Maximizing the Spread of Influence through a Social Network

David Kempe*
Dept. of Computer Science
Cornell University, Ithaca NY
kempe@cs.cornell.edu

Jon Kleinberg†
Dept. of Computer Science
Cornell University, Ithaca NY
kleinber@cs.cornell.edu

Éva Tardos‡
Dept. of Computer Science
Cornell University, Ithaca NY
eva@cs.cornell.edu

ABSTRACT

Models for the processes by which ideas and influence propagate through a social network have been studied in a number of domains, including the diffusion of medical and technological innovations, the sudden and widespread adoption of various strategies in game-theoretic settings, and the effects of "word of mouth" in the promotion of new products. Recently, motivated by the design of viral marketing strategies, Domingos and Richardson posed a fundamental algorithmic problem for such social network processes: if we can try to convince a subset of individuals to adopt a new product or innovation, and the goal is to trigger a large cascade of further adoptions, which set of individuals should we target?

We consider this problem in several of the most widely studied models in social network analysis. The optimization problem of selecting the most influential nodes is NP-hard here, and we provide the first provable approximation guarantees for efficient algorithms. Using an analysis framework based on submodular functions, we show that a natural greedy strategy obtains a solution that is provably within 63% of optimal for several classes of models; our framework suggests a general approach for reasoning about the performance guarantees of algorithms for these types of influence problems in social networks.

We also provide computational experiments on large collaboration networks, showing that in addition to their provable guarantees, our approximation algorithms significantly out-perform node-selection heuristics based on the well-studied notions of degree centrality and distance centrality from the field of social networks.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Non-numerical Algorithms and Problems

*Supported by an Intel Graduate Fellowship and an NSF Graduate Research Fellowship.

†Supported in part by a David and Lucile Packard Foundation Fellowship and NSF ITR/IM Grant IIS-0081334.

‡Supported in part by NSF ITR grant CCR-011337, and ONR grant N00014-98-1-0589.

Keywords

approximation algorithms, social networks, viral marketing, diffusion of innovations

1. INTRODUCTION

A social network — the graph of relationships and interactions within a group of individuals — plays a fundamental role as a medium for the spread of information, ideas, and influence among its members. An idea or innovation will appear — for example, the use of cell phones among college students, the adoption of a new drug within the medical profession, or the rise of a political movement in an unstable society — and it can either die out quickly or make significant inroads into the population. If we want to understand the extent to which such ideas are adopted, it can be important to understand how the dynamics of adoption are likely to unfold within the underlying social network: the extent to which people are likely to be affected by decisions of their friends and colleagues, or the extent to which "word-of-mouth" effects will take hold. Such network diffusion processes have a long history of study in the social sciences. Some of the earliest systematic investigations focused on data pertaining to the adoption of medical and agricultural innovations in both developed and developing parts of the world [8, 27, 29]; in other contexts, research has investigated diffusion processes for "word-of-mouth" and "viral marketing" effects in the success of new products [4, 7, 10, 13, 14, 20, 26], the sudden and widespread adoption of various strategies in game-theoretic settings [6, 12, 21, 32, 33], and the problem of cascading failures in power systems [2, 3].

In recent work, motivated by applications to marketing, Domingos and Richardson posed a fundamental algorithmic problem for such systems [10, 26]. Suppose that we have data on a social network, with estimates for the extent to which individuals influence one another, and we would like to market a new product that we hope will be adopted by a large fraction of the network. The premise of viral marketing is that by initially targeting a few "influential" members of the network — say, giving them free samples of the product — we can trigger a cascade of influence by which friends will recommend the product to other friends, and many individuals will ultimately try it. But how should we choose the few key individuals to use for seeding this process? In [10, 26], this question was considered in a probabilistic model of interaction; heuristics were given for choosing customers with a large overall effect on the network, and methods were also developed to infer the influence data necessary for posing these types of problems.

In this paper, we consider the issue of choosing influential sets of individuals as a problem in discrete optimization. The optimal solution is NP-hard for most models that have been studied, including the model of [10]. The framework proposed in [26], on the other hand, is based on a simple linear model where the solution to the optimization problem can be obtained by solving a system of linear equations. Here we focus on a collection of related, NP-hard models that have been extensively studied in the social networks community, and obtain the first provable approximation guarantees for efficient algorithms in a number of general cases. The generality of the models we consider lies between that of the polynomial-time solvable model of [26] and the very general model of [10], where the optimization problem cannot even be approximated to within a non-trivial factor.

We begin by departing somewhat from the Domingos-Richardson framework in the following sense: where their models are essentially *descriptive*, specifying a joint distribution over all nodes' behavior in a global sense, we focus on more *operational* models from mathematical sociology [15, 28] and interacting particle systems [11, 17] that explicitly represent the step-by-step dynamics of adoption. We show that approximation algorithms for maximizing the spread of influence in these models can be developed in a general framework based on *submodular functions* [9, 23]. We also provide computational experiments on large collaboration networks, showing that in addition to their provable guarantees, our algorithms significantly out-perform node-selection heuristics based on the well-studied notions of *degree centrality* and *distance centrality* [30] from the field of social network analysis.

Two Basic Diffusion Models. In considering operational models for the spread of an idea or innovation through a social network G, represented by a directed graph, we will speak of each individual node as being either *active* (an adopter of the innovation) or *inactive*. We will focus on settings, guided by the motivation discussed above, in which each node's tendency to become active increases monotonically as more of its neighbors become active. Also, we will focus for now on the *progressive* case in which nodes can switch from being inactive to being active, but do not switch in the other direction; it turns out that this assumption can easily be lifted later. Thus, the process will look roughly as follows from the perspective of an initially inactive node v: as time unfolds, more and more of v's neighbors become active; at some point, this may cause v to become active, and v's decision may in turn trigger further decisions by nodes to which v is connected.

Granovetter and Schelling were among the first to propose models that capture such a process; their approach was based on the use of node-specific *thresholds* [15, 28]. Many models of this flavor have since been investigated (see e.g. [5, 15, 18, 19, 21, 25, 28, 29, 31, 32, 33])but the following *Linear Threshold Model* lies at the core of most subsequent generalizations. In this model, a node v is influenced by each neighbor w according to a *weight* $b_{v,w}$ such that $\sum_{w \text{ neighbor of } v} b_{v,w} \leq 1$. The dynamics of the process then proceed as follows. Each node v chooses a *threshold* θ_v uniformly at random from the interval $[0, 1]$; this represents the weighted fraction of v's neighbors that must become active in order for v to become active. Given a random choice of thresholds, and an initial set of active nodes A_0 (with all other nodes inactive), the diffusion process unfolds deterministically in discrete *steps*: in step t, all nodes that were active in step $t-1$ remain active, and we activate any node v for which the total weight of its active neighbors is at least θ_v:

$$\sum_{w \text{ active neighbor of } v} b_{v,w} \quad \geq \quad \theta_v.$$

Thus, the thresholds θ_v intuitively represent the different latent tendencies of nodes to adopt the innovation when their neighbors do; the fact that these are randomly selected is intended to model our lack of knowledge of their values — we are in effect averaging over possible threshold values for all the nodes. (Another class of approaches hard-wires all thresholds at a known value like $1/2$; see for example work by Berger [5], Morris [21], and Peleg [25].)

Based on work in interacting particle systems [11, 17] from probability theory, we can also consider dynamic *cascade* models for diffusion processes. The conceptually simplest model of this type is what one could call the *Independent Cascade Model*, investigated recently in the context of marketing by Goldenberg, Libai, and Muller [13, 14]. We again start with an initial set of active nodes A_0, and the process unfolds in discrete steps according to the following randomized rule. When node v first becomes active in step t, it is given a single chance to activate each currently inactive neighbor w; it succeeds with a probability $p_{v,w}$ — a parameter of the system — independently of the history thus far. (If w has multiple newly activated neighbors, their attempts are sequenced in an arbitrary order.) If v succeeds, then w will become active in step $t+1$; but whether or not v succeeds, it cannot make any further attempts to activate w in subsequent rounds. Again, the process runs until no more activations are possible.

The Linear Threshold and Independent Cascade Models are two of the most basic and widely-studied diffusion models, but of course many extensions can be considered. We will turn to this issue later in the paper, proposing a general framework that simultaneously includes both of these models as special cases. For the sake of concreteness in the introduction, we will discuss our results in terms of these two models in particular.

Approximation Algorithms for Influence Maximization. We are now in a position to formally express the Domingos-Richardson style of optimization problem — choosing a good initial set of nodes to target — in the context of the above models. Both the Linear Threshold and Independent Cascade Models (as well as the generalizations to follow) involve an initial set of active nodes A_0 that start the diffusion process. We define the *influence* of a set of nodes A, denoted $\sigma(A)$, to be the expected number of active nodes at the end of the process, given that A is this initial active set A_0. The *influence maximization problem* asks, for a parameter k, to find a k-node set of maximum influence. (When dealing with algorithms for this problem, we will say that the chosen set A of k initial active nodes has been *targeted* for activation by the algorithm.) For the models we consider, it is NP-hard to determine the optimum for influence maximization, as we will show later.

Our first main result is that the optimal solution for influence maximization can be efficiently approximated to within a factor of $(1 - 1/e - \varepsilon)$, in both the Linear Threshold and Independent Cascade models; here e is the base of the natural logarithm and ε is any positive real number. (Thus, this is a performance guarantee slightly better than 63%.) The algorithm that achieves this performance guarantee is a natural greedy hill-climbing strategy related to the approach considered in [10], and so the main content of this result is the analysis framework needed for obtaining a provable performance guarantee, and the fairly surprising fact that hill-climbing is always within a factor of at least 63% of optimal for this problem. We prove this result in Section 2 using techniques from the theory of submodular functions [9, 23], which we describe in detail below, and which turn out to provide a natural context for reasoning about both models and algorithms for influence maximization.

In fact, this analysis framework allows us to design and prove guarantees for approximation algorithms in much richer and more realistic models of the processes by which we market to nodes. The deterministic activation of individual nodes is a highly simplified model; an issue also considered in [10, 26] is that we may in reality have a large number of different marketing actions available, each of which may influence nodes in different ways. The available budget can be divided arbitrarily between these actions. We show how to extend the analysis to this substantially more general framework. Our main result here is that a generalization of the hill-climbing algorithm still provides approximation guarantees arbitrarily close to $(1 - 1/e)$.

It is worth briefly considering the general issue of performance guarantees for algorithms in these settings. For both the Linear Threshold and the Independent Cascade models, the influence maximization problem is NP-complete, but it can be approximated well. In the linear model of Richardson and Domingos [26], on the other hand, both the propagation of influence *as well as* the effect of the initial targeting are linear. Initial marketing decisions here are thus limited in their effect on node activations; each node's probability of activation is obtained as a linear combination of the effect of targeting and the effect of the neighbors. In this fully linear model, the influence can be maximized by solving a system of linear equations. In contrast, we can show that general models like that of Domingos and Richardson [10], and even simple models that build in a fixed threshold (like $1/2$) at all nodes [5, 21, 25], lead to influence maximization problems that cannot be approximated to within any non-trivial factor, assuming P $\neq$ NP. Our analysis of approximability thus suggests a way of tracing out a more delicate boundary of tractability through the set of possible models, by helping to distinguish among those for which simple heuristics provide strong performance guarantees and those for which they can be arbitrarily far from optimal. This in turn can suggest the development of both more powerful algorithms, and the design of accurate models that simultaneously allow for tractable optimization.

Following the approximation and NP-hardness results, we describe in Section 3 the results of computational experiments with both the Linear Threshold and Independent Cascade Models, showing that the hill-climbing algorithm significantly out-performs strategies based on targeting high-degree or "central" nodes [30]. In Section 4 we then develop a general model of diffusion processes in social networks that simultaneously generalizes the Linear Threshold and Independent Cascade Models, as well as a number of other natural cases, and we show how to obtain approximation guarantees for a large sub-class of these models. In Sections 5 and 6, we also consider extensions of our approximation algorithms to models with more realistic scenarios in mind: more complex marketing actions as discussed above, and *non-progressive* processes, in which active nodes may become inactive in subsequent steps.

2. APPROXIMATION GUARANTEES IN THE INDEPENDENT CASCADE AND LINEAR THRESHOLD MODELS

The overall approach. We begin by describing our strategy for proving approximation guarantees. Consider an arbitrary function $f(\cdot)$ that maps subsets of a finite ground set U to non-negative real numbers.[1] We say that f is *submodular* if it satisfies a natural "diminishing returns" property: the marginal gain from adding an element to a set S is at least as high as the marginal gain from adding the same element to a superset of S. Formally, a submodular function satisfies

$$f(S \cup \{v\}) - f(S) \quad \geq \quad f(T \cup \{v\}) - f(T),$$

for all elements v and all pairs of sets $S \subseteq T$.

Submodular functions have a number of very nice tractability properties; the one that is relevant to us here is the following. Suppose we have a function f that is submodular, takes only non-negative values, and is *monotone* in the sense that adding an element to a set cannot cause f to decrease: $f(S \cup \{v\}) \geq f(S)$ for all elements v and sets S. We wish to find a k-element set S for which $f(S)$ is maximized. This is an NP-hard optimization problem (it can be shown to contain the Hitting Set problem as a simple special case), but a result of Nemhauser, Wolsey, and Fisher [9, 23] shows that the following greedy hill-climbing algorithm approximates the optimum to within a factor of $(1 - 1/e)$ (where e is the base of the natural logarithm): start with the empty set, and repeatedly add an element that gives the maximum marginal gain.

THEOREM 2.1. *[9, 23] For a non-negative, monotone submodular function f, let S be a set of size k obtained by selecting elements one at a time, each time choosing an element that provides the largest marginal increase in the function value. Let S^* be a set that maximizes the value of f over all k-element sets. Then $f(S) \geq (1-1/e) \cdot f(S^*)$; in other words, S provides a $(1-1/e)$-approximation.*

Due to its generality, this result has found applications in a number of areas of discrete optimization (see e.g. [22]); the only direct use of it that we are aware of in the databases and data mining literature is in a context very different from ours, for the problem of selecting database views to materialize [16].

Our strategy will be to show that for the models we are considering, the resulting influence function $\sigma(\cdot)$ is submodular. A subtle difficulty lies in the fact that the result of Nemhauser et al. assumes that the greedy algorithm can evaluate the underlying function exactly, which may not be the case for the influence function $\sigma(A)$. However, by simulating the diffusion process and sampling the resulting active sets, we are able to obtain arbitrarily close approximations to $\sigma(A)$, with high probability. Furthermore, one can extend the result of Nemhauser et al. to show that for any $\varepsilon > 0$, there is a $\gamma > 0$ such that by using $(1 + \gamma)$-approximate values for the function to be optimized, we obtain a $(1-1/e-\varepsilon)$-approximation.

As mentioned in the introduction, we can extend this analysis to a general model with more complex *marketing actions* that can have a probabilistic effect on the initial activation of nodes. We show in Section 6 how, with a more careful hill-climbing algorithm and a generalization of Theorem 2.1, we can obtain comparable approximation guarantees in this setting.

A further extension is to assume that each node v has an associated non-negative *weight* w_v, capturing how important it is that v be activated in the final outcome. (For instance, if we are marketing textbooks to college teachers, then the weight could be the number of students in the teacher's class, resulting in a larger or smaller number of sales.) If we let B denote the (random) set activated by the process with initial activation A, then we can define the weighted influence function $\sigma_w(A)$ to be the expected value over outcomes B of the quantity $\sum_{v \in B} w_v$. The influence function studied above is the special case obtained by setting $w_v = 1$ for all nodes v. The objective function with weights is submodular whenever the unweighted version is, so we can still use the greedy algorithm for obtaining a $(1-1/e-\varepsilon)$-approximation. Note, however, that a sampling algorithm to approximately choose the next element may need time that depends on the sizes of the weights.

[1]Note that the influence function $\sigma(\cdot)$ defined above has this form; it maps each subset A of the nodes of the social network to a real number denoting the expected size of the activated set if A is targeted for initial activation.

Independent Cascade

In view of the above discussion, an approximation guarantee for influence maximization in the Independent Cascade Model will be a consequence of the following

THEOREM 2.2. *For an arbitrary instance of the Independent Cascade Model, the resulting influence function $\sigma(\cdot)$ is submodular.*

In order to establish this result, we need to look, implicitly or explicitly, at the expression $\sigma(A \cup \{v\}) - \sigma(A)$, for arbitrary sets A and elements v. In other words, what increase do we get in the expected number of overall activations when we add v to the set A? This increase is very difficult to analyze directly, because it is hard to work with quantities of the form $\sigma(A)$. For example, the Independent Cascade process is underspecified, since we have not prescribed the order in which newly activated nodes in a given step t will attempt to activate their neighbors. Thus, it is not initially obvious that the process is even well-defined, in the sense that it yields the same distribution over outcomes regardless of how we schedule the attempted activations.

Our proof deals with these difficulties by formulating an equivalent view of the process, which makes it easier to see that there is an order-independent outcome, and which provides an alternate way to reason about the submodularity property.

Consider a point in the cascade process when node v has just become active, and it attempts to activate its neighbor w, succeeding with probability $p_{v,w}$. We can view the outcome of this random event as being determined by flipping a coin of bias $p_{v,w}$. From the point of view of the process, it clearly does not matter whether the coin was flipped at the moment that v became active, or whether it was flipped at the very beginning of the whole process and is only being revealed now. Continuing this reasoning, we can in fact assume that for *each* pair of neighbors (v, w) in the graph, a coin of bias $p_{v,w}$ is flipped at the very beginning of the process (independently of the coins for all other pairs of neighbors), and the result is stored so that it can be later checked *in the event* that v is activated while w is still inactive.

With all the coins flipped in advance, the process can be viewed as follows. The edges in G for which the coin flip indicated an activation will be successful are declared to be *live*; the remaining edges are declared to be *blocked*. If we fix the outcomes of the coin flips and then initially activate a set A, it is clear how to determine the full set of active nodes at the end of the cascade process:

CLAIM 2.3. *A node x ends up active if and only if there is a path from some node in A to x consisting entirely of live edges. (We will call such a path a* live-edge path*.)*

Consider the probability space in which each sample point specifies one possible set of outcomes for all the coin flips on the edges. Let X denote one sample point in this space, and define $\sigma_X(A)$ to be the total number of nodes activated by the process when A is the set initially targeted, and X is the set of outcomes of all coin flips on edges. Because we have fixed a choice for X, $\sigma_X(A)$ is in fact a deterministic quantity, and there is a natural way to express its value, as follows. Let $R(v, X)$ denote the set of all nodes that can be reached from v on a path consisting entirely of live edges. By Claim 2.3, $\sigma_X(A)$ is the number of nodes that can be reached on live-edge paths from *any* node in A, and so it is equal to the cardinality of the union $\cup_{v \in A} R(v, X)$.

Proof of Theorem 2.2. First, we claim that for each fixed outcome X, the function $\sigma_X(\cdot)$ is submodular. To see this, let S and T be two sets of nodes such that $S \subseteq T$, and consider the quantity $\sigma_X(S \cup \{v\}) - \sigma_X(S)$. This is the number of elements in $R(v, X)$ that are not already in the union $\cup_{u \in S} R(u, X)$; it is at least as large as the number of elements in $R(v, X)$ that are not in the (bigger) union $\cup_{u \in T} R(u, X)$. It follows that $\sigma_X(S \cup \{v\}) - \sigma_X(S) \geq \sigma_X(T \cup \{v\}) - \sigma_X(T)$, which is the defining inequality for submodularity. Finally, we have

$$\sigma(A) \quad = \sum_{\text{outcomes } X} \text{Prob}[X] \cdot \sigma_X(A),$$

since the expected number of nodes activated is just the weighted average over all outcomes. But a non-negative linear combination of submodular functions is also submodular, and hence $\sigma(\cdot)$ is submodular, which concludes the proof. ■

Next we show the hardness of influence maximization.

THEOREM 2.4. *The influence maximization problem is NP-hard for the Independent Cascade model.*

Proof. Consider an instance of the NP-complete *Set Cover* problem, defined by a collection of subsets $S_1, S_2, \ldots, S_m$ of a ground set $U = \{u_1, u_2, \ldots, u_n\}$; we wish to know whether there exist k of the subsets whose union is equal to U. (We can assume that $k < n < m$.) We show that this can be viewed as a special case of the influence maximization problem.

Given an arbitrary instance of the Set Cover problem, we define a corresponding directed bipartite graph with $n + m$ nodes: there is a node i corresponding to each set S_i, a node j corresponding to each element u_j, and a directed edge (i, j) with activation probability $p_{i,j} = 1$ whenever $u_j \in S_i$. The Set Cover problem is equivalent to deciding if there is a set A of k nodes in this graph with $\sigma(A) \geq n + k$. Note that for the instance we have defined, activation is a deterministic process, as all probabilities are 0 or 1. Initially activating the k nodes corresponding to sets in a Set Cover solution results in activating all n nodes corresponding to the ground set U, and if any set A of k nodes has $\sigma(A) \geq n + k$, then the Set Cover problem must be solvable. ■

Linear Thresholds

We now prove an analogous result for the Linear Threshold Model.

THEOREM 2.5. *For an arbitrary instance of the Linear Threshold Model, the resulting influence function $\sigma(\cdot)$ is submodular.*

Proof. The analysis is a bit more intricate than in the proof of Theorem 2.2, but the overall argument has a similar structure. In the proof of Theorem 2.2, we constructed an equivalent process by initially resolving the outcomes of some random choices, considering each outcome in isolation, and then averaging over all outcomes. For the Linear Threshold Model, the simplest analogue would be to consider the behavior of the process *after* all node thresholds have been chosen. Unfortunately, for a fixed choice of thresholds, the number of activated nodes is not in general a submodular function of the targeted set; this fact necessitates a more subtle analysis.

Recall that each node v has an influence weight $b_{v,w} \geq 0$ from each of its neighbors w, subject to the constraint that $\sum_w b_{v,w} \leq 1$. (We can extend the notation by writing $b_{v,w} = 0$ when w is not a neighbor of v.) Suppose that v picks at most one of its incoming edges at random, selecting the edge from w with probability $b_{v,w}$ and selecting no edge with probability $1 - \sum_w b_{v,w}$. The selected edge is declared to be "live," and all other edges are declared to be "blocked." (Note the contrast with the proof of Theorem 2.2: there, we determined whether an edge was live independently of the decision for each other edge; here, we negatively correlate the decisions so that at most one live edge enters each node.)

The crux of the proof lies in establishing Claim 2.6 below, which asserts that the Linear Threshold model is equivalent to reachability via live-edge paths as defined above. Once that equivalence is established, submodularity follows exactly as in the proof of Theorem 2.2. We can define $R(v, X)$ as before to be the set of all nodes reachable from v on live-edge paths, subject to a choice X of live/blocked designations for all edges; it follows that $\sigma_X(A)$ is the cardinality of the union $\cup_{v\in A} R(v, X)$, and hence a submodular function of A; finally, the function $\sigma(\cdot)$ is a non-negative linear combination of the functions $\sigma_X(\cdot)$ and hence also submodular. ■

CLAIM 2.6. *For a given targeted set A, the following two distributions over sets of nodes are the same:*

(i) The distribution over active sets obtained by running the Linear Threshold process to completion starting from A; and

(ii) The distribution over sets reachable from A via live-edge paths, under the random selection of live edges defined above.

Proof. We need to prove that reachability under our random choice of live and blocked edges defines a process equivalent to that of the Linear Threshold Model. To obtain intuition about this equivalence, it is useful to first analyze the special case in which the underlying graph G is directed and acyclic. In this case, we can fix a topological ordering of the nodes $v_1, v_2, \ldots, v_n$ (so that all edges go from earlier nodes to later nodes in the order), and build up the distribution of active sets by following this order. For each node v_i, suppose we already have determined the distribution over active subsets of its neighbors. Then under the Linear Threshold process, the probability that v_i will become active, given that a subset S_i of its neighbors is active, is $\sum_{w\in S_i} b_{v_i,w}$. This is precisely the probability that the live incoming edge selected by v_i lies in S_i, and so inductively we see that the two processes define the same distribution over active sets.

To prove the claim generally, consider a graph G that is not acyclic. It becomes trickier to show the equivalence, because there is no natural ordering of the nodes over which to perform induction. Instead, we argue by induction over the iterations of the Linear Threshold process. We define A_t to be the set of active nodes at the end of iteration t, for $t = 0, 1, 2, \ldots$ (note that A_0 is the set initially targeted). If node v has not become active by the end of iteration t, then the probability that it becomes active in iteration $t+1$ is equal to the chance that the influence weights in $A_t \setminus A_{t-1}$ push it over its threshold, *given* that its threshold was not exceeded already; this probability is $\frac{\sum_{u\in A_t\setminus A_{t-1}} b_{v,u}}{1-\sum_{u\in A_{t-1}} b_{v,u}}$.

On the other hand, we can run the live-edge process by revealing the identities of the live edges gradually as follows. We start with the targeted set A. For each node v with at least one edge from the set A, we determine whether v's live edge comes from A. If so, then v is reachable; but if not, we keep the source of v's live edge unknown, subject to the condition that it comes from outside A. Having now exposed a new set of reachable nodes A'_1 in the first stage, we proceed to identify further reachable nodes by performing the same process on edges from A'_1, and in this way produce sets $A'_2, A'_3, \ldots$. If node v has not been determined to be reachable by the end of stage t, then the probability that it is determined to be reachable in stage $t+1$ is equal to the chance that its live edge comes from $A_t \setminus A_{t-1}$, given that its live edge has not come from any of the earlier sets. But this is $\frac{\sum_{u\in A_t\setminus A_{t-1}} b_{v,u}}{1-\sum_{u\in A_{t-1}} b_{v,u}}$, which is the same as in the Linear Threshold process of the previous paragraph. Thus, by induction over these stages, we see that the live-edge process produces the same distribution over active sets as the Linear Threshold process. ■

Influence maximization is hard in this model as well.

THEOREM 2.7. *The influence maximization problem is NP-hard for the Linear Threshold model.*

Proof. Consider an instance of the NP-complete *Vertex Cover* problem defined by an undirected n-node graph $G = (V, E)$ and an integer k; we want to know if there is a set S of k nodes in G so that every edge has at least one endpoint in S. We show that this can be viewed as a special case of the influence maximization problem.

Given an instance of the *Vertex Cover* problem involving a graph G, we define a corresponding instance of the influence maximization problem by directing all edges of G in both directions. If there is a vertex cover S of size k in G, then one can deterministically make $\sigma(A) = n$ by targeting the nodes in the set $A = S$; conversely, this is the only way to get a set A with $\sigma(A) = n$. ■

In the proofs of both the approximation theorems in this section, we established submodularity by considering an equivalent process in which each node "hard-wired" certain of its incident edges as transmitting influence from neighbors. This turns out to be a proof technique that can be formulated in general terms, and directly applied to give approximability results for other models as well. We discuss this further in the context of the general framework presented in Section 4.

3. EXPERIMENTS

In addition to obtaining worst-case guarantees on the performance of our approximation algorithm, we are interested in understanding its behavior in practice, and comparing its performance to other heuristics for identifying influential individuals. We find that our greedy algorithm achieves significant performance gains over several widely-used structural measures of influence in social networks [30].

The Network Data. For evaluation, it is desirable to use a network dataset that exhibits many of the structural features of large-scale social networks. At the same time, we do not address the issue of inferring actual influence parameters from network observations (see e.g. [10, 26]). Thus, for our testbed, we employ a collaboration graph obtained from co-authorships in physics publications, with simple settings of the influence parameters. It has been argued extensively that co-authorship networks capture many of the key features of social networks more generally [24]. The co-authorship data was compiled from the complete list of papers in the high-energy physics theory section of the e-print arXiv (www.arxiv.org).[2]

The collaboration graph contains a node for each researcher who has at least one paper with co-author(s) in the arXiv database. For each paper with two or more authors, we inserted an edge for each pair of authors (single-author papers were ignored). Notice that this results in parallel edges when two researchers have co-authored multiple papers — we kept these parallel edges as they can be interpreted to indicate stronger social ties between the researchers involved. The resulting graph has 10748 nodes, and edges between about 53000 pairs of nodes.

[2] We also ran experiments on the co-authorship graphs induced by theoretical computer science papers. We do not report on the results here, as they are very similar to the ones for high-energy physics.

While processing the data, we corrected many common types of mistakes automatically or manually. In order to deal with aliasing problems at least partially, we abbreviated first names, and unified spellings for foreign characters. We believe that the resulting graph is a good approximation to the actual collaboration graph (the sheer volume of data prohibits a complete manual cleaning pass).

The Influence Models. We compared the algorithms in three different models of influence. In the linear threshold model, we treated the multiplicity of edges as weights. If nodes u, v have $c_{u,v}$ parallel edges between them, and degrees d_u and d_v, then the edge (u, v) has weight $\frac{c_{u,v}}{d_v}$, and the edge (v, u) has weight $\frac{c_{u,v}}{d_u}$.

In the independent cascade model, we assigned a uniform probability of p to each edge of the graph, choosing p to be 1% and 10% in separate trials. If nodes u and v have $c_{u,v}$ parallel edges, then we assume that for each of those $c_{u,v}$ edges, u has a chance of p to activate v, i.e. u has a total probability of $1 - (1 - p)^{c_{u,v}}$ of activating v once it becomes active.

The independent cascade model with uniform probabilities p on the edges has the property that high-degree nodes not only have a chance to influence many other nodes, but also to be influenced by them. Whether or not this is a desirable interpretation of the influence data is an application-specific issue. Motivated by this, we chose to also consider an alternative interpretation, where edges into high-degree nodes are assigned smaller probabilities. We study a special case of the Independent Cascade Model that we term "weighted cascade", in which each edge from node u to v is assigned probability $1/d_v$ of activating v. The weighted cascade model resembles the linear threshold model in that the expected number of neighbors who would succeed in activating a node v is 1 in both models.

The algorithms and implementation. We compare our greedy algorithm with heuristics based on nodes' degrees and centrality within the network, as well as the crude baseline of choosing random nodes to target. The degree and centrality-based heuristics are commonly used in the sociology literature as estimates of a node's influence [30].

The high-degree heuristic chooses nodes v in order of decreasing degrees d_v. Considering high-degree nodes as influential has long been a standard approach for social and other networks [30, 1], and is known in the sociology literature as "degree centrality".

"Distance centrality" is another commonly used influence measure in sociology, building on the assumption that a node with short paths to other nodes in a network will have a higher chance of influencing them. Hence, we select nodes in order of increasing average distance to other nodes in the network. As the arXiv collaboration graph is not connected, we assigned a distance of n — the number of nodes in the graph — for any pair of unconnected nodes. This value is significantly larger than any actual distance, and thus can be considered to play the role of an infinite distance. In particular, nodes in the largest connected component will have smallest average distance.

Finally, we consider, as a baseline, the result of choosing nodes uniformly at random. Notice that because the optimization problem is NP-hard, and the collaboration graph is prohibitively large, we cannot compute the optimum value to verify the *actual* quality of approximations.

Both in choosing the nodes to target with the greedy algorithm, and in evaluating the performance of the algorithms, we need to compute the value $\sigma(A)$. It is an open question to compute this quantity exactly by an efficient method, but very good estimates can be obtained by simulating the random process. More specifically, we simulate the process 10000 times for each targeted set, re-choosing thresholds or edge outcomes pseudo-randomly from $[0, 1]$ every time. Previous runs indicate that the quality of approximation after 10000 iterations is comparable to that after 300000 or more iterations.

The results. Figure 1 shows the performance of the algorithms in the linear threshold model. The greedy algorithm outperforms the high-degree node heuristic by about 18%, and the central node heuristic by over 40%. (As expected, choosing random nodes is not a good idea.) This shows that significantly better marketing results can be obtained by explicitly considering the dynamics of information in a network, rather than relying solely on structural properties of the graph.

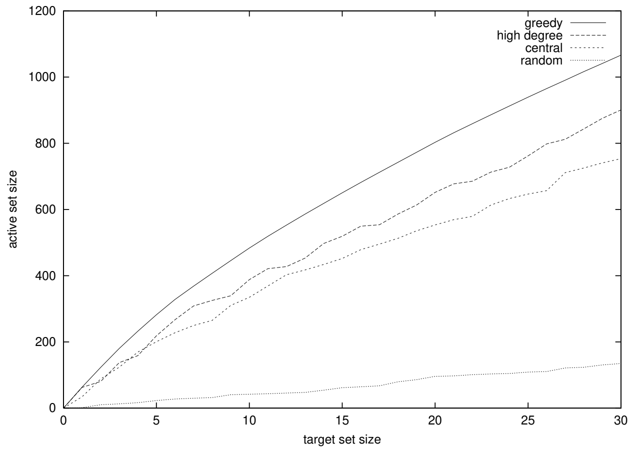

Figure 1: Results for the linear threshold model

When investigating the reason why the high-degree and centrality heuristics do not perform as well, one sees that they ignore such network effects. In particular, neither of the heuristics incorporates the fact that many of the most central (or highest-degree) nodes may be clustered, so that targeting all of them is unnecessary. In fact, the uneven nature of these curves suggests that the network influence of many nodes is not accurately reflected by their degree or centrality.

Figure 2 shows the results for the weighted cascade model. Notice the striking similarity to the linear threshold model. The scale is slightly different (all values are about 25% smaller), but the behavior is qualitatively the same, even with respect to the exact nodes whose network influence is not reflected accurately by their degree or centrality. The reason is that in expectation, each node is influenced by the same number of other nodes in both models (see Section 2), and the degrees are relatively concentrated around their expectation of 1.

The graph for the independent cascade model with probability 1%, given in Figure 3, seems very similar to the previous two at first glance. Notice, however, the very different scale: on average, each targeted node only activates three additional nodes. Hence, the network effects in the independent cascade model with very small probabilities are much weaker than in the other models. Several nodes have degrees well exceeding 100, so the probabilities on their incoming edges are even smaller than 1% in the weighted cascade model. This suggests that the network effects observed for the linear threshold and weighted cascade models rely heavily on low-degree nodes as multipliers, even though targeting high-degree

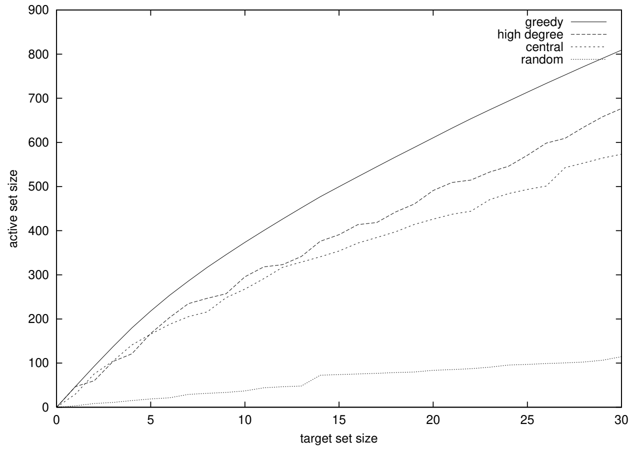

Figure 2: Results for the weighted cascade model

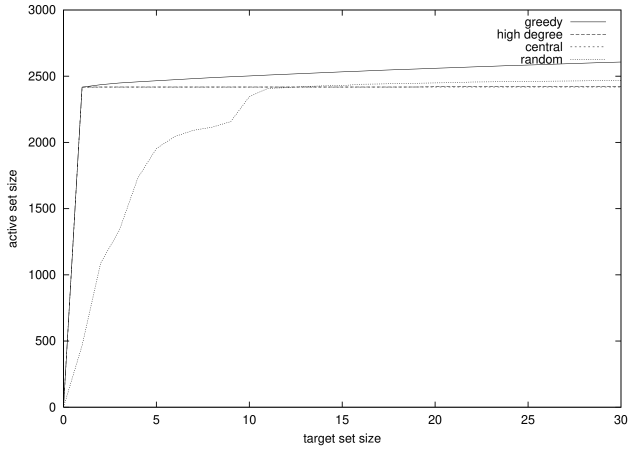

Figure 4: Independent cascade model with probability 10%

nodes is a reasonable heuristic. Also notice that in the independent cascade model, the heuristic of choosing random nodes performs significantly better than in the previous two models.

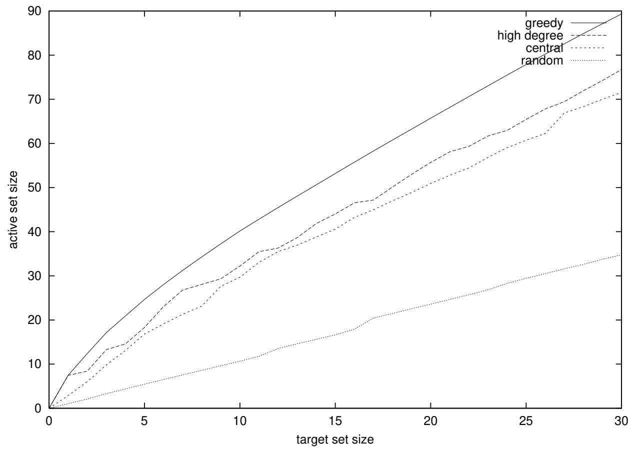

Figure 3: Independent cascade model with probability 1%

The improvement in performance of the "random nodes" heuristic is even more pronounced for the independent cascade model with probabilities equal to 10%, depicted in Figure 4. In that model, it starts to outperform both the high-degree and the central nodes heuristics when more than 12 nodes are targeted. It is initially surprising that random targeting for this model should lead to more activations than centrality-based targeting, but in fact there is a natural underlying reason that we explore now.

The first targeted node, if chosen somewhat judiciously, will activate a large fraction of the network, in our case almost 25%. However, any additional nodes will only reach a small additional fraction of the network. In particular, other central or high-degree nodes are very likely to be activated by the initially chosen one, and thus have hardly any marginal gain. This explains the shapes of the curves for the high-degree and centrality heuristics, which leap up to about 2415 activated nodes, but make virtually no progress afterwards. The greedy algorithm, on the other hand, takes the effect of the first chosen node into account, and targets nodes with smaller marginal gain afterwards. Hence, its active set keeps growing, although at a much smaller slope than in other models.

The random heuristic does not do as well initially as the other heuristics, but with sufficiently many attempts, it eventually hits some highly influential nodes and becomes competitive with the centrality-based node choices. Because it does not focus exclusively on central nodes, it eventually targets nodes with additional marginal gain, and surpasses the two centrality-based heuristics.

4. A GENERAL FRAMEWORK FOR INFLUENCE MAXIMIZATION

General Threshold and Cascade Models. We have thus far been considering two specific, widely studied models for the diffusion of influence. We now propose a broader framework that simultaneously generalizes these two models, and allows us to explore the limits of models in which strong approximation guarantees can be obtained. Our general framework has equivalent formulations in terms of thresholds and cascades, thereby unifying these two views of diffusion through a social network.

- **A general threshold model.** We would like to be able to express the notion that a node v's decision to become active can be based on an arbitrary monotone function of the set of neighbors of v that are already active. Thus, associated with v is a monotone *threshold function* f_v that maps subsets of v's neighbor set to real numbers in $[0, 1]$, subject to the condition that $f_v(\emptyset) = 0$. The diffusion process follows the general structure of the Linear Threshold Model. Each node v initially chooses θ_v uniformly at random from the interval $[0, 1]$. Now, however, v becomes active in step t if $f_v(S) \geq \theta_v$, where S is the set of neighbors of v that are active in step $t - 1$. Note that the Linear Threshold Model is the special case in which each threshold function has the form $f_v(S) = \sum_{u \in S} b_{v,u}$ for parameters $b_{v,u}$ such that $\sum_{u \text{ neighbor of } v} b_{v,u} \leq 1$.
- **A general cascade model.** We generalize the cascade model to allow the probability that u succeeds in activating a neighbor v to depend on the set of v's neighbors that have already tried. Thus, we define an *incremental function* $p_v(u, S) \in [0, 1]$, where S and $\{u\}$ are disjoint subsets of v's neighbor

set. A general cascade process works by analogy with the independent case: in the general case, when u attempts to activate v, it succeeds with probability $p_v(u, S)$, where S is the set of neighbors that have *already* tried (and failed) to activate v. The Independent Cascade Model is the special case where $p_v(u, S)$ is a constant $p_{u,v}$, independent of S. We will only be interested in cascade models defined by incremental functions that are *order-independent* in the following sense: if neighbors $u_1, u_2, \ldots, u_\ell$ try to activate v, then the probability that v is activated at the end of these ℓ attempts does not depend on the order in which the attempts are made.

These two models are equivalent, and we give a method to convert between them. First, consider an instance of the general threshold model with threshold functions f_v. To define an equivalent cascade model, we need to understand the probability that an additional neighbor u can activate v, given that the nodes in a set S have already tried and failed. If the nodes in S have failed, then node v's threshold θ_v must be in the range $\theta_v \in (f_v(S), 1]$. However, subject to this constraint, it is uniformly distributed.Thus, the probability that a neighbor $u \notin S$ succeeds in activating v, given that the nodes in S have failed, is

$$p_v(u, S) = \frac{f_v(S \cup \{u\}) - f_v(S)}{1 - f_v(S)}.$$

It is not difficult to show that the cascade process with these functions is equivalent to the original threshold process.

Conversely, consider a node v in the cascade model, and a set $S = \{u_1, \ldots, u_k\}$ of its neighbors. Assume that the nodes in S try to activate v in the order $u_1, \ldots, u_k$, and let $S_i = \{u_1, \ldots, u_i\}$. Then the probability that v is not activated by this process is by definition $\prod_{i=1}^{k}(1 - p_v(u_i, S_{i-1}))$. Recall that we assumed that the order in which the u_i try to activate v does not affect their overall success probability. Hence, this value depends on the set S only, and we can define $f_v(S) = 1 - \prod_{i=1}^{k}(1 - p_v(u_i, S_{i-1}))$. Analogously, one can show that this instance of the threshold model is equivalent to the original cascade process.

An Inapproximability Result. The general model proposed above includes large families of instances for which the influence function $\sigma(\cdot)$ is not submodular. Indeed, it may become NP-hard to approximate the optimization problem to within any non-trivial factor.

THEOREM 4.1. *In general, it is NP-hard to approximate the influence maximization problem to within a factor of $n^{1-\varepsilon}$, for any $\varepsilon > 0$.*

Proof. To prove this result, we reduce from the *Set Cover* problem. We start with the construction from the proof of Theorem 2.4, letting $u_1, \ldots, u_n$ denote the nodes corresponding to the n elements; i.e. u_i becomes active when at least one of the nodes corresponding to sets containing u_i is active. Next, for an arbitrarily large constant c, we add $N = n^c$ more nodes $x_1, \ldots, x_N$; each x_j is connected to all of the nodes u_i, and it becomes active only when *all* of the u_i are.

If there are at most k sets that cover all elements, then activating the nodes corresponding to these k sets will activate all of the nodes u_i, and thus also all of the x_j. In total, at least $N + n + k$ nodes will be active. Conversely, if there is no set cover of size k, then no targeted set will activate all of the u_i, and hence none of the x_j will become active (unless targeted). In particular, fewer than $n+k$ nodes are active in the end. If an algorithm could approximate the problem within $n^{1-\varepsilon}$ for any ε, it could distinguish between the cases where $N + n + k$ nodes are active in the end, and where fewer than $n + k$ are. But this would solve the underlying instance of Set Cover, and therefore is impossible assuming P $\neq$ NP. ∎

Note that our inapproximability result holds in a very simple model, in which each node is "hard-wired" with a fixed threshold.

Exploring the Boundaries of Approximability. Thus, the general threshold and cascade models are too broad to allow for non-trivial approximation guarantees in their full generality. At the same time, we have seen that the greedy algorithm achieves strong guarantees for some of the main special cases in the social networks literature. How far can we extend these approximability results?

We can generalize the proof technique used in Theorems 2.2 and 2.5 to a model that is less general (and also less natural) than the general threshold and cascade models; however, it includes our special cases from Section 2, and every instance of this model will have a submodular influence function. The model is as follows.

- **The Triggering 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 neighbor in its chosen triggering set T_v that is active at time $t - 1$. (Thus, v's threshold has been replaced by a latent subset of T_v of neighbors whose behavior *actually* affects v.)

It is useful to think of the triggering sets in terms of "live" and "blocked" edges: if node u belongs to the triggering set T_v of v, then we declare the edge (u, v) to be live, and otherwise we declare it to be blocked. As in the proofs of Theorems 2.2 and 2.5, a node v is activated in an instance of the Triggering Model if and only if there is a live-edge path from the initially targeted set A to v. Following the arguments in these proofs, we obtain the following

THEOREM 4.2. *In every instance of the Triggering Model, the influence function $\sigma(\cdot)$ is submodular.*

Beyond the Independent Cascade and Linear Threshold, there are other natural special cases of the Triggering Model. One example is the "Only-Listen-Once" Model. Here, each node v has a parameter p_v so that the first neighbor of v to be activated causes v to become active with probability p_v, and all subsequent attempts to activate v deterministically fail. (In other words, v only listens to the first neighbor that tries to activate it.) This process has an equivalent formulation in the Triggering Set Model, with an edge distribution defined as follows: for any node v, the triggering set T_v is either the entire neighbor set of v (with probability p_v), or the empty set otherwise. As a result, the influence function in the Only-Listen-Once Model is also submodular, and we can obtain a $(1 - 1/e - \varepsilon)$-approximation here as well.

However, we can show that there exist models with submodular influence functions that do not have equivalent formulations in terms of triggering sets, so it makes sense to seek further models in which submodularity holds.

One tractable special case of the cascade model is based on the natural restriction that the probability of a node u influencing v is non-increasing as a function of the set of nodes that have previously tried to influence v. In terms of the cascade model, this means that $p_v(u, S) \geq p_v(u, T)$ whenever $S \subseteq T$. We say that a process satisfying these conditions is an instance of the *Decreasing Cascade Model*. Although there are natural Decreasing Cascade instances that have no equivalent formulation in terms of triggering sets, we can show by a more intricate analysis that every instance of the Decreasing Cascade Model has a submodular influence function. We will include details of this proof in the full version of the paper.

A Conjecture. Finally, we state an appealing conjecture that would include all the approximability results above as special cases.

CONJECTURE 4.3. *Whenever the threshold functions f_v at every node are monotone and submodular, the resulting influence function $\sigma(\cdot)$ is monotone and submodular as well.*

It is not difficult to show that every instance of the Triggering Model has an equivalent formulation with submodular node thresholds. Every instance of the Decreasing Cascade Model has such an equivalent formulation as well; in fact, the Decreasing Cascade condition stands as a very natural special case of the conjecture, given that it too is based on a type of "diminishing returns." When translated into the language of threshold functions, we find that the Decreasing Cascade condition corresponds to the following natural requirement:

$$\frac{f_v(S \cup \{u\}) - f_v(S)}{1 - f_v(S)} \geq \frac{f_v(T \cup \{u\}) - f_v(T)}{1 - f_v(T)},$$

whenever $S \subseteq T$ and $u \notin T$. This is in a sense a "normalized submodularity" property; it is stronger than submodularity, which would consist of the same inequality on just the numerators. (Note that by monotonicity, the denominator on the left is larger.)

5. NON-PROGRESSIVE PROCESSES

We have thus far been concerned with the *progressive* case, in which nodes only go from inactivity to activity, but not vice versa. The *non-progressive* case, in which nodes can switch in both directions, can in fact be reduced to the progressive case.

The non-progressive threshold process is analogous to the progressive model, except that at each step t, each node v chooses a new value $\theta_v^{(t)}$ uniformly at random from the interval $[0, 1]$. Node v will be active in step t if $f_v(S) \geq \theta_v^{(t)}$, where S is the set of neighbors of v that are active in step $t - 1$.

From the perspective of influence maximization, we can ask the following question. Suppose we have a non-progressive model that is going to run for τ steps, and during this process, we are allowed to make up to k *interventions*: for a particular node v, at a particular time $t \leq \tau$, we can target v for activation at time t. (v itself may quickly de-activate, but we hope to create a large "ripple effect.") Which k interventions should we perform? Simple examples show that to maximize influence, one should not necessarily perform all k interventions at time 0; e.g., G may not even have k nodes.

Let A be a set of k interventions. The influence of these k interventions $\sigma(A)$ is the sum, over all nodes v, of the number of time steps that v is active. The *influence maximization problem* in the non-progressive threshold model is to find the k interventions with maximum influence.

We can show that the non-progressive influence maximization problem reduces to the progressive case in a different graph. Given a graph $G = (V, E)$ and a time limit τ, we build a layered graph G^τ on $\tau \cdot |V|$ nodes: there is a copy v_t for each node v in G, and each time-step $t \leq \tau$. We connect each node in this graph with its neighbors in G indexed by the previous time step.

THEOREM 5.1. *The non-progressive influence maximization problem on G over a time horizon τ is equivalent to the progressive influence maximization problem on the layered graph G^τ. Node v is active at time t in the non-progressive process if and only if v_t is activated in the progressive process.*

Thus, models where we have approximation algorithms for the progressive case carry over. Theorem 5.1 also implies approximation results for certain non-progressive models used by Asavathiratham et al. to model cascading failures in power grids [2, 3].

Note that the non-progressive model discussed here differs from the model of Domingos and Richardson [10, 26] in two ways. We are concerned with the sum over all time steps $t \leq \tau$ of the expected number of active nodes at time t, for a *given a time limit* τ, while [10, 26] study the limit of this process: the expected number of nodes active at time t as t goes to infinity. Further, we consider interventions for a particular node v, at a particular time $t \leq \tau$, while the interventions considered by [10, 26] permanently affect the activation probability function of the targeted nodes.

6. GENERAL MARKETING STRATEGIES

In the formulation of the problem, we have so far assumed that for one unit of budget, we can deterministically target any node v for activation. This is clearly a highly simplified view. In a more realistic scenario, we may have a number m of different *marketing actions* M_i available, each of which may affect some subset of nodes by *increasing* their probabilities of becoming active, without necessarily making them active deterministically. The more we spend on any one action the stronger its effect will be; however, different nodes may respond to marketing actions in different ways,

In a general model, we choose *investments* x_i into marketing actions M_i, such that the total investments do not exceed the budget. A *marketing strategy* is then an m-dimensional vector $\mathbf{x}$ of investments. The probability that node v will become active is determined by the strategy, and denoted by $h_v(\mathbf{x})$. We assume that this function is non-decreasing and satisfies the following "diminishing returns" property for all $\mathbf{x} \geq \mathbf{y}$ and $\mathbf{a} \geq \mathbf{0}$ (where we write $\mathbf{x} \geq \mathbf{y}$ or $\mathbf{a} \geq \mathbf{0}$ to denote that the inequalities hold in all coordinates):

$$h_v(\mathbf{x} + \mathbf{a}) - h_v(\mathbf{x}) \quad \leq \quad h_v(\mathbf{y} + \mathbf{a}) - h_v(\mathbf{y}) \qquad (1)$$

Intuitively, Inequality (1) states that any marketing action is more effective when the targeted individual is less "marketing-saturated" at that point.

We are trying to maximize the expected size of the final active set. As a function of the marketing strategy $\mathbf{x}$, each node v becomes active independently with probability $h_v(\mathbf{x})$, resulting in a (random) set of initial active nodes A. Given the initial set A, the expected size of the final active set is $\sigma(A)$. The expected revenue of the marketing strategy $\mathbf{x}$ is therefore

$$g(\mathbf{x}) \quad = \quad \sum_{A \subseteq V} \sigma(A) \cdot \prod_{u \in A} h_u(\mathbf{x}) \cdot \prod_{v \notin A} (1 - h_v(\mathbf{x})).$$

In order to (approximately) maximize g, we assume that we can evaluate the function at any point $\mathbf{x}$ approximately, and find a direction i with approximately maximal gradient. Specifically, let $\mathbf{e}_i$ denote the unit vector along the i^{th} coordinate axis, and δ be some constant. We assume that there exists some $\gamma \leq 1$ such that we can find an i with $g(\mathbf{x} + \delta \cdot \mathbf{e}_i) - g(\mathbf{x}) \geq \gamma \cdot (g(\mathbf{x} + \delta \cdot \mathbf{e}_j) - g(\mathbf{x}))$ for each j. We divide each unit of the total budget k into equal parts of size δ. Starting with an all-0 investment, we perform an approximate gradient ascent, by repeatedly (a total of $\frac{k}{\delta}$ times) adding δ units of budget to the investment in the action M_i that approximately maximizes the gradient.

The proof that this algorithm gives a good approximation consists of two steps. First, we show that the function g we are trying to optimize is non-negative, non-decreasing, and satisfies the "diminishing returns" condition (1). Second, we show that the hill-climbing algorithm gives a constant-factor approximation for any function g with these properties. The latter part is captured by the following theorem.

THEOREM 6.1. *When the hill-climbing algorithm finishes with strategy $\mathbf{x}$, it guarantees that $g(\mathbf{x}) \geq (1 - e^{-\frac{k \cdot \gamma}{k + \delta \cdot n}}) \cdot g(\hat{\mathbf{x}})$, where $\hat{\mathbf{x}}$ denotes the optimal solution subject to $\sum_i \hat{x}_i \leq k$.*

The proof of this theorem builds on the analysis used by Nemhauser et al. [23], and we defer it to the full version of this paper.

With Theorem 6.1 in hand, it remains to show that g is non-negative, monotone, and satisfies condition (1). The first two are clear, so we only sketch the proof of the third. Fix an arbitary ordering of vertices. We then use the fact that for any a_i, b_i,

$$\prod_i a_i - \prod_i b_i \quad = \quad \sum_i (a_i - b_i) \cdot \prod_{j<i} a_j \cdot \prod_{j>i} b_j, \qquad (2)$$

and change the order of summation, to rewrite the difference

$$\begin{aligned} & g(\mathbf{x}+\mathbf{a}) - g(\mathbf{x}) \\ = & \sum_u \Big((h_u(\mathbf{x}+\mathbf{a}) - h_u(\mathbf{x})) \cdot \sum_{A:u\notin A} (\sigma(A+u) - \sigma(A)) \cdot \\ & \prod_{j<u, j\in A} h_j(\mathbf{x}+\mathbf{a}) \cdot \prod_{j<u, j\notin A} (1 - h_j(\mathbf{x}+\mathbf{a})) \cdot \\ & \prod_{j>u, j\in A} h_j(\mathbf{x}) \cdot \prod_{j>u, j\notin A} (1 - h_j(\mathbf{x})) \Big). \end{aligned}$$

To show that this difference is non-increasing, we consider $\mathbf{y} \leq \mathbf{x}$. From the diminishing returns property of $h_u(\cdot)$, we obtain that $h_u(\mathbf{x}+\mathbf{a}) - h_u(\mathbf{x}) \leq h_u(\mathbf{y}+\mathbf{a}) - h_u(\mathbf{y})$. Then, applying again equation (2), changing the order of summation, and performing some tedious calculations, writing $\Delta(v, \mathbf{x}, \mathbf{y}) = h_v(\mathbf{x}+\mathbf{a}) - h_v(\mathbf{y}+\mathbf{a})$ if $v < u$, and $\Delta(v, \mathbf{x}, \mathbf{y}) = h_v(\mathbf{x}) - h_v(\mathbf{y})$ if $v > u$, we obtain that

$$\begin{aligned} & (g(\mathbf{x}+\mathbf{a}) - g(\mathbf{x})) - (g(\mathbf{y}+\mathbf{a}) - g(\mathbf{y})) \\ \leq & \sum_{u,v:u\neq v} \Big((h_u(\mathbf{y}+\mathbf{a}) - h_u(\mathbf{y})) \cdot \Delta(v, \mathbf{x}, \mathbf{y}) \cdot \\ & \sum_{A:u,v\notin A} \big(\sigma(A+\{u,v\}) - \sigma(A+v) - \sigma(A+u) + \sigma(A) \big) \cdot \\ & \prod_{j<\min(u,v), j\in A} h_j(\mathbf{x}+\mathbf{a}) \cdot \prod_{j<\min(u,v), j\notin A} (1 - h_j(\mathbf{x}+\mathbf{a})) \cdot \\ & \prod_{u<j<v, j\in A} h_j(\mathbf{x}) \cdot \prod_{u<j<v, j\notin A} (1 - h_j(\mathbf{x})) \cdot \\ & \prod_{v<j<u, j\in A} h_j(\mathbf{y}+\mathbf{a}) \cdot \prod_{v<j<u, j\notin A} (1 - h_j(\mathbf{y}+\mathbf{a})) \cdot \\ & \prod_{j>\max(u,v), j\in A} h_j(\mathbf{y}) \cdot \prod_{j>\max(u,v), j\notin A} (1 - h_j(\mathbf{y})) \Big) \end{aligned}$$

In this expression, all terms are non-negative (by monotonicity of the $h_v(\cdot)$), with the exception of $\sigma(A+\{u,v\}) - \sigma(A+u) - \sigma(A+v) + \sigma(A)$, which is non-positive because σ is submodular. Hence, the above difference is always non-positive, so g satisfies the diminishing returns condition (1).

7. REFERENCES

[1] R. Albert, H. Jeong, A. Barabasi. Error and attack tolerance of complex networks. *Nature* 406(2000), 378-382.

[2] C. Asavathiratham, S. Roy, B. Lesieutre, G. Verghese. The Influence Model. *IEEE Control Systems*, Dec. 2001.

[3] C. Asavathiratham. *The Influence Model: A Tractable Representation for the Dynamics of Networked Markov Chains.* Ph.D. Thesis, MIT 2000.

[4] F. Bass. A new product growth model for consumer durables. *Management Science* 15(1969), 215-227.

[5] E. Berger. Dynamic Monopolies of Constant Size. *Journal of Combinatorial Theory Series B* 83(2001), 191-200.

[6] L. Blume. The Statistical Mechanics of Strategic Interaction. *Games and Economic Behavior* 5(1993), 387-424.

[7] J. Brown, P. Reinegen. Social ties and word-of-mouth referral behavior. *Journal of Consumer Research* 14:3(1987), 350-362.

[8] J. Coleman, H. Menzel, E. Katz. *Medical Innovations: A Diffusion Study* Bobbs Merrill, 1966.

[9] G. Cornuejols, M. Fisher, G. Nemhauser. Location of Bank Accounts to Optimize Float. *Management Science*, 23(1977).

[10] P. Domingos, M. Richardson. Mining the Network Value of Customers. *Seventh International Conference on Knowledge Discovery and Data Mining*, 2001.

[11] R. Durrett. *Lecture Notes on Particle Systems and Percolation*. Wadsworth Publishing, 1988.

[12] G. Ellison. Learning, Local Interaction, and Coordination. *Econometrica* 61:5(1993), 1047-1071.

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

[14] J. Goldenberg, B. Libai, E. Muller. Using Complex Systems Analysis to Advance Marketing Theory Development. *Academy of Marketing Science Review* 2001.

[15] M. Granovetter. Threshold models of collective behavior. *American Journal of Sociology* 83(6):1420-1443, 1978.

[16] V. Harinarayan, A. Rajaraman, J. Ullman. Implementing Data Cubes Efficiently. *Proc. ACM SIGMOD* 1996.

[17] T.M. Liggett. *Interacting Particle Systems*. Springer, 1985.

[18] M. Macy. Chains of Cooperation: Threshold Effects in Collective Action. *American Sociological Review* 56(1991).

[19] M. Macy, R. Willer. From Factors to Actors: Computational Sociology and Agent-Based Modeling. *Ann. Rev. Soc.* 2002.

[20] V. Mahajan, E. Muller, F. Bass. New Product Diffusion Models in Marketing: A Review and Directions for Research. *Journal of Marketing* 54:1(1990) pp. 1-26.

[21] S. Morris. Contagion. *Review of Economic Studies* 67(2000).

[22] G. Nemhauser, L. Wolsey. *Integer and Combinatorial Optimization.* John Wiley, 1988. .

[23] G. Nemhauser, L. Wolsey, M. Fisher. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1978), 265–294.

[24] M. Newman. The structure of scientific collaboration networks. *Proc. Natl. Acad. Sci.* 98(2001).

[25] D. Peleg. Local Majority Voting, Small Coalitions, and Controlling Monopolies in Graphs: A Review. *3rd Colloq. on Structural Information and Communication*, 1996.

[26] M. Richardson, P. Domingos. Mining Knowledge-Sharing Sites for Viral Marketing. *Eighth Intl. Conf. on Knowledge Discovery and Data Mining*, 2002.

[27] E. Rogers. *Diffusion of innovations* Free Press, 1995.

[28] T. Schelling. *Micromotives and Macrobehavior*. Norton, 1978.

[29] T. Valente. *Network Models of the Diffusion of Innovations*. Hampton Press, 1995.

[30] S. Wasserman, K. Faust. *Social Network Analysis*. Cambridge University Press, 1994.

[31] D. Watts. A Simple Model of Global Cascades in Random Networks. *Proc. Natl. Acad. Sci.* 99(2002), 5766-71.

[32] H. Peyton Young. The Diffusion of Innovations in Social Networks. Santa Fe Institute Working Paper 02-04-018(2002).

[33] H. Peyton Young. *Individual Strategy and Social Structure: An Evolutionary Theory of Institutions*. Princeton, 1998.