Influence Maximization through Adaptive Seeding

YARON SINGER
Harvard University

In this monograph we survey the adaptive seeding methodology for influence maximization. Influence maximization is the challenge of spreading information effectively through influential users in a social network. In many applications, one is restricted to select influencers from a set of users who engaged with the topic being promoted, and due to the structure of social networks, these users often rank low in terms of their influence potential. An alternative approach one can consider is an adaptive method which selects users in a manner which targets their influential neighbors. The advantage of such an approach is that it leverages the friendship paradox in social networks: while users are often not influential themselves, they know someone who is.

Adaptive seeding is a two-stage stochastic optimization framework, designed to leverage the friendship paradox to seed high influential nodes. This framework encompasses a rich set of algorithmic challenges at the intersection of stochastic optimization and submodular optimization. We discuss this method here, along with algorithmic approaches, the friendship paradox in random graph models, and experiments on adaptive seeding.

Categories and Subject Descriptors: F.0 [**Theory of Computation**]: General

General Terms: Algorithms; Economics; Experimentation; Human factors;

Additional Key Words and Phrases: Influence maximization, submodularity, stochastic optimization, friendship paradox

1. INTRODUCTION

Influence maximization is the algorithmic challenge of selecting a fixed number of individuals who can serve as early adopters of a new idea, product, or technology in a manner that will trigger a large cascade in a social network. Since it was first posed by Domingos and Richardson [Domingos and Richardson 2001; Richardson and Domingos 2002] and elegantly formulated and further developed in seminal work by Kempe, Kleinberg, and Tardos [Kempe et al. 2003], numerous techniques and improvements have been developed ranging from sophisticated predictive models of influence [Leskovec et al. 2006; Gomez-Rodriguez et al. 2010; Bakshy et al. 2011; Abraham et al. 2013; Gomez-Rodriguez et al. 2013; Du et al. 2013] to fast approximation methods [Leskovec et al. 2007; Mossel and Roch 2007; Chen 2008; Mathioudakis et al. 2011; Borgs et al. 2012].

In many cases where influence maximization methods are applied, one cannot select any user in the network but is rather limited to some core sample of users. In marketing applications for example, merchants often apply influence maximization techniques on users who visit their online store, or have engaged in other ways (e.g. subscribe to a mailing list, follow the brand, or installed an application). Similarly, in many other cases, whether due to scale, privacy, or profile-based targeting, influence maximization algorithms are applied on relatively small core samples of the network. The problem is that when we select influential users from such core sam-

Supported by NSF grant CCF-1301976, CAREER CCF-1452961, a Google Faculty Research Award, and Facebook Faculty Award. Author's address: yaron@seas.harvard.edu

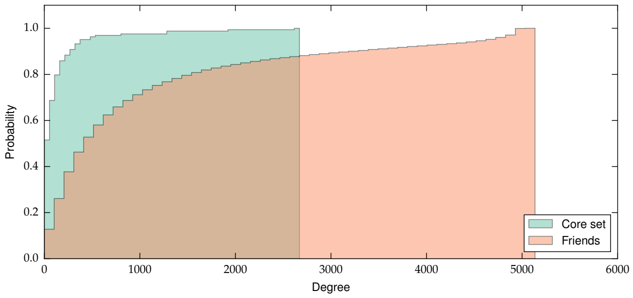


Fig. 1: CDF of the degree distribution of users who endored a message by Kiva on Facebook, and CDF of the degree distribution of their friends.

ples we may end up with poor outcomes, simply due to the structural properties of social networks. Due to the heavy-tailed degree distribution of social networks, high degree nodes are rare, and if we think of the core set as a random sample from a heavy-tailed degree distribution, it will likely not include high degree nodes. Since influence maximization techniques often depend on the ability to select nodes with high (not necessarily the highest) degree, a naive application of influence maximization on the core set may be ineffective.

Is it possible to design effective influence maximization strategies when influencers are rare?

An alternative to spending the entire budget on the core set is an adaptive two-stage approach. In the first stage, one can spend a fraction of the budget on the core users so that they invite their friends to participate in the campaign. Then, in the second stage, one can spend the rest of the budget on the influential friends who hopefully have arrived. The idea behind this approach is to leverage a structural phenomenon in social networks known as the *friendship paradox* [Feld 1991]. Intuitively, the friendship paradox says that individuals are likely to have far less friends than their friends do, on average. In Figure 1 we give an example of this phenomenon by plotting a CDF of the degree distribution of a core set of users who responded to a post on Facebook by the Kiva organization and the degree distribution of their friends.

1.1 An Illustrative Example

Suppose we are given a network, an arbitrary set of core users X and budget k, and the goal is to select a subset of nodes in X of any size $t \leq k$ which has the most influential set of neighbors of size $k - t$. For simplicity, assume for now that the influence of a set is simply the sum of the degrees of its nodes. If we take the $k/2$ highest-degree neighbors of X, then surely there is a set S of size at most $k/2$ in X connected to this set, and selecting S would be at least a 1/2-approximation to this problem. In comparison, the standard approach of influence maximization is to select the k highest degree nodes in X. Thus, standard influence maximization would have k of the most influential nodes in X while the approximation algorithm we propose has $k/2$ of the most influential nodes from its set of neighbors. Which is better? If we consider an input like the one portrayed in Figure 1, at least intuitively, the first approach will substantially dominate the naive influence maximization approach.

1.2 Adaptive Seeding

In this survey, we will introduce the adaptive seeding model recently formalized in [Seeman and Singer 2013], and highlight several key structural and algorithmic results. At a high level, adaptive seeding seeks to select users who have influential friends that likely can be selected to initiate a cascade. As we will later see, this approach presents deep algorithmic questions and inspires novel problems in stochastic optimization that transcend their application to influence maximization. In addition, this problem poses interesting questions regrading the friendship paradox phenomenon and structural properties of social networks.

1.2.1 *The model.* The adaptive seeding model is more general than the caricature model discussed in the illustrative example in two important aspects. First, influence is not necessarily measured as the sum of degrees but rather according to standard models studied in previous literature. Second, adaptive seeding assumes that every neighbor of X can be selected with some independent probability. These probabilities model the uncertainty we have about the neighbors' interest in promoting the product, as they are not in the core set. More formally, in the adaptive seeding model we are given a set of core nodes X, their set of neighbors $\mathcal{N}(X)$, each associated with a probability p_i, as well as a budget $k \in \mathbb{N}$ and a function $f : 2^{\mathcal{N}(X)} \to \mathbb{R}$. In the first stage, a set $S \subseteq X$ can be selected, which causes each one of its neighbors to realize independently with probability p_i. In the second stage, the remainder of the budget $k - |S|$ can be used to optimize an influence function f over the realized neighbors. The goal is to select a subset $S \subseteq X$ of size at most k s.t. the function can be optimized in expectation over all possible realizations of its neighbors with the remaining budget $k - |S|$. Equivalently, our goal is to select $S \subseteq X$ s.t. in expectation over all possible realizations $R_1, \ldots, R_m$ of $\mathcal{N}(X)$ the value of a set of its neighbors T_i of size $k - |S|$ that appears in the realization R_i is optimal. Thus, the objective is[1]:

$$\begin{aligned} \max_{S \subseteq X} \sum_{i=1}^{m} & f(T_i) \cdot p(R_i) \\ T_i \subseteq R_i \cap \mathcal{N}(S) & \qquad \forall i \in [m] \\ |S| + |T_i| \leq k & \qquad \forall i \in [m] \end{aligned}$$

1.3 Organization of this Survey

We will begin by motivating adaptive seeding by discussing the friendship paradox in Section 2. This section largely discusses Feld's work and the friendship paradox in random graph models based on recent work in [Lattanzi and Singer 2015]. In Section 3 we discuss the concept of influence as formulated by Kempe, Kleinberg,

[1] In this formulation we are not assuming that we receive any value from selecting the nodes $S \subseteq X$ in the first stage. We do so to remain faithful to the original formulation of influence of a set defined by Kempe, Kleinberg, and Tardos. If we wish wish to include the value of S we can slightly modify the problem so that the algorithmic problem is equivalent to the one formulated here.

and Tardos [Kempe et al. 2003] to provide context to the adaptive seeding model. We discuss diffusion models, influence functions, and submodularity, all of which are crucial for understanding the adaptive seeding problem. In section 4 we describe the various algorithmic approaches for solving adaptive seeding. We will discuss concepts of non-adaptivity, local adaptivity, as well as interesting new connections between convex optimization and submodular optimization. This section is based on work in [Seeman and Singer 2013; Rubinstein et al. 2015; Badanidiyuru et al. 2016]. Finally, in Section 5 we describe experiments of adaptive seeding using data sets collected purposefully for this task. We conclude with a brief discussion and open questions in Section 6.

2. THE FRIENDSHIP PARADOX

The friendship paradox describes a phenomenon first discovered in 1991 by Scott Feld [Feld 1991]. In his paper, Feld empirically observes that in various social networks an individual is likely to have a smaller degree than the average degree of her neighbors (hence "your friends have more friends than you"). Interestingly, this phenomenon was recently verified empirically in large online social networks as well [Ugander et al. 2011; Hodas et al. 2013]. For our purposes, the friendship paradox justifies the adaptive seeding approach: if an individual is indeed likely to have high degree neighbors, seeding friends of random nodes may lead to dramatic improvements in influence maximization.

2.0.1 *Proving the existence of the friendship paradox.* Ideally, we would like to *prove* that the friendship paradox exists in models that faithfully describe social networks. That is, we would like to show that with some constant probability, there is a significant gap between the degree of a random node in the graph and the average degree of her friends. In his paper, Feld provides analysis which conveys intuition about the phenomenon but does not prove the friendship paradox. Fortunately however, one can show that a slightly weaker version of the friendship paradox exists in graph models that are used to describe social networks. We will first discuss Feld's analysis followed by a discussion of a proof of the weaker version of friendship paradox based on recent work [Lattanzi and Singer 2015].

2.1 Feld's Analysis

To give intuition behind the idea that "your friends have more friends than you", Feld shows that the average degree of an individual (a node in the graph) is bounded from above by the average degree of a random neighbor (an endpoint of an edge), in expectation over all neighbors in the graph. To see this, let d_i denote the degree of node i in a graph of n nodes, and μ be the average degree:

$$\mu = \frac{\sum_{i=1}^{n} d_i}{n}$$

Now, let's calculate the average degree of neighbors in the graph. Here, we want to count the total degree and divide it by the number of neighbors. For every node:

$$\frac{\sum_{i=1}^{n} \text{total degrees of neighbors of } i}{\text{number of neighbors in the graph}}$$

For every node i, its degree d_i is added to every one of its d_i neighbors. Hence the total degree of neighbors is $\sum_i d_i^2$ and averaging this over the total number of neighbors we get that the average degree over all neighbors is $\sum_i d_i^2 / \sum_i d_i$. To see how this compares to the average degree of nodes we can manipulate this term:

$$\begin{aligned}
\frac{\sum_i d_i^2}{\sum_i d_i} =& \frac{\sum_i d_i^2}{\sum_i d_i} - \frac{\sum_i d_i}{n} + \frac{\sum_i d_i}{n} \\
=& \frac{\frac{\sum_i d_i^2}{n}}{\frac{\sum_i d_i}{n}} - \frac{\left(\frac{\sum_i d_i}{n}\right)^2}{\frac{\sum_i d_i}{n}} + \frac{\sum_i d_i}{n} \\
=& \frac{\frac{\sum_i d_i^2}{n} - \left(\frac{\sum_i d_i}{n}\right)^2}{\frac{\sum_i d_i}{n}} + \frac{\sum_i d_i}{n} \\
=& \frac{\sigma^2}{\mu} + \mu
\end{aligned}$$

where σ^2 denotes the variance of the degree distribution in the graph. Thus, since $\mu \leq \mu + \sigma^2/\mu$ we get that in expectation over all nodes in the graph, the degree of a random node is upper bounded by the average degree of a random neighbor, in expectation over all neighbors in the graph.

2.1.1 *The gap between Feld's analysis and the friendship paradox.* The friendship paradox is the idea that the degree of a node is likely to be bounded from above by the average degree of its neighbors. Feld's analysis, however, bounds the expected degree of a node against the expected degree of a *random* neighbor in the graph. That is, not only is it that Feld's analysis only holds in expectation, but more crucially it compares the average degree of a node x against the average degree of a neighbor y in the graph, where y is not necessarily a neighbor of x. To illustrate the difference, consider a *star graph* with n nodes. Note that there are $n-1$ nodes whose neighbor is the center of the star and thus the average degree of their neighbor is $n-1$, and a single node which is the center of the star and the average degree of its neighbors is 1. Therefore the average degree of a neighbor of a random node is:

$$\frac{1}{n}\left((n-1)\cdot(n-1)+1\right) = n - 2 + \frac{2}{n}$$

Feld's analysis considers the average degree of a random neighbor in the graph. To compute this quantity we can select an edge at random and average the degree of its endpoints. In the star there are n edges where one endpoint has degree 1 and the other has degree $n-1$. Thus, the average degree of a random neighbor is:

$$\frac{1}{n}\left(\frac{1+(n-1)}{2}\right) = \frac{n}{2}$$

Thus, in the star graph, the friendship paradox (ratio between degree of a random node and the average degree of its neighbor) and Feld's analysis (ratio between degree of a random node and the degree of a random neighbor) would differ by a factor of almost 2.

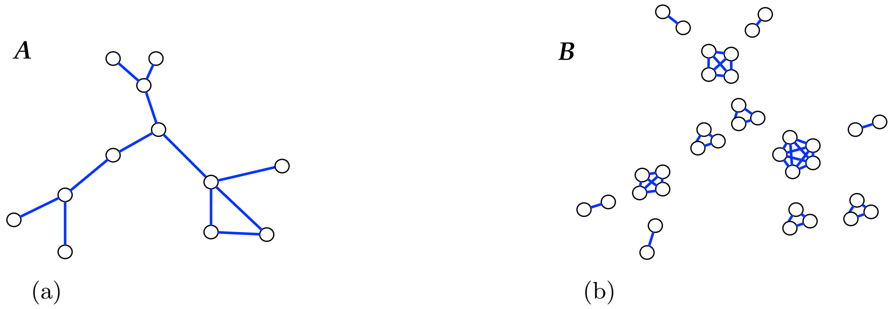


Fig. 2: An illustration of two networks whose degree distributions have high variance. The network on the right (B) is a set of disjoint cliques created using the network on the left (A): every node in A of degree d is represented as a clique of $d + 1$ in B. While B has a degree distribution with high variance, every one of its nodes has exactly the same degree as its neighbor.

A more dramatic gap is achieved on the network illustrated in Figure 2. Essentially, one can take any graph whose degree distribution has high variance (illustrated as network A) and generate a graph where each node of degree d in the original network is represented as an isolated clique of size $d + 1$ (illustrated as network B). The degree distribution of the resulting graph has high variance as well, and Feld's analysis indicates that there is a large gap between the average degree of a neighbor and the average degree of a node. But every node has *exactly* the same degree as its neighbor, and the friendship paradox does not hold.

2.2 The Friendship Paradox in Power Law Graphs

We are interested in understanding whether there are reasonable models of social networks where with *constant probability* a random node has a neighbor which has an asymptotically higher degree (i.e. the ratio between the degree of a node and its neighbor asymptotically grows with the size of the graph). We will refer to this phenomenon as the *existence of the friendship paradox.*

Does the friendship paradox exist in models that describe social networks?

In a regular network, where all nodes have the same degree, there is no difference between the degree of any node and that of its neighbors, so the friendship paradox does not holds. In a star network on the other hand, where a randomly selected node is likely to be an endpoint with its only neighbor being the center, with high probability the ratio between a node and its neighbor is linear in the size of the network, and the friendship paradox holds in the strongest sense possible. So what lies between these extremes that can serve as a good model for social networks?

2.2.1 *Power law graphs and long-range edges.* Perhaps the most common assumption about the structure of social networks and complex systems in general, is that their degree distribution is heavy-tailed and in particular follows a *power law*: the likelihood of observing a node with degree i in the network is proportional to $i^{-\beta}$, where β is a constant that depends on the network.[2] Empirical studies of social

[2] To handle finite graphs we need to define the power-law property for distributions with finite support. To do so, we can make the standard assumption which states that there is only one node

network topology typically report power law degree distributions with $\beta \in (1, 3)$. It is tempting to assume that the friendship paradox holds in any power-law network, but we already saw an example that contradicts this intuition. The network B in Figure 2 actually has a power-law degree distribution but since every node has the exact same degree as its neighbor the friendship paradox trivially does not hold. Thus, a power-law degree distribution is not a sufficient condition for the friendship paradox to hold. Fortunately, social networks are not only characterized by their degree distributions. An important property of social networks is that of long-range edges. Long-range links were first observed by Granovetter [Granovetter 1983] and are often modeled as random edges [Kleinberg 2000; Watts and Strogatz 1998].

To describe a general class of power law graphs with random edges we will make the following definition.

Definition. *Given a graph G whose degree distribution is power law and some $p \in [0, 1]$, its* ***perturbed power law graph*** *$G(p)$ is the graph G where every one of the edges in G is rewired with probability p.* [3]

A perturbed power law graph can be obtained from *any model of a social network* whose degree distribution follows a power law, by allowing every edge to have some fixed probability to be randomly rewired. For such graphs, the friendship paradox exists when considering a small sample.[4]

Theorem ([Lattanzi and Singer 2015])**.** *For any perturbed power-law graph, with constant probability there is an asymptotic gap between the average degree of a random set of polylogarithmic size and the average degree of its set of neighbors.*

2.2.2 *Models of social networks.* The family of perturbed power-law graphs covers a broad class of graphs that are used to describe social networks. The *configuration model* [Bollobás 1980] with power law degree distributions for finite graphs (e.g. [Aiello et al. 2000]) is a special case of this model. This result also holds for the preferential attachment model [Barabasi and Albert 1999], which is widely used in social network analysis (see e.g. [Aiello et al. 2000; Santos and Lenaerts 2006; Newman 2003]), as well as a similar model for generating connected graphs [Viger and Latapy 2005] and other heavy-tailed distribution models. [5]

of maximal degree in the graph [Aiello et al. 2000].

[3] Rewiring is done by first selecting all the edges that have to be rewired and then by applying the technique used in the configuration model [Bollobás 1980]: every edge selected to be rewired is split into two stubs attached to the nodes corresponding with the endpoints of that edge, and the set of all stubs are then connected uniformly at random. In case of self-loops, we count a node as a neighbor of itself (in most of the setting the number of self loops is extremely small).

[4] Ideally, we would have liked to prove that the friendship paradox exists even for a single node and not just samples of polylogarithmic size as stated in the above theorem. Unfortunately, while this is true for power law graphs with parameter $\beta \in (1, 2]$, this is no longer the case when $\beta \in (2, 3)$ and thus only the weaker version holds. Nevertheless, for the purpose of adaptive seeding, this implies that even when the power-law degree distribution parameter is in $(2, 3)$ range, as long as there are polylogarithmically-many nodes in the core set, there is a significant advantage in seeding their neighbors.

[5] There are other distributions that can be used to model degree distributions of social networks. Different papers support different heavy tail distributions: power law, log-normal, double Pareto, etc. For a survey of different heavy-tailed models and their comparison see [Mitzenmacher 2004].

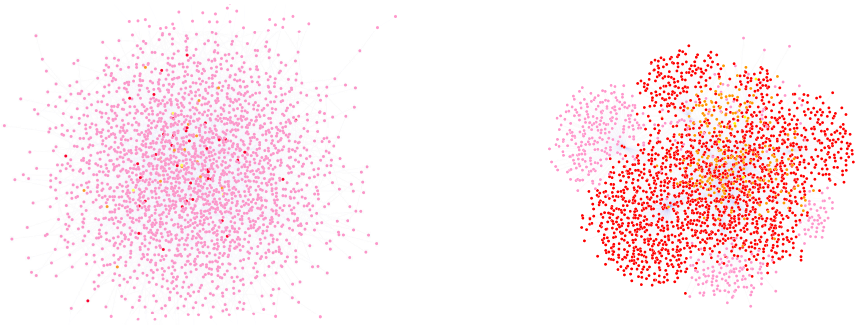


Fig. 3: An illustration of the friendship paradox in different networks. The two networks above are of the same size in terms of nodes and edges but with different topologies. The graph on the left has edges connected between nodes uniformly at random, and the graph on the right is a random graph with degree distribution close to a power law, i.e. the probability of observing a node of degree i is proportional to $i^{-\beta}$ for some constant β. In the first network the friendship paradox does not hold while in the second one it does. In each network we performed the following experiment: we selected five random nodes to represent the sample S, which are depicted in yellow, their neighbors $\mathcal{N}(S)$ are depicted in orange and their neighbors which is the potential influence of $\mathcal{N}(S)$ are depicted in red. The rest of the nodes are represented in pink.

2.2.3 *Your friends are also more influential than you.* Notice that we use the sum of degrees of a set as a proxy of its *influence*. The results we describe hold for both sum of degrees, i.e. $f(S) = \sum_{i \in S} d_i$ and coverage $f(S) = | \cup_{i \in S} T_i|$, where d_i, T_i are the degree and set of neighbors of a node i, respectively. The sum of degrees is the basis for the well-studied *voter model* (see e.g. [Holley and Liggett 1975; Domingos and Richardson 2001; Richardson and Domingos 2002; Even-Dar and Shapira 2011; Singer 2012]), and coverage is the basis of submodular influence models such as *independent cascade* and *linear threshold* [Kempe et al. 2003]. Although coverage does not account for diffusion, empirical evidence indicate that diffusion paths are often only one hop long [Yang and Counts 2010], which suggests that coverage may serve as a good proxy for richer influence models.

2.3 Algorithmic Implications

Showing that the friendship paradox exists in social networks implies that adaptive seeding algorithms can indeed enable dramatic improvements for information dissemination: if with constant probability a random sample is connected to a set of comparable size which can reach asymptotically many more nodes, this means that for exponentially many *arbitrary* (i.e. not necessarily random) samples, the expected number of nodes influenced through adaptive seeding algorithms could be asymptotically larger than the number of nodes influenced through applying influence maximization on the sample alone. More importantly, this would suggest that the well-studied problem of influence maximization can benefit from taking an alternative approach of two-stage optimization. We illustrate how the friendship paradox translates to potential influence of two-stage approaches in Figure 3.

3. THE ADAPTIVE SEEDING MODEL

We will devote this section to discussing the adaptive seeding model. We will first introduce the concept of *influence functions* as formulated in [Kempe et al. 2003]. An influence function quantifies the expected number of nodes that are influenced when a set of individuals initiates a cascade. Such a function is a consequence of a diffusion process that can be captured through elegant mathematical models, and respects an important structural property called *submodularity.*

3.1 Diffusion, Influence, and Submodularity

When we consider influence in social networks we wish to capture the idea that individuals are influenced by the actions of their peers. To model this, we can imagine a process measured in discrete time steps $T = \{0, 1, 2, \ldots\}$ where in every time step $t \in T$ an individual has an opinion that is either 0 or 1. A node is *activated* at time step t if it had opinion 0 at time step $t-1$ and its opinion changed to 1 at time step t. In this survey we will discuss four important models of diffusion defined on some finite graph $G = (V, E)$:

The voter model. This model is defined on an undirected graph with self loops [6], and at time step t, each node u adopts the opinion its neighbor v had at time $t-1$ with probability $1/d(u)$, where $d(u)$ denotes the degree of node u. This is one of the earliest models of diffusion studied [Holley and Liggett 1975], and leads to influence functions that are easy to optimize, as we will later discuss.

Independent cascade. This model was formulated in [Kempe et al. 2003], and defined on a directed edge-weighted graph. A node u that became activated at time step t attempts to activate its neighbor v at time step $t+1$ and succeeds (independently) with probability $p_{u,v}$, where $p_{u,v}$ is the weight encoded on the edge (u, v). Note that in this model, once a node is activated it remains activated, and every node attempts to activate its neighbor only once.

Linear threshold. This model is defined on a directed graph with weights on its nodes and edges. In this model each node v has some threshold θ_v chosen uniformly at random from the interval $[0, 1]$, and every edge (u, v) is assigned a weight $w_{u,v}$ s.t.:

$$\sum_{u \in \mathcal{N}(v)} w_{u,v} \leq 1$$

where $\mathcal{N}(v)$ denotes the set of neighbors of v. In this model v becomes activated at time step $t+1$ if the weighted sum of its neighbors' edges who were activated in previous time steps exceeds the threshold θ_v. This model was also proposed in [Kempe et al. 2003] and inspired by earlier work of Granovetter [Granovetter 1978] and Schelling [Schelling 1978].

Submodular threshold. This model is identical to the linear threshold model with the exception that the influence of neighbors is measured as a *monotone submodular* function. That is, in addition to a threshold, every node $v \in V$ has

[6] Observe that this is not a structural property of the graph, but rather a more convenient way to describe the model. That is, we can describe an equivalent process on graphs without self loops, but the notation will become messier.

a *monotone* and *submodular* function $f_v : 2^{\mathcal{N}(v)} \to \mathbb{R}$, and the node is activated only if the value of f_v over the neighbors influenced in previous time steps exceeds the threshold θ_v. Like independent cascade and linear threshold, this model was also defined in [Kempe et al. 2003] to better capture the idea that influence by our peers may exhibit diminishing return effects.

3.1.1 *The diffusion processes lead to submodular influence functions.* The models above all define an *influence function* $f : 2^V \to \mathbb{R}_+$ that can be used to quantify the expected number of nodes influenced as a result of selecting some subset of nodes to initiate a cascade. If all nodes are initially set to have opinion 0, we can select a subset of nodes S and set their opinion to be 1 at time step $t = 0$. For some time step $t > 0$, the influence $f(S)$ at time t would be the expected number of nodes who have opinion 1, where the expectation is over the randomness of the model [7]. An important structural property of these models is that their resulting influence functions are all *monotone* and *submodular.* A function $f : 2^V \to \mathbb{R}_+$ is *monotone* if $S \subseteq T$ implies $f(S) \leq f(T)$, for any $S, T \subseteq V$. The function is *submodular* if it has a natural diminishing returns property: for any $S \subseteq T \subseteq V$ and $a \in V \setminus T$ a function is submodular if:

$$f_S(a) \geq f_T(a)$$

where $f_A(B) = f(A \cup B) - f(A)$. The abstraction of the influence process as a set function allows defining the challenge of influence maximization as an elegant optimization problem: given a budget of k individuals to select and an influence function $f : 2^V \to \mathbb{R}_+$ that results from a diffusion model, find:

$$S \in \underset{\substack{T \subseteq V \\ |T| \leq k}}{\arg\max}\, f(T) \tag{1}$$

For functions that are monotone and submodular, it is well known that the problem is NP-hard. Fortunately however, a simple greedy algorithm with access to a value oracle of the function (given S the oracle returns $f(S)$) can obtain a $1 - 1/e$ approximation to the optimal solution [Nemhauser et al. 1978], which is tight unless P=NP [Feige 1998]. The algorithm begins with the empty set as its solution and iteratively adds the element whose marginal contribution is largest until exhausting its budget k.

3.1.2 *Influence in order of algorithmic difficulty.* Finally, before we continue to discussing the adaptive seeding model, we will outline the relationship between the different models, in terms of their "algorithmic difficulty".

[7] Notice that the influence function is also determined by the number of time steps t. In our discussion of influence functions we will not specify the time step in which we measure the cascade, and implicitly assume that it is some large polynomial in the size of the network. Notice that the independent cascade, linear threshold, and submodular threshold models are all *progressive*, i.e. once a node is activated it remains activated. Hence, these models stabilize (the function does not change its value) after at most n time steps. In the voter model, it is known that the influence function stabilizes after $O(n^3 \log n)$ time steps. We can therefore assume that the diffusion process is being run for polynomially-many time steps, and not mention the value of t when describing the influence function.

Voter model as additive functions. Influence in the voter model is an *additive* function, i.e. $f(S) = \sum_{a \in S} f(a)$, as proven by Even-Dar and Shapira [Even-Dar and Shapira 2011], and thus a special case of submodular functions. Maximizing influence of an additive function can be done in polynomial-time as it simply requires us to choose the k nodes whose individual value is largest.

Independent cascade and linear threshold as coverage functions. In their seminal paper, Kempe, Kleinberg, and Tardos defined a broader class of diffusion models called the *Triggering model* which includes *linear threshold* and *independent cascade* and proved that influence in this model is a monotone submodular function as well. Recall that a function $f : 2^V \to \mathbb{R}_+$ is called *coverage* if there exists some universe $\mathcal{U}$ where the elements in the ground set V are identified with $T_1, \ldots, T_{|V|}$ which are all subsets of $\mathcal{U}$ and

$$f(S) = |\cup_{i \in S} T_i|$$

The Triggering model can be thought of a special case of coverage functions. One can show that for every influence function f which results from diffusion in the Triggering model there is a coverage function $\tilde{f}$ whose representation is polynomial in the size of the graph and $\tilde{f}(S)$ is arbitrarily close to $f(S)$, for any set $S \subseteq V$, with high probability [Seeman and Singer 2013]. It is well known that maximizing a coverage function under a cardinality constraint is NP-hard and no algorithm can obtain an approximation strictly better than $1 - 1/e$ unless P=NP [Feige 1998]. As we will later see, coverage functions can be optimized via convex programs which plays an important role in adaptive seeding.

Submodular threshold as general monotone submodular functions. Finally, Mossel and Roch proved that influence in the more general *submodular threshold* model results in a monotone submodular function as well [Mossel and Roch 2007]. Given any monotone submodular function defined over a ground set of size n, we can always construct a graph where selecting the k most influential nodes in the graph in the Submodular threshold model is as hard as maximizing the monotone submodular function under a cardinality constraint k.

Intuitively, the above diffusion models generate influence functions of an increasing order of difficulty: the voter model, triggering model, and the submodular threshold model. As we will see in the following section, adaptive seeding in these models requires algorithmic approaches that increase in their level of difficulty.

3.2 The Adaptive Seeding Model

Recall that in the adaptive seeding model we are given a set of core nodes X, their set of neighbors $\mathcal{N}(X)$, each realizing independently with a probability p_i, as well as a budget $k \in \mathbb{N}$ and a function $f : 2^{\mathcal{N}(X)} \to \mathbb{R}$. Our goal is to select $S \subseteq X$ s.t. in expectation over all possible realizations $R_1, \ldots, R_m$ of $\mathcal{N}(X)$ the value of a set of its neighbors T_i of size $k - |S|$ that appears in the realization R_i is maximal:

$$\begin{aligned} \max_{S \subseteq X} \sum_{i=1}^{m} f(T_i) \cdot p(R_i) & \\ T_i \subseteq R_i \cap \mathcal{N}(S) & \quad \forall i \in [m] \\ |S| + |T_i| \leq k & \quad \forall i \in [m] \end{aligned}$$

Before we continue to discuss different algorithmic approaches, it would be useful to discuss this model and be convinced that this is indeed a problem worth solving.

A generalization of influence maximization. Notice that when $X = V$ adaptive seeding is a strict generalization of the influence maximization problem.

Incentives vs. diffusion. It may seem at first as if the two-stage approach is redundant since influential nodes may be reached through the diffusion process in the network. The crucial difference is that adaptive seeding relies on incentives rather a natural diffusion process to reach influential nodes. The underlying assumption in the influence maximization problem is that individuals have a standard bayesian utility model with no externalities: once receiving a unit of reward (e.g. free sample), a node forwards information with probability 1. Nodes who do not receive an incentive to forward information do so with probability that is dictated by the stochastic process associated with the influence model. Therefore, an influential node will forward information to her neighbors according to the diffusion model. In adaptive seeding however, the model incentivizes influential nodes to initiate a cascade by allocating unit of budgets in the second stage. In other words, in adaptive seeding the goal is to recruit the neighbors to become influencers using the budget (free samples), as opposed to influencing them to forward the information without incentives.

Unit costs. Although it seems quite plausible that the probabilities of attracting neighbors could depend on the rewards they receive, the simplification assuming unit costs is deliberate. This simplification is consistent with the celebrated KKT model [Kempe et al. 2003], and can be extended to the case where nodes take on different costs [Rubinstein et al. 2015]. Some might argue that the likelihood of becoming an early adopter is inversely proportional to one's influence, but then any influence maximization model loses substance since even the standard influence maximization problem then becomes trivial.

Beyond two stages. It is tempting to think of models that go beyond two stages, and consider a general n-stage approach. First, we found that there is no significant gain in going beyond two-stages. One can think of generalizations of the friendship paradox where we consider the average degree of nodes that are multiple hops away from the set of core nodes. From empirical observations it seems like asymptotic improvements cannot be obtained beyond one hop from a random sample [Lattanzi and Singer 2015]. Another caveat is that of computational complexity. One can show hardness results similar to those in [Rubinstein et al. 2015] for adaptive seeding with super-constant number of stages.

4. THE ALGORITHMIC PERSPECTIVE

The above formulation of adaptive seeding introduces a new kind of two-stage stochastic optimization problems. In this section we outline different algorithmic approaches for designing approximation algorithms with provable guarantees in this setting. In particular, we will discuss the main result summarized below.

Theorem ([Badanidiyuru et al. 2016]). *For every constant $\epsilon > 0$ and any monotone submodular function there is an algorithm that runs in polynomial time and returns a solution which is a $((1-1/e)^2 - \epsilon)$-approximation to the optimal adaptive solution.*

We will first sketch an algorithm for the case in which the underlying function is *additive*. Such functions include influence in the voter model, and the experiments we describe in the following section use a variant of this algorithm. We will then discuss three different approaches that can be applied to functions in the Triggering model, as well as general monotone submodular functions.

4.1 Special Case: Additive Functions

In the special case where the influence function is additive, i.e. $f(S) = \sum_{a\in S} f(a)$ the adaptive seeding problem reduces to maximizing a monotone submodular function under a cardinality constraint. To see this, we can first consider the non-stochastic case, i.e. the special case in which all nodes realize with probability 1, and define:

$$F^{(t)}(S) = \max_{\substack{|T|\leq t \\ T\subseteq N(S)}} f(T)$$

It is easy to show that when f is *additive* then for any $t > 0$ the function $F^{(t)}$ is monotone and submodular. Therefore for any $t \in [k]$ we can use the celebrated greedy algorithm to obtain a $1-1/e$ approximation to find:

$$S_t^\star \in \operatorname*{arg\,max}_{S\subseteq X:|S|\leq k-t} F^{(t)}(S)$$

We know that there exists some partition $(k-t, t)$ of the budget in which the optimal solution spends $k-t$ on the core set and t on their neighbors. Thus, running a greedy algorithm on all functions in $\{F^{(t)}\}_{t=1}^{k}$ with budget $k-t$ and selecting the solution whose value is largest is a $1-1/e$ approximation.

To solve the stochastic version we can define:

$$F^{(t)}(S) = \sum_{i=1}^{m} p(R_i)\left(\max_{\substack{T\subseteq S\cap R_i \\ |T|\leq t}} f(T)\right)$$

Since this is a positive-weighted sum of monotone submodular functions, this function is monotone submodular as well. Although this is a sum over exponentially many terms, one overcome this by sampling polynomially-many realizations and evaluating the marginal contribution of adding a node into the solution on the sum of the sample when running the greedy algorithm. So, the moral of the non-stochastic case applies: when the underlying function f is additive, there is an approximation algorithm that is arbitrarily close to the $1-1/e$ guarantee.

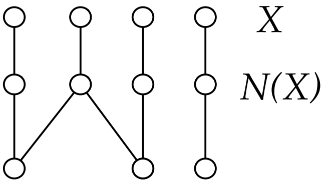


Fig. 4: An example of a graph where the two stage adaptive seeding is not submodular when the influence function is a coverage function. From left to right the nodes in X are a, b, c, d.

4.1.1 *$F^{(t)}$ is not submodular for coverage functions.* Unfortunately, for functions like coverage (which is a special case of influence in the independent cascade model where all probabilities are 1), the above approach is inapplicable as $F^{(t)}$ is no longer submodular, even in the non-stochastic case. To see this, consider the example illustrated in Figure 4. It is easy to see that $F^{(2)}(\{a,c\}) = 2$ and also that $F^{(2)}(\{a,c,d\}) = 2$. On the other hand, $F^{(2)}(\{a,b,c\}) = 2$ and also $F^{(2)}(\{a,b,c,d\}) = 3$. This contradicts the diminishing returns property of submodular functions since the marginal contribution of d to $\{a,b,c\}$ is larger than its marginal contribution to $\{a,c\}$.

4.2 An Overview of the Different Algorithmic Approaches

So, even for coverage functions the adaptive seeding objective is no longer submodular. In this section we will focus on developing algorithmic approaches that can be applied to coverage functions (and thus, for functions in the Triggering model), as well as for general monotone submodular functions. We will discuss three different algorithmic approaches:

Concave relaxation. We will first briefly sketch a concave relaxation approach developed in the first work on adaptive seeding [Seeman and Singer 2013]. The general idea is to relax the objective via a concave function and solve a mixed integer program. This approach can be used to obtain an approximation guarantee arbitrarily close to $(1-1/e)^2$ for functions in the Triggering model; [8]

Non-adaptive policies. The optimal solution is an *adaptive* policy: one which selects a subset from X, and after the realization of its neighbors, selects an optimal solution with its remaining budget. Since adaptive policies are notorious in stochastic optimization for their difficulty, the standard approach is to design non-adaptive policies which approximate adaptive policies well. Informally, by non-adaptive policy we mean a policy which selects the subset S and a distribution on its neighbors, a priori to their realization. Non-adaptive policies can also be used to obtain a guarantee arbitrarily close to $(1-1/e)^2$ for functions in the Triggering model, and the algorithm we give here is from [Badanidiyuru et al. 2016]. Unfortunately, this approach cannot be applied to general monotone submodular functions, as we later discuss;

Locally-adaptive policies. For general monotone submodular functions we can obtain a guarantee that is arbitrarily close to $(1-1/e)^2$ by what we call *locally-*

[8] The approximation ratio reported in [Seeman and Singer 2013] is a worse constant, but tighter analysis using the ideas described here yields the $(1-1/e)^2 - \epsilon$ guarantee, for any $\epsilon > 0$.

adaptive policies. Intuitively, a locally-adaptive policy consists of a set nodes, and each node is associated with a "block" of neighbors. Rather than committing to a distribution on the neighbors a priori (as non-adaptive policies do), a locally-adaptive policy adaptively decides which neighbors to seed from the block associated with node selected, after the realization of neighbors occurs. The "local" adaptivity within a block allows us to circumvent the impossibility of the non-adaptive approach, and leads to a $(1-1/e)^2$ guarantee for general monotone submodular functions.

At a very high level, the general idea behind the different algorithmic approaches is the same: we run a greedy algorithm which at every step seeks to add the *densest block*: the node from the core set X and a fractional portion of its neighbors who maximize value for budget spent. We will begin by describing a solution for the non-stochastic case as a warm up to convey the main idea, and continue to describe each one of the algorithmic approaches.

4.3 Warm Up: the Non-Stochastic Case

Let's begin by considering the non-stochastic version of the problem. That is, the version in which every node in the set of neighbors appears with probability 1. Here we are given a set X and its neighbors $\mathcal{N}(X)$, there is a monotone submodular function defined on $\mathcal{N}(X)$, and the goal is to select $t \leq k$ elements in X connected to a set of size at most $k-t$ in $\mathcal{N}(X)$ for which the submodular function has the largest value. Somewhat surprisingly, a variant of the celebrated greedy algorithm can yield a solution that is arbitrarily close to the optimal $1-1/e$ bound for this two-stage problem. The algorithm can be best described through what we call *ϵ-blocks*, which constitute the core idea behind the approaches we later describe. For each element $x \in X$ we will define its ϵ-block to be its set of children of size at most $1/\epsilon$ whose marginal contribution is *densest*, where the density of a set of nodes is simply the ratio between their marginal contribution and cardinality. More formally, given a set $T \subseteq \mathcal{N}(X)$ in the current solution, the ϵ-block of $x \in X$ is:

$$\operatorname*{arg\,max}_{\substack{B\subseteq\mathcal{N}(x)\\|B|\leq 1/\epsilon}} \frac{f_T(B)}{|B|}$$

The densest node $x \in X$ is simply the node whose ϵ-block is maximal. The algorithm is now simple: until exhausting the budget, add the densest node $x \in X$ and its ϵ-block. For any $\epsilon > 0$, this algorithm is a $(1-1/e-\epsilon)$-approximation. The idea behind the analysis is very simple: at every stage this algorithm selects a set of nodes in $\mathcal{N}(X)$ that is a $(1-\epsilon)$ approximation to the densest set in $\mathcal{N}(X)$, which is a sufficient condition to obtain a $(1-1/e-\epsilon)$ approximation. To see this, think of the optimal solution O as a set of t nodes in X and for each node $o_i \in O$ we can associate a set of children $C_i \in \mathcal{N}(X)$. The cost associated with each node and its children is simply $1+|C_i|$. For the maximal ϵ-block B we have that:

$$\frac{f_T(B)}{|B|} \geq \frac{f_T(B)}{1/\epsilon} \geq \frac{f_T(C_i)}{|C_i|}$$

which implies:

$$\frac{f_T(B)}{1+1/\epsilon} \geq \Big(\frac{1}{1+1/\epsilon}\Big)\frac{f_T(C_i)}{\epsilon|C_i|} = \Big(\frac{1}{1+\epsilon}\Big)\frac{f_T(C_i)}{|C_i|} \geq \Big(1-\epsilon\Big)\frac{f_T(C_i)}{1+|C_i|}$$

Through similar arguments as those in the analysis of standard (i.e. one stage) submodular maximization, one can show that the ability to find a $(1-\epsilon)$-approximation to the densest subset of nodes implies a $1-1/e-\epsilon$ approximation.

Although it only holds in the dummy non-stochastic version of the problem, the algorithm we just sketched encapsulates the core idea behind the approaches for solving the stochastic case.

4.4 Concave Relaxation

As conveyed in the warm up above, the main algorithmic challenge in adaptive seeding is due to the uncertainty regarding the realization in the second stage. The first approach we discuss manages the uncertainty using a concave program. For any influence function in the Triggering model, we create a concave objective function $L : [0,1]^n \to \mathbb{R}$ which approximates the influence function arbitrarily well, for any given integral solution. This objective is in fact a concave relaxation of a coverage function, and from this point onwards we can think of functions in the Triggering model as coverage functions. Using this concave objective we formulate the following mixed-integer program:

$$\max\ L(\mathbf{y}) \tag{2}$$

$$\text{s.t.}\ \sum_{i\in X} x_i + \sum_{i\in\mathcal{N}(X)} p_i y_i \leq k \tag{3}$$

$$y_i \leq \sum_{j:i\in\mathcal{N}(j)} x_j, \qquad \forall i \in \mathcal{N}(X) \tag{4}$$

$$x_i \in \{0,1\}, \qquad \forall i \in X \tag{5}$$

$$y_i \in [0,1], \qquad \forall i \in \mathcal{N}(X) \tag{6}$$

In the formulation above the x_i variables represent the nodes in X and the y_i variables represent their neighbors, $\mathcal{N}(X)$. Condition (3) ensures the budget is maintained in *expectation.* Condition (4) is a *dependency* constraint ensuring every nonzero assignment of a node in $\mathcal{N}(X)$ is connected to a node in X that has been selected. The conditions (5) and (6) ensure that a solution will have only integral components for nodes in X, while nodes in $\mathcal{N}(X)$ can be selected fractionally.

The solution to this program is a vector $(\mathbf{x},\mathbf{y}) \in [0,1]^{|X|+|\mathcal{N}(X)|}$ with an *integral* component $\mathbf{x} \in \{0,1\}^{|X|}$ and *fractional* component $\mathbf{y} \in [0,1]^{|\mathcal{N}(X)|}$. One can show that any feasible solution to this program can be converted to a feasible solution to the adaptive seeding problem: with high probability, for any realization of neighbors R, the resulting integral solution $\mathbf{x}$ is connected to a subset of R whose influence is arbitrarily close to $1-1/e$ of the of the optimal adaptive seeding policy in R.

What's left is to actually *solve* the program. We cannot use standard convex optimization methods as the integrality constraints break the convexity of the domain. Instead, we design a "densest ascent" algorithm: we consider each node in X as a direction, and the fraction to spend on its neighbors as the step size. The algorithm iteratively takes steps in the *densest direction*: the node in X and the fraction of its neighbors whose marginal value over budget spent is largest. We do so until exhausting the budget, and show that this algorithm obtains a $1 - 1/e$ approximation guarantee. Chaining both approximation factors, this approach gives a solution that is arbitrarily close to $(1 - 1/e)^2$.

So what's the intuition behind this program? It is actually the concave approximation of the *non-adaptive* solution, which is a central theme in all our techniques.

4.5 Non-Adaptive Policies

A non-adaptive policy is a pair of sets $(S, T) \subseteq X \times \mathcal{N}(X)$ where S represents the set selected in the first stage and $T \subseteq \mathcal{N}(S)$ is the set selected in the second stage. The natural definition for feasibility would be that the policy selects at most k nodes, though it is easy to construct examples where such solutions result in unbounded approximation ratios. We therefore consider *relaxed* non-adaptive policies that select at most k nodes *in expectation*, where the expectation is over the randomization in the model, i.e. the probabilities of nodes arriving in the second stage. We define the value of a non-adaptive policy (S, T) to be $F(T) = \sum_{i\in[m]} p(R_i)f(T \cap R_i)$ and its cost as $|S| + c(T)$, where $c(T) = \sum_{i\in T} p_i$. Finding the optimal non-adaptive policy requires solving the optimization problem:

$$\texttt{OPT}_{NA} = \max_{S,T} \{F(T) \ : \ |S| + c(T) \leq k, \ S \subseteq X, T \subseteq \mathcal{N}(S)\} .$$

The crucial difference between these policies and adaptive ones is that non-adaptive policies fix a set T a priori to seeing the realization, whereas adaptive policies have the luxury of selecting a different set T_i for every realization R_i. Note that these policies are not a relaxation of adaptive policies nor are they a special case, since on the one hand they fix one set T but on the other hand obey the budget only in expectation. Thus, it is not immediately clear they are useful for approximating adaptive policies. Fortunately, however, it turns out that non-adaptive policies can indeed be useful as the adaptivity gap (i.e. the ratio between the value of the optimal adaptive policy and that of the non-adaptive policy) is bounded. In particular, the adaptivity gap is *exactly* $1 - 1/e$.

Theorem ([Badanidiyuru et al. 2016]). *For every $\epsilon > 0$, given an algorithm that finds a* ***non-adaptive*** *policy with value at least $\alpha\texttt{OPT}_{NA}$, there is a $(1 - 1/e)\alpha - \epsilon$ approximation algorithm for the optimal* ***adaptive*** *policy, and this bound is tight.*

The proof utilizes an interesting connection between the optimal adaptive policy and what is known as the *concave closure* of the underlying submodular function [Calinescu et al. 2011]. Using this connection and properties of the concave closure from [Calinescu et al. 2011] we can bound the error of a *fractional* non-adaptive policy that we then round using contention resolution schemes developed in [Vondrák et al. 2011] to get our result. To see that the adaptivity gap is tight, consider an instance with a single node in X connected to $n = 1/\delta^2$ nodes, each appearing with probability δ, for some small $\delta > 0$. The function is:

$$f(T) = \begin{cases} 1 & \text{if } T \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

For a budget of 2, an optimal adaptive policy seeds the single node in X, waits for the realization of its neighbors, and seeds whichever node realizes. The optimal non-adaptive policy here seeds the single node in X and spends the rest of its budget on $1/\delta$ nodes in $\mathcal{N}(X)$, which has an expected utility of $1-(1-\delta)^{1/\delta} \approx (1-1/e)$. The adaptivity gap is therefore at least $1-1/e$.

4.6 Optimization via Non-Adaptive Policies

The bounded adaptivity gap is a powerful concept: instead of thinking about the complex adaptive problem, one can focus on the simpler non-adaptive version and only sacrifice a $1-1/e$ factor in the approximation guarantee. To solve the non-adaptive version we can consider a greedy algorithm, similar to the one sketched in Section 4.3: at each step, as long as it doesn't exceed the total budget, the algorithm adds the densest ϵ-block. For non-adaptive policies, an ϵ-block is a node $x \in X$ and a subset of its neighbors whose *expected* cardinality is at most $1/\epsilon$. By using standard inductive arguments one can show that for any submodular function this algorithm has an approximation ratio of $1-1/e^{\alpha}-\epsilon$, where α is the approximation guarantee of the procedure which finds the optimal ϵ-block. The problem therefore reduces to computing the best approximation for the densest ϵ-block.

4.6.1 *Finding dense ϵ-blocks reduces to optimization with small probabilities.* Note that if all the probabilities on nodes are at least some constant, finding the densest ϵ-block is easy: given some constant $\epsilon > 0$ the algorithm simply enumerates over all $x \in X$ and over all possible subsets of items $T \in \mathcal{N}(x)$ s.t. $c(T) \leq 1/\epsilon$. The challenge is when the probabilities are small, i.e. in $o(1)$. In this case enumerating over all possible solutions is computationally infeasible, and the problem of finding ϵ-blocks reduces to the following fundamental problem.

Definition. OPTIMIZATION-WITH-SMALL-PROBABILITIES *(OSP) problem: We are given a monotone submodular function $f : 2^{[n]} \to \mathbb{R}$ and probabilities of each element realizing $p_1, \ldots, p_n$ s.t. $\max_i p_i \leq \delta$. Our goal is to find a set T of expected size k that maximizes the expected value of f.*

Since we can exhaustively search over all sets of elements whose probability of realizing is some constant, finding arbitrarily dense ϵ-blocks reduces to solving OSP when $\delta \in o(1)$. In particular, we are interested in obtaining an approximation for OSP that becomes arbitrarily close to optimal as δ approaches 0. At a first glance, it may seem like no algorithm should be able to get an approximation better than $1-1/e$ for this problem: when $\delta = 1$ the problem identifies with submodular maximization under a cardinality constraint and no algorithm can do better than $1-1/e$ unless P=NP even for coverage functions [Feige 1998]. It seems like shrinking the constraint polytope by a factor of δ should not make a difference in the optimization. Surprisingly, it does, and one can obtain a $1-\delta/2$ approximation for this problem when the underlying function is a coverage function [9].

4.6.2 *A $1-\delta/2$ approximation for OSP with coverage functions.* We'll consider a fractional version of this problem in which the items of T are chosen independently with probabilities encoded in $\mathbf{q} \in [0,1]^n$. Since we are only interested in small values

[9] In fact, this result applies to a more general class of functions known as *matroid rank sum.*

of δ it is easy to round fractional solutions using the *pipage rounding* technique with an arbitrarily small loss [Ageev and Sviridenko 2004]. Formally, we want to solve:

$$\begin{aligned} \max_{\mathbf{q}} \quad & \hat{F}(\mathbf{q}) = \sum_{T \subseteq N} \left(\prod_{i \in T} q_i \prod_{i \notin T} (1 - q_i) \right) f(T) \\ \text{s.t.} \quad & \sum_i q_i \leq k \\ & q_i \in [0, p_i] \quad \forall i \in [n] \end{aligned}$$

Our main tool is to solve the above problem but with the following objective instead:

$$\widetilde{F}(\mathbf{q}) = \sum_{T \subseteq N} \left(\prod_{i \in T} (1 - e^{-q_i}) \prod_{i \notin T} e^{-q_i} \right) f(T)$$

This program is similar to the one used in [Dughmi et al. 2011] in their Poisson rounding technique. They show that $\widetilde{F}$ is a concave function when f is a coverage function, and hence the program can be solved efficiently. While in general, there is a $(1-1/e)$ gap between $\widetilde{F}$ and $\hat{F}$ (which is what is used in [Dughmi et al. 2011]), when δ is small, there is only a small gap between the functions.

Lemma. *If f is a coverage function, then $\forall \mathbf{q} \in [0, \delta]^n$:*

$$(1 - \delta/2)\hat{F}(\mathbf{q}) \leq \widetilde{F}(\mathbf{q}) \leq \hat{F}(\mathbf{q}).$$

Since $\widetilde{F}$ is concave when f is a coverage functions, and we can use a convex programming algorithm to find arbitrarily good approximations of the optimal solution in polynomial time. We can combine the case of small probabilities and large probabilities and obtain arbitrarily good approximations for the densest ϵ-block. We can therefore use the densest block in the greedy algorithm we sketched above and obtain an approximation arbitrarily close to $1 - 1/e$ for the non-adaptive problem. Since, as discussed above, functions in the Triggering model can be approximated arbitrarily well with coverage functions we get:

Theorem ([Badanidiyuru et al. 2016]). *For any $\epsilon > 0$ there is a polynomial-time algorithm that returns a $(1 - 1/e - \epsilon)$-approximation of the optimal **non-adaptive** policy for any function in the Triggering model.*

Since we know that the adaptivity gap is at most $1-1/e$, this $1-1/e-\epsilon$ guarantee for non-adaptive policies implies a $(1-1/e)^2 - \epsilon$ guarantee for the adaptive seeding problem. Unfortunately, for general monotone submodular functions approximating OSP arbitrary well with a constant budget (and thus computing optimal densest ϵ-blocks) is at least as hard as finding a planted clique in a graph.

Theorem ([Badanidiyuru et al. 2016]). *There are monotone submodular functions for which approximating OSP with a constant budget $k = 1.7$ within any constant factor better than $\left(1 - e^{-k/2}\right) / \left(1 - \left(\frac{k}{2} + 1\right) e^{-k}\right) \approx 0.865$, implies that there is a polynomial-time algorithm for the* Planted-Clique *problem that succeeds with high probability.*

This lower bound implies that the non-adaptive framework we use here cannot obtain the $(1 - 1/e)^2$ approximation ratio for general submodular functions. This motivates our use of ϵ-locally adaptive policies discussed in the following section.

4.7 Locally Adaptive Policies

We will conclude our technical exposition by sketching the $(1 - 1/e)^2 - \epsilon$ approximation algorithm for general monotone submodular function. The approach relies on the novel definition of a restricted class of adaptive policies which we call ϵ-*locally-adaptive*. Informally, we say that a policy is ϵ-locally-adaptive, if it can be divided into ϵ-blocks. In this context, an ϵ-block is a subset of X of constant size (for technical reasons these are not singletons as in previous cases), and for each realization an adaptively chosen set of constant size of its neighbors.

Definition. *An (adaptive) ϵ-block is a set $S \subseteq X$ of size at most $1/\epsilon^2$ and for each realization R_i a set $T_i \subseteq \mathcal{N}(S) \cap R_i$ of size at most $2/\epsilon$. The cost of a block B is $c(B) = |S| + \max_i(|T_i|)$. An* ϵ-locally-adaptive *policy is a set $\mathcal{B}$ of (not necessarily disjoint) ϵ-blocks.*

Let $T_{i,B}$ be the set seeded by block B in realization R_i and let $\mathcal{T}_i(\mathcal{B}) = \bigcup_{B \in \mathcal{B}} T_{i,B}$. We abuse notation and generalize the value and cost functions to be applied on these policies. That is, we let the value of such a policy be $F(\mathcal{B}) = \sum_{i=1}^{m} p(R_i) f(\mathcal{T}_i(\mathcal{B}))$ and its cost $c(\mathcal{B}) = \sum_{B \in \mathcal{B}} c(B)$. The optimal ϵ-locally-adaptive policy with budget k is then:

$$\max_{\mathcal{B}} \left\{ F(\mathcal{B}) \ : \ c(\mathcal{B}) \leq k, \forall B \in \mathcal{B} : |S_B| \leq 1/\epsilon^2, \forall i : T_{i,B} \subseteq \mathcal{N}(S_B) \cap R_i, |T_{i,B}| \leq 2/\epsilon \right\}$$

This adaptive variant of ϵ-blocks allows us to find the optimal subset for each realization (much in the same way as in the warmup presented in the introduction) and thus find the optimal block.[10] Thus, the non-adaptive block structure allows for greedy optimizations and the power of adaptivity within a block circumvents the hardness result for general monotone submodular functions of the previous section. We then can show that the optimal ϵ-*locally-adaptive* policy is a $(1 - 1/e)$-approximation to the optimal adaptive policy by using the fact that locally-adaptive policies strictly dominate non-adaptive policies. In particular, we show we can convert any non-adaptive policy to an ϵ-*locally-adaptive* policy, with arbitrarily small loss in value.

4.8 Separation Between ϵ-Locally Adaptive and Adaptive Policies

A natural question is then whether locally-adaptive policies are as powerful as adaptive policies. It turns out that there exists a ≈ 0.853 gap between the optimal

[10] Since there are possibly exponentially-many realizations we can't output an explicit description of a locally-adaptive policy. Instead, for each block we can output its first stage set S_i and a budget k_i for it to optimize in the second stage, as well as an order over the blocks. On every realization the policy seeds the second stage nodes by going over the blocks in order and optimizing the choices of each block given only the choices made by the previous blocks. Note that this implicitly determines the content of each block. In our algorithm we implicitly assume the order on the blocks of $\mathcal{B}$ is the order in which the algorithms adds them to $\mathcal{B}$. Note that we can approximate $F(\mathcal{B})$ and $F_{\mathcal{B}}(B)$ (the marginal value of block B for policy $\mathcal{B}$) for such a policy to any desired accuracy by sampling realizations and running this process on each of them (we thus assume in the analysis that we have an oracle for their value).

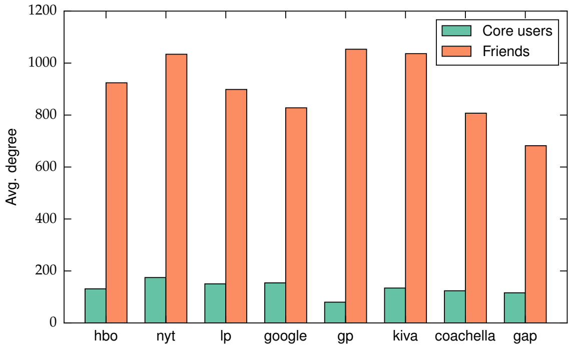


Fig. 5: Comparison of the average degree of core set users and average degree of their friends.

locally-adaptive policy and the optimal adaptive policy [Badanidiyuru et al. 2016].

5. EXPERIMENTS

So far we focused on theoretical models and guarantees that justify adaptive seeding. Our next and final goal for this survey is to discuss the performance of adaptive seeding in practice. To do so, we describe experiments we conducted on a data set we collected that resembles the scenario in which adaptive seeding can be applied. In our experiments we collected data from the Facebook social networking service. In particular, we used Facebook Pages (FP) which are webpages on Facebook operated by various commercial entities. A commercial entity can post messages on the FP it operates, and Facebook users can view and endorse the post ("like" it). We selected organizations from various verticals that operate a FP (e.g. Kiva, Peet's Coffee, GAP, HBO, etc.). For each page, we selected a post and then collected data about the users who endorsed the post and their friends. We collected this data set since it is representative of the scenario we study here. The users who endorsed the post represent the core users since social marketing campaigns often target users who have already expressed interest in the topic being promoted. More generally, users who endorse a post serve as a reasonable proxy for customers in an online or offline store, or more generally, users that one typically has access to in social marketing campaigns.

5.1 Experimental Setup

5.1.1 *The data set.* Each page is operated by an institution or an entity whose associated FP is regularly used for promotional posts. On each of these pages, we selected a posted message with approximately 1,000 "likes", i.e. roughly 1,000 unique users who endorsed the message. The set of users who endorsed those posts constitute our core set. We then crawled the social network of the users who endorsed the post: for each user, we collected her list of friends, and the degrees (number of friends) of these friends. For each FP this gave us a core set X of roughly 1,000 users and a set of roughly 100,000 neighbors. [11]

[11] Due to privacy settings of the core set users, it was not always possible to access their list of friends. In our experiments we removed these users from the data set since their ability to spread information could not be readily determined. This effect, combined with various errors

5.1.2 *Influence in the voter model.* In the experiments described here we use the sum of degrees to quantify influence. We use this measure since Facebook's restrictions on crawling made it infeasible to collect more than the neighbors of the core set and these neighbors' degrees. Since we did not have access to the entire network we could not simulate diffusion models like independent cascade or linear threshold. Fortunately, for diffusion models like the *voter model* using the sum of degrees suffices: after polynomially many time steps, the influence of a set is proportional to the sum of its degrees, with high probability [Even-Dar and Shapira 2011].[12] Since the Facebook network has over one billion users, convergence after a polynomial number of steps in the network size may seem potentially irrelevant in practice. Fortunately, our experiments on other large-scale networks where we have access to the entire network show convergence after fewer than 20 time steps. We discuss these experiments at the end of this section.

5.2 The Friendship Paradox in Facebook Pages

We first investigated the friendship paradox effect in the data sets. That is, we measured the average degree of users in the core set and compared it against the average degree of their friends. The results are plotted in Figure 5. The average degree of friends of the core set is about an order of magnitude larger than the average degree in the core set. This validates two ideas we used to motivate our models. First, the low and relatively homogeneous average degrees of the core sets indicates that the degree distribution of users who follow a post indeed resembles the degree distribution of a random sample from a power-law graph. Second, we see that a phenomenon similar to the friendship paradox indeed exists in these settings.

5.3 Performance of Adaptive Seeding

To test the performance of adaptive seeding we ran a two-stage optimization algorithm and compared it against other methods for influence maximization. Recall from our discussion in the previous section that when the influence function is additive, adaptive seeding can be reduced to monotone submodular maximization under a cardinality constraint. Thus, a variant of the standard greedy algorithm can be used to obtain an approximation guarantee that is arbitrarily close to $1 - 1/e$. In our experiments we applied a slightly more sophisticated version than the one described in the previous section whose approximation ratio is $1 - 1/e$ as well. The approach described in the previous section relies on sampling which becomes infeasible on large data sets. In the experiments described here we used a greedy algorithm which avoids sampling by solving a non-adaptive variant of the problem. Since the function is additive one can show that the adaptivity gap is arbitrarily close to 1 [Horel and Singer 2015].

In the first experiment we assumed that all probabilities in the adaptive seeding model equal 1. This implicitly assumes that every friend of a user who followed a

encountered during the data collection, accounts for an approximate 15% reduction between the users who endorsed a post and the number of users in the datasets we used.

[12] Since the influence process cannot be controlled by the designer, often the assumption is that the influence process runs until it stabilizes. In linear thresholds and independent cascades for example, the process terminates after n steps, and in the voter model the process converges after polynomially-many time steps.

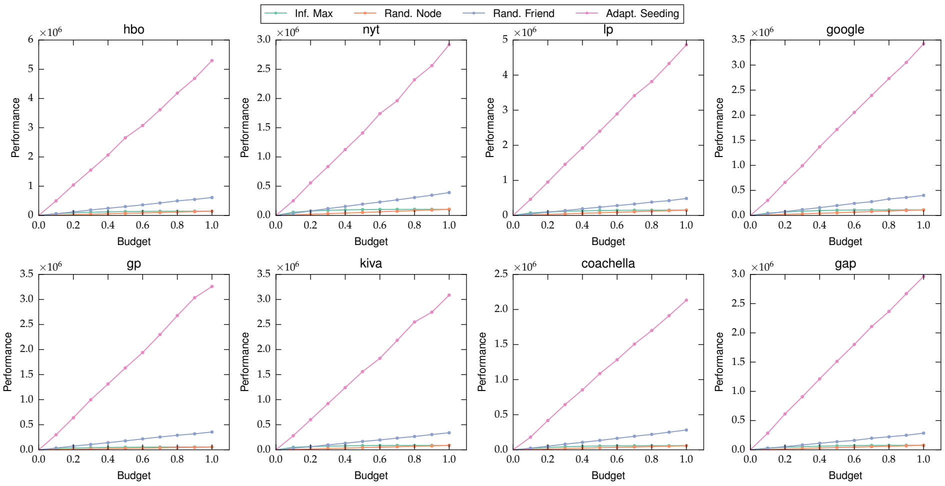


Fig. 6: Performance of adaptive seeding compared to other influence maximization approaches. The horizontal axis represents the budget used as a fraction of the size of the core set. The vertical axis is the expected influence measured by voter model.

certain topic is interested in promoting the topic given a reward. In doing so, we can outline an upper bound on the performance of adaptive seeding against other methods. The implicit logic is that if the gains from adaptive seeding would be significant when assuming all probabilities in the model are 1, this would merit further investigation into adaptive seeding. We tested our algorithm against the following benchmarks:

Influence Maximization (IM): an application of the optimal influence maximization algorithm on the core set. For the voter model, when the propagation time is polynomially large in the network size, the optimal solution is to simply take the k highest-degree nodes [Even-Dar and Shapira 2011];

Random Friend (RF): we implement a naive two-stage approach: randomly select $k/2$ nodes from the core set, and for each node select a random neighbor (hence spending k rewards overall). This method simply relies on the friendship paradox, and does not use any algorithmic machinery;

Random Node (RN): we randomly select k users from the core set. This is a typical baseline for influence maximization algorithms [Kempe et al. 2003].

We plot the results of the experiment in Figure 6. It is striking to see how well adaptive seeding does in comparison to other methods. Even when using a small budget (0.1 fraction of the core set, which in these cases is about 100 nodes), adaptive seeding improves influence by a factor of at least 10, across all verticals. To confirm this, we plot the relative improvements of adaptive seeding over IM in aggregate over the different pages. Returning to Figure 6, it is also interesting to note that the RF heuristic significantly outperforms the standard IM benchmark. Using the same budget, the degree gain induced by moving from the core set to its neighborhood is such that selecting at random among the core set users' friends already

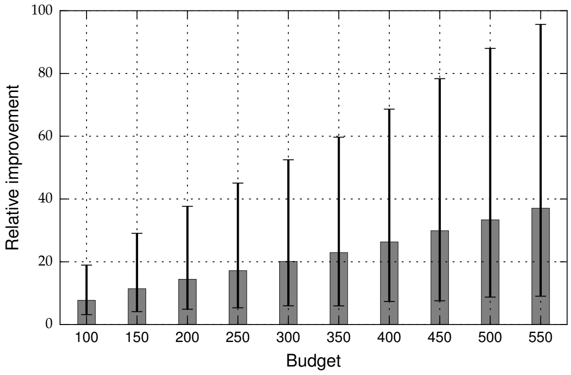


Fig. 7: Ratio of the performance of adaptive seeding to IM. Bars represents the mean improvement across all verticals, and the "error bar" represents the range of improvement across verticals.

does better than the best algorithm restricted only on the core set. Using adaptive seeding to optimize the choice of core set users based on their friends' degrees then results in an order of magnitude increase over RF, consistently for all the pages.

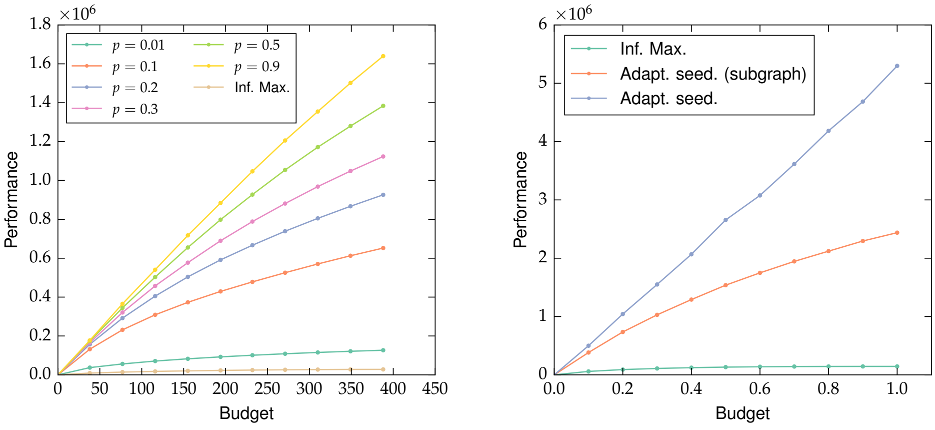
5.4 The Effect of the Second Stage Probabilities

The results presented in Section 5.3 were computed assuming the probabilities in the adaptive seeding model are 1, thus only giving an upper bound on its performance. Since this parameter critically affects the performance of adaptive seeding, we conducted two additional experiments.

5.4.1 *Impact of the Bernouilli parameter.* We computed the performance of *adaptive seeding* when each friend of a seeded user in the core set joins during the second stage independently with probability p, using different values of $p < 1$. In Figure 8a, for each value of p and each value of the budget k we plot the average influence across the different networks, as well as the average value of influence maximization. We see that even with $p = 0.01$, *adaptive seeding* outperforms IM. As p increases, the performance of *adaptive seeding* quickly increases and reaches 80% of the values of Figure 6 at $p = 0.5$.

5.4.2 *Coarse estimation of probabilities.* We conducted an experiment where we estimated the probabilities of neighbors joining the campaign in the second stage. Recall that the standard influence maximization model makes the simplistic assumption that the reward for spreading influence is valuable enough s.t. any user is willing to initiate a cascade in exchange for the reward. Since we wish to compare adaptive seeding to influence maximization we will stick to this modeling assumption. In this case, given that the neighbor has seen that there is an opportunity to be rewarded for initiating a cascade (after its neighbor in the core set was seeded in the first stage), the likelihood of that neighbor to realize depends on her affinity to the FP.

To estimate the affinity of the neighbor to the FP we used additional data at our disposal: for each neighbor we also know whether she has previously endorsed



(a)

(b)

Fig. 8: (a) Performance of adaptive seeding for various probabilities. (b) Performance of *adaptive seeding* when restricted to the subgraph of users who *endorsed* HBO (red line).

content by the FP (i.e. we see which pages she "liked" in the past). We therefore made the following coarse estimation of the probabilities: if the neighbor endorsed the FP in the past, we assumed that given a unit of reward she will be willing to promote for the FP with probability 1, and otherwise the probability of her promoting anything for the FP is 0. Figure 8b shows our results on the FP associated with the HBO entertainment channel. One can see that adaptive seeding still dramatically outperforms IM, and reaches approximately 50% of the performance when nodes in the second stage realize with probability 1.

5.5 Convergence of the Voter Model

The Facebook Pages data set we collected only describes the 2-hop neighborhood around the core users, and we therefore used the degree of users as a proxy for their influence. In the voter model, for a network with n nodes the influence of a set of nodes becomes proportional to the sum of their degrees after $O(n^3 \log n)$ time steps, with high probability [Even-Dar and Shapira 2011]. Thus if one considers influence after long-time horizons, using the sum of degrees accurately captures influence in the voter model. But a back-of-the-envelope calculation tells us that for a network of 1 billion users, if every time step of influence takes a second this bound implies that influence in the voter model becomes proportional to sum of degrees after more than 3×10^{19} years, which is several order of magnitude longer than the hypothesized age of the universe (roughly 13×10^9 years). This might be a bit too long for a campaign to reach fruition. We argue however, that in practice the convergence of the voter model will likely be faster. To demonstrate this, we used publicly available data sets of the Slashdot, and Epinions social networks obtained from [Leskovec and Sosič 2014]. In both cases we had the data of the entire network. Figure 9 shows the performance of adaptive seeding as a function of the time steps t compared to the performance of the IM benchmark. In this experiment, the core set is a random sample of 100 nodes from the network. We used the same parameters

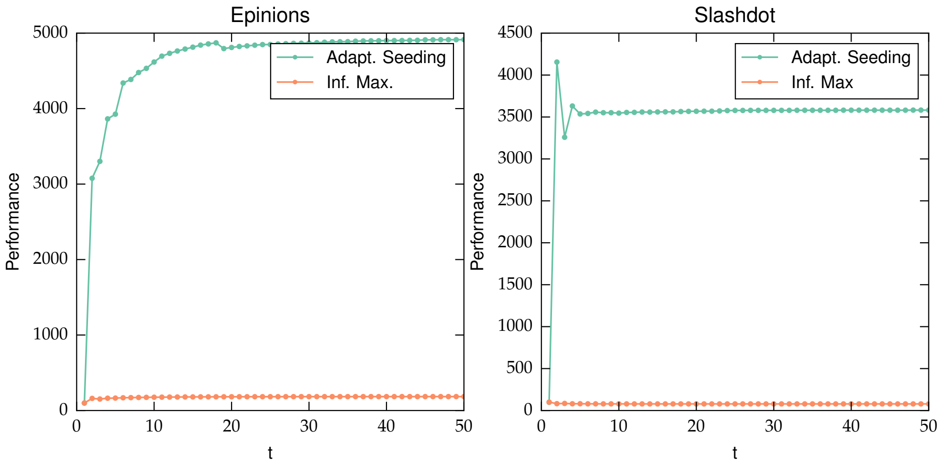


Fig. 9: Performance of adaptive seeding compared to IM for the voter influence model with t steps.

as those described in Section 5.3 and fixed the budget to be half the size of the core set. For each value of $t \in \{1, \ldots 50\}$ we ran adaptive seeding and IM using a voter model that runs for t time steps. We see that in both adaptive seeding and IM the value of influence in the voter model quickly converges (roughly 4 time steps for Slashdot, and 20 time steps for Epinions).

6. CONCLUSIONS AND OPEN QUESTIONS

Adaptive seeding seems like a natural approach that can be implemented in practice and may lead to substantial improvements in influence maximization. Perhaps the major open question involves its approximation guarantees. What is the optimal approximation guarantee for adaptive seeding? For coverage functions for example, a reasonable conjecture is that the best polynomial-time algorithm is non-adaptive. In this case $(1 - 1/e)^2$ is a natural lower bound since the problem is a (strong) generalization of Max-Cover, a loss of $(1 - 1/e)$ is inevitable, unless P=NP [Feige 1998]. The other factor of $(1 - 1/e)$ is due to the tight adaptivity gap between adaptive and non-adaptive policies discussed above. There are two caveats with this statement however. First, it may be that the hard instances for Max-Cover are not necessarily those where the adaptivity gap is largest. Second, as we witnessed with the locally-adaptive approach, there are more things in heaven and earth than are dreamt of in our non-adaptive philosophy.

Acknowledgement

This survey is a summary of more than three years of collaborative work on adaptive seeding. I'm indebted to Ashwin Badanidiyuru, Thibaut Horel, Silvio Lattanzi, Christos Papadimitriou, Aviad Rubinstein, and Lior Seeman for sharing their brilliance and patience along this journey.

REFERENCES

Abraham, I., Chechik, S., Kempe, D., and Slivkins, A. 2013. Low-distortion inference of latent similarities from a multiplex social network. In *SODA*. 1853–1872.

Ageev, A. A. and Sviridenko, M. 2004. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *J. Comb. Optim. 8,* 3.

Aiello, W., Chung, F., and Lu, L. 2000. A random graph model for power law graphs. In *Symposium on Theory of Computing (STOC).*

Badanidiyuru, A., Papadimitriou, C. H., Rubinstein, A., Seeman, L., and Singer, Y. 2016. Locally adaptive optimization: Adaptive seeding for monotone submodular functions. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2016, Arlington, VA, USA, January 10-12, 2016*. 414–429.

Bakshy, E., Hofman, J. M., Mason, W. A., and Watts, D. J. 2011. Everyone's an influencer: quantifying influence on twitter. In *WSDM.*

Barabasi, A. L. and Albert, R. 1999. Emergence of scaling in random networks. *Science 286*, 509–512.

Bollobás, B. 1980. A probabilistic proof of an asymptotic formula for the number of labelled regular graphs. *European J. Combin.*.

Borgs, C., Brautbar, M., Chayes, J., and Lucier, B. 2012. Influence maximization in social networks: Towards an optimal algorithmic solution. *arXiv preprint arXiv:1212.0884*.

Calinescu, G., Chekuri, C., Pál, M., and Vondrák, J. 2011. Maximizing a submodular set function subject to a matroid constraint. *SIAM J. Computing 40,* 6.

Chen, N. 2008. On the approximability of influence in social networks. In *SODA*.

Domingos, P. and Richardson, M. 2001. Mining the network value of customers. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 57–66.

Du, N., Song, L., Gomez-Rodriguez, M., and Zha, H. 2013. Scalable influence estimation in continuous-time diffusion networks. In *NIPS '13: Advances in Neural Information Processing Systems.*

Dughmi, S., Roughgarden, T., and Yan, Q. 2011. From convex optimization to randomized mechanisms: toward optimal combinatorial auctions. In *STOC*. 149–158.

Even-Dar, E. and Shapira, A. 2011. A note on maximizing the spread of influence in social networks. *Information Processing Letters 111,* 4, 184–187.

Feige, U. 1998. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM 45,* 4, 634–652.

Feld, S. 1991. Why your friends have more friends than you do. *American Journal of Sociology*.

Gomez-Rodriguez, M., Leskovec, J., and Krause, A. 2010. Inferring networks of diffusion and influence. In *KDD.*

Gomez-Rodriguez, M., Leskovec, J., and Schölkopf, B. 2013. Modeling information propagation with survival theory. In *ICML.*

Granovetter, M. 1978. Threshold models of collective behavior. *The American Journal of Sociology 83,* 6, 1420–1443.

Granovetter, M. 1983. The strength of weak ties: A network theory revisited. *Sociological Theory 1.*

Hodas, N. O., Kooti, F., and Lerman, K. 2013. Friendship paradox redux: Your friends are more interesting than you. *CoRR abs/1304.3480.*

Holley, R. A. and Liggett, T. M. 1975. Ergodic Theorems for Weakly Interacting Infinite Systems and the Voter Model. *The Annals of Probability 3,* 4, 643–663.

Horel, T. and Singer, Y. 2015. Scalable methods for adaptively seeding a social network. In *Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015*. 441–451.

Kempe, D., Kleinberg, J., and Tardos, E. 2003. Maximizing the spread of influence through a social network. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD).*

KLEINBERG, J. 2000. The small-world phenomenon: An algorithmic perspective. In *Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing*. STOC '00. ACM, New York, NY, USA, 163–170.

LATTANZI, S. AND SINGER, Y. 2015. The friendship paradox in power law networks.

LESKOVEC, J., ADAMIC, L. A., AND HUBERMAN, B. A. 2006. The dynamics of viral marketing. In *ACM Conference on Electronic Commerce*.

LESKOVEC, J., KRAUSE, A., GUESTRIN, C., FALOUTSOS, C., VANBRIESEN, J. M., AND GLANCE, N. S. 2007. Cost-effective outbreak detection in networks. In *KDD*.

LESKOVEC, J. AND SOSIČ, R. 2014. SNAP: A general purpose network analysis and graph mining library in C++. `http://snap.stanford.edu/snap`.

MATHIOUDAKIS, M., BONCHI, F., CASTILLO, C., GIONIS, A., AND UKKONEN, A. 2011. Sparsification of influence networks. In *KDD*.

MITZENMACHER, M. 2004. A brief history of generative models for power law and lognormal distributions. *Internet mathematics 1 (2)*.

MOSSEL, E. AND ROCH, S. 2007. On the submodularity of influence in social networks. In *STOC*.

NEMHAUSER, G. L., WOLSEY, L. A., AND FISHER, M. L. 1978. An analysis of approximations for maximizing submodular set functions ii. *Math. Programming Study 8*.

NEWMAN, M. 2003. The structure and function of complex networks. *SIAM review 45,* 2, 167–256.

RICHARDSON, M. AND DOMINGOS, P. 2002. Mining knowledge-sharing sites for viral marketing. In *KDD*. 61–70.

RUBINSTEIN, A., SEEMAN, L., AND SINGER, Y. 2015. Approximability of adaptive seeding under knapsack constraints. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation, EC '15, Portland, OR, USA, June 15-19, 2015*. 797–814.

SANTOS, F. AND LENAERTS, J. P. T. 2006. Evolutionary dynamics of social dilemmas in structured heterogeneous populations. *PNAS*.

SCHELLING, T. C. 1978. *Micromotives and Macrobehavior*. Norton.

SEEMAN, L. AND SINGER, Y. 2013. Adaptive seeding in social networks. In *IEEE Symposium on Foundations of Computer Science (FOCS)*.

SINGER, Y. 2012. How to win friends and influence people, truthfully: influence maximization mechanisms for social networks. In *WSDM*. 733–742.

UGANDER, J., KARRER, B., BACKSTROM, L., AND MARLOW, C. 2011. The anatomy of the facebook social graph. *CoRR abs/1111.4503*.

VIGER, F. AND LATAPY, M. 2005. Efficient and simple generation of random simple connected graphs with prescribed degree sequence. In *Proceedings of the 11th Annual International Conference on Computing and Combinatorics*. COCOON'05. Springer-Verlag, Berlin, Heidelberg, 440–449.

VONDRÁK, J., CHEKURI, C., AND ZENKLUSEN, R. 2011. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*. STOC '11. ACM, New York, NY, USA, 783–792.

WATTS, D. AND STROGATZ, S. 1998. Collective dynamics of 'small-world' networks. *Nature 393*.

YANG, J. AND COUNTS, S. 2010. Predicting the speed, scale, and range of information diffusion in twitter. In *ICWSM*.