned}$$

since $f_{\hat{v}}$ is normalized.

- $\hat{f}_{\hat{v}}$ monotone. Let $S \subseteq T \subseteq \hat{V}$. Then we have both $S \cap V \subseteq T \cap V$ and $S \cap A_{\hat{v}} \subseteq T \cap A_{\hat{v}}$. Thus, we can see that

$$\begin{aligned} f_{\hat{v}}(V \cap S) &\leq f_{\hat{v}}(V \cap T) \\ |A_{\hat{v}} \cap S| &\leq |A_{\hat{v}} \cap T|, \end{aligned}$$

where the former follows by the monotonicity of $f_{\hat{v}}$. Combining these, we get that

$$f_{\hat{v}}(V \cap S) + \delta|A_{\hat{v}} \cap S| \leq f_{\hat{v}}(V \cap T) + \delta|A_{\hat{v}} \cap T|.$$

Thus, we may conclude that $\hat{f}_{\hat{v}}(S) \leq \hat{f}_{\hat{v}}(T)$, since it follows from the above inequality that

$$f_{\hat{v}}(V \cap S) + \delta|A_{\hat{v}} \cap S| > \min\left(f_{\hat{v}}(V \cap T) + \delta|A_{\hat{v}} \cap T|, 1\right)$$

implies

$$\begin{aligned} & \min \quad \left(f_{\hat{v}}(V \cap S) + \delta|A_{\hat{v}} \cap S|, 1\right) = 1 \\ = & \min \quad \left(f_{\hat{v}}(V \cap T) + \delta|A_{\hat{v}} \cap T|, 1\right). \end{aligned}$$

- $\hat{f}_{\hat{v}}$ submodular. Let $S \subseteq T \subseteq \hat{V}$, and $\hat{u} \in \hat{V} \setminus T$. Now, we have three cases, depending on the choice of $\hat{u}$. If $\hat{u} \in V$, we have that $\hat{u} \notin A_{\hat{v}}$, and so

$$\begin{aligned} (f_{\hat{v}}(S \cup \{\hat{u}\}) - \delta|(S \cup \{\hat{u}\}) \cap A_{\hat{v}}|) - (f_{\hat{v}}(S) - \delta|S \cap A_{\hat{v}}|) \\ = f_{\hat{v}}(S \cup \{\hat{u}\}) - f_{\hat{v}}(S); \end{aligned}$$

and

$$\begin{aligned} (f_{\hat{v}}(T \cup \{\hat{u}\}) - \delta|(T \cup \{\hat{u}\}) \cap A_{\hat{v}}|) - (f_{\hat{v}}(T) - \delta|T \cap A_{\hat{v}}|) \\ = f_{\hat{v}}(T \cup \{\hat{u}\}) - f_{\hat{v}}(T). \end{aligned}$$

Thus, the submodularity of $f_{\hat{v}}$ implies the former is greater than or equal to the latter. On the other hand, if $\hat{u} \in A_{\hat{v}}$, then $\hat{u} \notin V$, and we can see that

$$\begin{aligned} (f_{\hat{v}}(S \cup \{\hat{u}\}) - \delta|(S \cup \{\hat{u}\}) \cap A_{\hat{v}}|) = \delta = \\ = (f_{\hat{v}}(T \cup \{\hat{u}\}) - \delta|(T \cup \{\hat{u}\}) \cap A_{\hat{v}}|). \end{aligned}$$

Lastly, if $\hat{u} \notin V \cup A_{\hat{v}}$, we can immediately see that

$$\begin{aligned} (f_{\hat{v}}(S \cup \{\hat{u}\}) - \delta|(S \cup \{\hat{u}\}) \cap A_{\hat{v}}|) = 0 = \\ = (f_{\hat{v}}(T \cup \{\hat{u}\}) - \delta|(T \cup \{\hat{u}\}) \cap A_{\hat{v}}|). \end{aligned}$$

Thus, in every case, we may conclude that

$$\begin{aligned} (f_{\hat{v}}(S \cup \{\hat{u}\}) - \delta|(S \cup \{\hat{u}\}) \cap A_{\hat{v}}|) \geq \\ \geq (f_{\hat{v}}(T \cup \{\hat{u}\}) - \delta|(T \cup \{\hat{u}\}) \cap A_{\hat{v}}|), \end{aligned}$$

and hence (by the same reasoning as for monotonicity) we may conclude that

$$\hat{f}_{\hat{v}}(S \cup \{\hat{u}\}) - \hat{f}_{\hat{v}}(S) \geq \hat{f}_{\hat{v}}(T \cup \{\hat{u}\}) - \hat{f}_{\hat{v}}(T).$$

Thus, we can see that $\hat{\mathcal{F}}$ is normalized, monotone, and submodular on $\hat{V}$, exactly as desired.

As such, we can apply Theorem 1 to our function and get that for our modified instance $\hat{\mathcal{I}} = (\hat{V}, \hat{\mathcal{F}}, \hat{w})$, the corresponding function $\hat{\sigma}$ must normalized, monotone, and submodular. All that remains is to demonstrate our claimed mapping from (discretized) fractional activations for $\mathcal{I}$ to integral activations for $\hat{\mathcal{I}}$.

We do so as follows. For each $v \in V$ and each $d \in \Delta$, let $A_v^d = \{v^1, v^2, \ldots, v^d\}$. Then, given the vector $\mathbf{x} \in \Delta^n$, we set

$$S^{\mathbf{x}} = \bigcup_{v\in V} A_v^{x_v},$$

where x_v is the component of $\mathbf{x}$ corresponding to the node v.

We first show that under this definition we have that $\sigma(\mathbf{x}) = \hat{\sigma}(S^{\mathbf{x}})$. In fact, as we will see the sets influenced will be the same not just in expectation, but for every set of thresholds Θ for the vertices V. Note that in the modified setting $\hat{\mathcal{I}}$ we also have thresholds for each vertex in $\hat{V} \setminus V$; however, since we chose $\hat{f}_{\hat{v}} \equiv 0$ for all $\hat{v} \in \hat{V} \setminus V$, and thresholds are independent draws from $\mathcal{U}[0,1]$, we can see that with probability 1 we have $\hat{f}_{\hat{v}}(S) < \theta_{\hat{v}}$ for all S and all $\hat{v} \in \hat{V} \setminus V$. Thus, in the following discussion we do not bother to fix these thresholds, as their precise values have no effect on the spread of influence.

Fix some vector Θ of thresholds for the vertices in V. Let $S_1^{\Theta}, \dots, S_n^{\Theta}$ and $\hat{S}_1^{\Theta}, \dots, \hat{S}_n^{\Theta}$ be the influenced sets in each round in the setting $\mathcal{I}$ with influence vector $\mathbf{x}$ and in the setting $\hat{\mathcal{I}}$ with influence set $S^{\mathbf{x}}$, respectively. We show by induction that for all $i = 0, 1, \dots, n$, $\hat{S}_i^{\Theta} \cap V = S_i^{\Theta}$. By the definition of $\hat{w}$, this immediately implies that $w(S_n^{\Theta}) = \hat{w}(\hat{S}_n^{\Theta})$, as desired.

While we give full details of the induction in the full version, we sketch the key ideas here. The key observation is that for each vertex $v \in V$, the only difference between $\mathcal{I}$ and $\hat{\mathcal{I}}$ is that in every stage v has x_v influence directly applied to it in the former but not the latter, and experiences x_v influence from elements of A_v in the latter but not the former. Since these have equivalent effects, the vertices in $\mathcal{I}$ and $\hat{\mathcal{I}}$ activate similarly. Observing that our definitions ensure $\hat{S}_0 \cap V = S^{\mathbf{x}} \cap V = \emptyset = S_0$ completes the induction.

We have now shown that for all vectors of thresholds Θ for vertices in V, with probability 1 we have that $\hat{S}_i^{\Theta} \cap V = S_i^{\Theta}$ for $i = 0, 1, \dots, n$. In particular, note that $\hat{S}_n^{\Theta} \cap V = S_n^{\Theta}$, and so $\hat{w}(\hat{S}_n^{\Theta}) = w(S_n^{\Theta})$. Thus, we may conclude that $\hat{\sigma}(S^{\mathbf{x}}) = \sigma(\mathbf{x})$.

Lastly, we need to show that for our given mapping from (discretized) fractional activation vectors $\mathbf{x}$ to set $S^{\mathbf{x}}$, we have that the desired properties for σ are satisfied if the corresponding properties are satisfied for $\hat{\sigma}$. So we assume that $\hat{\sigma}$ is normalized, monotone, and submodular (as, in fact, it must be by the above argument and Theorem 1), and show that σ is as well. First, note that $\mathbf{x} = \mathbf{0}$ implies $S^{\mathbf{x}} = \emptyset$, and so $\sigma(\mathbf{x}) = \hat{\sigma}(\emptyset) = 0$. Second, let $\mathbf{x}, \mathbf{y} \in \Delta^n$ such that $\mathbf{x} \leq \mathbf{y}$ componentwise. Then we can see that $S^{\mathbf{x}} \subseteq S^{\mathbf{y}}$ and so

$$\sigma(\mathbf{x}) = \hat{\sigma}(S^{\mathbf{x}}) \leq \hat{\sigma}(S^{\mathbf{y}}) = \sigma(\mathbf{y}).$$

Finally, pick some $v \in V$ such that $y_v < 1$. Recall our definition of $\hat{f}_{\hat{v}}$; by inspection, we can see that we have $\hat{f}_{\hat{v}}(S) = \hat{f}_{\hat{v}}(T)$ any time both $S \cap V = T \cap V$ and $|S \cap A_v| = |T \cap A_v|$, for any $S, T \in \hat{V}$. Thus, we can see that $S^{\mathbf{x}+\delta_i} = S^{\mathbf{x}} \cup \{v^{x_v+1}\}$ and $S^{\mathbf{y}} = S^{\mathbf{y}} \cup \{v^{y_v+1}\}$. So we have

$$\begin{aligned}
\sigma(\mathbf{x}+\delta_i) - \sigma(\mathbf{x}) &= \hat{\sigma}(S^{\mathbf{x}} \cup \{v^{x_v+1}\}) - \hat{\sigma}(S^{\mathbf{x}}) \\
&= \hat{\sigma}(S^{\mathbf{x}} \cup \{v^{y_v+1}\}) - \hat{\sigma}(S^{\mathbf{x}}) \\
&\geq \hat{\sigma}(S^{\mathbf{y}} \cup \{v^{y_v+1}\}) - \hat{\sigma}(S^{\mathbf{y}}) \\
&= \sigma(\mathbf{y}+\delta_i) - \sigma(\mathbf{y}).
\end{aligned}$$

Thus, σ has exactly the claimed properties on Δ^n, and the theorem follows. □

In fact, we can use the same technique as achieve the following extension to fully continuous versions of our properties. We define the following properties for σ on the continuous domain $[0,1]^n$:

- we say σ is *normalized* if $\sigma(\mathbf{0}) = 0$;
- we say σ is *monotone* if $\mathbf{x} \leq \mathbf{y}$ implies $\sigma(\mathbf{x}) \leq \sigma(\mathbf{y})$; and
- we say σ is *submodular* if for any $\mathbf{x} \leq \mathbf{y}$, any $v \in V$, and for any $\varepsilon > 0$ such that $y_v + \varepsilon \leq 1$, we have that $\sigma(\mathbf{x} + \varepsilon_v) - \sigma(\mathbf{x}) \geq \sigma(\mathbf{y} + \varepsilon_v) - \sigma(\mathbf{y})$,

where ε_v is the vector with a value of ε in the coordinate corresponding to v and a value of 0 in all other coordinates. As before, all comparisons and additions between vectors above are componentwise. The same techniques immediately give us the following theorem; in the interests of space, we defer the proof to the full version of the paper.

THEOREM 3. *Let $\mathcal{I} = (V, \mathcal{F}, w)$ be an instance of our problem. If both w and $\mathcal{F}$ are normalized, monotone, and submodular, then σ is normalized, monotone, and submodular on $[0,1]^n$.*

One concern with discretizing the space we optimize over as in Theorem 2 is what effect the discretization has on the objective values that can be achieved. As the following theorem shows, however, that we can only lose an δn factor from our objective when we discretize the space to multiples of δ.

THEOREM 4. *Let $\mathcal{I} = (V, \mathcal{F}, w)$ be an instance of our problem. Then for any discretization Δ^n of $[0,1]^n$ (as defined above), if σ is normalized, monotone, and submodular on Δ^n, we have that*

$$\max_{\substack{\mathbf{x} \in \Delta^n: \\ \|\mathbf{x}\|_1 \leq K}} \sigma(\mathbf{x}) \geq (1 - \delta n) \max_{\substack{\mathbf{x} \in [0,1]^n: \\ \|\mathbf{x}\|_1 \leq K}} \sigma(\mathbf{x}),$$

for any K.

PROOF. Let $\mathbf{x}^*$ be an optimal solution to our problem on $[0,1]^n$, i.e. we have

$$\operatorname*{argmax}_{\mathbf{x} \in [0,1]^n : \|\mathbf{x}\|_1 \leq K} \sigma(\mathbf{x}).$$

Let $\bar{\mathbf{x}}^*$ be the result of rounding $\mathbf{x}^*$ up componentwise to the nearest element of Δ^n. Formally, we define $\bar{\mathbf{x}}^*$ by $\bar{\mathbf{x}}_v^* = \min\{d \in \Delta : d \geq \mathbf{x}^*\}$. Note that by monotonicity, we must have that $\sigma(\bar{\mathbf{x}}^*) \geq \sigma(\mathbf{x}^*)$; we also have that $\|\bar{\mathbf{x}}^*\|_1 \leq \|\mathbf{x}^*\|_1 + \delta n$. Now, consider constructing $\bar{\mathbf{x}}^*$ greedily by adding δ to a single coordinate in each step. Formally, set $\mathbf{x}^0 = \mathbf{0}$, and for each $i = 1, 2, \dots, \|\bar{\mathbf{x}}^*\|_1 / \delta$ set

$$\mathbf{x}^i = \mathbf{x}^{i-1} + \delta_v \text{ for some } v \in \operatorname*{argmax}_{v:\, \mathbf{x}_v^{i-1} < 1} (\sigma(\mathbf{x}^{i-1} + \delta_v) = \sigma(\mathbf{x}^{i-1})),$$

where (as before) δ_v is a vector with δ in the component corresponding to v and 0 in all other components. Note that the submodularity of σ implies that $\sigma(\mathbf{x}^i) - \sigma(\mathbf{x}^{i-1})$ is decreasing in i. An immediate consequence of this is that, for any i, we have that

$$\sigma(\mathbf{x}^i) \geq \frac{i}{\|\bar{\mathbf{x}}^*\|_1} \sigma(\bar{\mathbf{x}}^*).$$

Invoking the above for $i = K/\delta$ we get that

$$\sigma(\mathbf{x}^{K/\delta}) \geq \frac{K/\delta}{\|\bar{\mathbf{x}}^*\|_1} \sigma(\bar{\mathbf{x}}^*) \geq \frac{K}{K + \delta n} \sigma(\bar{\mathbf{x}}^*) \geq (1 - \delta n) \sigma(\bar{\mathbf{x}}^*).$$

We observe that $\|\mathbf{x}^{K/\delta}\|_1 = K$, and $\mathbf{x}^{K/\delta} \in \Delta^n$, and so the desired theorem follows. □

4. DAGS

In this section, we focus on a special case of fractional influence model called the linear influence model, and argue that some aspects of the problem become simpler on DAGs. In the linear variant of the problem, our influence functions are computed as follows. We are given a digraph $G = (V, E)$ and a weight function $\mathbf{w}$ on edges. We use $\delta^-(v)$ and $\delta^+(v)$ to denote the sets of nodes with edges to and edges from v, respectively. Then, we denote the influence function f_v for v by

$$f_v(S) = \sum_{u \in S \cap \delta^-(v)} w_{uv}.$$

In this model, we assume that $\sum_{u\in\delta^-(v)} w_{uv} \leq 1$ always. Similar to the fractional influence model, our goal is to pick an influence vector $\mathbf{x} \in [0,1]^{|V|}$ indexed by V to maximize

$$\sigma(\mathbf{x}) = \mathop{\mathrm{E}}_{\Theta}[\ |S_n^\Theta|\ |\text{ we apply direct influences } \mathbf{x}\ ],$$

where $S_1^\Theta, \ldots, S_n^\Theta$ is the sequence of sets of nodes activated under thresholds Θ and direct influence $\mathbf{x}$. We sometimes abuse notation and use $\sigma(S)$ to denote σ applied to the characteristic vector of the set $S \in 2^V$. Given a DAG $G = (V, E)$ and a fractional influence vector $\mathbf{x} \in [0,1]^{|V|}$ indexed by V, we define the sets

$$I(\mathbf{x}) = \{v \in V : x_v > 0\}, \text{ and}$$
$$S(\mathbf{x}) = \{v \in V : x_v + \textstyle\sum_{u\in\delta^-(v)} w_{uv} > 1\},$$

as the sets of nodes *influenced* by $\mathbf{x}$ and *(over-)saturated* by $\mathbf{x}$. Note that $S(\mathbf{x}) \subseteq I(\mathbf{x})$. Following, we show that under specific circumstances, σ becomes a linear function and therefore the influence maximization problem tractable.

THEOREM 5. *Given a DAG G and influence vector $\mathbf{x}$, if G contains no path from an element of $I(\mathbf{x})$ to any element of $S(\mathbf{x})$, then we have that*

$$\sigma(\mathbf{x}) = \textstyle\sum_{v\in V} x_v \sigma(\mathbf{1}_v)$$

, and therefore the influence maximization problem can be solved efficiently for these instances.

PROOF. We prove this by induction on the number of vertices. In the case that V contains only a single vertex, the claim is trivial. Otherwise, let $G = (V, E)$ and $\mathbf{x}$ satisfy our assumptions, with $|V| = n > 1$, and assume out claim holds for any DAG with $(n-1)$ or fewer nodes. Let $s \in V$ be a source vertex (i.e. have in-degree 0) in G. Now, if $s \notin I(\mathbf{x})$, we know that s is never activated. Let $\hat{\sigma}$ and $\hat{\mathbf{x}}$ be σ on G restricted to $V \setminus s$ and $\mathbf{x}$ restricted to $V \setminus s$, respectively, and observe that we may apply our induction hypothesis to $\hat{\sigma}(\hat{\mathbf{x}})$ since removing s from G cannot cause any of the requirements for our theorem to become violated. Thus, we can see that

$$\sigma(\mathbf{x}) = \hat{\sigma}(\hat{\mathbf{x}}) = \sum_{v\in V\setminus s} x_v \hat{\sigma}(\mathbf{1}_v) = \sum_{v\in V} x_v \sigma(\mathbf{1}_v),$$

since $x_s = 0$.

Now, assume that $s \in I(\mathbf{x})$. Recall that by our conditions on G, therefore, we know that G contains no path from s to any elements of $S(\mathbf{x})$. One critical implication of this is that none of the nodes in $\delta^+(s)$ have paths to elements of $S(\mathbf{x})$ either, and so we made apply influence to them without violating the assumptions of our inductive hypothesis, as long as we are careful not to add so much weight that they become saturated.

In order to prove our claim, we focus on G restricted to $V \setminus \{s\}$, call it $\hat{G}$. Let $\hat{\sigma}$ be σ over $\hat{G}$, and consider the following two influence vectors for $\hat{G}$. Define $\hat{\mathbf{x}}$ to simply be the restriction of $\mathbf{x}$ to $\hat{G}$; define $\hat{\mathbf{y}}$ by $y_v = w_{sv}$ if $v \in \delta^+(s)$ and 0 otherwise. Letting $\hat{I}$ and $\hat{Sat}$ be I and S, respectively, restricted to $\hat{G}$, we can see that

$$\left.\begin{array}{l} \hat{I}(\hat{\mathbf{x}}), \hat{I}(\hat{\mathbf{y}}), \hat{I}(\hat{\mathbf{x}}+\hat{\mathbf{y}}) \subseteq I(\mathbf{x}) \cup \delta^+(s), \text{ and} \\ \hat{S}(\hat{\mathbf{x}}), \hat{S}(\hat{\mathbf{y}}), \hat{S}(\hat{\mathbf{x}}+\hat{\mathbf{y}}) \subseteq S(\mathbf{x}). \end{array}\right\} \quad (1)$$

The observation that gives the above is that, compared to $\mathbf{x}$, the only vertices with increased influence applied to them are the elements of $\delta^+(s)$, and the amounts of these increases are precisely balanced by the removal of s (and its outgoing edges) from $\hat{G}$. In particular, note that for any $v \in V \setminus \{s\}$, by our definition of $]\hat{y}s$ we have that

$$x_v + \sum_{u\in\delta^-(v)} w_{uv} = \hat{x_v} + \hat{y_v} + \sum_{\substack{u\in\delta^-(v)\\ u\neq s}} w_{uv}.$$

As previously noted G contains no paths from an element of $\delta^+(s)$ to any element of $S(\mathbf{x})$; this combined with (1) allows us to conclude that we may apply our induction hypothesis to $\hat{G}$ with any of $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, or $\hat{\mathbf{x}} + \hat{\mathbf{y}}$. We proceed by showing that for any vector Θ of thresholds for G (and its restriction to $\hat{G}$), we have that the set activated under $\mathbf{x}$ in G always corresponds closely to one of the sets activated by $\hat{\mathbf{x}}$ or $(\hat{\mathbf{x}} + \hat{\mathbf{y}})$ in $\hat{G}$. To that end, fix any vector Θ. We consider the cases where $x_s \geq \theta_s$ and $x_s < \theta_s$ separately.

We begin with the case where $x_s < \theta_s$, since it is the simpler of the two. Let $S_0^\Theta, \ldots, S_n^\Theta$ and $\hat{S}_0^\Theta, \ldots, \hat{S}_n^\Theta$ denote the sets activated in G under $\mathbf{x}$ and in $\hat{G}$ under $\hat{\mathbf{x}}$, respectively, in stages $0, \ldots, n$. Note that since s is a source, and $x_s < \theta_s$, we know that $s \notin S_i^\Theta$ for all i. However, this means that every node in $V \setminus \{s\}$ has both the same direct influence applied to it under $\mathbf{x}$ and $\hat{\mathbf{x}}$, and the same amount of influence applied by any activated set in both G and $\hat{G}$. So we can immediately see that since $S_0^\Theta = \emptyset = \hat{S}_0^\Theta$, by induction we will have that $S_i^\Theta = \hat{S}_i^\Theta$ for all i, and in particular for $i = n$.

The case where $x_s \geq \theta_s$ requires more care. Let $S_0^\Theta, \ldots, S_n^\Theta$ and $\hat{S}_0^\Theta, \ldots, \hat{S}_n^\Theta$ denote the sets activated in G under $\mathbf{x}$ and in $\hat{G}$ under $\hat{\mathbf{x}} + \hat{\mathbf{y}}$, respectively, in stages $0, \ldots, n$. Note that our assumption implies that s will be activated by our direct influence in the first round, and so we have $s \in \hat{S}_i^\Theta$ for all $i \geq 1$. Fix some $v \in V$, $v \neq s$, and let $f_v(S)$ and $\hat{f}_v(S)$ denote the total influence – both direct and cascading – applied to in G and $\hat{G}$, respectively, when the current active set is S. Then, we can see that for any $S \subseteq V \setminus \{s\}$ we have that

$$\hat{f}_v(S) = \hat{x_v} + \hat{y_v} + \sum_{\substack{u\in\delta^-(v)\\ u\in S}} w_{uv} = x_v + \sum_{\substack{u\in\delta^-(v)\\ u\in S\cup\{s\}}} w_{uv} = f_v(S \cup \{s\}). \quad (2)$$

Furthermore, note that both f_v and $\hat{f}_v$ are always monotone nondecreasing. While we cannot show that $S_i^\Theta = \hat{S}_i^\Theta$ for all i in this case, we will instead show that $S_i^\Theta \setminus \{s\} \subseteq \hat{S}_i^\Theta \subseteq S_{i+1}^\Theta \setminus \{s\}$ for all $i = 0, \ldots, n-1$. Recall that the propagation of influence converges by n steps. That is, if we continued the process for an additional step to produce activated sets S_{n+1}^Θ and $\hat{S}_{n+1}^\Theta$, we would have that $S_{n+1}^\Theta = S_n^\Theta$ and $\hat{S}_{n+1}^\Theta = \hat{S}_n^\Theta$. However, our claim would extend to this extra stage as well, and so we conclude that we must have that $S_n^\Theta = \hat{S}_n^\Theta \cup \{s\}$. We prove our claim inductively. First, observe that it holds trivially for $i = 0$, since we have $S_0^\Theta = \hat{S}_0^\Theta = \emptyset$, and previously observed that $s \in S_1^\Theta$. Now, the claim holds for some i. Note, however, that by (2) and monotonicity we must have that for all $v \in V$, $v \neq s$

$$\begin{aligned} f_v(S_i^\Theta) &= \hat{f}_v(S_i^\Theta \setminus \{s\}) \leq \hat{f}_v(\hat{S}_i^\Theta) \\ &\leq \hat{f}_v(S_{i+1}^\Theta \setminus \{s\}) = f_v(S_{i+1}^\Theta). \end{aligned}$$

But from the above, we can see that $S_{i+1} \setminus \{s\} \subseteq \hat{S}_{i+1}^\Theta \subseteq S_{i+2}^\Theta \setminus \{s\}$ since such a v in included in each of the above sets if and only if $f_v(S_i^\Theta)$, $\hat{f}_v(\hat{S}_i^\Theta)$, or $f_v(S_{i+1}^\Theta)$, respectively, exceeds θ_v.

Thus, by observing that θ_s is an independent draw from $\mathcal{U}[0,1]$, we can see that taking expectations over Θ and conditioning on which of θ_s and x_s is larger, gives us that

$$\sigma(\mathbf{x}) = (1 - x_s)\hat{\sigma}(\hat{\mathbf{x}}) + x_s(1 + \hat{\sigma}(\hat{\mathbf{x}} + \hat{\mathbf{y}}))$$
$$= \sum_{\substack{v \in V \\ v \neq s}} x_v \sigma(\mathbf{1}_v) + x_s(1 + \hat{\sigma}(\hat{\mathbf{y}})).$$

We complete our proof by observing that, in fact, $\sigma(\mathbf{1}_s)$ is precisely equal to $1 + \hat{\sigma}(\hat{\mathbf{y}})$. We can see this by once again coupling the activated sets under any vector Θ of thresholds. In particular, let $S_0^\Theta, \ldots, S_n^\Theta$ and $\hat{S}_0^\Theta, \ldots, \hat{S}_n^\Theta$ denote the sets activated in G under $\mathbf{1}_s$ and in $\hat{G}$ under $\hat{\mathbf{y}}$, respectively, in stages $0, \ldots, n$. Arguments identical to those made above allow us to conclude that for all i, we have that $S_{i+1}^\Theta = \hat{S}_i^\Theta \cup \{s\}$. Thus, by again noting that influence cascades converge after n steps we see that $S_n^\Theta = \hat{S}_n^\Theta \cup \{s\}$, and taking expectations with respect to Θ gives precisely the desired equality.

Since we have the linearity of σ, to maximize $\sigma(\mathbf{x})$, we just need to sort vertices based on $\sigma(\mathbf{1}_v)$, and put the influences on vertices with higher $\sigma(\mathbf{1}_v)$ until the budget is exhausted. Estimating $\sigma(\mathbf{1}_v)$ can be done by letting the process run with influence vector $\mathbf{1}_v$ for several times, and taking the average number of activated nodes in these trials. □

5. HARDNESS

In this section, we present NP-hardness and inapproximability results in the linear influence model. We assume that thresholds are not chosen from a distribution, and they are fixed and given as part of the input. We note that this is the main assumption that makes our problem intractable, and to achieve reasonable algorithms, one has to make some stochastic (distributional) assumptions on the thresholds. In Section 4, we introduced the linear influence model as a special case of fractional influence model, but it makes sense to define it as a special case of integral influence model as well. In the fractional linear influence model, we are allowed to apply any influence vector $\mathbf{x} \in [0, 1]^n$ on nodes. By restricting the influence vector $\mathbf{x}$ to be in $\{0, 1\}^n$ (a binary vector), we achieve the integral version of linear influence model. Our hardness results in Theorem 6, and Corollary 7 work for both fractional and integral versions of linear influence model. We start by proving that the linear influence model is NP-hard with a reduction from independent set problem in Theorem 6. We strengthen this hardness result in Corollary 7 by showing that an $n^{1-\varepsilon}$ approximation algorithm for the linear influence problem yields an exact algorithm for it as well for any constant $\varepsilon > 0$, and therefore even an $n^{1-\varepsilon}$ approximation algorithm is NP-hard to achieve. At the end, we show the that it is NP-hard to achieve any better than $1 - 1/e$ approximation in the Triggering model which is introduced in [15]. We will elaborate on the Triggering Model and this hardness result at the end of this section. We note that the proofs are omitted due to lack of space.

THEOREM 6. *If we allow arbitrary, fixed thresholds, it is NP-hard to compute for a given instance of the integral linear influence problem (G, k, T) (graph G, budget k, and a target goal T) whether or not there exists a set S of k vertices in G such that $\sigma(S) \geq T$. Furthermore, the same holds in the factional version of the problem (instead of a set S of size k, we should look for a influence vector with ℓ_1 norm equal to k in the fractional case).*

COROLLARY 7. *If we allow arbitrary, fixed thresholds, it is NP-hard to approximate the linear influence problem to within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$. Furthermore, the same holds for the fractional version of our problem.*

Network	# nodes	# edges	Avg. deg.	Directed
NetHEPT	15,233	58,891	7.73	No
NetPHY	37,154	231,584	12.46	No
Facebook	4,039	88,234	21.84	No
Amazon	262,111	1,234,877	4.71	Yes

Table 1: Information about the real-world networks we use.

Before stating Theorem 8, we should define the triggering model introduced in [15]. In this model, each node v independently chooses a random triggering set T_v according to some distribution over subsets of its neighbors. To start the process, we target a set A for initial activation. After this initial iteration, an inactive node v becomes active in step t if it has a neighor in its chosen triggering set T_v that is active at time $t - 1$. For our purposes, the distributions of triggering sets have support size one (deterministic triggering sets). We also show that our hardness result even holds when the size of these sets is two.

THEOREM 8. *It is NP-hard to approximate linear influence problem to within any factor better than $1 - 1/e$, even in the Triggering model where triggering sets have size at most 2.*

6. EXPERIMENTAL RESULTS

Datasets. We use the following real-world networks for evaluating our claims. Table 1 gives some statistical information about these networks.

- NetHEPT: An academic collaboration network based on "High Energy Physics — Theory" section of the e-print arXiv[2] with papers from 1991 to 2003. In this network, nodes represent authors and edges represent co-authorship relationships. This network is available at `http://research.microsoft.com/en-us/people/weic/graphdata.zip`.
- NetPHY: Another academic collaboration network, taken from the full "Physics" section of the e-print arXiv. Again, nodes represent authors and edges represent co-authorship relationships. The network is available at `http://research.microsoft.com/en-us/people/weic/graphdata.zip`.
- Facebook: A surveyed portion of the Facebook friend network. The nodes are anonymized Facebook users and edges represents friendship relationships. The data is available at `http://snap.stanford.edu/data/egonets-Facebook.html`.
- Amazon: Produced by crawling the Amazon website based on the following observation: customers who bought product i also bought product j. In this network, nodes represent products and there is a directed edge from node i to node j if product i is frequently co-purchased with product j. This network is based on Amazon data in March 2003. The data is available at `http://snap.stanford.edu/data/amazon0302.html`.

Algorithms. We compare the following algorithms in this study. The first three algorithms are for the integral influence model, and the last three algorithms work for the fractional influence model.

[2] `http://www.arXiv.org`

- DegreeInt: A simple greedy algorithm which selects nodes with the largest degrees. This method was used by Kempe et al. [15] and Chen et al. [5] as well.
- DiscountInt: A variant of DegreeInt which selects node u with the highest degree in each step. Moreover, after adding node u to the seed set, the algorithm decreases the degrees of neighbors of u by 1. This method was proposed and evaluated by Chen et al. [5].
- RandomInt: This algorithm randomly adds B nodes to the seed set, i.e., by spending 1 on each of them. We use this algorithm as a baseline in our comparisons. Other works [5, 6, 15] also use this algorithm as a baseline.
- DegreeFrac: This algorithm selects each node fractionally proportional to its degree. In particular, this algorithm spends $\min\{\frac{Bd_i^-}{m}, 1\}$ on node i where B is the budget, d_i^- is the out-degree of node i, and m is the total number of edges[3].
- DiscountFrac: A heuristic for the fractional case given by Algorithm 1. Let $\Gamma_v^-(A)$ be the total sum of the weight of edges from node v to set A, and $\Gamma_v^+(A)$ be the total sum of the weight of edges from set A to node v. This algorithm starts with an empty seed set S, and in each step it adds node $v \notin S$ with the maximum $\Gamma_v^-(V - S)$ to seed set S by spending $\max\{0, 1 - \Gamma_v^+(S)\}$ on node v. Note that in each step the total influence from the current seed set S to node v is $\Gamma_v^+(S)$, and it is enough to spend $1 - \Gamma_v^+(S)$ for adding node v to the current seed set S. Note that no node would pay a positive amount, and the algorithm spends $\max\{0, 1 - \Gamma_v^+(S)\}$ on node v.
- UniformFrac: This algorithm distributes the budget equally among all nodes. We use this algorithm as another baseline in our comparisons.

Algorithm 1 DiscountFrac

Input: Graph $G = (V, E)$ and budget B
Output: Influencing vector $\mathbf{x}$

1: $S \leftarrow 0$
2: $b \leftarrow B$
3: $\mathbf{x} \leftarrow \mathbf{0}$
4: **while** $b > 0$ **do**
5: $\quad u \leftarrow \text{argmax}_{v \in V-S}\{\Gamma_v^-(V - S)\}$
6: $\quad x_u \leftarrow \min\{b, \max\{0, 1 - \Gamma_u^+(S)\}\}$
7: $\quad b \leftarrow b - x_u$
8: $\quad S \leftarrow S \cup \{u\}$
9: **end while**
10: **return** $\mathbf{x}$

All these heuristic algorithms are fast and are designed for running on large real-world networks. In particular, algorithms DegreeInt and DegreeFrac only need the degree of nodes. We can use a Fibonacci heap to implement DiscountInt, resulting in a running time of $O(B \log n + m)$. Similarly, the running time of DiscountFrac is $O(n \log n + m)$ using a Fibonacci heap.[4] Algorithms RandomInt and UniformFrac are linear-time algorithms.

[3] If the graph is undirected, the cost is $2m$ instead of m

[4] In DiscountFrac, the while loop (lines 4–9 of Algorithm 1) may run for n steps even when budget B is less than n. Hence, the running time is $O(\max\{n, B\} \log n + m) = O(n \log n + m)$ instead of $O(B \log n + m)$.

It also has been shown that the performance of DiscountInt almost matches the performance of the greedy algorithm which maximizes a submodular function [5]. Hence, it seems that DiscountInt is an appropriate candidate for evaluating the power of the integral influence model.

Results. We have implemented all algorithms in C++, and have run all experiences on a server with two 6-core/12-thread 3.46GHz Intel Xeon X5690 CPUs, with 48GB 1.3GHz RAM. We run all of the aforementioned algorithms for finding the activation vector/set, and compute the performance of each algorithm by running 10,000 simulations and taking the average of the number of adopters.

We first examine the performance of a fractional activation vector in the *weighted cascade model*, where the weight of the edge from u to v is $\frac{1}{d_v^-}$, where d_v^- is the in-degree of node v. In this model, the total sum of weight of incoming edges of each node is $\sum_{uv} w_{uv} = \sum_{uv} \frac{1}{d_v^-} = 1$. This model was proposed by Kempe et al. [15], and it has been used in the literature [5, 6, 7]. Results are shown in Figure 1.

We then compare the performance of various algorithms when the weight of edges are determined by the *TRIVALENCY* model, in which the weight of each edge is chosen uniformly at random from the set $\{0.001, 0.01, 0.1\}$. Here 0.001, 0.01, and 0.1 represent low, medium, and high influences. In this model, the total sum of the weights of incoming edges of each node may be greater than 1. This model and its variants have been used in [5, 6, 15]. We run all proposed algorithms on real-world networks when their weights are defined by TRIVALENCY model. Results are shown in Figure 2.

Discussion. In most of the plots, algorithms for the fractional influence model do substantially better than algorithms for the integral influence model. Overall, for most datasets, DiscountFrac is the best algorithm, with the only exception being the Facebook dataset. As a simple metric of the power of the fractional model versus the integral model, we consider the pointwise performance gain of fractional model algorithms versus the integral model algorithms. i.e., for a given budget, we compute the ratio of expected number of adopters for the fractional model with the most adopters and the expected number of adopters for the integral model algorithm with the most adopters. Depending on the dataset, we get a mean pointwise performace gain between 3.4% (Facebook dataset, TRIVALENCY model) and 142.7% (Amazon dataset, weighted cascade model) with the mean being 31.5% and the median being 15.7% over all the datasets and both models (weighted cascade and TRIVALENCY). Among the heuristics presented for the integral model, DiscountInt is probably the best. If we compare just it to its fractional adaptation, DiscountFrac, we get a similar picture: the range of average performace gain is between 9.1% (Facebook, TRIVALENT model) and 397.6% (Amazon, weighted cascade model) with the mean being 64.1% and the median being 15.6%.

In summary, the experimental results clearly demonstrate that the fractional model leads to a significantly higher number of adopters across a wide range of budgets on diverse datasets.

References

[1] Hessameddin Akhlaghpour, Mohammad Ghodsi, Nima Haghpanah, Vahab S Mirrokni, Hamid Mahini, and Afshin Nikzad. Optimal iterative pricing over social networks. In *WINE*, pages 415–423. 2010.

[2] David Arthur, Rajeev Motwani, Aneesh Sharma, and Ying Xu. Pricing strategies for viral marketing on social networks. In *WINE*, pages 101–112. 2009.

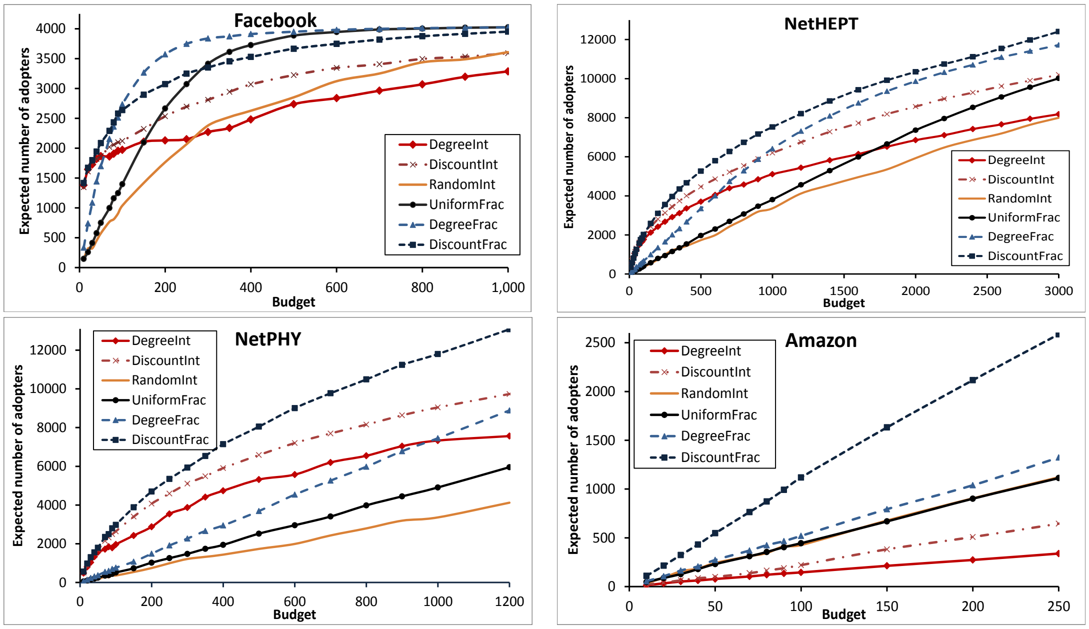

Figure 1: Performance of different algorithms on Facebook, NetHEPT, NetPHY, and Amazon. The weights of edges are defined based on the weighted cascade model. The x axis is the budget and the y axis is the expected number of adopters.

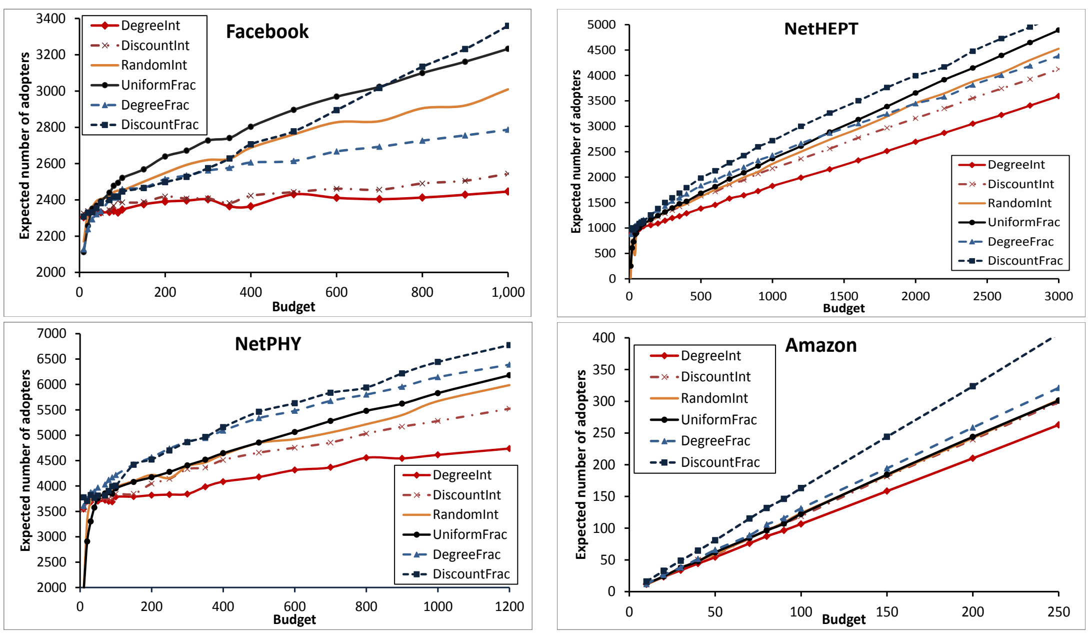

Figure 2: Performance of different algorithms on Facebook, NetHEPT, NetPHY, and Amazon. The weights of edges are defined based on the TRIVALENCY model. The x axis is the budget and the y axis is the expected number of adopters.

[3] Sushil Bikhchandani, David Hirshleifer, and Ivo Welch. A theory of fads, fashion, custom, and cultural change as informational cascades. *Journal of Political Economy*, 100:992–1026, 1992.

[4] Christian Borgs, Michael Brautbar, Jennifer T. Chayes, and Brendan Lucier. Maximizing social influence in near optimal time. to appear in SODA 2014.

[5] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *KDD*, pages 199–208. ACM, 2009.

[6] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD*, pages 1029–1038. ACM, 2010.

[7] Wei Chen, Yifei Yuan, and Li Zhang. Scalable influence maximization in social networks under the linear threshold model. In *ICDM*, pages 88–97. IEEE, 2010.

[8] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *KDD*, pages 57–66. ACM, 2001.

[9] Shayan Ehsani, Mohammad Ghodsi, Ahmad Khajenezhad, Hamid Mahini, and Afshin Nikzad. Optimal online pricing with network externalities. *Information Processing Letters*, 112(4):118 – 123, 2012. ISSN 0020-0190.

[10] Mark Granovetter. Threshold models of collective behavior. *American journal of sociology*, 83:1420–1443, 1978.

[11] Dilek Gunnec. *Integrating social network effects in product design and diffusion*. Ph.d. thesis, University of Maryland, College Park, 2012.

[12] Dilek Gunnec and S. Raghavan. Integrating social network effects in the share-of-choice problem. Technical report, University of Maryland, College Park, 2012.

[13] Dilek Gunnec, S. Raghavan, and Rui Zhang. The least cost influence problem. Technical report, University of Maryland, College Park, 2013.

[14] Jason Hartline, Vahab Mirrokni, and Mukund Sundararajan. Optimal marketing strategies over social networks. In *WWW*, pages 189–198. ACM, 2008.

[15] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146. ACM, 2003.

[16] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *KDD*, pages 420–429. ACM, 2007.

[17] Elchanan Mossel and Sebastien Roch. On the submodularity of influence in social networks. In *STOC*, pages 128–134. ACM, 2007.

[18] Elchanan Mossel and Sébastien Roch. Submodularity of influence in social networks: From local to global. *SIAM J. Comput.*, 39(6):2176–2188, 2010.

[19] Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD*, pages 61–70. ACM, 2002.

[20] Everett M Rogers. *Diffusion of innovations*. Simon and Schuster, 2010.

[21] Thomas C Schelling. *Micromotives and macrobehavior*. WW Norton & Company, 2006.

[22] Thomas W Valente. *Network models of the diffusion of innovations (Quantitative methods in communication series)*. Hampton Press, 1995.