Adaptive Influence Maximization in Social Networks: Why Commit when You can Adapt?

Sharan Vaswani
University of British Columbia
Vancouver, Canada
sharanv@cs.ubc.ca

Laks V.S. Lakshmanan
University of British Columbia
Vancouver, Canada
laks@cs.ubc.ca

ABSTRACT

Most previous work on influence maximization in social networks is limited to the non-adaptive setting in which the marketer is supposed to select all of the seed users, to give free samples or discounts to, up front. A disadvantage of this setting is that the marketer is forced to select all the seeds based solely on a diffusion model. If some of the selected seeds do not perform well, there is no opportunity to course-correct. A more practical setting is the adaptive setting in which the marketer initially selects a batch of users and observes how well seeding those users leads to a diffusion of product adoptions. Based on this market feedback, she formulates a policy for choosing the remaining seeds. In this paper, we study adaptive offline strategies for two problems: (a) MaxSpread – given a budget on number of seeds and a time horizon, maximize the spread of influence and (b) MinTss – given a time horizon and an expected number of target users to be influenced, minimize the number of seeds that will be required. In particular, we present theoretical bounds and empirical results for an adaptive strategy and quantify its practical benefit over the non-adaptive strategy. We evaluate adaptive and non-adaptive policies on three real data sets. We conclude that while benefit of going adaptive for the MaxSpread problem is modest, adaptive policies lead to significant savings for the MinTss problem.

1. INTRODUCTION

Recently, there has been tremendous interest in the study of influence propagation in social and information networks, motivated by applications such as the study of spread of infections and innovations, viral marketing, and feed ranking to name a few (e.g., see [15, 8, 20, 21]). A prototypical problem that has received much attention in the literature is *influence maximization* (MaxSpread): given a directed network G, with edge weights denoting probabilities of influence between nodes, find k nodes, such that activating them in the beginning leads to the maximum *expected spread*, i.e., expected number of activated nodes as according

to a given diffusion model. For the viral marketing application, nodes may model users, activation may correspond to product adoption, and seed users are given free or price discounted samples of the product, with the aim of achieving the maximum expected number of product adoptions.

Kempe, Kleinberg and Tardös [17] formalized this as a discrete optimization problem and studied several discrete-time diffusion models including independent cascade and linear threshold (details in Section 2). They showed that MaxSpread under these models is NP-hard but the expected spread function satisfies the nice properties of monotonicity and submodularity. Exploiting these properties, they showed that a simple greedy algorithm, which repeatedly adds the seed with the largest marginal gain, i.e., increase in expected spread, until the budget k is reached, achieves a $(1 - 1/e)$-approximation to the optimum. There has been an explosion of research activity around this problem, including development of scalable heuristics, alternative diffusion models, and scalable approximation algorithms (e.g., see [5] [23] [18] [13] [12] [22]). For space limitations, we refer the reader to [3] for a more detailed survey.

The majority of the work in influence maximization has confined itself to a *non-adaptive* setting where, in viral marketing terms, the marketer must commit to choosing all the k seeds up front. This means that the choice of every single seed is driven completely by the diffusion model used for capturing the propagation phenomena. In practice, it may happen that the *actual spread* resulting from the seeds chosen may fall short of the expected spread predicted by the diffusion model. Recent work by Goyal et al. [12] shows that most diffusion models tend to over-predict the actual spread. Thus, committing to the choice of all k seeds in one shot can result in a sub-optimal performance in actuality. A more realistic setting is one where the marketer chooses a subset of seeds and activates them. She monitors how their activation spreads through the network and observes the actual spread thus far. She can then take into account this market feedback in making subsequent seed selections. We call this setting an *adaptive* setting, as choices of subsequent seeds are adapted to observations made so far about the actual spread achieved by previous selections. Hence, the adaptive setting introduces a *policy* π which specifies which node(s) to seed at a given time. It is very intuitive that adaptive seed selection should lead to a higher actual spread compared to non-adaptive seed selection, since it benefits from market feedback and tailors seed selections accordingly.

Adaptive seed selection raises several major challenges. For instance, in the adaptive setting, in practice, there is

a finite time horizon T within which the marketer wishes to conduct her viral marketing campaign. Suppose k is the seed budget of a marketer. The marketer must then consider the following questions. How many seeds to select at a given time, that is, what is the *batch* size? Which nodes should be selected in each intervention ? How long should she wait between seeding successive batches (interventions)? If T is sufficiently long, it seems intuitive that selecting one seed at a time and waiting until the diffusion completes, before choosing the next seed, should lead to the maximum spread. The reason is that we do not commit any more seeds than necessary to start a fresh diffusion and every seed selection takes full advantage of market feedback. We refer to the above case as *unbounded time horizon*. Another natural question is, what if the time horizon T is not long enough to allow many small batches to be chosen and/or diffusions to be observed in full. In this case, which we call *bounded time horizon*, the marketer has to choose a strategy in which the budget k is spent within the time horizon T and every seed selection benefits from as much feedback as possible.

Instead of maximizing the spread, the marketer may have a certain expected spread as the *target* that she wants to achieve. This target may be derived from the desired sales volume for the product. A natural problem is to find the minimum number of seeds needed to achieve the target. This problem, called *minimum targeted seed selection* (MINTSS for short), has been studied in the non-adaptive setting [11], where it was shown that the classic greedy algorithm leads to a bi-criteria approximation to the optimal solution. An interesting question is whether an adaptive strategy for seed selection can significantly cut down on the number of seeds needed to reach a given target spread.

Adaptive MAXSPREAD has been studied recently in [14, 10, 14], and adaptive MINTSS has been studied in [7]. While a more detailed comparison with these papers appears in Section 2, here are the key differences with our work. The market feedback model assumed by these papers is that when a node is activated (seeded), a subset of the out-edges from the node become active or "live" while others stay inactive or become "dead". This amounts to saying we get to observe which active nodes succeeded in activating which other nodes. We refer to this as "edge level" feedback. *Edge level feedback assumption is unrealistic*, since in practice, we may only know which other nodes activated as a result of choosing certain seeds, rather than who succeeded in activating whom.

The experiments conducted in these papers (if at all) are on small toy networks with 1000 nodes and they do not clarify the *practical* benefits of going adaptive for real large networks. All previous studies are confined to the setting of *unbounded time horizon*, which means the horizon is long enough for the diffusion started by each batch to complete. In practice, the horizon may be bounded and not leave enough time for successive diffusions to complete. The theoretical results in these papers bound the performance of the greedy adaptive policy compared to the optimal adaptive policy. Notice that the optimal (adaptive) policy cannot be computed in polynomial time. The only practical options for both non-adaptive and adaptive settings are greedy approximations (possibly with techniques for scaling up to large datasets). Thus, a real question of practical interest is what do we gain by going adaptive, i.e., what is the gain in performance of the greedy approximation algorithm when it

is made adaptive? In contrast, we study MAXSPREAD and MINTSS under both unbounded and bounded time horizon and quantify the benefits of going adaptive with reference to the greedy approximation algorithm, as opposed to the optimal algorithm which is not practical. Furthermore, we propose a novel node level feedback model which we use for adaptive seed selection. Node level feedback is in line what is really observable in practice: *which users became active as a result of seeding the last batch of users?*

In this paper, we address the aforementioned questions and make the following contributions.

- We define the problems of *adaptive influence maximization* under bounded or unbounded time horizon and *minimum adaptive targeted seed selection* by generalizing their non-adaptive counterparts (Section 3).
- We propose a novel *node level feedback* model for capturing market feedback for adaptive seed selection and show that as long as the time horizon is unbounded, i.e., long enough to allow diffusions to complete, the spread function for our node level feedback model is adaptive monotone and adaptive submodular (Section 4).
- We establish a bound on the spread achieved by a greedy adaptive strategy for seed selection compared to both an optimal adaptive strategy and a greedy non-adaptive strategy. The former shows that the greedy algorithm continues to provide a guaranteed approximation, while the latter formally establishes the benefits of practical adaptive strategies over practical non-adaptive ones (Section 4).
- We establish a similar bound on the number of seeds required by the greedy adaptive policy compared to the greedy non-adaptive one, in order to meet a given target expected spread and establish the practical advantage of going adaptive (Section 4).
- For the unbounded horizon, we scale up the classic greedy adaptive policy by leveraging the recent state-of-the-art non-adaptive randomized algorithm based on reverse reachable sets and adapt it to the adaptive setting to achieve superior performance (Section 5).
- We argue that the expected spread function is computationally hard to optimize f or the case of bounded time horizon and propose an alternative algorithm based on sequential model-based optimization (Section 5).
- We conduct a comprehensive set of experiments on 3 real datasets to measure the performance of our algorithms and their advantages over non-adaptive policies. We report our results (Section 6).

Related work is discussed in Section 2. We summarize the paper and present directions for future work in Section 7.

2. RELATED WORK

Non-adaptive: Two classical models of diffusion that have been extensively studied are Independent Cascade (IC) and the Linear Threshold (LT) [17]. Both are discrete-time models. The expected spread function under both these models is monotone and submodular. A real-valued set function $f : 2^U \to R$ is *monotone* if $f(S) \le f(S'), \forall S \subset S' \subseteq U$. It is *submodular* if $\forall S \subset S' \subset U$ and $x \in U \setminus S'$, $f(S' \cup \{x\}) - f(S') \le f(S \cup \{x\}) - f(S)$, i.e., the marginal gain (increase in the objective function) by adding an element to a set cannot increase as the set grows. While MAXSPREAD under both IC and LT models is NP-hard, a simple greedy algorithm [19] provides a $(1 - 1/e)$-approximation to the op-

timal solution. The greedy algorithm involves successively selecting the node with the highest marginal gain. Since computing the expected spread of a given set (and hence marginal gain) is #P-hard for both IC and LT models [6, 4], Kempe et al. [17] advocated using MCMC simulations to estimate marginal gains. Using MCMC estimation of the marginal gain, the greedy algorithm yields a $(1 - 1/e - \epsilon)$-approximation to the optimum, where $\epsilon > 0$ is the error in marginal gain estimation. Tang et al. [22] propose a near-optimal (w.r.t. time complexity) randomized greedy $(1 - 1/e - \epsilon)$-approximation algorithm for MaxSpread. It uses the concept of random reverse reachable (RR) sets to achieve this. We briefly review their procedure in Section 5. Here, we note that it is currently the state of the art for MaxSpread and has been shown to scale to a billion-edge network [22].

Adaptive: Adaptive influence maximization has been proposed previously in [10, 14, 7]. Golovin and Krause [10] extend the definitions of submodularity and monotonicity to the adaptive setting. In the adaptive setting, batches of nodes are seeded at different intervals. When a batch is seeded, an *actual* diffusion (called realization in [10]) unfolds as per the classical IC model. The next batch is chosen based on the previously observed cascade. An objective function is *adaptive monotone* and *adaptive submodular* if the marginal gain of every element is non-negative and non-increasing in every possible realization, as the size of the set (alternatively length of the policy) increases. We wish to choose a policy that maximizes such an objective function in the adaptive setting. As before, the greedy policy consists of selecting the node with the maximum marginal gain. Golovin and Krause [10] derive average case bounds on the performance of greedy adaptive policies. They also prove bounds on the greedy adaptive policy for adaptive submodular functions under matroid constraints [9]. They assume an edge level feedback mechanism with the IC model and show that the expected spread is adaptive monotone and adaptive submodular, guaranteeing an approximation algorithm. Guillory et al. [14] study the problem of submodular set cover in the adaptive setting in which the objective is to minimize the total number of sets required to cover a certain target set and prove worst case bounds for the greedy adaptive policy. They briefly describe how their framework can be used for influence maximization in a social network with hidden information (e.g., hidden preferences of users). In this paper, we consider the more traditional influence maximization problem and assume that users do not have any hidden preferences. We establish average case guarantees similar to [10]. Finally, [7] addresses the adaptive MinTss problem and shows that under certain conditions, the batch-greedy adaptive policy, in which the seeds are chosen in batches in a greedy manner, is competitive not only against the sequential greedy policy (choosing one seed at a time) but also against the optimal adaptive policy. As explained in the introduction, the key difference between these papers and our work is that unlike them, we adopt a more realistic node level feedback, establish bounds relating adaptive greedy policy with the non-adaptive greedy algorithm, thus answering the question, in practice what does one gain by going adaptive. This question is not answered by simply comparing adaptive greedy with the optimal policy which anyway could not be used in a real network owing to its intractability.

3. PROBLEM DEFINITION

We consider a directed social network $G = (V, E)$, with $|V| = n$ and $|E| = m$ with the edge weights giving the influence probabilities between two users. In this paper, we assume the independent cascade (IC) diffusion model. In this model, time proceeds in discrete steps. At time $t = 0$, the seed nodes are active. Each active user gets one chance to influence/activate her neighbor in the next time step. This activation attempt succeeds with the corresponding influence probability between the two users. An edge along which an activation attempt succeeded is said to be "live" whereas the other edges are said to be "dead". This leads to $2^{|E|}$ *possible worlds* of the network. One of these possible worlds is the *true* world, which reflects the reality of which activation attempts succeeded/failed and of the whole cascade/diffusion that occurred starting from the batches of seeds chosen at various times. Clearly, we don't know the true world, although as described below, in an adaptive setting, it may be revealed partially from successive seed selections.

We consider influence maximization in the adaptive setting, where we are given a seed budget k and a time horizon T. The *state* of a network at time t is used to determine the seed(s) which will be selected at that time. The precise definition of state depends on the type of feedback model. Golovin and Krause [10] consider an edge level feedback, where they assume after a batch is seeded, the precise status (live/dead?) of every edge is observable. We will instead adopt the *node level* feedback model, whereby we assume the status (active/inactive?) of every node in the network is observable. A *network state* is a mapping $\psi_t : V \rightarrow \{0, 1\}$. For convenience, ψ_t can be treated as the set of active nodes at time t. At $t = 0$, the nodes seeded at $t = 0$ are the only active nodes. Let G_ℓ denote the true world. It is easy to observe which nodes become active: e.g., which users buy a product or share a particular story on their Facebook page. These are precisely the nodes reachable from seeds in G_ℓ within a given *time horizon* T. However, we cannot observe which edges of the network G are actually present (i.e., are live) in G_ℓ: e.g., we can't trace which friend of a particular user influenced her to buy the product. Hence node level feedback is a more realistic feedback model. Notice that it makes weaker assumptions about what is observable, than edge level feedback.

By a *policy*, we mean a seed selection strategy, i.e., a mapping $\pi : \psi_t \rightarrow 2^V$, where ψ_t is the network state at current time t. Thus, a policy specifies which nodes to seed next, given the current network state. It is possible to have $\pi(\psi_t) = \{\}$. This means the policy chooses to not seed any nodes and just "wait". The policy thus transforms a network from its current state ψ_t to the new state $\psi_{t+1} := \psi_t \cup \pi(\psi_t)$. The network state can change from seeding as well as from the spreading of influence from current active nodes under the diffusion model. We use $f(\pi)$ to denote the spread of a policy π within a given time horizon T in the true world (the world G_ℓ which is realized in reality). The seeds selected by an adaptive policy depend on the true world whereas for the non-adaptive case, the selected seeds are independent of the true world. We care about the performance of both strategies in the true world. Since *we* don't know the true world, we quantify the *average* gain obtained by going adaptive, and do so by generating a number of candidate true worlds, finding the spread $f(\pi)$ in each of these worlds and taking the average. We denote the average performance of a policy

π as $f_{avg}(\pi)$. σ refers to the expected spread without any feedback from the network whereas f_{avg} is the average true spread given a feedback model. Note that σ is equal to f_{avg} for a non-adaptive policy since it does not use any feedback.

We use π_k to denote a policy constrained to select k seeds. When the policy π seeds a node, i.e., when $\pi(\psi_t) \neq \emptyset$, we refer to it as an *intervention*. Since the diffusion model we use is discrete, the diffusion process completes in a maximum of D time-steps where D is the length of the longest simple path in the network. If the last intervention occurred at time t and there are no subsequent interventions, then we have $\psi_{t+D+1} = \psi_{t+D}$.

Since we have a model of the diffusion process, we can simulate reality (i.e., generate or sample a set of candidate true worlds) and gauge policies by measuring their performance against the set of candidate true worlds. Thus, policy design is based on maximizing the average performance over the set of candidate worlds. These worlds constitute a training set in standard machine learning. The true world (the test sample in this analogy) will be generated from the same diffusion process, i.e., it will be drawn from the same probability distribution. Since the policy is known before we actually implement it in the real world, these are *offline* policies, as opposed to *online* policies, which are determined dynamically depending on the state of the network and the time left. In this paper, we focus on offline policies.

We next formalize the two problems studied in this paper. We assume $\mathcal{C}$ is a set of candidate true worlds, chosen by some oracle, which will be used for computing f_{avg}.

Problem 1 (MaxSpread)**.** *Given a directed probabilistic network $G = (V, E)$, seed budget k, and time horizon T, find the optimal policy $\pi_{opt,k} = arg\, max\{f_{avg}(\pi_k)\}$.*

Notice that in the calculation of spread f (and hence f_{avg}), the diffusion is restricted to a length of at most T, where T is the given time horizon. When $T \geq kD$, we say the time horizon is *unbounded*. Otherwise, it is *bounded*. We consider MaxSpread under both bounded and unbounded horizons.

MinTss in the non-adaptive case aims to find a seed set of the smallest size such that the resulting expected spread is above a given threshold Q. The best known result states there is a bi-criteria approximation: given $\beta > 0$, the greedy algorithm yields a seed set of size $\geq OPT(1 + \ln\lceil \frac{Q}{\beta}\rceil)$, where OPT is the optimal seed set size and the expected spread of the greedy selection is $\geq Q - \beta$. We generalize it to the adaptive setting. Let $c_{avg}(\pi)$ denote the average number of seeds chosen by policy π across the true worlds $\mathcal{C}$ on which it is tested.

Problem 2 (MinTss)**.** *Given a directed probabilistic network $G = (V, E)$, time horizon T, and spread threshold Q, find the policy π_{opt} that leads to the smallest seed set, i.e., $\pi_{opt} = arg\, min\{c_{avg}(\pi) \mid f(\pi) \geq Q\}$.*

Notice that policy π may end up using different numbers of seeds in different choices of true worlds. $c_{avg}(\pi)$ is the average number of seeds chosen by π across those worlds.

Finally, notice that we require that the spread achieved by π_{opt} must be $\geq Q$ in *every* true world chosen and not on an average across candidate true worlds. This is motivated by practical considerations: if the policy underperforms (i.e., $f < Q$) in some worlds and the true world is among them, the marketer won't be happy!

4. THEORETICAL RESULTS

Recall, we say time horizon is unbounded if $H \geq kD$, where k is the seed budget, D is the length of the longest path of G, and H is the time horizon. We consider unbounded time horizon up to Section 4.2. Bounded time horizon is addressed in subsection 4.3. Our first result is that our spread function based on node level feedback is adaptive monotone and adaptive submodular, thus affording an efficient approximation algorithm.

Theorem 1. *For unbounded time horizon, if the diffusion process is allowed to complete after every intervention, node level feedback is equivalent to edge level feedback w.r.t. marginal gain computation and therefore the expected spread function is adaptive submodular and adaptive monotone under node level feedback.*

Proof. We will show that node level feedback is equivalent to edge level feedback from the perspective of marginal gain computation. In [10], the authors show that the expected spread function under edge level feedback is adaptive monotone and adaptive submodular. The above theorem will follow from this. Specifically, we prove that (a) for every edge level feedback based network state, there is a corresponding state based on node level feedback, which preserves marginal gains of nodes, and (b) vice versa.

Given edge level feedback, we clearly know which nodes are active. These are precisely nodes reachable from the seeds via live edge paths in the revealed network. In the rest of the proof, we show that for each node level feedback state, there is a corresponding edge level feedback state that preserves marginal gains. Let S_0 be the set of seeds chosen at time $t = 0$. Given node level feedback, we can infer the corresponding edge level feedback based network state using the following rules. Consider an edge from an active node u to node v. Notice that the status of an edge leaving an inactive node is unknown in either feedback model.

Rule 1: If node u is active, v is inactive, and there is an edge from u to v, then infer that edge (u, v) is dead.

Rule 2: If nodes u and v are both active and u is the only in-neighbor of v, then conclude that the edge (u, v) is live.

Rule 3: If nodes u and v are both active and u has more than one in-neighbor, arbitrarily set the status of the edge (u, v) to be live or dead.

We now show that the way edge status is chosen to be live or dead in Rule 3 plays no role in determining the marginal gains of nodes. We make the observation that if the diffusion process is allowed to complete after each intervention, the only extra information about the network that is observed using edge level feedback over node level feedback is the status of edges between 2 active nodes. Given that the node u is active, we need to calculate the marginal gain of every other node in the network for the next intervention. Next we show that the status of edges between 2 active nodes does not matter in the marginal gain computation for any node.

For the rest of the argument, we consider the both u and v are active and that v has multiple active in-neighbours, i.e., the case that is addressed by Rule 3. Consider an arbitrary node w the marginal gain of which we need to calculate. There maybe multiple paths from w to a node reachable from w. These paths can be classified into those which involve the edge (u, v) and ones which don't. The marginal gain computation involving the latter paths is independent of the status of the edge (u, v). Since the diffusion pro-

cess is allowed to complete, all nodes which can be reached (in the "true" possible world) from w through (u,v) have already been activated. Hence paths going through (u,v) do not contribute to the marginal gain for w. Thus, the status of the edge (u,v) does not matter. Since w is any arbitrary node, we can conclude that the marginal gain of every node remains the same under states based on both feedback models. Adaptive monotonicity and submodularity are properties of marginal gains. Since marginal gains are preserved between edge level and node level feedback, it follows that these properties carry over to our node level feedback model. □

4.1 MAXSPREAD

There are four types of policies – the greedy non-adaptive policy (abbreviated GNA), the greedy adaptive policy (GA), the optimal non-adaptive policy (ONA) and the optimal adaptive policy (OA). We use $\pi_{GA,k}$ to denote the greedy adaptive policy constrained to select k seeds and $\sigma(\pi_{GA,k})$ to refer to the expected spread for this policy. While previous results bound the performance of greedy (adaptive) policies in relation to optimal adaptive policies, they do not shed light on practically implementable policies under either setting. These previous results do not quite answer the question "What do we gain in practice by going adaptive?" since both the optimal non-adaptive or optimal adaptive policies are intractable. We establish relations between two key practical (and hence implementable!) kinds of policies – the greedy non-adaptive policy and the greedy adaptive policy – for both MAXSPREAD and MINTSS. These relations quantify the average "adaptivity gain", i.e., the average benefit one can obtain by going adaptive.

We first restate Theorem 7 from [7]. This theorem gives a relation between the spreads obtained using a batch greedy adaptive policy which is constrained to select seeds in batches of size b and the optimal adaptive policy.

Fact 1. *If $\sigma(\pi_{GA,lb})$ is the average spread obtained by using a greedy batch policy with a batchsize b and $\sigma(\pi_{OA,mb})$ is the spread using an optimal sequential policy (the optimal policy if we are able to select one seed per intervention) constrained to selecting a number of seeds divisible by the batchsize b, then*

$$\sigma(\pi_{GA,lb}) > (1 - e^{\frac{-l}{\alpha\gamma m}})\sigma(\pi_{OA,mb}) \quad (1)$$

where α is the multiplicative error in calculating the marginal gains. gamma is a constant and equal to $(\frac{e}{e-1})^2$.

Proposition 1. *Let the horizon be unbounded. Let $\pi_{GA,n_{GA}}$ be the greedy batch policy that select n_{GA} seeds overall in batches of size b_{GA}, and let $\pi_{OA,n_{OA}}$ be the optimal adaptive policy that selects n_{OA} seeds overall in batches of size b_{OA}. Then*

$$\sigma(\pi_{GA,n_{GA}}) \geq \left[1 - \exp\left(-\frac{\lceil\frac{n_{GA}}{b_{GA}}\rceil}{\alpha\gamma\lceil\frac{n_{OA}}{b_{OA}}\rceil}\right)\right]\sigma(\pi_{OA,n_{OA}}) \quad (2)$$

where $\alpha \geq 1$ is the multiplicative error in calculating the marginal gains and $\gamma = (\frac{e}{e-1})^2$ is a constant.

Proof. Fact 1 gives a relation between the spreads obtained by a batch greedy adaptive policy constrained to select lb seeds and the optimal adaptive policy constrained to select mb seeds. Both these policies are constrained to select seeds

in batches of size b. The relation is in terms of the number of batches used by the policies. Let l and m be the number of batches for the greedy and optimal policies respectively. We make the following observations. First, the two policies can be constrained to select seeds in different batchsizes, b_{GA} and b_{OA} respectively. Next, the number of seeds selected by the policies need not be divisible by the batchsizes. We can follow a similar proof procedure as Theorem 7 in [7] and replace l by $\lceil\frac{n_{GA}}{b_{GA}}\rceil$ and m by $\lceil\frac{n_{OA}}{b_{OA}}\rceil$. □

Theorem 2. *Let $\pi_{GNA,k}$ be a greedy non-adaptive policy, $\pi_{GA,k}$ and $\pi_{OA,k}$ be the greedy and optimal adaptive policies respectively with batch-sizes equal to one i.e. the adaptive policies are sequential. All policies are constrained to select k seeds. Then we have the following relations:*

$$\sigma(\pi_{GA,k}) \geq (1 - e^{-1/\alpha\gamma})\sigma(\pi_{OA,k}) \quad (3)$$

$$\sigma(\pi_{GNA,k}) \geq (1 - \frac{1}{e} - \epsilon)^2\sigma(\pi_{OA,k}) \quad (4)$$

Proof. Proposition 1 gives us bounds on the ratio of the spread achieved by batch-greedy adaptive policy and that achieved by the optimal adaptive policy. We set $n_{OA} = k$ and $b_{GA} = b_{OA} = 1$ and obtain equation 3 of the theorem.

Theorem 2 of [1] states that for a submodular monotone function, there exists a non-adaptive policy which obtains $(1 - 1/e - \epsilon)$ fraction of the value of the optimal adaptive policy. In our context, this implies that the spread due to an optimal non-adaptive policy constrained to select n_{ONA} seeds is within a $(1 - e^{-n_{ONA}/n_{OA}} - \epsilon)$ factor of the spread of an optimal adaptive policy selecting n_{OA} seeds. More precisely,

$$\sigma(\pi_{ONA,n_{ONA}}) \geq (1 - e^{-n_{ONA}/n_{OA}} - \epsilon)\sigma(\pi_{OA,n_{OA}}) \quad (5)$$

The classical result from Nemhauser [19] states that the greedy non-adaptive algorithm obtains a $(1 - 1/e - \epsilon)$ fraction of the value of the optimal non-adaptive algorithm, where ϵ is the additive error made in the marginal gain computation. Moreover if the greedy non-adaptive policy is constrained to select n_{GNA} seeds and the optimal non-adaptive policy selects n_{ONA} seeds we have the following:

$$\sigma(\pi_{GNA,n_{GNA}}) \geq (1 - e^{-n_{GNA}/n_{ONA}} - \epsilon)\sigma(\pi_{ONA,n_{ONA}}) \quad (6)$$

Combining equations 5 and 6, we obtain the following result

$$\sigma(\pi_{GNA,n_{GNA}}) \geq (1 - e^{-n_{GNA}/n_{OA}} - \epsilon)(1 - e^{-n_{GNA}/n_{OA}} - \epsilon)\sigma(\pi_{OA,n_{OA}}) \quad (7)$$

Setting $n_{GNA} = n_{GA} = n_{OA} = k$, we obtain equation 4 of the theorem. □

Discussion: To clarify what this theorem implies, lets assume that we can estimate the marginal gains perfectly. Let's set $\epsilon = 0$ and $\alpha = 1$. We thus obtain the following relations: $\sigma(\pi_{GA,k}) \geq (1 - e^{-1/\gamma})\sigma(\pi_{OA,k})$ and $\sigma(\pi_{GNA,k}) \geq (1 - \frac{1}{e})^2\sigma(\pi_{OA,k})$. These two factors are almost equal (in fact non-adaptive is slightly better) and in the case of perfect marginal estimation, there is not much gain in going adaptive. This intuition is confirmed by our experiments in section **??**.

4.2 MinTss

Given that it takes the optimal adaptive policy n_{OA} seeds to achieve a spread of Q, we seek to find the number of seeds that it will take the greedy adaptive and traditional greedy non-adaptive policy to achieve the same spread. Since the non-adaptive policy can be guaranteed to achieve the target spread only in expectation, we allow it to have a small shortfall β_{ONA}. In addition, we allow both the greedy policies to have a small shortfall against their optimal variants. We formalize these notions in the following theorem.

Theorem 3. *Let the target spread to be achieved by the optimal adaptive policy be Q. Let the allowable shortfall for the optimal non-adaptive policy over the optimal adaptive policy be β_{ONA}. Let β_{GA} and β_{GNA} be the shortfall for the greedy adaptive and non-adaptive policies over their optimal variants. Let the number of seeds required by the four policies - OA, ONA, GA and GNA be n_{OA}, n_{ONA}, n_{GA} and n_{GNA}. Then we have the following relations*

$$n_{GA} \le n_{OA}(\alpha\gamma\ln(Q/\beta_{GA})) \tag{8}$$

$$n_{GNA} \le n_{OA}\ln\left(\frac{Q}{\beta_{ONA} - Q\epsilon}\right)\ln\left(\frac{Q - \beta_{ONA}}{\beta_{GNA} - \epsilon(Q - \beta_{ONA})}\right) \tag{9}$$

$$n_{GNA} \le n_{OA}\ln\left(\frac{Q}{\beta_{GA} - \beta_{GNA} - Q\epsilon}\right)\ln\left(\frac{Q - \beta_{GA} + \beta_{GNA}}{\beta_{GNA} - \epsilon(Q - \beta_{GA} + \beta_{GNA})}\right) \tag{10}$$

Proof. If in proposition 1, we set $b_{GA} = b_{OA} = 1$, $\sigma(\pi_{OA,n_{OA}}) = Q$ and $\sigma(\pi_{GA,n_{GA}}) = Q - \beta_{GA}$, after some algebraic manipulation we can obtain equation 8 of the theorem. Setting $\sigma(\pi_{ONA,n_{ONA}}) = Q - \beta_{ONA}$, $\sigma(\pi_{OA,n_{OA}}) = Q$ in equation 5, we obtain the intermediate relation 11.

$$n_{ONA} \le n_{OA}\ln\left(\frac{Q}{Q - \beta_{ONA}}\right) \tag{11}$$

Setting $\sigma(\pi_{ONA,n_{ONA}}) = Q - \beta_{ONA}$ and $\sigma(\pi_{GNA,n_{GNA}}) = Q - \beta_{ONA} - \beta_{GNA}$, we obtain the following relation.

$$n_{GNA} \le n_{ONA}\ln\left(\frac{Q - \beta_{ONA}}{\beta_{GNA} - \epsilon(Q - \beta_{ONA})}\right) \tag{12}$$

We constrain the spreads for the greedy adaptive and greedy non-adaptive policies to be the same. Hence, $Q - \beta_{GA} = Q - \beta_{ONA} - \beta_{GNA}$. Hence $\beta_{ONA} = \beta_{GA} - \beta_{GNA}$. By combining equations 11 and 12 and substituting β_{ONA} as $\beta_{GA} - \beta_{GNA}$, we obtain equation 10 of the theorem. $\square$

Discussion: To understand the implications of this theorem, set $\alpha = 1$, $\epsilon = 0$. Let the $\beta_{GNA} = 2$ and $\beta_{GA} = 1$, thus allowing for a shortfall of only 2 nodes in the spread. We obtain the following relations: $n_{GA} \le n_{OA}\gamma\ln(Q/2)$ and $n_{GNA} \le n_{OA}\ln(Q)\ln((Q-1)/2)$.

Figure 1 shows the growth of these functions with Q. We can see that as Q increases, the ratio $\frac{n_{GA}}{n_{OA}}$ grows much slower than $\frac{n_{GNA}}{n_{OA}}$. Hence, for the MinTss problem, there is clearly an advantage on going adaptive. This is confirmed by our experiments in section **??**.

4.3 Bounded Time Horizon

In discrete diffusion models (e.g., IC), each time-step represents one hop in the graph, so the time needed for a diffusion to complete is bounded by D, the longest simple path
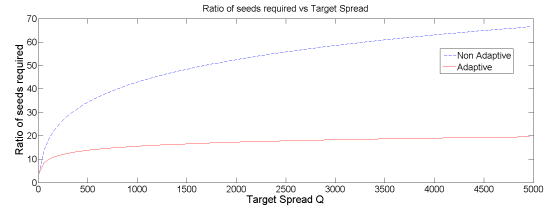

Figure 1: Theoretical comparison of adaptive and non-adaptive strategies

in the network. In networks where this length is small [**?**], most diffusions complete within a short time. This is also helped by the fact that in practice, influence probabilities are small. However, if we are given a very short time horizon, the diffusion process may not complete. In this case, seed selection is forced to be based on observations of incomplete diffusions. We show that the spread function in this case is no longer adaptive submodular.

Theorem 4. *The spread function with the IC diffusion model is not adaptive submodular if the diffusion process after each intervention is not allowed to complete.*
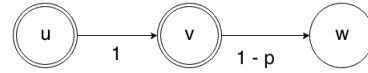

Figure 2: Counterexample to show that the spread is not adaptive submodular under incomplete diffusion

Proof. We give a counterexample. Consider the network shown in Figure 2 and the true world, where the edge (u, v) is live and (v, w) is dead. Let $H = 2$, $k = 2$. Suppose at $t = 0$, we choose the seed set $S = \{u\}$, so the next intervention must be made at time $t = 1$. Based on the true world, we observe that nodes u and v are active at time $t = 1$. Hence we infer the edge (u, v) to be live. We do not know the status of edge (v, w). Even though w is reachable in the network G, there is incomplete information in the realization revealed at $t = 1$ to decide if the node w is active or not, since the observed diffusion is incomplete. Thus, the expected spreads w.r.t. the realization above are as follows: $\sigma(S) = 2 + (1 - p)$ and $\sigma(S \cup \{w\}) = 3$. Let $S' = \{u, v\}$. Then $\sigma(S') = 2$ and $\sigma(S' \cup \{w\}) = 3$. This is because w is one hop away from $v \in S'$ and the realization tells us that w is not active. Thus, we have $\sigma(S \cup \{w\}) - \sigma(S) < \sigma(S' \cup \{w\}) - \sigma(S')$. This was to be shown. $\square$

What are our options, given that the spread under bounded time horizon is in general not adaptive submodular? Theorem 24 in [10] shows that if a function is not adaptive submodular, no polynomial algorithm can approximate the optimal expected spread within any reasonable factor. Thus, we may continue to use adaptive greedy policy, but without any guarantees in general. In our experiments **??**, we use a novel Sequential Model Based Optimization (SMBO) approach for finding a reasonably good policy when the time horizon is bounded.

5. ALGORITHMS

To obtain a greedy adaptive policy, we need to repeatedly select nodes with the maximum marginal gain at every intervention. This implies that we need to run the greedy influence maximization algorithm to compute the marginal gain over the entire network multiple times. Fortunately, this can be done efficiently by exploiting the recent work [22] which describes a near-optimal and efficient greedy algorithm – Two-phase Influence Maximization (TIM) for non-adaptive influence maximization. We first review TIM and describe the modifications we made to it for the adaptive case.

5.1 Two phase Influence Maximization

Overview of TIM: Given a budget of k seeds, a network with m edges and n nodes and an appropriate diffusion model such as IC, TIM obtains a $(1 - 1/e - \epsilon)$ fraction of the optimal spread in the non-adaptive case, incurring a near-optimal runtime complexity of $\mathcal{O}(k+l)(n+m)logn/\epsilon^2$. TIM operates by generating a large number of random Reverse Reachable (RR) sets. An RR set is defined for a particular node v and a possible world W of the network. It consists of the set of nodes that can reach the node v in the possible world W. Given enough number (see [22] for an explicit bound) of RR sets, the nodes which cover a large number of RR sets are chosen as the seed nodes: the node u which appears in the maximum number of RR sets is chosen as the first seed. Once a node u is selected as a seed, we remove all the RR sets containing u and the next seed is the node which covers the maximum of the remaining RR sets and so on until a seed set S with k nodes is chosen. Tang et al. [22] show that this simple strategy is enough to guarantee a $(1 - 1/e - \epsilon)$-approximation factor in near optimal time.

Adaptive TIM: In a greedy adaptive policy, we need to select seed nodes in every intervention. After each intervention, a certain number of nodes are influenced and become active. These already active nodes should not be selected as seeds. To ensure this, we eliminate all RR sets covered by any of these active nodes. If the number of nodes which became active is large, it brings the number of remaining RR sets below the required bound, which in turn can invalidate the theoretical guarantees of TIM, as the marginal gain of seeds selected in the next intervention may not be estimated accurately. Hence after each intervention, we need to re-generate the RR sets to effectively select seeds for the next intervention. To avoid this expensive repeated RR set generation, we instead eliminate all active nodes from the original network, by making all the incoming and outgoing edges have a zero probability, and generating the required number of RR sets for the new modified network. This guarantees that the resulting RR sets do not contain the already active nodes. This is equivalent to running the greedy non-adaptive algorithm multiple times on modified networks and results in retaining preserves the theoretical guarantees of TIM. For the unbounded time horizon, the optimal policy consists of selecting one seed per intervention and letting the diffusion complete. For the IC model, the diffusion can take a maximum of D time steps where D is the lenght of the longest simple path in the network.

5.2 Sequential Model Based Optimization

In the case of bounded time horizon (i.e., $T < kD$), as discussed at the end of Section 4.3, there is no straightforward strategy to find or approximate the optimal policy. The policy depends on the precise values of time horizon T and properties of the network. For MAXSPREAD the two extreme cases are the non-adaptive policy and the completely sequential greedy policy. The non-adaptive policy does not take any feedback into account and is hence suboptimal. For a sequential policy, the inter-intervention time T/k will be less than D. Hence the completely sequential policy will result in incomplete diffusions and from 4 will be suboptimal. A similar reasoning applies for MINTSS. For both problems, we are either forced to seed more than one node per intervention or wait for less than D time-steps between interventions, or both. We split the problem of finding the optimal policy into two parts - finding the intervention times and the number of nodes to be seeded at each intervention and which nodes need to be seeded at each intervention. Using the logic in 4, we solve the latter problem by using the adaptive TIM algorithm described above. For the former problem, we resort to a heuristic approach since the expected spread function we need to optimize does not have any nice algebraic properties w.r.t. time. In order to find the best offline policy, we need to calculate f_{avg} for each candidate policy. Calculating f_{avg} across all the candidate possible worlds is expensive. Thus we need to maximize an expensive function without any nice mathematical properties. Hence we resort to a bayesian optimization technique known as sequential model based optimization (SMBO)[16]. The SMBO approach narrows down on the promising configurations (in our case, policies) to optimize a certain function. It iterates between fitting models and using them to make choices about which configurations to investigate.

We now show the above problems can be encoded for solving these problems using SMBO. Consider MAXSPREAD. We have a maximum of k interventions. Some of these interventions may seed multiple nodes whereas other might not seed any. There are another $k-1$ variables corresponding to the inter-intervention times. Since the number of variables is $2k-1$, SMBO techniques will slow down as k increases. It is also non-trivial to add the constraint that the sum of seeds across all interventions will add to k. Since this leads to an unmanageable number of variables for large k, we introduce a parameter p which we refer to as the policy complexity. Essentially, p encodes the number of degrees of freedom a policy can have. For every $i < p$, we have a variable s_i which is the number of nodes to be seeded at a particular intervention. We have also have a variable t_i which encodes waiting time before making the next intervention. For example, if $p = 2$ and $s1 = 2, t1 = 5, s2 = 3, t2 = 7$ we initially seed 2 nodes, wait for 5 time-steps, then seed 3 nodes, wait for 7 time-steps before the next intervention. In the next intervention, we repeat the above procedure, until we run out of time, i.e., reach T or get too close (within $s1$ or $s2$) to the budget of k seeds. In the latter case, the last intervention just consists of using the remaining seeds. We use the same strategy to encode policies for MINTSS. In this case, however, we stop if the time reaches T or if $\geq Q$ nodes become active. Since we have a manageable number of parameters, we can easily use SMBO techniques to optimize over these parameters. The objective function for the first problem is to maximize the spread. The constraint is covered by the encoding. For the second problem, the objective function is to minimize the seeds to achieve a spread of Q. This can be modelled by introducing penalty parameters λ_1 and λ_2. The function can be written as,

$$minimize\ g(x) + \lambda_1(Q - f(x)) + \lambda_2(f(x) - Q) \qquad (13)$$

where x is the parameter vector, $g(x)$ is the number of seeds, $f(x)$ is the spread, Q is the target spread. The parameter λ_1 penalizes not achieving the target spread whereas λ_2 penalizes over-shooting the target spread. λ_1 encodes the hard constraint whereas λ_2 is used to direct the search.

6. EXPERIMENTS

6.1 Datasets

We run all our experiments on 3 real datasets – the author collaboration network NetHEPT (15k nodes and 62k edges), the trust network Epinions (75k nodes and 500k edges) and Flixster. On NetHEPT and Epinions where real influence probabilities are not available, we set the probability of an edge into a node v to $1/indegree(v)$, following the popular approach [5, 23, 4]. We use the Flixster network under the topic-aware independent cascade model of diffusion [2] for which the authors learned the probabilities using Expectation Maximization. Their processed network has 29k nodes and 10 topics. We choose the topic which results in the maximum number of non-zero edge weights. The resulting sub-network of Flixster consists of 29k nodes and 200k edges.

6.2 Experimental Setup

As mentioned earlier, we consider only the IC model of diffusion. We compare between greedy non-adaptive, greedy sequential adaptive and the batch-greedy adaptive policies. Since the actual true world is not known, we sample each edge in the network according to its influence probability and generate multiple true worlds. Since we are interested in the performance of a policy on an average, we randomly generate generate 100 true worlds and average our results across them. For either problem, the seeds selected by the non-adaptive policy is based on expected case calculations and remain the same irrespective of the true world. Only the performance of the policy is affected by the true possible world. Also note that for MINTSS in some true worlds the spread of the non-adaptive policy might be less than the target Q. The shortfall can be modelled by the factor β introduced in Section 4.

6.3 Sequential Model Based Optimization

We use Sequential Model-Based Optimization for General Algorithm Configuration (SMAC) [16]. SMAC is the state of the art tool used for automatic algorithm configuration for hard problems including SAT and Integer Linear Programming. Based on the performance of a configuration on certain kinds of benchmark instances characterized by problem specific properties, SMAC creates a random forest model and uses it to choose promising candidate configurations to be evaluated. SMAC uses the model's predictive distribution to compute its expected positive improvement over the the incumbent (current best solution). This approach automatically trades exploitation for exploration. SMAC can easily handle both numerical and categorical parameters.

For our case, we need to optimize an expensive black-box function (as defined in the previous section) over $2p$ configurations where p is the policy complexity. Because the function is hard to evaluate a simple brute-force grid search over the parameter space is not feasible. SMAC is implicitly able to leverage the structure present in the problem and come up with promising solutions to the problem.

The benchmark instances consist of seeds for the random process generating 10 true worlds at a time. Hence, the evaluation of each configuration on each instance involves running the algorithm 10 times. We use a training set of 1000 such instances and a separate test set of 50 instances to evaluate the policies found by SMAC. We restrict the number of function evaluations SMAC can make to 500 and set the tuner timeout (the maximum time that can be spent by SMAC in building the random forest model and deciding which configuration to evaluate next) is set to 100 seconds.

6.4 MAXSPREAD

6.4.1 Unbounded time horizon

For MAXSPREAD we vary the number of seeds k over $\{1, 10, 20, 50, 100\}$. For the unbounded horizon, we compute the spread obtained using the greedy non-adaptive and the greedy adaptive sequential policies. We set $\epsilon = 0.1$.
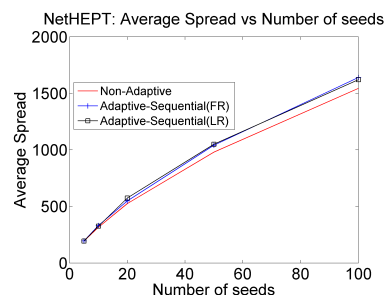
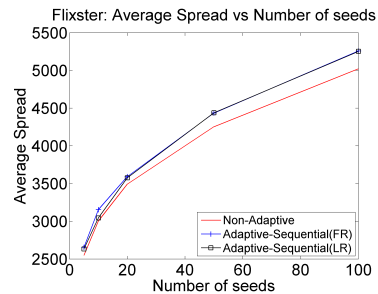**Figure 3: NetHEPT: Average Spread vs Number of seeds**

Figure 4: Flixster: Average Spread vs Number of seeds

Figures 3 and 4 show the average spread f_{avg} across 100 possible true worlds as the number of seeds is varied in the given range. We quantify the the effect of adaptivity by the ratio $\frac{f_{avg}(\pi_{GA})}{f_{avg}(\pi_{GNA})}$, which we call the average adaptivity gain. We see that the average adaptivity gain remains constant as the number of seeds are varied. We obtain similar results even with higher (100 to 500) values of k. This finding is consistent with the observations made in Section 4.

For the adaptive greedy sequential strategy in which we select one seed at a time, we generate RR sets for $k = 1$ and regenerate the RR sets between each pair of interventions. The run-time graphs are shown in Figures 5 and 6. As can be seen, although this method scales linearly with the number of seeds, it is much slower than the non-adaptive case and will prohibitive for larger datasets. Instead we can generate a large number of RR sets upfront and use these sets to
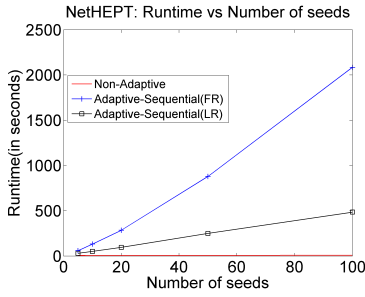
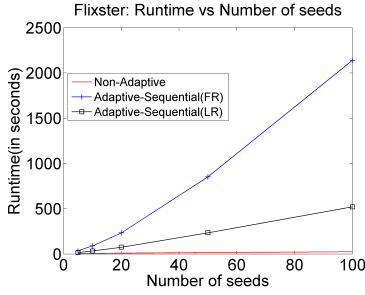**Figure 5: NetHEPT: Runtime vs Number of seeds**

Figure 6: Flixster: Runtime vs Number of seeds

select seeds for the first few interventions. The RR sets are regenerated as soon as the change in the number of active nodes becomes greater than a certain threshold (the regeneration threshold θ). The intuition is that if the number of active nodes has not increased much in a few interventions, the number of RR sets does not decline significantly and they still represent information about the state of the network well. We call this optimization trick lazy RR(LR) set regeneration to contrast it with the full RR(FR) set regeneration. We observe that because of submodularity, the frequency of RR set (re)generation decreases as the number of seeds (and time) increases. For our experiments, we empirically set θ equal to 10. Higher values of θ lead to lower runtimes but to a smaller average adaptivity gain.
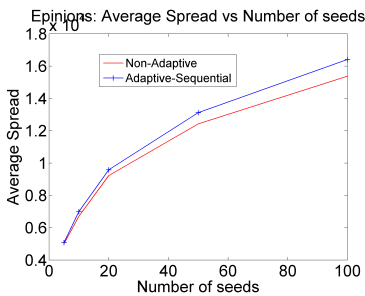

Figure 7: Epinions: Average Spread vs Number of seeds

We use this strategy to find the spread for both NetHEPT and Flixster. As can be seen from the runtime graphs and average spread graphs, this strategy does not decrease the spread much but leads to significant computational savings. After verifying this strategy, we use it to compare the 2 policies on the larger Epinions dataset, where the same trend is observed – see Figures 7 and 8. The average adaptivity gain is small even for the greedy adaptive sequential policy in case of unbounded time horizon.
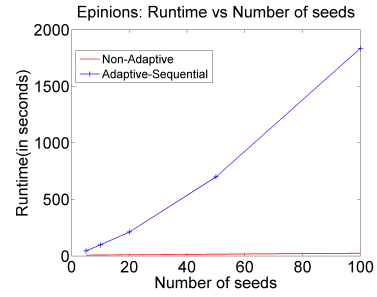

Figure 8: Epinions: Runtime vs Number of seeds

6.4.2 Bounded time horizon

For the bounded time horizon, the policy will be forced to group sets of seeds together to form a batch. From Fact 1, we know that the average spread for such a policy will be lower and hence the average adaptivity gain will further decrease. To verify this, we conduct an experiment on the NetHEPT dataset in which we decrease the time horizon T from a large value (corresponding to unbounded time horizon) to low values of the order of the length of the longest path in the network. We vary the policy complexity p to be 1 or 2 in this case. We aim to find the best configuration by varying the batch-size in the range 1 to 100 and the inter-intervention time between 1 and the D of the network. Since the difference between the spreads for the non-adaptive policy vs. the greedy adaptive sequential policy is so small, for the bounded time horizon, SMAC is unable to find a unique optimal policy. Different runs of SMAC yield different policies for the same number of seeds, sometimes converging to the non-adaptive policy even for reasonably large time horizons! A higher configuration time for SMAC might lead to stable results or alternatively we might need to encode the problem differently. We leave this for future work.

6.5 MINTSS

6.5.1 Unbounded time horizon

For all 3 datasets, for the unbounded time horizon, we compare the greedy non-adaptive and greedy adaptive policies with different batch sizes in the range $\{1, 10, 50, 100\}$. Because a large number of seeds may be required to saturate a certain fraction of the network, we use the lazy RR set generation approximation explained above and set ϵ to 0.2. Figures 9, 10 show the comparison between the non-adaptive and various adaptive greedy policies for the NetHEPT and Flixster datasets. Epinions shows a similar trend.
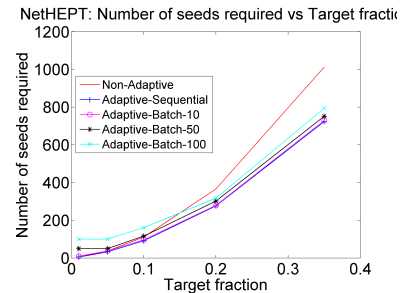

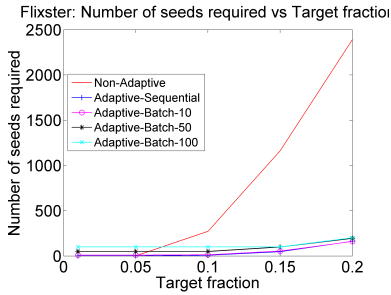**Figure 9: NetHEPT: Number of seeds required vs Target fraction**

Figure 10: Flixster: Number of seeds required vs Target fraction

As can be seen, the non-adaptive policy is competitive for smaller number of target nodes. But as the target fraction increases, the adaptive policies are better able to exploit the market feedback mechanism and lead to large savings in the number of seeds. This again agrees with our theoretical results which showed that the adaptivity gain increases as the number of target nodes increases. As the size of the network increases, the estimated spread calculation in the non-adaptive case is averaged across greater number of true worlds and hence becomes less efficient. We observed that in many cases, the final true spread for the non-adaptive policy either overshoots the target spread or misses the target spread by a large amount. We conclude that adaptive policies are particularly useful if we want to influence a significant fraction of the network.

We give some intuition for the difference in the adaptivity gains for the two problems. For adaptive policies, the rate of increase in the expected spread is fast in the beginning before the effects of submodularity take over. Hence adaptive policies require fewer seeds than non-adaptive to reach a comparable target spread. However, once submodularity kicks in, the additional seeds added contribute relatively little to the spread. Hence for MINTSS, where the objective is to reach a target spread with minimum seeds, the adaptivity gain is higher. However for MAXSPREAD, even though the adaptive policy reaches a high spread with fewer seeds, the remaining seeds in the budget don't add much more to the true spread.
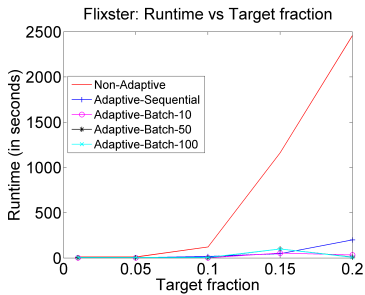

Figure 11: Flixster: Runtime vs Target fraction

We also plot the runtime graph for the Flixster dataset. The non-adaptive time dominates because it needs to choose a larger number of seeds. Since the batch-greedy policies select batches of seeds and consider feedback less often, they have a lower running time which decreases as the batch-size increases. Figure 11 shows the runtime variation for Flixster. Results on other datasets show a similar trend.

6.5.2 Bounded time horizon

T	10	50	100	1000
ShortFall (β)	709	174.98	10.54	0
Number of seeds	200	177.33	171.11	168
Objective function	7290	1927.1	276.51	168
Policy(s,t)	(100,6)	(28,8)	(20,11)	(3,12)

Table 1: Policies of $p = 1$ recovered by SMAC for varying time horizons(T) for Flixster with $Q = 5800$

We now consider the important question, how good is the effect of adaptivity for a bounded time horizon for the MINTSS problem. For this, we vary the time horizon T from 10 to 1000 and the policy complexity p is set to either 1 or 2. We use the Flixster dataset and fix the target fraction of nodes to 0.2. As in the previous problem, we aim to find the best configuration by varying the batch size in the range 1-100 and the inter-intervention time between 1 and the D of the network. Since each configuration run involves solving MINTSS 500 times, to save computation time we use a relatively high $\epsilon = 0.5$. We verified that similar results hold for smaller values of ϵ. The optimal policy returned by SMAC is evaluated on a different set of instances (possible true worlds) averaging the results over 50 such instances.

Table 1 shows the results for this experiment. For both $p = 1, 2$, as the time horizon increases, the shortfall goes to zero and the objective function is just the number of seeds required. We see that even for a low time horizon, SMAC is able to find a policy for which the number of seeds is close to the policy (which uses 163 seeds) for an unbounded time horizon. It is still much better than the non-adaptive version of the policy which uses a large number of seeds even for unbounded time horizon. As T increases, in the policy found by SMAC, the number of seeds/interventions decreases and inter-intervention time increases. In fact, for $T = 1000$ the $p = 1$, the policy found by SMAC seeded 3 nodes per intervention and had a inter-intervention time equal to 12 (which is greater than D of the graph). For extremely small T, the policy found by SMAC had 100 nodes per intervention and a very short inter-intervention time of 3. We observe similar behaviour even for $p = 2$ and with the NetHEPT dataset as well. Note that as long as $T > D$, the non-adaptive version will require the same number of seeds it needs for the unbounded horizon case. This shows us the benefit of adaptivity even when the time horizon is severely constrained. These experiments show the effectiveness of SMAC in finding reasonably good policies for any time horizon for MINTSS.

7. CONCLUSION

We studied adaptive influence maximization in social networks and focused on the MAXSPREAD and MINTSS problems. We considered both the unbounded and bounded time horizon. For unbounded horizon, we derived theoretical bounds on the performance of the greedy adaptive over greedy non-adaptive policies, thus quantifying the practical benefit of going adaptive. We studied how the adaptivity gain is affected by batch-size and number of seeds for MAXSPREAD and by target spread for MINTSS. From our experiments on real networks, we conclude that while the benefit of going adaptive is modest for the MAXSPREAD

problem, adaptive policies lead to significant savings (i.e., gain) for the MINTSS problem. For bounded time horizon, we argued that finding the optimal policy is hard and used sequential model based optimization (SMBO) techniques to find a good policy for both the problems.

Several interesting directions for future work remain. Extending our framework to the LT model and also to continuous time models is interesting. We believe that with continuous time models and the use of queries to infer the state of the network, adaptive influence maximization will bring the theory much closer to the practical needs of a real viral marketer.

8. REFERENCES

[1] A. Asadpour, H. Nazerzadeh, and A. Saberi. Maximizing stochastic monotone submodular functions. *arXiv preprint arXiv:0908.2788*, 2009.

[2] N. Barbieri, F. Bonchi, and G. Manco. Topic-aware social influence propagation models. *Knowledge and information systems*, 37(3):555–584, 2013.

[3] W. Chen, L. V. Lakshmanan, and C. Castillo. Information and influence propagation in social networks. *Synthesis Lectures on Data Management*, 5(4):1–177, 2013.

[4] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1029–1038. ACM, 2010.

[5] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208. ACM, 2009.

[6] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear îĂĂreshold model. In *Proc. 2010 IEEE Int. Conf. on Data Mining*, pages 88âĂŞ–97, 2010.

[7] Y. Chen and A. Krause. Near-optimal batch mode active learning and adaptive submodular optimization. In *Proceedings of The 30th International Conference on Machine Learning*, pages 160–168, 2013.

[8] P. Domingos and M. Richardson. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66. ACM, 2001.

[9] D. Golovin and A. Krause. Adaptive Submodular Optimization under Matroid Constraints. *Computing Research Repository*, abs/1101.4, 2011.

[10] D. Golovin and A. Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research*, 42(1):427–486, 2011.

[11] A. Goyal, F. Bonchi, L. Lakshmanan, and S. Venkatasubramanian. On minimizing budget and time in influence propagation over social networks. *Social Network Analysis and Mining*, 3(2):179–192, 2013.

[12] A. Goyal, F. Bonchi, and L. V. Lakshmanan. A data-based approach to social influence maximization. *Proceedings of the VLDB Endowment*, 5(1):73–84, 2011.

[13] A. Goyal, W. Lu, and L. V. Lakshmanan. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pages 211–220. IEEE, 2011.

[14] A. Guillory and J. Bilmes. Interactive submodular set cover. In *ICML*, pages 415–422, 2010.

[15] H. W. Hethcote. The mathematics of infectious diseases. *SIAM review*, 42(4):599–653, 2000.

[16] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization*, pages 507–523. Springer, 2011.

[17] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

[18] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429. ACM, 2007.

[19] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

[20] J. J. Samper, P. A. Castillo, L. Araujo, J. Merelo, O. Cordon, and F. Tricas. Nectarss, an intelligent rss feed reader. *Journal of Network and Computer Applications*, 31(4):793–806, 2008.

[21] X. Song, Y. Chi, K. Hino, and B. L. Tseng. Information flow modeling based on diffusion rate for prediction and ranking. In *Proceedings of the 16th international conference on World Wide Web*, pages 191–200. ACM, 2007.

[22] Y. Tang, X. Xiao, and S. Yanchen. Influence maximization: Near-optimal time complexity meets practical efficiency. 2014.

[23] C. Wang, W. Chen, and Y. Wang. Scalable influence maximization for independent cascade model in large-scale social networks. *Data Mining and Knowledge Discovery*, 25(3):545–576, 2012.