Efficient Algorithms for Adaptive Influence Maximization

Kai Han *
University of Science and Technology of China
hankai@ustc.edu.cn

Keke Huang *
Nanyang Technological University
khuang005@e.ntu.edu.sg

Xiaokui Xiao *
National University of Singapore
xkxiao@nus.edu.sg

Jing Tang
National University of Singapore
isejtang@nus.edu.sg

Aixin Sun
Nanyang Technological University
axsun@ntu.edu.sg

Xueyan Tang
Nanyang Technological University
asxytang@ntu.edu.sg

ABSTRACT

Given a social network G, the influence maximization (IM) problem seeks a set S of k seed nodes in G to maximize the expected number of nodes activated via an influence cascade starting from S. Although a lot of algorithms have been proposed for IM, most of them only work under the *non-adaptive* setting, i.e., when all k seed nodes are selected before we observe how they influence other users. In this paper, we study the *adaptive* IM problem, where we select the k seed nodes in batches of equal size b, such that the choice of the i-th batch can be made after the influence results of the first $i-1$ batches are observed. We propose the first practical algorithms for adaptive IM with an approximation guarantee of $1-\exp(\xi-1)$ for $b=1$ and $1-\exp(\xi-1+1/e)$ for $b>1$, where ξ is any number in $(0,1)$. Our approach is based on a novel AdaptGreedy framework instantiated by non-adaptive IM algorithms, and its performance can be substantially improved if the non-adaptive IM algorithm has a small *expected* approximation error. However, no current non-adaptive IM algorithms provide such a desired property. Therefore, we further propose a non-adaptive IM algorithm called EPIC, which not only has the same worst-case performance bounds with that of the state-of-the-art non-adaptive IM algorithms, but also has a reduced expected approximation error. We also provide a theoretical analysis to quantify the performance gain brought by instantiating AdaptGreedy using EPIC, compared with a naive approach using the existing IM algorithms. Finally, we use real social networks to evaluate the performance of our approach through extensive experiments, and the experimental experiments strongly corroborate the superiorities of our approach.

PVLDB Reference Format:
Kai Han, Keke Huang, Xiaokui Xiao, Jing Tang, Aixin Sun, and Xueyan Tang. Efficient Algorithms for Adaptive Influence Maximization. *PVLDB*, 11(9): 1029-1040, 2018.
DOI: https://doi.org/10.14778/3213880.3213883

*These authors contributed equally to the paper.

Proceedings of the VLDB Endowment, Vol. 11, No. 9

DOI: https://doi.org/10.14778/3213880.3213883

1. INTRODUCTION

The proliferations of online social networks such as Facebook and Twitter have motivated considerable research on viral marketing as an optimization problem. For example, an advertiser could provide a few individuals (referred to as "seed nodes") in a social network with free product samples, in exchange for them to spread the good word about the product, so as to create a large cascade of influence on other social network users via word-of-mouth recommendations. The Influence Maximization (IM) problem in such a scenario aims to select a number of seed nodes to maximize the influence propagation created.

Formally, the input to IM consists of a social network $G=(V,E)$, a budget k, and a influence model M. The influence model M captures the uncertainty of influence propagation in G, and it defines a set $\mathcal{W}$ of *possible worlds*, each of which represents a possible scenario of the influence among the nodes in G. The problem seeks to *activate* (i.e., influence) a seed set S of k nodes that can maximize the expected number of influenced individuals over all the possible worlds in $\mathcal{W}$.

A plethora of techniques have been proposed for IM [7, 12, 13, 15–17, 20–24]. Almost all techniques, however, require that the seed set S should be decided before the influence propagation process, which means that they work in a "non-adaptive" manner. In other words, if an advertiser has k product samples, she would have to commit all samples to k chosen social network users before observing how they may influence other users. In practice, however, an advertiser could employ a more *adaptive* strategy to disseminate the product samples. For example, she may choose to give out half of the samples, and then wait for a while to find out which users are influenced; after that, she could examine the set U of users that have not been influenced, and then disseminate the remaining samples to $k/2$ users that have a large influence on U. This strategy is likely to be more effective than giving out all k samples all at once, since the dissemination of the second batch of products is optimized using the knowledge obtained from the first batch's results.

In fact, the above adaptive approach has been applied in HEALER [26], a software agent deployed in practice since 2016, which recommends sequential intervention plans for homeless shelters. HEALER aims to raise awareness about HIV among homeless youth by maximizing the spread of awareness in the social network of the target population. It chooses people as the seed nodes, who are "activated" by participating the intervention plans for HIV. The choices of seed nodes are adaptive, i.e., they are selected in batches and the choice of a batch depends on the observed results of all previous batches.

Golovin et al. [11] are the first to study IM under the adaptive setting, assuming that the k seed nodes are chosen in a *sequential* manner, such that the selection of the $(i+1)$-th node is performed after the influence of the first i nodes has been observed. Specifically, they consider that (i) the social network conforms to a possible world w sampled from $\mathcal{W}$, but (ii) w is not known to the advertiser before the selection of the first seed node. Then, after the i-th seed node v_i is chosen, the part of w relevant to $\{v_1, v_2, \ldots, v_i\}$ (i.e., the nodes that they can influence in w) is revealed to the advertiser, based on which she can (i) eliminate the possible worlds in $\mathcal{W}$ that contradict what she observes, and (ii) select the next seed node as one that has a large expected influence over the remaining possible worlds.

Golovin et al. [11] propose a simple greedy algorithm for adaptive IM that returns a seed set S whose influence is at least $1 - 1/e$ of the optimum. under the case that only one seed is selected in each batch (i.e., $b = 1$). Nevertheless, the algorithm requires knowing the *exact* expected influence of every node, which is impractical since the computation of expected spread is #P-hard in general [8]. Vaswani and Lakshmanan [25] extend Golovin et al.'s model by allowing selecting $b \geq 1$ seed nodes in each batch, and by accommodating errors in the estimation of expected spreads. Their method returns an $(1 - \exp\left(-\frac{(1-1/e)^2}{\eta}\right))$-approximation under this setting, where η is certain number bigger than 1. However, even this relaxed approach is still impractical, its requirements on the accuracy of expected spread estimation cannot be met by any existing algorithms without incurring prohibitive processing costs (see Section 2.3 for a discussion).

Contributions. Motivated by the deficiency of existing techniques, we study the adaptive IM problem under the general setting that each batch contains $b \geq 1$ seed nodes, and propose the *first* practical solution for adaptive IM. Specifically, our contributions include the following.

First, We propose AdaptGreedy, a framework that enables us to construct strong approximation solutions for adaptive IM using existing non-adaptive IM methods as building blocks. In particular, we prove that AdaptGreedy achieves an approximation guarantee of $1 - \exp(\xi - 1)$ for $b = 1$ and $1 - \exp(\xi - 1 + 1/e)$ for $b > 1$, where $\xi \in (0, 1)$ is a user-specified parameter. The derivation of this approximation result requires non-trivial extension of the existing theoretical results on adaptive algorithms (e.g., [11] and [25]), since AdaptGreedy imposes far fewer constraints on the building blocks used for adaptive IM.

Second, we conduct an in-depth analysis on how AdaptGreedy could be instantiated with the state-of-the-art non-adaptive IM algorithms, and provide an interesting insight: the overall approximation guarantee of AdaptGreedy could be improved if the *expected* approximation guarantee of the non-adaptive IM algorithm used by AdaptGreedy is much better than the *worst-case* approximation guarantee of the algorithm. Existing non-adaptive IM algorithms, however, do not benefit AdaptGreedy in this regard, as there is no known result on their expected approximation guarantee. Motivated by this, we develop a new non-adaptive IM method, EPIC, that provides not only an attractive expected approximation ratio, but also the same worst-case guarantees as the state-the-art non-adaptive IM techniques. We establish AdaptGreedy's performance guarantee when it is instantiated with EPIC, based on a non-trivial theoretical analysis utilizing Azuma's inequality [10].

Third, We conduct extensive experiments to test the performance of AdaptGreedy and EPIC, and the experimental results strongly corroborate the effectiveness and efficiency of our approach.

2. PRELIMINARIES

2.1 IM and Possible Worlds

Let $G = (V, E)$ be a social network with a node set V and an edge set E, such that $|V| = n$ and $|E| = m$. We assume that the propagation of influence on G follows the *independent cascade (IC)* model [16], in which each edge (u, v) in G is associated with a probability $p(u, v)$, and the influence propagation process is defined as a discrete-time stochastic process as follows. At timestamp 0, we activate a set S of *seed nodes*. Then, at each subsequent timestamp t, each node u that is newly activated at timestamp $t-1$ has a chance to activate each v of its neighbors, such that the probability of activation equals $p(u, v)$. After that, u stays active, but cannot activate any other nodes. The propagation process terminates when no node is newly activated at a certain timestamp, and the total number of nodes activated then is defined as the *influence spread* of S, denoted as $I_G(S)$. The *vanilla* influence maximization (IM) problem asks for a seed set S of k nodes that maximizes the expected value of influence spread $\mathbb{E}[I_G(S)]$.

As demonstrated in [16], the IC model also has an interpretation based on *possible worlds*. Let w be a graph generated by removing each edge (u, v) in G with $1 - p(u, v)$ probability, and let $\mathcal{W}$ be the set of all possible choices of w. Then, w can be regarded as a possible world sampled from a distribution over $\mathcal{W}$ that is defined by G and the edge removal process. For example, Figure 1 shows a social network and three of its possible worlds. For convenience, we abuse notation and use $\mathcal{W}$ to denote both the universe of possible worlds and the aforementioned distribution over it. For any seed set S, let $I_w(S)$ be the number of nodes in w (including those in S) that can be reached from S via a directed path starting from S, and $\mathbb{E}_{w\sim\mathcal{W}}[I_w(S)]$ be the expectation of $I_w(S)$ over $\mathcal{W}$. It is shown in [16] that

$$\mathbb{E}_{w\sim\mathcal{W}}[I_w(S)] = \mathbb{E}[I_G(S)].$$

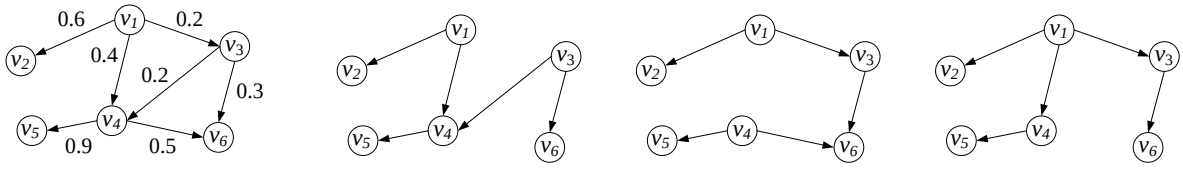
In other words, if we are to address the vanilla IM problem, it suffices to identify a seed set S whose expected influence over the possible worlds in $\mathcal{W}$ is the largest.

Remark. We note that the algorithms presented in this paper can be extended to other influence models such as the *linear threshold* model [16] or the topic-aware models [5]. We focus on the IC model, however, as it simplifies the exposition of our technical details.

2.2 Adaptive IM

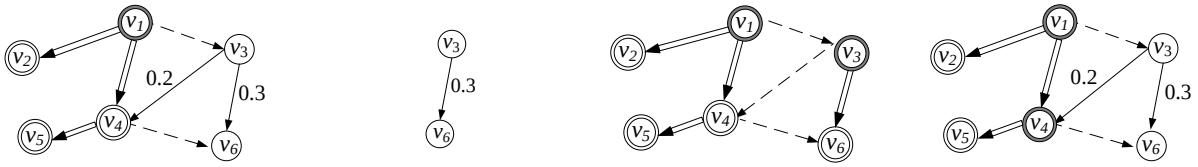
Suppose that the influence propagation on G conforms to a possible world w that is sampled from $\mathcal{W}$, i.e., for any seed set S, the nodes that it can influence are exactly the nodes that it can reach in w. The *adaptive influence maximization (IM)* problem [11] considers that w is unknown in advanced, but can be partially revealed after we choose some nodes as seeds. For example, consider the social network in Figure 1(a), and suppose that the possible world sampled from $\mathcal{W}$ is $w = w_1$, as shown in 1(b). Assume that we choose v_1 as the first seed node. In that case, we can observe v_1's influence on v_2 and v_4, since v_1 has two outgoing edges (v_1, v_2) and (v_1, v_4) in w_1. Similarly, we can observe v_4's influence on v_5. In addition, we can also observe that v_1 (resp. v_4) cannot influence v_3 (resp. v_6), as w_1 does not contain an edge from v_1 to v_3 (resp. v_4 to v_6). Figure 2(a) shows the results of the influence propagation from v_1, with each double-line (dashed-line) arrow denoting a successful (resp. failed) step of influence.

In general, after choosing a partial set S' of seed nodes, we can learn all nodes that S' can reach in w, as well as the out-edges



(a) A social network (b) Possible world w_1 (c) Possible world w_2 (d) Possible world w_3

Figure 1: A social network and three of its possible worlds



(a) v_1 as the first seed (b) First residue graph (c) v_3 as the second seed (d) Non-adaptive IM

Figure 2: Adaptive vs. non-adaptive seed selection with $k = 2$

of those nodes in w. This enables us to optimize the choices of the remaining seed nodes since we can focus on the nodes that have not been influenced by S'. For instance, consider that selecting another seed node based on the result in Figure 2(a). In that case, we can omit the nodes that have been influenced (i.e., v_1, v_2, v_4, and v_5), and focus on the subgraph induced by the remaining nodes, as shown in Figure 2(b). Based on this, we can choose v_3 as the second seed node, which yields the results in Figure 2(c), where we have 6 nodes influenced in total. In contrast, if we are to non-adaptively choose two seed nodes from the social network in Figure 1(a), we may end up choosing v_1 and v_4, in which case we would obtain the result in Figure 2(d) when the underlying possible world is w_1 in Figure 1(b). In other words, we can only influence 4 nodes instead of 6.

Assume that we are to choose k seed nodes in r batches of equal size $b = k/r$, and that we are allowed to observe the influence propagation in w for r times in total, once after the selection of each batch. The adaptive IM problem asks for r seed sets $S_1, S_2, \ldots, S_r$, such that selecting S_i ($i \in [1, r]$) as the i-th batch maximizes the expected influence spread over the choices of $w \sim \mathcal{W}$. Observe that when $b = k$ (i.e., $r = 1$), the problem degenerates to the vanilla IM problem.

We aim to develop algorithms for adaptive IM that provide non-trivial worst-case guarantees in terms of both accuracy (i.e., the expected influence of $\bigcup_i S_i$) and efficiency (i.e., the time required to identify S_i). We do not consider the "waiting time" required to observe the influence of a seed node batch S_i before the selection of the next batch S_{i+1}, since it is independent of the algorithms used. That is, we target at helping the advertiser to identify S_{i+1} as quickly as possible after the effects of S_i have been observed.

Table 1 lists the notations that are frequently used in the remainder of the paper.

Table 1: Frequently used notations

Notation	Description
$G = (V, E)$	A social network with node set V and edge set E.
n, m	the numbers of nodes and edges in G, respectively
k	the total number of selected seed nodes
b	the number of nodes selected in each batch
G_i	the i-th residue graph
n_i, m_i	the numbers of nodes and edges in G_i, respectively
S_i	the seed set selected from G_i
S_i^o	the optimal seed set in G_i
c	$c = 1$ when $b = 1$ and $c = 1 - 1/e$ otherwise.
$OPT_{k,b}$	the optimal expected influence spread of k seed nodes under the setting of selecting b nodes in each batch.
$OPT_b(G_i)$	the optimal expected influence spread of b seed nodes in G_i.
$I_G(S)$	the number of nodes activated by S in G
ϵ_i, δ_i	the parameters for the worst-case approximation guarantee in the ith batch.
ξ_i	the absolute error factor in the ith batch
$Cov_{\mathcal{R}}(S)$	the number of RR-sets in $\mathcal{R}$ that overlap S
$F_{\mathcal{R}}(S)$	the fraction of RR-sets in $\mathcal{R}$ that overlap S

2.3 Existing Solutions

The first solution to adaptive IM is by [11]. It assumes that $b = 1$ (i.e., each batch consists of only one seed node), and adopts a greedy approach as follows. Given G, it first identifies the node v_1 whose expected spread $\mathbb{E}[I_G(\{v_1\})]$ on G is the largest, and selects it as the first seed. Then, it observes the nodes that are influenced by v_1 (which are in accordance to the possible world w_0), and removes them from G. Let G_2 denote the subgraph of G induced by the remaining nodes. After that, for the i-th ($i > 1$) batch, it (i) selects the node v_i with the maximum expected spread $\mathbb{E}[I_{G_i}(\{v_i\})]$ on G_i, (ii) observes the influence of v_i on G_i, and then (iii) generates a new graph G_{i+1} by removing from G_i those nodes that are influenced by v_i. For convenience, we refer to G_i as the i-th *residue graph*, and let $G_1 = G$.

Let $OPT_{k,b}$ denote the expected spread of the optimal solution to the adaptive IM problem parameterized with k and b. Golovin et al. [11] show that the above greedy approach returns a solution whose expected spread is at least $(1 - 1/e) \cdot OPT_{k,1}$. This approximation guarantee, however, cannot be achieved in polynomial time because (i) in the i-th batch, it requires identifying a node v_i with the maximum largest expected spread $\mathbb{E}[I_{G_i}(\{v_i\})]$ on G_i, but (ii) computing the exact expected spread of a node is #P-hard in general [8].

To remedy the above deficiency, Vaswani and Lakshmanan [25] propose a relaxed approach that allows errors in the estimation of expected spreads. In particular, they assume that for any node set S and any residue graph G_i, we can derive an estimation $\tilde{\mathbb{E}}[I_{G_i}(S)]$

Algorithm 1: AdaptGreedy

Input: social network G, seed set size k, batch number r
Output: adaptively selected seed sets $S_1, \cdots, S_r$

$b \leftarrow k/r$;
$G_1 \leftarrow G$;
if $r = k$ **then**
  $c \leftarrow 1$;
else
  $c \leftarrow 1 - 1/e$;
for $i = 1$ **to** r **do**
  Identify a size-b seed set S_i from G_i, such that the expected spread of S_i on G_i is at least $c - \xi_i$ times the largest expected spread of any size-b seed set on G_i;
  Observe the influence of S_i in G_i;
  Remove all nodes in G_i that are influenced by S_i, and denote the resulting graph as G_{i+1};
return $S_1, \cdots, S_r$

of $\mathbb{E}[I_{G_i}(S)]$, such that

$$c^{\perp} \cdot \mathbb{E}[I_{G_i}(S)] \leq \tilde{\mathbb{E}}[I_{G_i}(S)] \leq c^{\top} \cdot \mathbb{E}[I_{G_i}(S)], \qquad (1)$$

with $c^{\top}/c^{\perp}$ bounded from above by a parameter η. They show that, by feeding such estimated expected spreads to the greedy approach in [11], it can achieve an approximation guarantee of $1 - \exp(-1/\eta)$. In addition, they show that the greedy approach can be extended to the case when $b > 1$, with one simple change: in the i-th batch, instead of selecting only one node, we select a size-b seed set S_i whose estimated expected spread on G_i is at least $1 - 1/e$ fraction of the largest estimated expected spread on G_i. In that case, they show that the resulting approximation guarantee is $1 - \exp\left(-\frac{(1-1/e)^2}{\eta}\right)$.

Unfortunately, the accuracy requirement in Equation 1 is so stringent that no existing algorithm for evaluating expected spread can meet the requirement without incurring prohibitive computation costs. To understand this, observe that when $\mathbb{E}[I_{G_i}(S)]$ is very small, Equation (1) allows only a tiny amount of estimation error in $\tilde{\mathbb{E}}[I_{G_i}(S)]$, in which case the derivation of $\tilde{\mathbb{E}}[I_{G_i}(S)]$ is extremely challenging. Due to this issue, Vaswani and Lakshmanan [25] propose to trade accuracy for efficiency and adopt algorithms that do not enforce Equation 1, but fail to establish any non-trivial approximation guarantees accordingly.

3. SOLUTION FRAMEWORK

Algorithm 1 illustrates the framework of our solution for adaptive IM, referred to as AdaptGreedy. At the first glance, AdaptGreedy may seem similar to Vaswani and Lakshmanan's method [25], since both techniques (i) adaptively select seed nodes in r batches and (ii) do not require exact computation of expected spreads. However, there is a crucial difference between the two: Vaswani and Lakshmanan's method requires that the expected spread of *every* node set should be estimated with a small *relative* error, whereas AdaptGreedy allows a random *absolute* error of $\xi_i \cdot OPT_b(G_i)$, where $OPT_b(G_i)$ denotes the maximum expected spread of any size-b seed set on G_i. The error requirement of AdaptGreedy is much more lenient than that of Vaswani and Lakshmanan's method, and it can be achieved by several state-of-the-art solutions [17,23,24] for vanilla influence maximization, i.e., it admits practical implementations. In addition, AdaptGreedy provides a strong approximation guarantee, as shown in the following theorem.

THEOREM 1. *Let $\mathcal{G}_i$ be the set of possible choices for G_i. Let $\Pr[\xi_i \mid G_1, \ldots, G_i]$ be the probability that S_i achieves an approximation ratio of $c - \xi_i$ conditioned on the event that the first i residue graphs are $G_1, \ldots, G_i$, and*

$$\xi = \frac{1}{r} \sum_{i=1}^{r} \sum_{G_1 \in \mathcal{G}_1, \ldots, G_i \in \mathcal{G}_i} (\xi_i \cdot \Pr[\xi_i \mid G_1, \ldots, G_i] \cdot \Pr[G_1, \ldots, G_i]). \qquad (2)$$

Then, the approximation guarantee of AdaptGreedy *is at least*

$$\begin{cases} 1 - \exp(\xi - 1), & \text{if } b = 1 \\ 1 - \exp\left(\xi - 1 + \frac{1}{e}\right), & \text{otherwise} \end{cases} \qquad (3)$$

Intuitively, Theorem 1 states that the approximation guarantee of ξ depends on the average value of ξ_i ($i \in [1, r]$) conditioned on the possibilities of $G_1, \ldots, G_r$.

Now recall that the adaptive IM method in [25] provides the following approximation guarantee for certain $\eta > 1$:

$$\begin{cases} 1 - \exp(-1/\eta), & \text{if } b = 1 \\ 1 - \exp\left(-\frac{(1-1/e)^2}{\eta}\right), & \text{otherwise} \end{cases}$$

In comparison, the approximation guarantee of AdaptGreedy is significantly better when $b > 1$, and is comparable when $b = 1$. In addition, AdaptGreedy is *flexible* in that it allows each batch of seed nodes S_i to be selected with different (even random) approximation guarantee $c - \xi_i$, whereas the existing solutions (e.g., [11]) for adaptive IM require that all seed sets $S_1, \ldots, S_r$ should be processed with identical accuracy assurance. As we show in Section 4, the flexibility of AdaptGreedy is crucial in improving the efficiency of our adaptive IM algorithms.

The proof of Theorem 1 is rather intricate as it requires non-trivial extensions of the theoretical results developed for adaptive submodular optimization [11]. We refer interested readers to Appendix A for the details.

4. INSTANTIATIONS OF ADAPTGREEDY

4.1 Instantiation using Existing Algorithms

As shown in Algorithm 1, AdaptGreedy requires identifying a size-b seed set S_i from the i-th residue graph G_i, such that

$$\mathbb{E}[I_{G_i}(S_i)] \geq (c - \xi_i) \cdot OPT_b(G_i),$$

where $\mathbb{E}[I_{G_i}(S_i)]$ is the expected spread of S_i on G_i, $OPT_b(G_i)$ is the maximum spread of any size-b seed set on G_i, and c equals 1 if $b = 1$ and $1 - 1/e$ otherwise. Such an approach achieves a provable approximation guarantee represented by ξ as long as $\xi_1, \cdots, \xi_i, \xi$ satisfy the condition shown in Theorem 1. We observe that such a seed set S_i could be obtained by applying the state-of-the-art algorithms (e.g., [17, 23, 24]) for vanilla influence maximization (IM) on G_i. In particular, these algorithms are randomized, and they provide a *worst-case* approximation guarantee as follows: given a seed set size b, an error threshold ϵ_i and a failure probability δ_i, they output a size-b seed set S_i in G_i whose expected spread is $c - \rho_i$ times the maximum expected spread of any size-b seed set on G_i, such that $\rho_i \leq \epsilon_i$ with at least $1 - \delta_i$ probability. For convenience, we refer to ρ_i as the *absolute error factor*.

By applying such algorithms on each residue graph G_i with any given parameters ϵ_i and δ_i, we obtain an instantiation of

AdaptGreedy achieving an approximation ratio of $1 - \exp(\xi - c)$ (see Theorem 1), with

$$\begin{aligned}
&\Pr\left[\xi > \tfrac{1}{r}\textstyle\sum_{i=1}^{r}\epsilon_i\right] \\
&= \Pr\left[\tfrac{1}{r}\textstyle\sum_{i=1}^{r}\epsilon_i < \tfrac{1}{r}\sum_{i=1}^{r}\sum_{G_1,\ldots,G_i}\left(\rho_i \cdot \Pr\left[\rho_i, G_1, \ldots, G_i\right]\right)\right] \\
&\leq \textstyle\sum_{i=1}^{r}\Pr\left[\epsilon_i < \sum_{G_1,\ldots,G_i}\left(\rho_i \cdot \Pr[\rho_i, G_1, \ldots, G_{i-1}]\right)\right] \\
&\leq \textstyle\sum_{i=1}^{r}\sum_{G_1,\ldots,G_i}\left(\Pr\left[\epsilon_i < \rho_i \mid G_1, \ldots, G_i\right] \cdot \Pr[G_1, \ldots, G_i]\right) \\
&\leq \textstyle\sum_{i=1}^{r}\sum_{G_1,\ldots,G_i}\left(\delta_i \cdot \Pr[G_1, \ldots, G_i]\right) \\
&= \textstyle\sum_{i=1}^{r}\delta_i
\end{aligned}$$

In other words, the instantiation yields an approximation guarantee of $1 - \exp\left(\frac{1}{r}\sum_{i=1}^{r}\epsilon_i - c\right)$ with at least $1 - \sum_{i=1}^{r}\delta_i$ probability.

But how efficient is the above instantiation? To answer the above question, we need to investigate the time complexity of the vanilla IM algorithms in [23]. The theoretical analysis in [23] show that if we are to achieve $(c - \epsilon_i)$-approximation on G_i with at least $1 - \delta_i$ probability, then the expected computation cost is $O((b \log n_i + \log \frac{1}{\delta_i})(m_i + n_i)/\epsilon_i^2)$, where n_i and m_i denotes the numbers of nodes and edges in G_i. Since $n_i \leq n$ and $m_i \leq m$, the expected time required to process G_i is $O((b \log n + \log \frac{1}{\delta_i})(m + n)/\epsilon_i^2)$. As such, all r batches of seed nodes can be identified in $O(\sum_{i=1}^{r}(b \log n + \log \frac{1}{\delta_i})(m + n)/\epsilon_i^2)$ expected time.

4.2 Rationale for an Improved Approach

The instantiation of AdaptGreedy mentioned in Section 4.1 is simple and intuitive, but is far from optimized in terms of its approximation guarantee. To explain, recall that it requires each seed set S_i to achieve $(c - \epsilon_i)$-approximation on G_i with at least $1 - \delta_i$, based on which it provides an overall approximation ratio of $1 - \exp\left(\frac{1}{r}\sum_{i=1}^{r}\epsilon_i - c\right)$ with at least $1 - \sum_{i=1}^{r}\delta_i$ probability. In other words, it imposes a stringent worst-case approximation guarantee on each seed set S_i. This, however, might be overly conservative. For example, suppose that one S_j of the seed sets has an expected spread that is $c - \rho_j$ times the optimum, with $\rho_j > \epsilon_j$, i.e., it fails to achieve $(c-\epsilon_j)$-approximation. Even in that case, the overall approximation ratio of AdaptGreedy could still be $c - \frac{1}{r}\sum_{i=1}^{r}\epsilon_i$, as long as there exists another seed set S_i whose expected spread is $(c-\rho_i)$ times the maximum, with $\rho_i - \epsilon_i \geq \epsilon_j - \rho_j$. In other words, the deficiency of one seed set can be compensated, as long as there exist other seed sets whose quality is above the bar by a sufficient margin.

Formally, if we regard each seed set S_i's approximation ratio $c - \rho_i$ as a random variable, then the overall approximation guarantee of AdaptGreedy, namely, $1 - \exp\left(\frac{1}{r}\sum_{i=1}^{r}\rho_i - c\right)$, depends on the mean of all r variables. Intuitively, when r is sizable, $\frac{1}{r}\sum_{i=1}^{r}\rho_i$ should be concentrated to its expectation, i.e., $\frac{1}{r}\sum_{i=1}^{r}\mathbb{E}[\rho_i]$. That is, instead of formulating the approximation ratio of AdaptGreedy based on the worst-case guarantee of each S_i, we might derive it based on each S_i's *expected* approximation ratio, which could lead to much tighter results.

To make the above idea work, however, there are several challenges that we need to address. First, there is no known result for vanilla IM with expected approximation guarantees. This motivates us to develop a vanilla IM method that is tailored for AdaptGreedy, as we show in Section 4.3. Second, as the selection of the i-th seed set S_i is dependent on the results of the $(i-1)$-th seed set S_{i-1}, the random variables $\rho_1, \rho_2, \ldots, \rho_r$ are *correlated*, which makes it rather non-trivial to derive concentration results for $\frac{1}{r}\sum_{i=1}^{r}\rho_i$. We circumvent this issue with a theoretical analysis leveraging Azuma's inequality [10] in Section 4.4. Finally, even if we are given a concentration bound on $\frac{1}{r}\sum_{i=1}^{r}\rho_i$, we still need to carefully tune each ρ_i, so as to yield a strong theoretical guarantee while achieving superior practical efficiency.

4.3 Vanilla IM with Expected Approximation

As discussed in Section 4.2, the existing IM algorithms provide only a worst-case approximation guarantee $c - \epsilon_i$, i.e., they ensure that their absolute error factor ρ_i is no more than the input threshold ϵ_i with high probability. To optimize the performance of AdaptGreedy, however, we are in need of non-adaptive IM algorithm $\mathcal{A}$ with two properties:

1. The worst-case approximation guarantee and time complexity of $\mathcal{A}$ should be at least as good as those of the state-of-the-art non-adaptive IM algorithms.

2. The expected value of $\mathcal{A}$'s absolute error factor ρ_i should be much smaller than the input threshold ϵ_i.

In the following, we present a new non-adaptive IM algorithm, referred to as EPIC[1], that satisfies both of the above requirements. Towards that end, we first introduce the concept of *reverse reachable sets (RR-sets)* [7], which is the basis of our algorithm.

RR-Sets. In a nutshell, RR-sets are subgraph samples of G that can be used to efficiently estimate the expected spreads of any given seed sets. Specifically, a random RR-set of G is generated by first selecting a node $v \in V$ uniformly at random, and then taking the nodes that can reach v in a random graph generated by by independently removing each edge $e \in E$ with probability $1 - p(e)$. If a seed node set S has large expected influence spread, then the probability that S intersects with a random RR-set is high, as shown in the following equation [7]:

$$\mathbb{E}\{I_G(S)\} = n \cdot \mathbb{P}\{R \cap S \neq \emptyset\}, \tag{4}$$

where R is a random RR-set. This result suggests a simple method for estimating the expected influence spread of any node set S: we can use a set $\mathcal{R}$ of random RR-sets to estimate the value of $\mathbb{P}\{R \cap A \neq \emptyset\}$ and hence $\mathbb{E}\{I_G(S)\}$. In particular, let $Cov_{\mathcal{R}}(S)$ denote the number of RR-sets in $\mathcal{R}$ that overlap S. Then the value of $\mathbb{E}\{I_G(S)\}$ can be unbiasedly estimated by $n \cdot F_{\mathcal{R}}(S)$, where

$$F_{\mathcal{R}}(S) = Cov_{\mathcal{R}}(S)/|\mathcal{R}| \tag{5}$$

By the law of large numbers, $n \cdot F_{\mathcal{R}}(S)$ should converge to $\mathbb{E}\{I_G(S)\}$ when $|\mathcal{R}|$ is large, which provides a way to estimate $\mathbb{E}\{I_G(S)\}$ to any desired accuracy level. However, due to the cost of generating RR-sets, there is a tradeoff between accuracy and efficiency in any algorithm using RR-set sampling.

The EPIC Algorithm Algorithm 2 shows the pseudo-code of our EPIC algorithm. At the fist glance, EPIC is similar to the *SSA* algorithm in [17] as they both (i) start from a small number of RR-sets and (ii) iteratively increase the RR-set number until a satisfactory solution is identified. The main difference between the two algorithm lies in the way that they generate RR-sets in each iteration. In particular, in SSA [17], the number of RR-sets generated in each iteration is a random number, which makes it rather difficult to derive the algorithm's time complexity or its expected approximation guarantee. (This could explain the absence of formal time complexity analysis in [17].) In contrast, in each iteration of EPIC, it uses a number of RR-sets that is fixed based on the number of preceding iterations, which enables us to derive rigorous bounds on its worst-case approximation guarantee, expected time complexity,

[1] Expected approximation for influence maximization.

Algorithm 2: The EPIC Algorithm

input : G_i, ϵ_i, δ_i, and b.
output: The seed set S_i selected in the ith batch.

1 $\gamma_{i,1} = \frac{\epsilon_i}{6},\ \gamma_{i,3} = \frac{\epsilon_i}{2},\ \gamma_{i,2} = \frac{\epsilon_i - \gamma_{i,1} - c\gamma_{i,3}}{1+\gamma_{i,1}}$
2 $\Upsilon_1 = \frac{(4e-8)(1+\gamma_{i,1})(1+\gamma_{i,2})}{\gamma_{i,3}^2} \ln(3/\delta_i)$
3 $T_{max} = \frac{(8+2\epsilon_i)n_i}{b\epsilon_i^2} \left(\ln \frac{2}{\delta_i} + \ln \binom{n_i}{b}\right), \omega = \left\lceil \log_2(\frac{T_{max}}{\Upsilon_1}) \right\rceil$
4 $\Upsilon_2 = 1 + \frac{(4e-8)(1+\gamma_{i,2})}{\gamma_{i,2}^2} \ln \frac{3\omega}{\delta_i}$
5 Generate a set $\mathcal{R}_1$ of Υ_1 random RR sets
6 **repeat**
7 $\quad \langle S_i, F_{\mathcal{R}_1}(S_i)\rangle \leftarrow$ Max-Coverage$(\mathcal{R}_1, b)$
8 $\quad$ **if** $|\mathcal{R}_1| \cdot F_{\mathcal{R}_1}(S_i) \geq \Upsilon_1$ **then**
9 $\quad\quad$ Generate $|\mathcal{R}_1|$ random RR sets into $\mathcal{R}_2$
10 $\quad\quad$ Calculate $F_{\mathcal{R}_2}(S_i)$ of S_i in $\mathcal{R}_2$
11 $\quad\quad$ **if** $|\mathcal{R}_2| \cdot F_{\mathcal{R}_2}(S_i) \geq \Upsilon_2$ **then**
12 $\quad\quad\quad$ **if** $F_{\mathcal{R}_1}(S_i) \leq (1+\gamma_{i,1}) F_{\mathcal{R}_2}(S_i)$ **then**
13 $\quad\quad\quad\quad$ **return** S_i
14 $\quad \mathcal{R}_1 = \mathcal{R}_1 \cup \mathcal{R}_2$
15 **until** $|\mathcal{R}_1| \geq T_{max}$;
16 **return** S_i

Algorithm 3: The MaxCover Algorithm

input : A set $\mathcal{R}$ of random RR sets, and b.
output: A node set S_i, and the fraction of RR sets in $\mathcal{R}$ covered by S_i.

1 $S_i = \emptyset$
2 **for** $i = 1$ *to* b **do**
3 $\quad v = \arg\max_{v' \in V} (Cov_{\mathcal{R}}(S_i \cup \{v'\}) - Cov_{\mathcal{R}}(S_i))$
4 $\quad$ Insert v into S_i
5 **return** $\langle S_i, Cov_{\mathcal{R}}(S_i)/\mathcal{R}\rangle$

and expected approximation ratio. In what follows, we discuss the details of EPIC and its subroutine MaxCover (in Algorithm 3).

Based on the RR-set sampling method described previously, a simple approach for selecting S_i with a large expected influence spread is to first generate a set $\mathcal{R}$ of RR-sets, and then invoke the MaxCover algorithm on $\mathcal{R}$. In particular, MaxCover uses a simple greedy approach to identify $S_i \subseteq V$ such that S_i overlaps with as many RR-sets in $\mathcal{R}$ as possible. Since $F_{\mathcal{R}}(\cdot)$ is a submodular function for any set $\mathcal{R}$ of RR-sets [7], the set S_i found by such an approach ensures that

$$F_{\mathcal{R}}(S_i) \geq cF_{\mathcal{R}}(S_i^o), \tag{6}$$

where S_i^o is an optimal seed set in G_i. Note that $n \cdot F_{\mathcal{R}}(S_i)$ and $n \cdot F_{\mathcal{R}}(S_i^o)$ are unbiased estimations of the expected influence spread of S_i and S_i^o, respectively. Therefore, when $|\mathcal{R}|$ is large, the approximation guarantee of S_i converges to c according to Equation (6).

To strike a balance between the quality of S_i and the number of RR-sets used to derive S_i, EPIC iterates in a careful manner as follows. In each iteration, it maintains two sets of random RR-sets $\mathcal{R}_1$ and $\mathcal{R}_2$ with $|\mathcal{R}_1| = |\mathcal{R}_2|$. It invokes MaxCover on $\mathcal{R}_1$ to identify a seed set S_i, and then utilizes $\mathcal{R}_2$ to test whether S_i provides a good approximation guarantee. Initially, the cardinalities of $\mathcal{R}_1$ and $\mathcal{R}_2$ are small constants determined by the parameter Υ_1 in Line 2 in the first iteration of EPIC. Then, whenever EPIC finds that the quality of the seed set S_i generated in an iteration is not satisfactory, it doubles the sizes of $\mathcal{R}_1$ and $\mathcal{R}_2$. This process repeats until a satisfying solution is found or $\mathcal{R}_1$ and $\mathcal{R}_2$ reaches an upper bound T_{max} (Line 15).

As explained before, one of the main designing goals for EPIC is to achieve a worst-case approximation ratio of $c - \epsilon_i$, as with the state-of-the-art IM algorithms. EPIC achieves this goal by a series of operations in each iteration, whose implications are briefly explained in the following.

In each iteration, EPIC first applies MaxCover on $\mathcal{R}_1$ (Line 7), which returns a seed set S_i satisfying

$$F_{\mathcal{R}_1}(S_i) \geq cF_{\mathcal{R}_1}(S_i^o) \tag{7}$$

After that, EPIC uses $\mathcal{R}_2$ to estimate the expected spread of S_i (i.e., $\mathbb{E}\{I_{G_i}(S_i)\}$). Observe that $|\mathcal{R}_2|F_{\mathcal{R}_2}(S_i)$ is a binomial random variable due to Equation (5). Accordingly, EPIC uses the Chernoff bound to set a threshold Υ_2 (Line 4) such that, if the condition $|\mathcal{R}_2| \cdot F_{\mathcal{R}_2}(S_i) \geq \Upsilon_2$ in Line 11 is satisfied, then

$$n_i F_{\mathcal{R}_2}(S_i) \leq (1+\gamma_{i,2})\mathbb{E}\{I_{G_i}(S_i)\} \tag{8}$$

should hold with high probability. Intuitively, Equation (8) implies that $|\mathcal{R}_2|$ is large enough such that $n_i F_{\mathcal{R}_2}(S_i)$ is a sufficiently accurate estimation of the expected influence spread of S_i in G_i. After that, EPIC further checks whether

$$F_{\mathcal{R}_1}(S_i) \leq (1+\gamma_{i,1}) F_{\mathcal{R}_2}(S_i) \tag{9}$$

holds in Line 12. Intuitively, if Equation (9) is true, then we know that $n_i \cdot F_{\mathcal{R}_1}(S_i)$ is also a sufficiently accurate estimation of the expected spread of S_i in G_i. Note that $\mathbb{E}\{I_{G_i}(S_i)\} \leq OPT_b(G_i)$. Therefore, if the estimation of $\mathbb{E}\{I_{G_i}(S_i)\}$ using $\mathcal{R}_1$ is sufficiently accurate, then the estimation of $OPT_b(G_i)$ using $\mathcal{R}_1$ should also be sufficiently accurate due to the Chernoff Bound. Thus, when Equation (9) and the inequality $|\mathcal{R}_1| \cdot F_{\mathcal{R}_1}(S_i) \geq \Upsilon_1$ in Line 8 hold, then

$$n_i \cdot F_{\mathcal{R}_1}(S_i^o) \geq (1-\gamma_{i,3}) OPT_b(G_i) \tag{10}$$

holds with high probability. Combining Equations. (7)–(10), we can derive a quantitative relationship between $\mathbb{E}\{I_{G_i}(S_i)\}$ and $OPT_b(G_i)$ when S_i is returned:

$$\begin{aligned} & (1+\gamma_{i,2})\mathbb{E}\{I_{G_i}(S_i)\} \geq n_i F_{\mathcal{R}_2}(S_i) \\ \geq\ & \frac{n_i F_{\mathcal{R}_1}(S_i)}{1+\gamma_{i,1}} \geq \frac{cn_i F_{\mathcal{R}_1}(S_i^o)}{1+\gamma_{i,1}} \\ \geq\ & \frac{c(1-\gamma_{i,3}) OPT_b(G_i)}{1+\gamma_{i,1}}. \end{aligned} \tag{11}$$

This proves the $(c - \epsilon_i)$ worst-case approximation ratio of $\mathbb{E}\{I_{G_i}(S_i)\}$ as $\epsilon_i = \gamma_{i,1} + \gamma_{i,2} + \gamma_{i,1}\gamma_{i,2} + c\gamma_{i,3}$.

4.4 Theoretical Analysis

Based on the discussions in Section 4.3, we prove the worst-case approximation guarantee and time complexity of EPIC as follows.

THEOREM 2. *With a probability of at least* $1-\delta_i$, EPIC *returns a seed set* S_i *satisfying*

$$\mathbb{E}\{I_{G_i}(S_i)\} \geq (c-\epsilon_i) OPT_b(G_i)$$

for any G_i. *In addition, the expected time complexity of* EPIC *is* $O((b \log n_i + \log \frac{1}{\delta_i})(m_i + n_i)/\epsilon_i^2)$, *where* n_i *and* m_i *are the numbers of nodes and edges of* G_i, *respectively.*

Due to the space constraint, we omit the proofs of Theorem 2 and its corollary (i.e., Lemma 2), and include them in our technical

report [1]. In what follows, we analyze the expected approximation guarantee of EPIC, and show that instantiating AdaptGreedy using EPIC can lead to improved performance for adaptive IM.

4.4.1 Expected Approximation Guarantee

To derive the expected approximation guarantee of EPIC, we would need to compute the expectation of the absolute error factor ξ_i of EPIC. We observe that ξ_i consists of two components: the estimation error on $I_{G_i}(S_i)$ and the estimation error on $I_{G_i}(S_i^o)$, where S_i^o is an optimal seed set in G_i. In what follows, we analyze these two components in detail.

Recall that EPIC utilizes the set $\mathcal{R}_1$ of random RR-sets to find S_i, and then employs $\mathcal{R}_2$ to estimate $\mathbb{E}\{I_{G_i}(S_i)\}$. This ensures that $\mathcal{R}_2$ is independent of S_i and $\mathcal{R}_1$ is independent of S_i^o. Based on this property and the definition of $F_{\mathcal{R}}(\cdot)$ in Equation (5), we know that $|\mathcal{R}_1|F_{\mathcal{R}_1}(S_i^o)$ and $|\mathcal{R}_2|F_{\mathcal{R}_2}(S_i)$ are both random variables following binomial distributions $Bin\left(|\mathcal{R}_1|, \frac{\mathbb{E}\{I_{G_i}(S_i^o)\}}{n_i}\right)$ and $Bin\left(|\mathcal{R}_2|, \frac{\mathbb{E}\{I_{G_i}(S_i)\}}{n_i}\right)$, respectively. By the well known bound on the mean absolute deviation of binomial random variables (see Lemma 4 in the appendix), we have

$$\mathbb{E}\left\{\left|F_{\mathcal{R}_1}(S_i^o) - \frac{OPT_b(G_i)}{n_i}\right|\right\} \leq \sqrt{\frac{OPT_b(G_i)}{n_i|\mathcal{R}_1|}} \tag{12}$$

$$\mathbb{E}\left\{\left|F_{\mathcal{R}_2}(S_i) - \frac{\mathbb{E}\{I_{G_i}(S_i)\}}{n_i}\right|\right\} \leq \sqrt{\frac{OPT_b(G_i)}{n_i|\mathcal{R}_2|}} \tag{13}$$

Intuitively, Equation (12) provides an upper bound of the "estimation error" of $\mathbb{E}\{I_{G_i}(S_i^o)\}$ using $|\mathcal{R}_1|$, while Equation (13) gives an upper bound of the estimation error of $\mathbb{E}\{I_{G_i}(S_i)\}$ using $|\mathcal{R}_2|$. Moreover, the absolute error factor ξ_i of EPIC can be represented by these estimation errors, due to Equations (7)–(10). Combining these results, we obtain the following lemma:

LEMMA 1. *Suppose that $\mathcal{R}_1$ is the set of random RR-sets used by* EPIC *to identify S_i in the last iteration of* EPIC. *Let $\gamma_{i,1}$ be the parameter set in Line 1 of* EPIC. *Then,*

$$\mathbb{E}\{\xi_i \mid \mathcal{R}_1\} \leq (c+1)\sqrt{\frac{n_i}{|\mathcal{R}_1| \cdot OPT_b(G_i)}} + c\gamma_{i,1} \tag{14}$$

Meanwhile, the following theorem shows that, when EPIC terminates, the size of $\mathcal{R}_1$ is likely to be large.

LEMMA 2. *Let $\gamma_{i,3}$ be the parameter set in Line 1 of* EPIC. *When* EPIC *stops, we must have*

$$|\mathcal{R}_1| \geq \frac{n_i(4e-8)}{\gamma_{i,3}^2 \cdot OPT_b(G_i)} \ln(3/\delta_i)$$

with the probability of at least $1 - \delta_i/3$.

Combining Lemma 1 and Lemma 2, we immediately get the following bound on the expected approximation ratio of EPIC in each batch i:

THEOREM 3. *For each batch i, define λ_i, β_i as*

$$\lambda_i = (c+1)\frac{\gamma_{i,3}}{2\epsilon_i}\frac{1}{\sqrt{(e-2)\ln(3/\delta_i)}} + \frac{\gamma_{i,1}c}{\epsilon_i} \tag{15}$$

$$\beta_i = \lambda_i + (c - \lambda_i\epsilon_i)\delta_i/(3\epsilon_i) \tag{16}$$

where $\gamma_{i,3}$ and $\gamma_{i,1}$ are the parameters set in Line 1 of EPIC, *and ϵ_i, δ_i are the input parameters to* EPIC *in batch i. Then, we have $\mathbb{E}\{\xi_i\} \leq \beta_i\epsilon_i$, and hence, the expected approximation guarantee of* EPIC *is at least $c - \beta_i\epsilon_i$.*

PROOF. Using Lemma 1 and Lemma 2, we can get

$$\begin{aligned}\mathbb{E}\{\xi_i\} &= \mathbb{E}\{\mathbb{E}\{\xi_i \mid \mathcal{R}_1\}\} \\ &\leq \lambda_i\epsilon_i(1-\delta_i/3) + c\delta_i/3 = \beta_i\epsilon_i\end{aligned} \tag{17}$$

Hence, the lemma follows. □

By Theorem 3, the expected approximation guarantee of EPIC can be much better than its worst-case approximation guarantee (i.e., $c - \epsilon_i$), as long as β_i is considerably smaller than 1. In Section 4.4.2, we investigate how this property can be exploited to develop an improved instantiation of AdaptGreedy based on EPIC.

4.4.2 Performance Improvement for AdaptGreedy

In this section, we consider the instantiation of AdaptGreedy with EPIC, and aim to derive an improved approximation guarantee for AdaptGreedy based on the results in Section 4.4.1. Towards this end, we utilize Azuma's inequality:

LEMMA 3. *(Azuma's inequality [10]) Let $Y_1, \cdots, Y_r$ be any sequence of random variables satisfying $Y_i \leq \alpha$ and*

$$\mathbb{E}\{Y_i \mid Y_1, \cdots, Y_{i-1}\} \leq \vartheta$$

for every $i \in [r]$. Then we have

$$\Pr\left[\sum_{i=1}^{r} Y_i > r\vartheta + z\sqrt{r}\alpha\right] \leq \exp\{-z^2/2\} \tag{18}$$

Observe that Azuma's inequality provides a concentration bound for possibly correlated random variables. Recall that we have shown in Theorem 1 that the approximation guarantee of AdaptGreedy is determined by the summation of the absolute error factors of the non-adaptive IM algorithm (i.e., EPIC in our case) used to select each batch of seed nodes, and these absolute error factors could be correlated. Therefore, if we consider them as the random variables in Azuma's inequality, we can get a bound on their summation, based on which we can derive the overall approximation guarantee by Theorem 1. However, there is one issue in this approach: the Azuma's inequality requires that the random variables considered have a deterministic upper bound (i.e., α in Lemma 3), which is not the case for our absolute error factors. Fortunately, due to the worst-case approximation guarantee of EPIC, its absolute error factor is bounded by an input parameter (i.e., ϵ_i) with high probability. We leverage this property to apply Azuma's inequality, by considering a "truncated" version of each absolute error factor, which is defined as the minimum of the absolute error factor and the input parameter ϵ_i. We derive a concentration result for such truncated variables, and then extend it to prove the following theorem that establishes the approximation guarantee of AdaptGreedy when instantiated by EPIC.

THEOREM 4. *Suppose that we instantiate* AdaptGreedy *using* EPIC *with the parameters (ϵ_i, δ_i) in each batch i. Define $\epsilon = \max\{\epsilon_1, \cdots, \epsilon_r\}$ and $\beta = \max\{\beta_1, \cdots, \beta_r\}$, where β_i is defined in Theorem 3. For any given $\delta' \in (0,1)$ and $\xi \in (0,1)$, if*

$$\epsilon = \max\{(\beta + \sqrt{(2/r)\ln(1/\delta')})^{-1}\xi, \xi\} \tag{19}$$

then AdaptGreedy *achieves the approximation ratio shown in Theorem 1 with a probability of at least $1 - \delta' - \sum_{i=1}^{r}\delta_i$.*

Recall that the expected approximation guarantee of EPIC can be better than its worst-case approximation guarantee, in which case the parameter β in Equation (19) can be much smaller than 1. In that case, Equation (19) indicates that ϵ can be larger than ξ, especially when the round number $r \geq 2\ln(1/\delta') = 2\ln n$ (assuming

$\delta' = 1/n$). Note that $\epsilon = \max\{\epsilon_1, \cdots, \epsilon_r\}$, and we have proved in Theorem 2 that EPIC has the same time complexity with the existing IM algorithms under the same input parameters. This indicates that we can use some $\epsilon_1, \cdots, \epsilon_r$ satisfying $\sum_{i=1}^{r} \epsilon_i/r > \xi$ as the input of EPIC in each batch i, while still achieving the approximation ratio shown in Theorem 1, but under a smaller time complexity compared with that of using the existing IM algorithms. In other words, Theorem 4 shows that instantiating AdaptGreedy using EPIC can achieve tighter approximation guarantee under the same time complexity compared with that of the existing IM algorithms.

5. RELATED WORK

Non-Adaptive Influence Maximization: The IM problem under the non-adaptive setting has been extensively studied. The seminal work of Kempe et al. [16] shows that there is a $1-1/e-\epsilon$ approximation guarantee for the non-adaptive IM problem, and it proposes a monte carlo simulation algorithm to achieve this approximation ratio with high time complexity. After that, a lot of studies have appeared to improve Kempe et al.'s work in terms of time efficiency. Among these works, Borgs. et al. [7] propose the RR-set sampling method for influence spread estimation, and several later studies [17, 20, 23, 24] use this method to find more efficient algorithms for the IM problem. However, all these studies concentrate on the non-adaptive IM problem, and hence their approximation guarantees do not hold for the adaptive IM problem.

Adaptive Influence Maximization: Compared with the studies on non-adaptive IM, the studies on adaptive IM are relatively few. Golovin et al. [11] derive a $1 - 1/e$ approximation ratio under the case that only one seed node can be selected in each batch. Chen et al. [9], Vaswani and Lakshmanan [25] study adaptive seed selection under the case that more than one seed nodes can be selected in each batch. Nevertheless, Chen et al. [9] aim to minimize the cost of the selected seeds under the constraint that the influence spread is larger than a given threshold, which is a different goal from ours. Vaswani and Lakshmanan [25] derive an approximation guarantee $1 - \exp\left(-\frac{(1-1/e)^2}{\eta}\right)$ for certain $\eta > 1$. Unfortunately, none of the studies listed above provide a practical algorithm to achieve the claimed approximation ratios. More specifically, Golovin et al. [11] and Chen et al. [9] assume that the expected influence spread can be exactly computed in polynomial time (which is not true due to [8]), while Vaswani and Lakshmanan [25] did not provide a method to bound the key parameter η appearing in their approximation ratio.

We also notice that Seeman et al. [19], Horel et al. [14] and Badanidiyuru et al. [4] consider an influence maximization problem called "adaptive seeding", but with totally different implication from ours. More specifically, they assume that the seed nodes can be selected in two stages. In the first stage, a set S can be selected from a given node set $X \subseteq V$. In the second stage, another seed set T can be selected from the influenced neighboring nodes of S. The goal of their problem is to maximize the expected influence spread of T, under the constraint that the total number of nodes in $S \cup T$ is no more than k. However, the problem model and optimization goal of these studies are both very different from ours, and hence their methods cannot be applied to our problem.

6. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed approach using extensive experiments. The goal of our experiments is to test the efficiency and effectiveness of AdaptGreedy using real social networks. All of our experiments are conducted on a Linux machine with an Intel Xeon 2.6GHz CPU and 256GB RAM.

Table 2: Dataset details. ($\mathbf{K = 10^3, M = 10^6, G = 10^9}$)

Dataset	n	m	Type	Avg. degree
NetHEPT	15.2K	31.4K	undirected	4.18
Epinions	132K	841K	directed	13.4
DBLP	655K	1.99M	undirected	6.08
LiveJournal	4.85M	69.0M	directed	28.5
Orkut	3.07M	117M	undirected	76.2

6.1 Experimental Setting

Datasets: We use five real datasets in our experiments, as shown by Table 2. All these datasets are downloaded from [2]. To the best of our knowledge, only Vaswani and Lakshmanan [25] have used real social networks to test adaptive IM algorithms (but without an approximation guarantee), and the largest network used by them only has 75k nodes and 500k edges [25]. Note that the number of edges in Orkut is about 234 times of that of the largest dataset in [25]. Therefore, as far as we know, our datasets are the largest ones for testing adaptive IM algorithms in the literature. We also generate 20 possible worlds for each dataset to test the performance of our algorithms, and the reported data are the average results on these possible worlds.

Algorithms: As we discuss in Section 2.3, there are only two existing methods [11, 25] for adaptive IM. They both require using a non-adaptive IM algorithm that is both efficeint and extremely accurate, but none of the existing non-adaptive IM algorithms satisfy such requirements. Consequently, the methods in [11, 25] do not allow practical implementations without invlidating their theoretical results. Instead, we implement two adaptive IM algorithms *AdaptIM-1* and *AdaptIM-2* by instantiating AdaptGreedy using EPIC. The difference between *AdaptIM-1* and *AdaptIM-2* is that *AdaptIM-1* achieves the approximation guarantee shown in Theorem 1 by leveraging the worst-case approximation guarantee of EPIC (in the way explained by Sec. 4.1), while *AdaptIM-2* leverages Theorem 4 to achieve the same approximation guarantee. The purpose of implementing *AdaptIM-1* and *AdaptIM-2* is to test whether the method proposed in Sec. 4.4.2 is effective, i.e., whether the performance of AdaptGreedy can be improved by leveraging the small approximation error of EPIC.

We also test two state-of-the-art non-adaptive IM algorithms (i.e., *D-SSA* [18] and *IMM* [23]) in our experiments. *IMM* is obtained from [3], and *D-SSA* is obtained from [18]. The purpose of using *D-SSA* and *IMM* in our experiments is to test whether we can achieve larger influence spread by adaptively selecting seed nodes, compared with the non-adaptive IM algorithms such as *D-SSA* and *IMM*.

Parameter settings: We use the popular independent cascade (IC) model [16] in our experiments. Following a large body of existing work on influence maximization [7,16,17,23,24], we set the propagation probability of each edge $e = (u, v)$ to $\frac{1}{d_{in}(v)}$, where $d_{in}(v)$ is the in-degree of the node v.

Given any $\xi \in (0, 1)$ and $\delta \in (0, 1)$, the goal of both *AdaptIM-1* and *AdaptIM-2* is to find a $1 - \exp\{\xi - c\}$ approximation solution with probability of at least $1 - \delta$. To achieve this goal, we set $\delta_1 = \cdots = \delta_r = \delta/r$ and $\epsilon_1 = \cdots = \epsilon_r = \xi$ for *AdaptIM-1*. As *AdaptIM-2* leverages Theorem 4 to improve its performance, we set $\delta_1 = \cdots = \delta_r = \delta/2r$, $\delta' = \delta/2$ and $\epsilon_1 = \cdots = \epsilon_r = \epsilon$

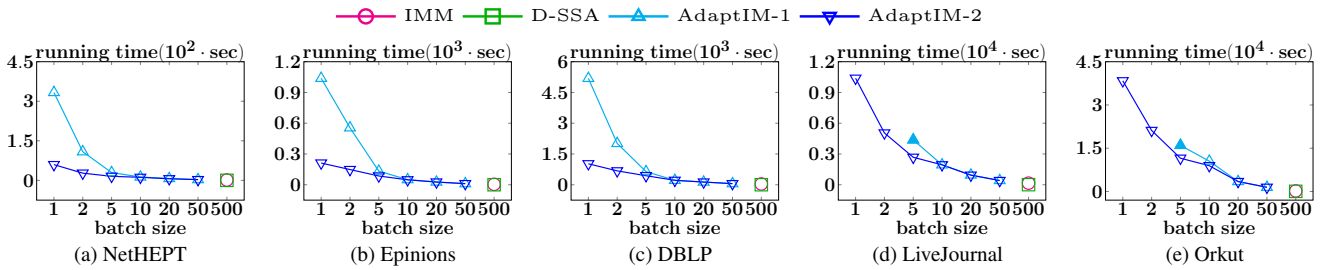


Figure 3: Running time vs. batch size

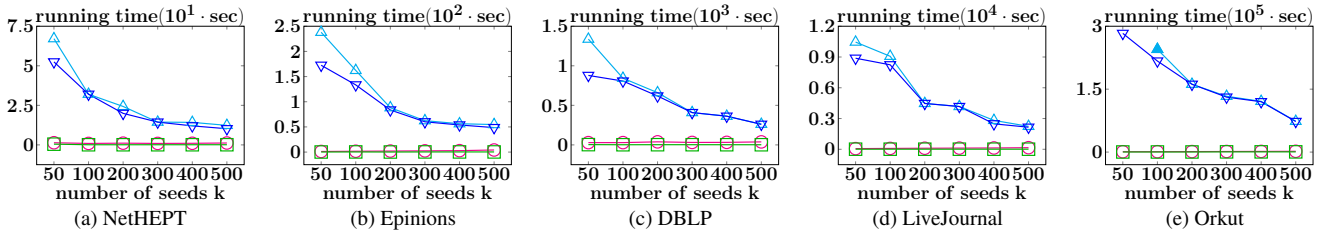


Figure 4: Running time vs. seed size

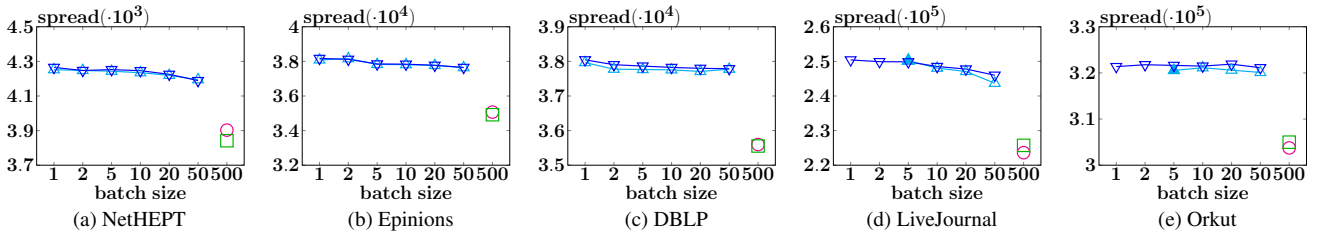


Figure 5: Spread vs. batch size

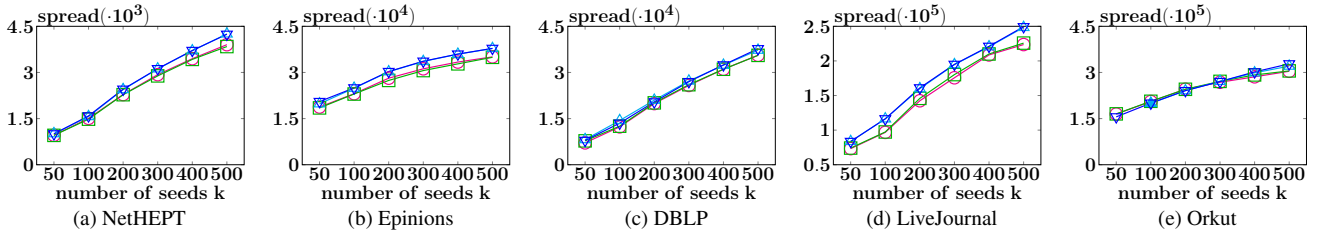


Figure 6: Spread vs. seed size

for *AdaptIM-2*, where ϵ, δ' and ξ satisfy the relationship shown in Eqn. (19). We also set $\delta = 1/n$ and $\xi = 0.1$ in all our experiments.

Recall that we need to select k nodes in r batches in adaptive IM, where $b = k/r$ nodes are selected in each batch. To see how the performance of our algorithms is impacted by k, b and r, we set these parameters according to the *b-setting* and *k-setting* explained as follows. Under the *b-setting*, we fix $k = 500$ and vary b such that $b \in \{1, 2, 5, 10, 20, 50, 500\}$. Under the *k-setting*, we fix $r = 50$ and vary k such that $k \in \{50, 100, 200, \cdots, 500\}$.

6.2 Comparing the Running Time

In this section, we compare the time efficiency of the implemented algorithms by varying b, k and r.

We first plot our experimental results under the *b-setting* in Fig. 3, where k is fixed to 500 and b scales from 1 to 500. As both *IMM* and *D-SSA* are non-adaptive IM algorithms that select all 500 seed nodes in one batch, we can only test their performance under the *b-setting* for $b = 500$. For *AdaptIM-1* and *AdaptIM-2*, we test their running time under the case of $b < 500$ in Fig. 3. The experimental results in Fig. 3 show that *IMM* and *D-SSA* have the smallest running time, which is not surprising given that they only run for one batch, and hence, generate a smaller number of RR-sets. Besides, it can be seen that the time efficiency of *AdaptIM-2* is significantly better than *AdaptIM-1*, especially when b is small (i.e., r is large). For example, *AdaptIM-2* runs almost 4 times faster than *AdaptIM-1* on the Epinions dataset when $b = 1$. We also notice that, *AdaptIM-1* even cannot finish under the case of $b < 5$ for the largest datasets LiveJournal and Orkut, due to the memory overflow. To explain, reall that *AdaptIM-2* leverages Theorem 4 to set its input parameter ϵ, due to which ϵ can be much larger than ξ, especially when r is large. Consequently, the number of RR-sets generated in *AdaptIM-2* can be much smaller than that in *AdaptIM-1*, and hence *AdaptIM-2* achieves better time efficiency and less memory consumption.

In Fig. 4, we plot our experimental results under the *k-setting*, where r is fixed to 50 and k varies from 50 to 500. The results show similar trends as those in Fig. 3, which can be explained by similar reason as that for Fig. 3. Besides, Fig. 4 reveals that the performance gain of *AdaptIM-2* with respect to *AdaptIM-1* can be more prominent when k is small under the *k-setting*. This can be

explained as follows. As r is fixed to 50 under the *k-setting*, b must increase with k, due to which the optimal influence spread $OPT_b(G_i)$ also tends to increase with k for each batch i. Consequently, *AdaptIM-1* and *AdaptIM-2* are both less sensitive to their input parameter ϵ_i when k gets larger, as they both generate fewer RR-sets in each batch; thus, they both achieve better time efficiency when $OPT_b(G_i)$ gets larger.

6.3 Comparing the Influence Spread

In this section, we study the performance of the implemented algorithms on the influence spread, and the experimental results are shown in Fig. 5 and Fig. 6. The parameter settings in Fig. 5 and Fig. 6 are the same with those in Fig. 3 and Fig. 4, respectively.

We first study the performance of the implemented algorithms under the *b-setting* in Fig. 5, where k is fixed to 500 and b scales from 1 to 500. It can be seen that *AdaptIM-1* and *AdaptIM-2* achieve similar influence spreads, which proves the effectiveness of *AdaptIM-2*, as *AdaptIM-2* achieves better time efficiency than *AdaptIM-1*. Moreover, the influence spreads of *AdaptIM-1* and *AdaptIM-2* both tend to decrease when b increases. This can be explained as follows. Under the *b-setting*, r decreases when b increases, which implies that both *AdaptIM-1* and *AdaptIM-2* become "less adaptive" when b increases. Consequently, they both could activate fewer nodes. In fact, when $b = 500$, all the seed nodes must be non-adaptively selected in one batch. This explains why *IMM* and *D-SSA* achieve much worse influence spread than *AdaptIM-1* and *AdaptIM-2* do.

Finally, we study the influence spread under the *k-setting* in Fig. 6, where r is fixed to 50 and k scales from 50 to 500. It can be seen that all the influence spreads of the implemented algorithms increase with k, which is due to the reason that selecting more seed nodes causes a larger influence spread. Moreover, both the influence spreads of *AdaptIM-1* and *AdaptIM-2* outperform those of *IMM* and *D-SSA*, as they can activate more nodes by adaptively selecting seed nodes. Indeed, Fig. 6 shows that *AdaptIM-1* and *AdaptIM-2* can achieve more than ten percentage gain on the influence spread for LiveJournal. We also note that the gain on the influence spread brought by adaptively selecting seed nodes can be affected by the network itself, as different social networks have different topologies and different possible worlds.

7. CONCLUSION

We have studied the adaptive Influence Maximization (IM) problem, where the seed nodes can be selected in multiple batches to maximize their influence spread. We have proposed the first practical algorithms to address the adaptive IM problem that achieve both time efficiency and provable approximation guarantee. Our approach is based on a novel AdaptGreedy framework instantiated by a new non-adaptive IM algorithm EPIC, which has a provable expected approximation guarantee. We also have conducted extensive experiments using real social network to test the performance of our algorithms, and the experimental results strongly corroborate the superiorities and effectiveness of our approach.

8. ACKNOWLEDGEMENT

This work is partially supported by National Natural Science Foundation of China under Grant No.61772491, No.61472460, by Natural Science Foundation of Jiangsu Province under Grant No.BK20161256, by MOE, Singapore under grant MOE2017-T1-002-024, MOE2015-T2-2-069, by NUS, Singapore under an SUG, and by NRF, Singapore under grant NRF-RSS2016-004.

APPENDIX

A. MISSING LEMMAS AND PROOFS

LEMMA 4. *[6] Suppose that $X \sim Bin(n,p)$ is a binomial random variable. Then the mean absolute deviation (MAD) of X (i.e., $\mathbb{E}\{|X - \mathbb{E}\{X\}|\}$) is no more than $\sqrt{np(1-p)}$.*

A.1 Proof of Theorem 1

In this section, we first introduce some definitions and lemmas, and then use them to prove Theorem 1. Our theoretical analysis borrows some concepts from [11], but is considerably different from that in [11]. This is mainly due to the reason that our AdaptGreedy framework allows different and even random approximation guarantee in different batches, while [11] requires that the approximation guarantees in all batches are identical and fixed constants.

For convenience, we call any strategy that selects seed nodes in the way explained by Section. 2.2 as an *adaptive seeding policy*. To analyze the performance ratio of AdaptGreedy, we formally define the expected influence spread of any adaptive seeding policy in Definition 1, and then introduce the "truncation" and "concatenation" operations on policies in Definition 2 and Definition 3, respectively.

DEFINITION 1. *(Influence spread of policy) Given any adaptive seeding policy Λ, let $N(\Lambda)$ denote the set of all seed nodes that would be selected by Λ. The expected influence spread of Λ is defined as $\pi(\Lambda) = \mathbb{E}_{w\sim\mathcal{W}}\{I_w(N(\Lambda))\}$, where w and $\mathcal{W}$ are explained in Section 2.1.*

DEFINITION 2. *(Policy truncation) For any adaptive seeding policy Λ, the policy truncation $\Lambda_{[i]}$ denotes an adaptive policy that performs exactly the same as Λ, except that $\Lambda_{[i]}$ only selects the first i ($i \leq r$) batches of nodes.*

DEFINITION 3. *(Policy concatenation) For any two adaptive seeding policy Λ and Λ', the policy concactenation $\Lambda \oplus \Lambda'$ denotes an adaptive policy that first executes the policy Λ, and then executes Λ' on the residue graph output by Λ, but without any knowledge about Λ.*

With the above definitions, we further introduce the concept of optimal marginal gain of any adaptive seeding policy in Definition 4, and then introduce Lemma 5 and Lemma 6, which are useful for proving Theorem 1.

DEFINITION 4. *(Optimal marginal gain) For any adaptive policy Λ, define $G[\Lambda]$ as the graph generated by removing the nodes activated by Λ in G. Define $\Delta^*(\Lambda)$ as the maximum expected influence spread of any b seed nodes in $G[\Lambda]$, which is called the optimal marginal gain of Λ.*

LEMMA 5. *For any adaptive seeding policy Λ and any $i \geq 1$, we have*

$$\pi(\Lambda_i) - \pi(\Lambda_{[i-1]}) \leq \mathbb{E}_{w\sim\mathcal{W}}\{\Delta^*(\Lambda_{[i-1]})\} \tag{20}$$

PROOF. Suppose that C is the set of nodes selected by Λ in the ith batch. Let $K = \mathbb{E}\{I_{G[\Lambda_{[i-1]}]}(C)\}$ be the expected influence spread of C in $G[\Lambda_{[i-1]}]$. So K is always no more than than $\Delta^*(\Lambda_{[i-1]})$. Therefore, we have

$$\begin{aligned} & \pi(\Lambda_{[i]}) - \pi(\Lambda_{[i-1]}) = \mathbb{E}\{K\} \leq \mathbb{E}\{\Delta^*(\Lambda_{[i-1]})\} \quad (21) \\ = \;& \mathbb{E}_{w\sim\mathcal{W}}\{\Delta^*(\Lambda_{[i-1]})\} \end{aligned}$$

where the expectations in (21) are taken with respect to the randomness of $G[\Lambda_{[i-1]}]$. □

LEMMA 6. *Given any adaptive seeding policy Λ and any $i \leq j$, we have*

$$\mathbb{E}_{w\sim\mathcal{W}}\{\Delta^*(\Lambda_{[j]})\} \leq \mathbb{E}_{w\sim\mathcal{W}}\{\Delta^*(\Lambda_{[i]})\} \tag{22}$$

PROOF. Note that the nodes activated by $\Lambda_{[i]}$ are always activated by $\Lambda_{[j]}$ due to $i \leq j$. Therefore, $G[\Lambda_{[j]}]$ is always a subgraph of $G[\Lambda_{[i]}]$. As $\Delta^*(\Lambda_{[j]})$ and $\Delta^*(\Lambda_{[i]})$ are the maximum expected influence spread of any b seed nodes in $G[\Lambda_{[j]}]$ and $G[\Lambda_{[i]}]$, respectively, we must have $\Delta^*(\Lambda_{[j]}) \leq \Delta^*(\Lambda_{[i]})$ for any possible world $w \sim \mathcal{W}$. Hence the lemma follows. □

Intuitively, the above lemmas reveal an interesting submodular property of any adaptive policy Λ, i.e., the optimal marginal gain of Λ satisfies the "diminishing returns" properties under truncation. Using Lemma 5 and Lemma 6, we can build a quantitative relationship between AdaptGreedy's expected influence spread and the expected influence spread of the optimal adaptive seeding policy, as shown by Lemma 7:

LEMMA 7. *Let Γ and Γ^{opt} denote the* AdaptGreedy *policy and the optimal adapt seeding policy, respectively. For any $1 \leq i \leq r$, let*

$$X_i = \sum_{G_1\in\mathcal{G}_1,\ldots,\mathcal{G}_i\in\mathcal{G}_i} (\xi_i \cdot \Pr[\xi_i \mid G_1,\ldots,G_i] \cdot \Pr[G_1,\ldots,G_i])$$

Then we have

$$\pi(\Gamma_{[i]}) - \pi(\Gamma_{[i-1]}) \geq \frac{c - X_i}{r}(\pi(\Gamma^{opt}) - \pi(\Gamma_{[i-1]})) \tag{23}$$

PROOF. For any $1 \leq i \leq r$, we have:

$$\begin{aligned}
&\mathrm{OPT} - \pi(\Gamma_{[i]}) \leq \pi(\Gamma_{[i]} \oplus \Gamma^{opt}) - \pi(\Gamma_{[i]}) \\
=\ & \sum_{j=1}^{r} \left(\pi(\Gamma_{[i]} \oplus \Gamma^{opt}_{[j]}) - \pi(\Gamma_{[i]} \oplus \Gamma^{opt}_{[j-1]})\right) \\
\leq\ & \sum_{j=1}^{r} \mathbb{E}_{w\sim\mathcal{W}}\{\Delta^*(\Gamma_{[i]} \oplus \Gamma^{opt}_{[j-1]})\} && (24)\\
\leq\ & \sum_{j=1}^{r} \mathbb{E}_{w\sim\mathcal{W}}\{\Delta^*(\Gamma_{[i-1]})\} && (25)\\
=\ & r\mathbb{E}_{w\sim\mathcal{W}}\{\Delta^*(\Gamma_{[i-1]})\}, && (26)
\end{aligned}$$

where Eqn. (24) is due to Lemma 5 and Eqn. (25) is due to Lemma 6. Besides, we have

$$\begin{aligned}
&\mathbb{E}_{w\sim\mathcal{W}}\{\xi_i\Delta^*(\Gamma_{[i-1]})\} \\
=\ & \mathbb{E}\{X_i\Delta^*(\Gamma_{[i-1]})\} = X_i\mathbb{E}\{\Delta^*(\Gamma_{[i-1]})\} && (27)\\
=\ & X_i\mathbb{E}_{w\sim\mathcal{W}}\{\Delta^*(\Gamma_{[i-1]})\} && (28)
\end{aligned}$$

where the expectations in Eqn. (27) are taken with respect to the randomness of G_i, and Eqn. (27) holds because that X_i is only determined by the algorithm used to select the seed nodes in the ith batch. Using Eqn. (28), we can get

$$\begin{aligned}
&\pi(\Gamma_{[i]}) - \pi(\Gamma_{[i-1]}) \\
\geq\ & \mathbb{E}_{w\sim\mathcal{W}}\left\{c\Delta^*(\Gamma_{[i-1]}) - \xi_i\Delta^*(\Gamma_{[i-1]})\right\} && (29)\\
\geq\ & c\mathbb{E}\left\{\Delta^*(\Gamma_{[i-1]})\right\} - X_i\mathbb{E}\left\{\Delta^*(\Gamma_{[i-1]})\right\} \\
\geq\ & (c - X_i)\mathbb{E}\left\{\Delta^*(\Gamma_{[i-1]})\right\} && (30)
\end{aligned}$$

where Eqn. (29) is due to the reason that AdaptGreedy achieves an $c - \xi_i$ approximation ratio in each batch. So the lemma follows by combining Eqn. (30) and Eqn. (26). □

Intuitively, Lemma 7 reveals that the expected marginal gain of AdaptGreedy in each batch i "covers" a sufficiently large portion of the optimal expected influence spread, which leads to the proof of the AdaptGreedy's approximation ratio:

PROOF. (of Theorem 1) According to Lemma 7, we have

$$\begin{aligned}
&\pi(\Gamma^{opt}) - \pi(\Gamma) \leq \left(1 - \frac{c - X_r}{r}\right)(\pi(\Gamma^{opt}) - \pi(\Gamma_{[r-1]})) \\
\leq\ & \cdots \leq \prod_{i=1}^{r}\left(1 - \frac{c - X_i}{r}\right)\pi(\Gamma^{opt}) && (31)
\end{aligned}$$

Therefore, we have

$$\begin{aligned}
&\pi(\Gamma) \geq \left(1 - \prod_{i=1}^{r}\left(1 - \frac{c - X_i}{r}\right)\right)\pi(\Gamma^{opt}) \\
\geq\ & \left(1 - \exp\left(-\sum_{i=1}^{r}\frac{c - X_i}{r}\right)\right)\pi(\Gamma^{opt}) && (32)\\
\geq\ & (1 - \exp(\xi - c))\pi(\Gamma^{opt}) && (33)
\end{aligned}$$

Recall that $c = 1$ and $c = 1 - 1/e$ when $b = 1$ and $b > 1$, respectively. Hence the theorem follows. □

A.2 Proof of Lemma 1

PROOF. Let n_i be the number of nodes in G_i. Recall that $\mathcal{R}_1$ is used to find S_i, while $\mathcal{R}_2$ is used to estimate S_i. Define

$$\xi_{i,1} = |n_i F_{\mathcal{R}_1}(S^o_i) - OPT_b(G_i)\}|/OPT_b(G_i) \tag{34}$$
$$\xi_{i,2} = |n_i F_{\mathcal{R}_2}(S_i) - \mathbb{E}\{I_{G_i}(S_i)\}|/OPT_b(G_i) \tag{35}$$

So we have

$$\begin{aligned}
&\mathbb{E}\{I_{G_i}(S_i)\} \geq n_i F_{\mathcal{R}_2}(S_i) - \xi_{i,2}OPT_b(G_i) \\
\geq\ & \frac{n_i}{1+\gamma_{i,1}}F_{\mathcal{R}_1}(S_i) - \xi_{i,2}OPT_b(G_i) && (36)\\
\geq\ & \frac{cn_i}{1+\gamma_{i,1}}F_{\mathcal{R}_1}(S^o_i) - \xi_{i,2}OPT_b(G_i) && (37)\\
\geq\ & \frac{c}{1+\gamma_{i,1}}(1-\xi_{i,1})OPT_b(G_i) - \xi_{i,2}OPT_b(G_i) \\
\geq\ & cOPT_b(G_i) - \left[\frac{c}{1+\gamma_{i,1}}\xi_{i,1} + \xi_{i,2} + \frac{\gamma_{i,1}c}{1+\gamma_{i,1}}\right]OPT_b(G_i)
\end{aligned}$$

where Eqn. (36) is due to Line 12 of EPIC, and Eqn. (37) is due to $F_{\mathcal{R}_1}(S_i) \geq cF_{\mathcal{R}_1}(S^o_i)$. Therefore, we have $\xi_i \leq \frac{c}{1+\gamma_{i,1}}\xi_{i,1} + \xi_{i,2} + \frac{\gamma_{i,1}c}{1+\gamma_{i,1}}$. Note that $|\mathcal{R}_1|F_{\mathcal{R}_1}(S^o_i)$ and $|\mathcal{R}_2|F_{\mathcal{R}_2}(S_i)$ are both random variables following the binomial distributions $Bin(|\mathcal{R}_1|, OPT_b(G_i)/n_i)$ and $Bin(|\mathcal{R}_2|, \mathbb{E}\{I_{G_i}(S_i)\}/n_i)$, respectively. So we can use Lemma 4 to get

$$\mathbb{E}\left\{\frac{|\mathcal{R}_1|}{n_i}\xi_{i,1}OPT_b(G_i)\right\} \leq \sqrt{|\mathcal{R}_1|OPT_b(G_i)/n_i}$$
$$\mathbb{E}\left\{\frac{|\mathcal{R}_2|}{n_i}\xi_{i,2}OPT_b(G_i)\right\} \leq \sqrt{|\mathcal{R}_2|OPT_b(G_i)/n_i}$$

Note that $|\mathcal{R}_1| = |\mathcal{R}_2|$ when EPIC returns. Therefore, we have

$$\begin{aligned}
&\mathbb{E}\{\xi_i \mid \mathcal{R}_1\} \leq \mathbb{E}\left\{\frac{c}{1+\gamma_{i,1}}\xi_{i,1} + \xi_{i,2} + \frac{\gamma_{i,1}c}{1+\gamma_{i,1}}\right\} \\
\leq\ & \left(\frac{c}{1+\gamma_{i,1}} + 1\right)\sqrt{\frac{n_i}{|\mathcal{R}_1|OPT_b(G_i)}} + \frac{\gamma_{i,1}c}{1+\gamma_{i,1}} \\
\leq\ & (c+1)\sqrt{\frac{n_i}{|\mathcal{R}_1|OPT_b(G_i)}} + \gamma_{i,1}c && (38)
\end{aligned}$$

Hence the lemma follows. □

A.3 Proof of Theorem 4

PROOF. Note that Theorem 2 has proved that EPIC achieves a $c - \epsilon_i$ worst-case approximation ratio. Therefore, if $\epsilon = \xi$, then we can use the reasoning similar to that in Sec. 4.1 to prove that AdaptGreedy satisfies the approximation guarantee shown in Theorem 1 with a probability of at least $1 - \sum_{i=1}^{r} \delta_i$, and hence the theorem follows. In the sequel, we consider the case of $\epsilon > \xi$.

let $Y_i = \min\{X_i, \epsilon_i\}$ for any $i \in \{1, \cdots, r\}$, where

$$X_i = \sum_{G_1 \in \mathcal{G}_1, \ldots, \mathcal{G}_i \in \mathcal{G}_i} (\xi_i \cdot \Pr[\xi_i \mid G_1, \ldots, G_i] \cdot \Pr[G_1, \ldots, G_i])$$

then we must have $Y_i \leq \epsilon$ and

$$\begin{aligned} & \mathbb{E}\{Y_i \mid Y_1, \cdots, Y_{i-1}\} \\ \leq \; & \mathbb{E}\{X_i \mid Y_1, \cdots, Y_{i-1}\} \leq \mathbb{E}\{X_i\} \\ \leq \; & \mathbb{E}\{\xi_i\} \leq \beta_i \epsilon_i \leq \beta\epsilon \end{aligned} \tag{39}$$

for any $i \in \{1, \cdots, r\}$. Therefore, using the Azuma's inequality, we can prove

$$\Pr\left[\sum_{i=1}^{r} Y_i > r\xi\right] \leq \delta' \tag{40}$$

Let $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4$ be the following events:

$$\mathcal{E}_1 = \{\sum_{i=1}^{r} X_i > r\xi \wedge \forall i : X_i \leq \epsilon\};\ \mathcal{E}_2 = \{\exists i : X_i > \epsilon\};$$
$$\mathcal{E}_3 = \{\sum_{i=1}^{r} Y_i > r\xi\};\ \ \mathcal{E}_4 = \{\exists i : \xi_i > \epsilon\}$$

Note that we have $\forall i : X_i = Y_i$ when $\mathcal{E}_1$ happens. So we have

$$\Pr[\mathcal{E}_1] \leq \Pr[\mathcal{E}_3] \leq \delta' \tag{41}$$

due to Eqn. (40). Moreover, as EPIC achieves a $c - \epsilon_i$ worst-case approximation guarantee with probability of at least $1 - \delta_i$ (see Theorem 2), we have $\mathbb{P}\{X_i > \epsilon\} = \mathbb{P}\{\xi_i > \epsilon\} \leq \mathbb{P}\{\xi_i > \epsilon_i\} \leq \delta_i$ for all i. Using the union bound, we have

$$\Pr[\mathcal{E}_2] = \Pr[\mathcal{E}_4] \leq \sum_{i=1}^{r} \delta_i \tag{42}$$

Combining Eqn. (41) and Eqn. (42), we get

$$\Pr[\sum_{i=1}^{r} X_i > r\xi] \leq \Pr[\mathcal{E}_1] + \Pr[\mathcal{E}_2] \leq \delta' + \sum_{i=1}^{r} \delta_i$$

Hence the lemma follows. □

2. REFERENCES

[1] https://bit.ly/2I3A99p.

[2] http://snap.stanford.edu.

[3] https://sourceforge.net/projects/im-imm/.

[4] A. Badanidiyuru, C. Papadimitriou, A. Rubinstein, L. Seeman, and Y. Singer. Locally adaptive optimization: adaptive seeding for monotone submodular functions. In *SODA*, 2016.

[5] N. Barbieri, F. Bonchi, and G. Manco. Topic-aware social influence propagation models. In *ICDM*, pages 81–90, 2012.

[6] D. Berend and A. Kontorovich. A sharp estimate of the binomial mean absolute deviation with applications. *Statistics & Probability Letters*, 83(4):1254–1259, 2013.

[7] C. Borgs, M. Brautbar, J. T. Chayes, and B. Lucier. Maximizing social influence in nearly optimal time. In *SODA*, pages 946–957, 2014.

[8] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *KDD*, pages 1029–1038, 2010.

[9] Y. Chen and A. Krause. Near-optimal batch mode active learning and adaptive submodular optimization. In *Proc. of ICML*, pages 160–168, 2013.

[10] D. P. Dubhashi and A. Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, 2009.

[11] 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.

[12] K. Han, Y. He, X. Xiao, S. Tang, F. Gui, C. Xu, and J. Luo. Budget-constrained organization of influential social events. In *ICDE*, pages 917–928, 2018.

[13] K. Han, C. Xu, F. Gui, S. Tang, H. Huang, and J. Luo. Discount allocation for revenue maximization in online social networks. In *MobiHoc*, 2018.

[14] T. Horel and Y. Singer. Scalable methods for adaptively seeding a social network. In *WWW*, pages 623–624, 2015.

[15] K. Huang, S. Wang, G. S. Bevilacqua, X. Xiao, and L. V. S. Lakshmanan. Revisiting the stop-and-stare algorithms for influence maximization. *PVLDB, 10(9):913–924*, 2017.

[16] D. Kempe, J. M. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD*, pages 137–146, 2003.

[17] H. T. Nguyen, M. T. Thai, and T. N. Dinh. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *SIGMOD*, pages 695–710, 2016.

[18] H. T. Nguyen, M. T. Thai, and T. N. Dinh. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. *CoRR*, abs/1605.07990v3, Feb 22, 2017.

[19] L. Seeman and Y. Singer. Adaptive seeding in social networks. In *FOCS*, pages 459–468, 2013.

[20] J. Tang, X. Tang, X. Xiao, and J. Yuan. Online processing algorithms for influence maximization. In *SIGMOD*, 2018.

[21] J. Tang, X. Tang, and J. Yuan. Influence maximization meets efficiency and effectiveness: A hop-based approach. In *ASONAM*, pages 64–71, 2017.

[22] J. Tang, X. Tang, and J. Yuan. An efficient and effective hop-based approach for inluence maximization in social networks. *Social Network Analysis and Mining*, 8(10), 2018.

[23] Y. Tang, Y. Shi, and X. Xiao. Influence maximization in near-linear time: A martingale approach. In *SIGMOD*, pages 1539–1554, 2015.

[24] Y. Tang, X. Xiao, and Y. Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. In *SIGMOD*, pages 75–86, 2014.

[25] S. Vaswani and L. V. S. Lakshmanan. Adaptive influence maximization in social networks: Why commit when you can adapt? *arXiv: 1604.08171*, 2016.

[26] A. Yadav, H. Chan, A. Xin Jiang, H. Xu, E. Rice, and M. Tambe. Using social networks to aid homeless shelters: Dynamic influence maximization under uncertainty. In *AAMAS*, pages 740–748, 2016.