

# Online Contextual Influence Maximization in Social Networks

Ömer Sarıtaş  
 Department of  
 Industrial Engineering  
 Bilkent University

Altuğ Karakurt  
 Department of  
 Electrical and Electronics Engineering  
 Bilkent University

Cem Tekin  
 Department of  
 Electrical and Electronics Engineering  
 Bilkent University

**Abstract**—In this paper, we propose the *Online Contextual Influence Maximization Problem* (OCIMP). In OCIMP, the learner faces a series of epochs in each of which a different *influence campaign* is run to promote a certain product in a given social network. In each epoch, the learner first distributes a limited number of free-samples of the product among a set of seed nodes in the social network. Then, the influence spread process takes place over the network, other users get influenced and purchase the product. The goal of the learner is to maximize the expected total number of influenced users over all epochs. We depart from the prior work in two aspects: (i) the learner does not know how the influence spreads over the network, i.e., it is unaware of the influence probabilities; (ii) influence probabilities depend on the context. We develop a learning algorithm for OCIMP, called *Contextual Online INfluence maximization* (COIN). COIN can use any approximation algorithm that solves the offline influence maximization problem as a subroutine to obtain the set of seed nodes in each epoch. When the influence probabilities are Hölder continuous functions of the context, we prove that COIN achieves sublinear regret with respect to an approximation oracle that knows the influence probabilities for all contexts. Moreover, our regret bound holds for any sequence of contexts. We also test the performance of COIN on several social networks, and show that it performs better than other methods.

## I. INTRODUCTION

In recent years, there has been growing interest in understanding how influence spreads in a social network. This interest is motivated by the proliferation of viral marketing in social networks. For instance, nowadays many companies promote their products on social networks by giving free samples of certain products to a set of seed nodes/users, expecting them to influence people in their social circles into purchasing these products. The objective of these companies is to find out the set of nodes that can collectively influence the greatest number of other nodes in the social network. This problem is called the *influence maximization* (IM) problem.

In the IM problem, the spread of influence is modeled by an *influence graph*, where directed edges between nodes represent the paths that the influence can propagate through and the weights on the directed edges represent the likelihood of the influence, i.e., the influence probability. Numerous models are proposed for the spread of influence, with the most popular ones being *independent cascade* (IC) and *linear threshold* (LT) models [1]. In IC model, the influence propagates on each edge independently from the other edges of the network. An influenced node has only a single chance to influence its neighbors. Hence, only recently influenced nodes can

propagate the influence. Thus, the influence stops to spread when the recently influenced nodes fail to influence their neighbors. On the other hand, in LT model, a node's chance to get influenced depends on whether the sum of weights of its active neighbors exceeds a threshold or not.

Most of the prior work in IM assume that the influence probabilities of the influence graph are known [2]–[7] and focus on designing computationally efficient algorithms to maximize the influence spread. However, in many practical settings, it is impossible to know beforehand the influence probabilities exactly. For instance, a firm that wants to introduce a new product or to advertise its existing products in a new social network may not know the influence probabilities on the edges of the network. In contrast to the prior works mentioned above, our focus is to design an optimal advertising strategy when the influence probabilities are unknown.

In the marketing example given above, influence depends on the product that is being advertised as well as the identities of the users. Hence, the characteristics (context) of the product affects the influence probabilities. The strand of literature that is closest to the problem we consider in this paper in terms of the dependence of the influence probabilities on the context is called *topic-aware IM* [8]–[11]. To the best of our knowledge, none of the prior works in topic-aware IM develop learning algorithms with provable performance guarantees for the case when the influence probabilities are unknown.

Motivated by the real-world challenges described above, in this paper, we define a new learning model for influence maximization, called the *Online Contextual Influence Maximization Problem* (OCIMP). In contrast to IM, which is a single-shot problem, OCIMP is a sequential decision making problem. In OCIMP, the learner/agent (e.g., the firm in the above example), faces a series of epochs in each of which a different influence campaign is run. At the beginning of each epoch, the learner observes the context of that epoch. For instance, the context can be the type of the influence campaign (e.g., one influence campaign might promote a sports equipment, while another influence campaign might promote a mobile data plan). After observing the context, the learner chooses a set of  $k$  seed nodes/users to influence. We call these nodes *exogenously influenced nodes*. Then, the influence spreads according to the IC model (explained in detail in Section III-A). The nodes that are influenced as a result of this process are called *endogenously influenced nodes*. The learner observes how the influence spreads, and

receives as its reward the number of endogenously influenced nodes.

The goal of the learner is to maximize its long term reward over epochs. In this paper, we propose a new learning algorithm called *Contextual Online INfluence maximization* (COIN) to maximize the learner's reward for any given number of epochs. COIN can use any approximation algorithm for the offline influence maximization problem as a subroutine to obtain the set of seed users in each epoch. When the influence probabilities are Hölder continuous functions of the context, we prove that COIN achieves sublinear regret with respect to an approximation oracle that knows the influence probabilities for all contexts. Moreover, the proven regret bound holds for any sequence of contexts.

The contributions are summarized as follows:

- We propose the *Online Contextual Influence Maximization Problem* (OCIMP). In OCIMP, the influence probabilities are unknown a priori and depend on the context, hence they need to be learned by repeated interaction.
- We propose an online learning algorithm named COIN to solve OCIMP. COIN only needs to keep a summary of the past observations in order to select the set of seed nodes.
- We prove a regret bound for COIN that is sublinear in the number of epochs when the influence probabilities are Hölder continuous functions of the context.
- We empirically evaluate the performance of COIN on several social networks, and show that it outperforms the other methods.

All proofs are given in the appendices.

## II. RELATED WORK

### A. Influence Maximization

The IM problem is proposed in [1], where it is proven to be NP-Hard, and an approximately optimal solution is given. However, the solution given in [1] does not scale well because it often requires thousands of Monte Carlo samples to estimate the expected influence spread of each seed set. This motivated the development of many heuristic methods with smaller computational complexity [3], [7], [12], [13].

In numerous other works, algorithms with approximation guarantees are developed for the IM problem: CELF [4], CELF++ [5], and NewGreedy [7]. In addition to these works, in [14], an approximation algorithm based on reverse influence sampling is proposed and its run-time optimality is proven. In [2], the authors improved the scalability of this algorithm by proposing two new algorithms TIM and TIM+. More recently, [6] developed IMM which is an improvement on TIM in terms of efficiency while preserving its theoretical guarantees. In none of the works mentioned above, context information is considered. IM based on context information is studied in several other works such as [8], [10], [11]. However, these works assume that the influence probabilities are known and topics/context are discrete. In OCIMP, the context is

represented by a collection of continuous features (which can be discretized if necessary) and the influence probabilities are unknown.

### B. Multi Armed Bandits (MAB)

Several recent works use MAB-based methods to solve the IM problem when the influence probabilities are unknown. In these works, as in ours, the set of arms chosen at each epoch corresponds to the seed set of nodes and this choice brings a reward, which is the number of endogenously influenced nodes.

For instance, [15] presents a combinatorial MAB problem where multiple arms are chosen at each epoch, where they probabilistically trigger the other arms. In our terminology, multiple arms chosen at each epoch correspond to the set of seed nodes and probabilistically triggered arms correspond to nodes other than the set of seed nodes. For this problem, a logarithmic regret bound is proven with respect to an approximation oracle. However, the problem in [15] does not involve any contexts (side information). Another general MAB model that uses greedy algorithms to solve the IM problem with unknown graph structure and influence probabilities is proposed in [16]. [17] considers a non-stationary IM problem, in which the influence probabilities are unknown and time varying. In OCIMP, the context can be used to model the time-varying nature of the influence probabilities (for instance, the context can be the time).

An online method for the IM problem that uses an *upper confidence bound* (UCB) based algorithm is proposed in [18]. In another related work [19], the IM problem is defined on an undirected graph, and a UCB-based algorithm is proposed to solve it. Most of the prior works described above assume that the influence on each edge in the network is observed by the learner. Recently, another observation model, called node-level feedback, is proposed in [20]. This model assumes that only the influenced nodes are observable while the spread of influence over the edges are not.

In conclusion, none of the works mentioned above consider the effect of the context (side information) on the influence probabilities. The differences between our work and the prior works are summarized in Table I.

|                 | Our Work | [15]–[17], [19] | [8]–[11] | [1]–[5], [7], [12]–[14] | [18] |
|-----------------|----------|-----------------|----------|-------------------------|------|
| Context         | Yes      | No              | Yes      | No                      | No   |
| Online Learning | Yes      | Yes             | No       | No                      | Yes  |
| Regret Bound    | Yes      | Yes             | No       | No                      | No   |

**TABLE I:** Comparison of the proposed method with prior work

### III. PROBLEM DESCRIPTION

#### A. Definition of the Influence

Consider a learner (e.g., a viral marketing engine) operating on a social network with  $n$  nodes/users and  $m$  edges. The set of nodes is denoted by  $V$ , and the set of edges is denoted by  $E$ . The (directed) *network graph* is denoted by  $G(V, E)$ . Set of children of node  $i$  is given by  $\mathcal{N}_i = \{j \in V : (i, j) \in E\}$ . Set of parents of node  $i$  is given by  $\mathcal{V}_i = \{j \in V : (j, i) \in E\}$ .

Ads arrive to the learner sequentially over time in discrete epochs, denoted by  $t = 1, 2, \dots$ . Without loss of generality, context of the ad at  $t^{th}$  epoch comes from the  $d$  dimensional context space  $\mathcal{X} := [0, 1]^d$ , and is denoted by  $x_t$ . The *influence graph* at epoch  $t$  is denoted by  $G(V, E, p^{x_t})$ , where  $p^{x_t} := \{p_{i,j}^{x_t}\}_{(i,j) \in E}$  is the set of context-dependent influence probabilities, and  $p_{i,j}^{x_t} \in [0, 1]$ , denotes the probability that node  $i$  influences node  $j$  when the context is  $x_t$ . These influence probabilities are not known a priori.

At the beginning of  $t^{th}$  epoch, the learner exogenously influences  $k$  out of the  $n$  nodes in the network. The set of these nodes is denoted by  $S_t$ .  $S_t$  is also called *action* at epoch  $t$ . An action is an element of the set of  $k$ -element subsets of  $V$ , which is denoted by  $\mathcal{M}$ . Nodes in  $S_t$  influence the other nodes according to the IC model. A node that has not been influenced yet is called an *inactive* node, whereas a node that has been influenced is called an *active* node. In the IC model, each epoch consists of a sequence of time slots indexed by  $s \in \{1, 2, \dots\}$ . Let  $\mathcal{A}_t^s$  denote the set of nodes that are already active at the beginning of time slot  $s$  of epoch  $t$ ,  $\mathcal{R}_t^s$  denote the set of nodes that are activated for the first time at time slot  $s$  of epoch  $t$ , and  $\mathcal{C}_t^s$  denote the set of nodes that might be activated at time slot  $s$  of epoch  $t$ . In the IC model, we have  $\mathcal{A}_t^1 = \emptyset$ ,  $\mathcal{R}_t^1 = S_t$ ,  $\mathcal{A}_t^{s+1} = \mathcal{A}_t^s \cup \mathcal{R}_t^s$  and  $\mathcal{C}_t^{s+1} = \{j \in \cup_{i \in \mathcal{R}_t^s} \mathcal{N}_i\} - \mathcal{A}_t^{s+1}\}$ . For  $j \in \mathcal{C}_t^{s+1}$ , let  $\tilde{\mathcal{V}}_t^{s+1}(j) = \{i \in \mathcal{V}_j \cap \mathcal{R}_t^s\}$  denote the set of nodes in  $\mathcal{R}_t^s$  that can influence  $j$ . In the IC model, we have

$$\Pr(j \in \mathcal{R}_t^{s+1} | j \in \mathcal{C}_t^{s+1}) = 1 - \prod_{i \in \tilde{\mathcal{V}}_t^{s+1}(j)} (1 - p_{i,j}^{x_t}). \quad (1)$$

Suppose that the influence propagation process started from a seed set  $S$  of nodes. We denote the expected number of endogenously activated nodes or *expected influence spread* given context  $x \in \mathcal{X}$  and action  $S$  as  $\sigma(x, S)$ , where the expectation is taken over the randomness of the influence spread process given  $S$ .

We assume that similar contexts have similar effects on the influence probabilities. This similarity is formalized in the following assumption.

**Assumption 1.** There exists  $L > 0$ ,  $\theta > 0$  such that for all  $(i, j) \in E$  and  $x \in \mathcal{X}$ ,  $|p_{i,j}^{x'} - p_{i,j}^x| \leq L \|x' - x\|^\theta$ , where  $\|\cdot\|$  denotes the Euclidian norm in  $\mathbb{R}^d$ .

#### B. Definition of the Reward and the Regret

The reward of action  $S$  at epoch  $t$  is equal to the *expected influence spread*  $\sigma(x_t, S)$ . For a given network graph

$G(V, E)$ , let  $\hat{p} = \{\hat{p}^x\}_{x \in \mathcal{X}}$  denote the set of estimated and  $p = \{p^x\}_{x \in \mathcal{X}}$  denote the set of true influence probabilities. We define  $\hat{\sigma}(x, S)$  as the expected influence spread of action  $S$  on  $G(V, E, \hat{p}^x)$ . First, we define the *omnipotent oracle*.

**Definition 1.** The omnipotent oracle knows the influence probabilities  $p_{i,j}^x \forall (i, j) \in E$  and  $\forall x \in \mathcal{X}$ . Given context  $x$ , it chooses  $S^*(x) \in \arg \max_{S \in \mathcal{M}} \sigma(x, S)$  as the seed set.

The *expected total reward* of the omnipotent oracle by epoch  $T$  is given by

$$\text{Rew}^*(T) := \sum_{t=1}^T \sigma(x_t, S^*(x_t)).$$

Next, we define the  $(\alpha, \beta)$ -approximation oracle, where  $0 < \alpha, \beta < 1$ .

**Definition 2.** The  $(\alpha, \beta)$ -approximation oracle knows the influence probabilities  $p_{i,j}^x \forall (i, j) \in E$  and  $\forall x \in \mathcal{X}$ . Given  $x$ , it generates an  $\alpha$ -approximate solution with probability at least  $\beta$ , i.e., it chooses the seed set  $S^{(\alpha, \beta)}(x)$  from  $\mathcal{M}$  such that  $\sigma(x, S^{(\alpha, \beta)}(x)) \geq \alpha \times \sigma(x, S^*(x))$  with probability at least  $\beta$ .

Next, we define the  $(\alpha, \beta)$ -approximation algorithm, which takes as input a set of estimated influence probabilities.

**Definition 3.** The  $(\alpha, \beta)$ -approximation algorithm takes as input the estimated influence probabilities  $\hat{p}_{i,j}^x \forall (i, j) \in E$  and  $\forall x \in \mathcal{X}$ . Given  $x$ , it chooses the seed set  $\hat{S}^{(\alpha, \beta)}(x)$  from  $\mathcal{M}$  such that  $\hat{\sigma}(x, \hat{S}^{(\alpha, \beta)}(x)) \geq \alpha \times \hat{\sigma}(x, \hat{S}^*(x))$  with probability at least  $\beta$ , where  $\hat{S}^*(x)$  is the seed set chosen by the omnipotent oracle given  $G(V, E, \hat{p}^x)$ .

For a sequence of context arrivals  $\{x_t\}_{t=1}^T$  the  $(\alpha, \beta)$ -regret of the learner, which chooses the sequence of actions  $\{S_t\}_{t=1}^T$ , with respect to the  $(\alpha, \beta)$ -approximation oracle by epoch  $T$  is defined as

$$R^{(\alpha, \beta)}(T) := \alpha \beta \times \text{Rew}^*(T) - \sum_{t=1}^T \sigma(x_t, S_t).$$

Our goal in this work is to design an online learning algorithm whose expected  $(\alpha, \beta)$ -regret, i.e.,  $\mathbb{E}[R^{(\alpha, \beta)}(T)]$  grows slowly in time and in the cardinality of the action space, without making any statistical assumptions on the context arrival process. Our algorithm can work together with any approximation algorithm designed for the offline IM problem.

### IV. APPROXIMATION GUARANTEE

The maximum difference between the true ( $p$ ) and the estimated ( $\hat{p}$ ) influence probabilities given context  $x$  is defined as  $\Delta_x(p, \hat{p}) := \max_{(i,j) \in E} |p_{i,j}^x - \hat{p}_{i,j}^x|$ , and the maximum difference between the true and the estimated influence probabilities over all contexts is defined as  $\Delta(p, \hat{p}) = \sup_{x \in \mathcal{X}} \Delta_x(p, \hat{p})$ . The following theorem, originally given as Lemma 6 in [15] provides a relation between the expected influence spread of action  $S$  on  $G(V, E, \hat{p}^x)$  and  $G(V, E, p^x)$ .

**Theorem 1.** (Lemma 6 in [15]) If  $\Delta_x(\mathbf{p}, \hat{\mathbf{p}}) = \Delta_x$ , then  $|\hat{\sigma}(x, S) - \sigma(x, S)| \leq mn\Delta_x$  for all  $S \in \mathcal{M}$ .

The next theorem gives an approximation guarantee for the  $(\alpha, \beta)$ -approximation algorithm when it runs using  $\hat{\mathbf{p}}$ .

**Theorem 2.** If  $\Delta(\mathbf{p}, \hat{\mathbf{p}}) = \Delta$ , then

$$\mathbb{E} [\sigma(x, \hat{S}^{(\alpha, \beta)}(x))] \geq \alpha\beta \times \sigma(x, S^*(x)) - \beta(1 + \alpha)mn\Delta$$

for all  $x \in \mathcal{X}$ .

## V. CONTEXTUAL ONLINE INFLUENCE MAXIMIZATION

In this section, we describe COIN. COIN is an online learning algorithm, which can utilize any  $(\alpha, \beta)$ -approximation algorithm for the IM problem as a subroutine. In order to exploit the context information efficiently, COIN aggregates information gained from epochs with similar contexts when estimating the influence probabilities. This aggregation is performed by creating a partition  $\mathcal{Q}$  of the context space  $\mathcal{X}$  based on the similarity information given in Assumption 1. Each set in the partition has a radius that is less than a time-horizon dependent threshold. This implies that the influence probability estimates formed by observations in a certain set of the partition do not deviate too much from the actual influence probabilities that correspond to contexts that are in the same set.

For each  $Q \in \mathcal{Q}$ , COIN keeps sample mean estimates of the influence probabilities. For any  $x \in Q$  and  $(i, j) \in E$ , the estimate of  $p_{i,j}^x$  at epoch  $t$  is denoted by  $\hat{p}_{i,j}^Q(t)$ .<sup>1</sup> This estimate is updated whenever an observation about the influence on edge  $(i, j)$  is made by COIN for some context  $x \in Q$ .

At the beginning of epoch  $t$ , COIN observes  $x_t$  and finds the set  $Q \in \mathcal{Q}$  that contains  $x_t$ , which is denoted by  $Q_t$ . Then, COIN decides on which seed set of nodes  $S_t$  to choose solely based on  $\{\hat{p}_{i,j}^{Q_t}(t)\}_{(i,j) \in E}$ . Since these values are noisy estimates of the true influence probabilities, two factors play a role in the accuracy of these estimates: estimation error and approximation error. Estimation error is due to the noise introduced by the randomness of the influence samples. It decreases with the number of samples that are used to estimate the influence probabilities. On the other hand, approximation error is due to the noise introduced by quantization of the context space. It increases with the radius of  $Q_t$ . There is an inherent tradeoff between these errors. In order to decrease the approximation error, partition  $\mathcal{Q}$  must be refined. This will create more sets in  $\mathcal{Q}$ , and hence, will result in smaller number of samples in each set, which will cause the estimation error to increase. In order to optimally balance these errors, size of the sets in  $\mathcal{Q}$  and the number of observations that fall into each of these sets must be adjusted carefully. COIN achieves this by using a time-horizon dependent partitioning parameter  $q_T$ , which is used to partition  $\mathcal{X}$  into  $q_T^d$  identical hypercubes with border lengths  $1/q_T$ . The value of  $q_T$  given in Theorem 3 achieves this balance. In order to minimize the loss due to estimation errors over different epochs, COIN

alternates between two phases of operation: *exploration* and *exploitation*. Let  $\hat{\mathbf{p}}_t$  denote the set of estimated influence probabilities at epoch  $t$ , whose elements are sample-mean estimates of influence probabilities. When COIN exploits in epoch  $t$ , it calls the  $(\alpha, \beta)$ -approximation algorithm with  $\hat{\mathbf{p}}_t$  to select  $S_t$ . As we stated in Theorem 2, choice of  $S_t$  does not guarantee the expected influence spread to be  $(\alpha, \beta)$ -approximate. When  $\hat{\mathbf{p}}_t$  is far away from  $\mathbf{p}$ , the estimation error is large. Thus, COIN will be highly suboptimal if it exploits when the estimation error is large. In order to achieve sublinear regret, the estimate  $\hat{\mathbf{p}}_t$  should improve over epochs for all edges  $(i, j) \in E$  and for all  $Q \in \mathcal{Q}$ . This is achieved by the exploration phase, which is discussed below.

Recall from (1) that in the IC model, at each time slot  $s + 1$  of epoch  $t$ , nodes in  $\mathcal{R}_t^s$  attempt to influence their children by activating the edges connecting them to their children. We call such an attempt in any time slot of epoch  $t$  as an *activation attempt*. Let  $F_t$  be the set of edges with activation attempts at epoch  $t$ . For  $(i, j) \in F_t$ , we call  $a_{i,j} = 1$  implies that node  $j$  is influenced by node  $i$  while  $a_{i,j} = 0$  implies that node  $j$  is not influenced by node  $i$ . COIN keeps two counters  $f_{i,j}^Q(t)$  and  $s_{i,j}^Q(t)$  for each  $(i, j) \in E$  and each  $Q \in \mathcal{Q}$ . The former denotes the number of failed activation attempts on edge  $(i, j)$  at epochs prior to epoch  $t$  when the context was in  $Q$ , while the latter denotes the number of successful activation attempts on edge  $(i, j)$  at epochs prior to epoch  $t$  when the context was in  $Q$ . COIN also keeps a deterministic, non-decreasing function  $D(t)$ , which is called the *control function*. Decision to explore or exploit at epoch  $t$  is based on the values of  $\{f_{i,j}^{Q_t}(t)\}_{(i,j) \in E}$ ,  $\{s_{i,j}^{Q_t}(t)\}_{(i,j) \in E}$  and  $D(t)$ . In order to form accurate influence probability estimates for the approximation algorithm, COIN explores edges whose activation attempts are not observed sufficiently many times by choosing  $S_t$  from

$$U_{Q_t}(t) := \{i \in V | \exists j \in \mathcal{N}_i : f_{i,j}^{Q_t} + s_{i,j}^{Q_t} < D(t)\}$$

which is the set of under-explored nodes of  $Q_t \in \mathcal{Q}$  at epoch  $t$ .

COIN has sufficiently accurate estimates of the influence probabilities when  $U_{Q_t}(t) = \emptyset$ . Therefore, in this case, it does exploitation by running an  $(\alpha, \beta)$ -approximation algorithm on  $G(V, E, \hat{\mathbf{p}}_t)$ . In the exploitation phase, COIN does *implicit exploration* by probabilistically activating nodes that are not in the seed set. However, because its current information on influence probabilities is not accurate, it may repeatedly explore the same edges and may not sample (explore) some of the edges that are actually worth exploring, and hence, may end up not learning some of the influence probabilities. To resolve this issue, it does exploration when  $U_{Q_t}(t) \neq \emptyset$ . We divide the seed set selection process for the exploration phase into two cases: (i) when  $|U_{Q_t}(t)| < k$  and (ii) when  $|U_{Q_t}(t)| \geq k$ . In the first case, all of the nodes in  $U_{Q_t}(t)$  are chosen and the remaining  $k - |U_{Q_t}(t)|$  nodes are chosen by calling an  $(\alpha, \beta)$ -approximation algorithm. In the latter case,  $k$  nodes are chosen randomly from  $U_{Q_t}(t)$ .

<sup>1</sup>We will drop the epoch index when it is clear from the context.

The value of  $D(t)$  (which controls  $U_{Q_t}(t)$ ) for which the regret is minimized is given in Theorem 3. This theorem shows that the regret of COIN is sublinear in time for any sequence of context arrivals  $x_1, \dots, x_T$ .

**Theorem 3.** When COIN runs with  $q_T = \lceil T^{1/(3\theta+d)} \rceil$  and  $D(t) = t^{2\theta/(3\theta+d)}$ , we have for any sequence of contexts  $x_1, \dots, x_T$

$$\mathbb{E}[R^{(\alpha,\beta)}(T)] = O(T^{\frac{2\theta+d}{3\theta+d}}).$$

### Algorithm 1 Contextual Online Influence Maximization (COIN)

**Require:**  $T, q_T, G(V, E), D(t) \quad t \in \{1, \dots, T\}$

```

Initialize sets: Create the partition  $\mathcal{Q}$  of  $\mathcal{X}$  such that  $\mathcal{X}$  is divided
into  $(q_T)^d$  identical hypercubes with edge lengths  $1/q_T$ 
Initialize counters:  $f_{i,j}^{Q_t} = 0$  and  $s_{i,j}^{Q_t} = 0, \forall (i, j) \in E, \forall Q \in \mathcal{Q}$ 
Initialize estimates:  $\hat{p}_{i,j}^{Q_t} = 0, \forall (i, j) \in E, \forall Q \in \mathcal{Q}$ 

1: while  $t \leq T$  do
2:   Find the set  $Q_t \in \mathcal{Q}$  that  $x_t$  belongs to
3:   Compute the set of under-explored nodes  $U_{Q_t}(t)$ 
4:   if  $|U_{Q_t}(t)| \geq k$  then {Explore}
5:     Select  $S_t$  randomly from  $U_{Q_t}(t)$ , such that,  $|S_t| = k$ 
6:   else if  $U_{Q_t}(t) \neq \emptyset$  and  $|U_{Q_t}(t)| < k$  then
7:     Select the  $|U_{Q_t}(t)|$  elements of  $S_t$  as  $U_{Q_t}(t)$  and the
       remaining  $k - |U_{Q_t}(t)|$  elements of  $S_t$  by using an  $(\alpha, \beta)$ -
       approximation algorithm.
8:   else {Exploit}
9:     Select  $S_t$  by using an  $(\alpha, \beta)$ -approximation algorithm for
       the IM problem on  $G(V, E, \hat{p}^{Q_t})$ 
10:  end if
11:  Observe the set of edges with activation attempts, i.e.,  $F_t$ 
12:  Update the successes and failures  $\forall (i, j) \in F_t$ :
13:  for  $(i, j) \in F_t$  do
14:    if  $a_{i,j} = 1$  then
15:       $s_{i,j}^{Q_t} = s_{i,j}^{Q_t} + 1$ 
16:    else if  $a_{i,j} = 0$  then
17:       $f_{i,j}^{Q_t} = f_{i,j}^{Q_t} + 1$ 
18:    end if
19:     $\hat{p}_{i,j}^{Q_t} = \frac{s_{i,j}^{Q_t}}{s_{i,j}^{Q_t} + f_{i,j}^{Q_t}}$ 
20:  end for
21:   $t = t + 1$ 
22: end while

```

## VI. EXPERIMENTS

### A. Improved Exploration Phase

To improve the exploration phases of COIN, we consider an additional exploration strategy in which  $S_t$  is not chosen randomly from  $U_{Q_t}(t)$ . In this variant, TIM+ [2] is used to select the seed nodes from  $U_{Q_t}(t)$ . This variant of COIN is called COIN+. It enjoys the same theoretical regret guarantees as COIN. Since, it incurs less regret on exploration phases, we use COIN+ instead of COIN in our experiments.

### B. Setup

For the social network, we use NetHEPT and NetHEPT-. NetHEPT is extensively used in IM literature [2], [18], [20] and NetHEPT- is a random subgraph of NetHEPT where all of the nodes have a positive in-degree. In NetHEPT, roughly a third of the nodes have an in-degree value of 0, which means that they cannot be activated endogenously whereas in NetHEPT-, all of the nodes can be activated by a choice of seed set.

We set  $q_T = 2$  and  $d = 2$  in COIN+. For initialization, all influence probability estimates are set to 0 and  $k$  is set to 50, which is a typical choice in the online IM literature [18], [20]. In order to reduce the number of explorations, we set  $D(t) = \frac{t^{2/(3+d)}}{10}$ . We set  $T = 5000$  for both graphs.

| Dataset  | V   | E     | Average In-degree |
|----------|-----|-------|-------------------|
| NetHEPT  | 15K | 59K   | 3.86              |
| NetHEPT- | 4K  | 10.5K | 2.63              |

TABLE II: Properties of the social networks used

L2 error at epoch  $t$  is given by  $L_2^t := \sqrt{\sum_{(i,j) \in E} (p_{i,j}^{x_t} - \hat{p}_{i,j}^{Q_t})^2}$ . We report both the time averaged regret (i.e.,  $R^{(\alpha,\beta)}/t$ ) and L2 error of the influence probability estimates. TIM+ is chosen as the approximation oracle, which uses the true influence probabilities for seed set selection.

### C. Defining the Influence Probabilities

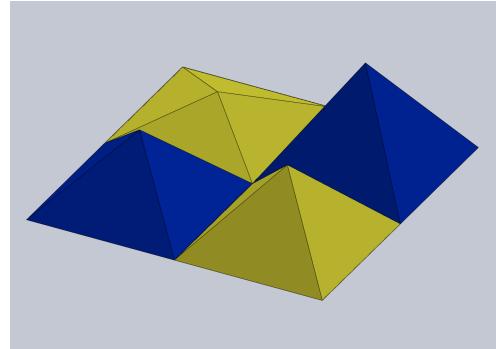


Fig. 1: An example of a pyramidal surface

The context in each epoch is sampled uniformly at random from  $[0, 1]^2$ . Context-dependent influence probabilities are generated according to a Hölder-continuous pyramidal surface defined over  $[0, 1]^2$ . For this task, we divide  $[0, 1]^2$  into four identical squares as shown in Fig. 1. Each square forms the base of a pyramid, whose extrema is located at the center of the square. Consider edge  $(u, v) \in E$ . The influence probability of this edge on the boundary of the squares is set to  $1/d_v$ . The height of the centers of pyramids are chosen randomly from  $[1/(4d_v), 7/(4d_v)]$  when  $d_v > 1$  and from  $[1/(4d_v), d_v]$  when  $d_v = 1$  to keep the influence probabilities between 0 and 1. The influence probability of  $(u, v) \in E$  for a context  $x \in [0, 1]^2$  is set to be equal to the height of

the pyramid at that context. With this setup, we are able to introduce variations within the same set in the context partition as well as among different sets in the context partition.

#### D. Algorithms

We compare COIN+ with the following algorithms, which do not exploit the presence of the context.

1) *ThompsonG*: This is a variant of Thompson sampling, in which the parameters of the beta distribution from which the influence estimates are drawn are calculated using global priors as explained in [18]. These global priors are updated at each epoch  $t$  according to the feedback,  $F_t$ .

2) *Pure Exploitation*: This algorithm is COIN with no exploration. Therefore, it only does exploitation, which results in implicit exploration.

3) *High-Degree*: High-Degree exploits only the graph structure and does not use the influence probabilities. It chooses nodes that have the highest out-degrees as its seed set.

4) *CB+MLE*: CB+MLE is a UCB-based algorithm for online IM problem explained in [18].

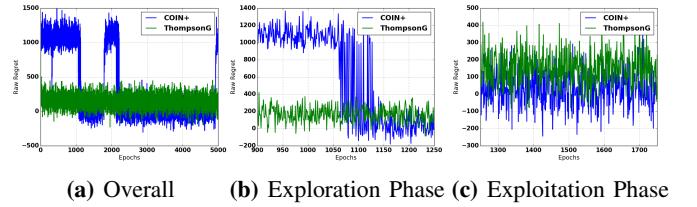
#### E. Results

Our experiments demonstrate that COIN+ outperforms the other mentioned algorithms in the long run (see Fig. 2). In NetHEPT, it takes longer for COIN+ to become better at average regret than ThompsonG (while this is not seen in Fig. 2 since  $T$  is not taken to be sufficiently large, Fig. 3 shows that it is performing better than ThompsonG when it exploits). COIN+ explores in the first epochs where it suffers from high exploration regret. Because of this, the average regret of COIN+ is higher than that of other algorithms.

To compare the two best performing algorithms, ThompsonG and COIN+, we investigate their raw regret, which is defined as the difference between spreads of  $(\alpha, \beta)$ -approximation oracle and the  $(\alpha, \beta)$ -approximation algorithm in each epoch. Fig. 3(a) depicts this metric throughout the experiment, in which the exploration phases of COIN+ are seen as intervals of high raw regret. Notice however that on Fig. 3(b), COIN+ outperforms ThompsonG in terms of raw regret as soon as its first exploration phase is over, performing on par with the oracle. The oscillatory behavior in this transition is due to the randomness in the context arrival process. Fig. 3(c) depicts the exploitation phase of COIN+ where the superior performance of the algorithm prevails. However, the effect of low exploitation regret is not reflected in the average regret right away because of the high exploration regret. On the other hand, the L2 error of COIN+ (in Fig. 2) keeps decreasing even after the first wave of exploration phases is over.

#### APPENDIX A PROOF OF THEOREM 2

Recall that  $\hat{S}^*(x)$  denotes the seed set chosen by the omnipotent oracle given context  $x$  and  $\hat{p}$ . By definition of the  $(\alpha, \beta)$ -approximation algorithm, we have  $\hat{\sigma}(x, \hat{S}^{(\alpha, \beta)}(x)) \geq$



**Fig. 3:** Raw regret of COIN+ and ThompsonG for NetHEPT

$\alpha \times \hat{\sigma}(x, \hat{S}^*(x))$  with probability at least  $\beta$ . Theorem 1 implies that for any seed set  $S$ ,  $|\hat{\sigma}(x, S) - \sigma(x, S)| \leq mn\Delta$ . Using the results above, we obtain

$$\begin{aligned} \sigma(x, \hat{S}^{(\alpha, \beta)}(x)) &\geq \hat{\sigma}(x, \hat{S}^{(\alpha, \beta)}(x)) - mn\Delta \\ &\geq \alpha \hat{\sigma}(x, \hat{S}^*(x)) - mn\Delta \\ &\geq \alpha \hat{\sigma}(x, S^*(x)) - mn\Delta \\ &\geq \alpha(\sigma(x, S^*(x)) - mn\Delta) - mn\Delta \\ &= \alpha\sigma(x, S^*(x)) - (1 + \alpha)mn\Delta \end{aligned}$$

with probability at least  $\beta$ . Since  $\sigma(x, \hat{S}^{(\alpha, \beta)}(x))$  is non-negative, we obtain the following bound by taking the expectation:

$$\mathbb{E}[\sigma(x, \hat{S}^{(\alpha, \beta)}(x))] \geq \alpha\beta \times \sigma(x, S^*(x)) - \beta(1 + \alpha)mn\Delta.$$

#### APPENDIX B PROOF OF THEOREM 3

As preliminaries, we introduce the following notations: for  $Q \in \mathcal{Q}$  let  $\bar{p}_{i,j}^Q := \sup_{x \in Q} p_{i,j}^Q(x)$  and  $\underline{p}_{i,j}^Q := \inf_{x \in Q} p_{i,j}^Q(x)$ .

Consider COIN with partitioning parameter  $q_T = \lceil T^z \rceil$  and control function  $D(t) = t^\gamma$ , where  $0 < \gamma, z < 1$ . For any context arrival process  $\{x_t\}_{t=1,\dots}$ , let  $\mathcal{T}_T^s$  be the set of epochs by epoch  $T$  in which COIN exploits and  $\mathcal{T}_T^o$  be the set of epochs by epoch  $T$  in which COIN explores. Since the activation attempts are random variables,  $\mathcal{T}_T^s$  is a random set. By the definition of the exploration and exploitation phases of COIN, we have for any  $t \in \mathcal{T}_T^s$

$$f_{i,j}^{Q_t}(t) + s_{i,j}^{Q_t}(t) \geq D(t) \quad \forall (i, j) \in E. \quad (2)$$

The simple  $(\alpha, \beta)$ -regret of COIN at epoch  $t$  is defined as

$$r^{(\alpha, \beta)}(t) := \alpha\beta \times \sigma(x_t, S^*(x_t)) - \sigma(x_t, S_t).$$

Let  $R^s(T) := \sum_{t \in \mathcal{T}_T^s} r^{(\alpha, \beta)}(t)$  be the regret incurred over epochs in which COIN exploits, and  $R^o(T) := \sum_{t \in \mathcal{T}_T^o} r^{(\alpha, \beta)}(t)$  be the regret incurred over epochs in which COIN explores. We have the following decomposition for regret:

$$\mathbb{E}[R^{(\alpha, \beta)}(T)] = \mathbb{E}[\mathbb{E}[R^s(T) + R^o(T) | \mathcal{T}_T^s]]. \quad (3)$$

The following lemma bounds the number of epochs in which COIN explores.

**Lemma 1.** *When COIN runs with control function  $D(t) = t^\gamma$  and partitioning parameter  $q_T = \lceil T^z \rceil$ ,  $0 < \gamma, z < 1$ , we have  $|\mathcal{T}_T^o| \leq (\frac{m}{k} + 1) \lceil T^z \rceil^d \lceil T^\gamma \rceil$  with probability 1.*

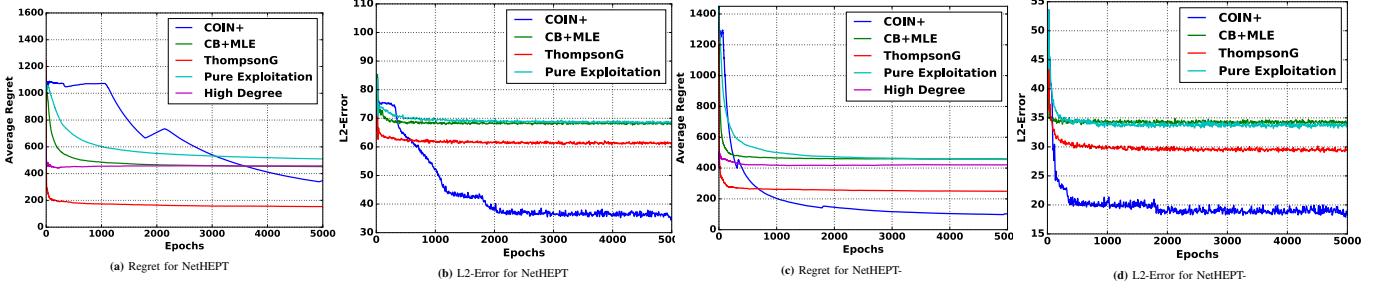


Fig. 2: Average regret and L2 error on NETHEPT and NETHEPT-

*Proof.* Let  $Y_{Q_t}(t) = \{(i, j) \in E \mid f_{i,j}^{Q_t} + s_{i,j}^{Q_t} < D(t)\}$  denote the set of under-explored edges at epoch  $t$ . We separate the set of epochs,  $\mathcal{T}_T^o$ , into two sets: Let  $\mathcal{T}_T^o = \mathcal{T}_T^H \cup \mathcal{T}_T^L$  where  $\mathcal{T}_T^H$  denotes the set of exploration phases where  $|Y_{Q_t}(t)| \geq k$  and  $\mathcal{T}_T^L$  denotes the set of exploration phases where  $|Y_{Q_t}(t)| < k$ . We will bound  $\mathcal{T}_T^H$  and  $\mathcal{T}_T^L$  separately.

Firstly, we bound the term  $|\mathcal{T}_T^L|$ . Note that for each  $Q \in \mathcal{Q}$ , there will be at most  $\lceil D(T) \rceil$  many epochs where  $Q_t = Q$  and  $|Y_Q(t)| < k$ . Therefore

$$|\mathcal{T}_T^L| < q_T^d \lceil D(T) \rceil \quad (4)$$

with probability 1.

Secondly, we bound  $\mathcal{T}_T^H$ . Let  $u(t) := |F_t \cap Y_{Q_t}(t)|$ . Note that for a given  $T$ , there are  $m q_T^d$  many context set-edge pairs. Hence the total number of explorations can be at most  $m q_T^d \lceil D(T) \rceil$ . Thus, we get

$$\sum_{t \in \mathcal{T}_T^H} u(t) \leq m q_T^d \lceil D(T) \rceil \quad (5)$$

with probability 1. From the definition of  $\mathcal{T}_T^H$ , we know that  $u(t) \geq k$  for all  $t \in \mathcal{T}_T^H$ . Using this together with (5), we obtain  $k |\mathcal{T}_T^H| \leq m q_T^d \lceil D(T) \rceil$  and hence,

$$|\mathcal{T}_T^H| \leq \frac{m q_T^d \lceil D(T) \rceil}{k} \quad (6)$$

with probability 1. The result follows from (4) and (6).  $\square$

At each epoch when COIN explores, in the worst case, it will fail to influence  $(n - k)$  remaining nodes. Given that the omnipotent oracle potentially influences all of the  $(n - k)$  nodes at these epochs, we get  $r^{(\alpha, \beta)}(t) \leq \alpha \beta (n - k)$  for each  $t \in \mathcal{T}_T^o$ . Using this together with the regret decomposition in (3) and Lemma 1 we get

$$\begin{aligned} \mathbb{E}[R^{(\alpha, \beta)}(T)] &\leq \alpha \beta (n - k) \left( \frac{m}{k} + 1 \right) \lceil T^z \rceil^d \lceil T^\gamma \rceil \\ &\quad + \mathbb{E}[\mathbb{E}[R^s(T) | \mathcal{T}_T^s]]. \end{aligned} \quad (7)$$

The following lemma bounds  $r^{(\alpha, \beta)}(t)$  when COIN exploits at epoch  $t$ .

**Lemma 2.** When COIN runs with control function  $D(t) = t^\gamma$  and partitioning parameter  $q_T = \lceil T^z \rceil$ , where  $0 < \gamma, z < 1$ ,

we have for any  $t \in \mathcal{T}_T^s$

$$\begin{aligned} \mathbb{E}[r^{(\alpha, \beta)}(t)] &\leq \beta(1 + \alpha)mnLd^{\theta/2}q_T^{-\theta} \\ &\quad + \frac{\beta(1 + \alpha)\pi m^2 nt^{-\gamma/2}}{\sqrt{2}} \end{aligned}$$

*Proof.* Consider epoch  $t$ . Let  $\Delta_t := \Delta_{x_t}(\mathbf{p}, \hat{\mathbf{p}}_t)$ . By Theorem 2, we have

$$\begin{aligned} \mathbb{E}[r^{(\alpha, \beta)}(t)] &= \alpha \beta \times \sigma(x_t, S^*(x_t)) - \mathbb{E}[\sigma(x_t, S_t)] \\ &\leq \beta(1 + \alpha)mn\mathbb{E}[\Delta_t]. \end{aligned}$$

Since  $\Delta_t \in [0, 1]$ ,

$$\mathbb{E}[r^{(\alpha, \beta)}(t)] \leq \beta(1 + \alpha)mn \int_0^1 \Pr(\Delta_t \geq y) dy. \quad (8)$$

Note that

$$\begin{aligned} \{\Delta_t \geq y\} &= \left\{ \max_{(i,j) \in E} (|\hat{p}_{i,j}^{Q_t}(t) - p_{i,j}^{x_t}| \geq y) \right\} \\ &= \bigcup_{(i,j) \in E} \{|\hat{p}_{i,j}^{Q_t}(t) - p_{i,j}^{x_t}| \geq y\} \\ &= \bigcup_{(i,j) \in E} \{\hat{p}_{i,j}^{Q_t}(t) - p_{i,j}^{x_t} \leq -y\} \cup \bigcup_{(i,j) \in E} \{\hat{p}_{i,j}^{Q_t}(t) - p_{i,j}^{x_t} \geq y\} \\ &\subset \bigcup_{(i,j) \in E} \{\hat{p}_{i,j}^{Q_t}(t) - \bar{p}_{i,j}^{Q_t} \leq -y\} \cup \bigcup_{(i,j) \in E} \{\hat{p}_{i,j}^{Q_t}(t) - \underline{p}_{i,j}^{Q_t} \geq y\}. \end{aligned}$$

Hence, by the union bound we get

$$\begin{aligned} \Pr(\Delta_t \geq y) &\leq \sum_{(i,j) \in E} \Pr(\hat{p}_{i,j}^{Q_t}(t) - \bar{p}_{i,j}^{Q_t} \leq -y) \\ &\quad + \sum_{(i,j) \in E} \Pr(\hat{p}_{i,j}^{Q_t}(t) - \underline{p}_{i,j}^{Q_t} \geq y) \end{aligned}$$

By Assumption 1, we have  $\bar{p}_{i,j}^{Q_t} - \underline{p}_{i,j}^{Q_t} \leq Ld^{\theta/2}q_T^{-\theta}$  for all  $(i, j) \in E$  and  $Q \in \mathcal{Q}$ . Hence, we have  $\bar{p}_{i,j}^{Q_t} \leq \mathbb{E}[\hat{p}_{i,j}^{Q_t}(t)] + Ld^{\theta/2}q_T^{-\theta}$  and  $\underline{p}_{i,j}^{Q_t} \geq \mathbb{E}[\hat{p}_{i,j}^{Q_t}(t)] - Ld^{\theta/2}q_T^{-\theta}$  for all  $(i, j) \in E$  and  $Q \in \mathcal{Q}$ . Using the fact above, we obtain for  $y \geq Ld^{\theta/2}q_T^{-\theta}$ ,  $\Pr(\hat{p}_{i,j}^{Q_t}(t) - \bar{p}_{i,j}^{Q_t} \leq -y) \leq \Pr(\hat{p}_{i,j}^{Q_t}(t) - \mathbb{E}[\hat{p}_{i,j}^{Q_t}(t)] \leq Ld^{\theta/2}q_T^{-\theta} - y)$  and  $\Pr(\hat{p}_{i,j}^{Q_t}(t) - \underline{p}_{i,j}^{Q_t} \geq y) \leq \Pr(\hat{p}_{i,j}^{Q_t}(t) - \mathbb{E}[\hat{p}_{i,j}^{Q_t}(t)] \geq y - Ld^{\theta/2}q_T^{-\theta})$ . Since (2) holds for  $t \in \mathcal{T}_T^s$ , by using the above inequalities together with

Hoeffding's inequality, we obtain for  $y \geq Ld^{\theta/2}q_T^{-\theta}$

$$\sum_{(i,j) \in E} \Pr(\hat{p}_{i,j}^{Q_t}(t) - \underline{p}_{i,j}^{Q_t} \geq y) \leq me^{-2(y-Ld^{\theta/2}q_T^{-\theta})^2 t^\gamma} \quad (9)$$

$$\sum_{(i,j) \in E} \Pr(\hat{p}_{i,j}^{Q_t}(t) - \bar{p}_{i,j}^{Q_t} \leq -y) \leq me^{-2(y-Ld^{\theta/2}q_T^{-\theta})^2 t^\gamma}. \quad (10)$$

In order to bound (8), we separate the integral into two parts. For  $0 \leq y < Ld^{\theta/2}q_T^{-\theta}$ , we have  $\Pr(\Delta_t \geq y) \leq 1$ . For  $Ld^{\theta/2}q_T^{-\theta} \leq y \leq 1$  by (9) and (10), we have  $\Pr(\Delta_t \geq y) \leq 2me^{-2(y-Ld^{\theta/2}q_T^{-\theta})^2 t^\gamma}$ . Hence,

$$\begin{aligned} & \int_0^1 \Pr(\Delta_t \geq y) dy \\ &= \int_0^{Ld^{\theta/2}q_T^{-\theta}} 1 dy + \int_{Ld^{\theta/2}q_T^{-\theta}}^1 2me^{-2(y-Ld^{\theta/2}q_T^{-\theta})^2 t^\gamma} dy \\ &\leq Ld^{\theta/2}q_T^{-\theta} + 2m \int_{Ld^{\theta/2}q_T^{-\theta}}^1 \frac{dy}{1+2(y-Ld^{\theta/2}q_T^{-\theta})^2 t^\gamma} \\ &= Ld^{\theta/2}q_T^{-\theta} + 2m \frac{t^{-\gamma/2}}{\sqrt{2}} (\arctan(\sqrt{2}t^{\gamma/2}(1-Ld^{\theta/2}q_T^{-\theta}))) \\ &\leq Ld^{\theta/2}q_T^{-\theta} + \frac{m\pi t^{-\gamma/2}}{\sqrt{2}} \end{aligned} \quad (11)$$

since  $e^{-y} \leq \frac{1}{1+y}$  for all  $y \geq 0$  and that  $\arctan(z) \leq \frac{\pi}{2}$  for all  $z \in \mathbb{R}$ . The result is obtained by substituting (11) in (8).  $\square$

The next lemma uses Lemma 2 to bound  $\mathbb{E}[R^s(T)]$ .

**Lemma 3.** *When COIN runs with control function  $D(t) = t^\gamma$  and partitioning parameter  $q_T = \lceil T^z \rceil$ , where  $0 < \gamma, z < 1$ , we have*

$$\begin{aligned} \mathbb{E}[R^s(T)] &\leq \beta(1+\alpha)mnLd^{\theta/2}T^{1-\theta z} \\ &\quad + \frac{\beta(1+\alpha)\pi m^2 n}{\sqrt{2}} \times \frac{T^{1-\gamma/2} - \gamma/2}{(1-\gamma/2)}. \end{aligned}$$

*Proof.* We utilize the following inequalities in the proof:  $|\mathcal{T}_T^s| \leq T$  with probability one and  $\sum_{t=1}^T t^{-x} \leq (T^{1-x} - x)/(1-x) \forall x \in (0, 1)$ . For any realization of  $\mathcal{T}_T^s$  denoted by  $\mathcal{T} \subset \{1, \dots, T\}$  we have

$$\begin{aligned} \mathbb{E}[R^s(T)|\mathcal{T}_T^s = \mathcal{T}] &= \sum_{t \in \mathcal{T}} \mathbb{E}[r^{(\alpha, \beta)}(t)] \\ &\leq \beta(1+\alpha) \sum_{t \in \mathcal{T}} \left( \frac{mnLd^{\theta/2}}{\lceil T^z \rceil^\theta} + \frac{\pi m^2 nt^{-\gamma/2}}{\sqrt{2}} \right) \\ &\leq \beta(1+\alpha)mnLd^{\theta/2}T^{1-\theta z} \\ &\quad + \frac{\beta(1+\alpha)\pi m^2 n}{\sqrt{2}} \times \frac{T^{1-\gamma/2} - \gamma/2}{(1-\gamma/2)}. \end{aligned}$$

$\square$

We obtain a regret bound for our algorithm by substituting the result of Lemma 3 into (7). The optimal solution is given by  $z = 1/(3\theta + d)$  and  $\gamma = 2\theta/(3\theta + d)$ .

## REFERENCES

- [1] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *Proc. 9th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 137–146, 2003.
- [2] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proc. 2014 ACM SIGMOD Int. Conf. Management of Data*, pp. 75–86, 2014.
- [3] K. Jung, W. Heo, and W. Chen, "IRIE: Scalable and robust influence maximization in social networks," in *Proc. 12th IEEE Int. Conf. Data Mining (ICDM)*, pp. 918–923, 2012.
- [4] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *Proc. 13th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 420–429, 2007.
- [5] A. Goyal, W. Lu, and L. V. Lakshmanan, "CELF++: Optimizing the greedy algorithm for influence maximization in social networks," in *Proc. 20th Int. Conf. Companion on World Wide Web (WWW)*, pp. 47–48, 2011.
- [6] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *Proc. 2015 ACM SIGMOD Int. Conf. Management of Data*, pp. 1539–1554, 2015.
- [7] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *Proc. 15th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 199–208, 2009.
- [8] S. Chen, J. Fan, G. Li, J. Feng, K.-I. Tan, and J. Tang, "Online topic-aware influence maximization," *Proc. VLDB Endow.*, vol. 8, no. 6, pp. 666–677, 2015.
- [9] N. Barbieri, F. Bonchi, and G. Manco, "Topic-aware social influence propagation models," in *Proc. 12th IEEE Int. Conf. Data Mining (ICDM)*, pp. 81–90, 2012.
- [10] C. Aslay, N. Barbieri, F. Bonchi, and R. A. Baeza-Yates, "Online topic-aware influence maximization queries," in *Proc. 17th Int. Conf. Extending Database Technology (EDBT)*, pp. 295–306, 2014.
- [11] W. Chen, T. Lin, and C. Yang, "Efficient topic-aware influence maximization using preprocessing," *arXiv preprint arXiv:1403.0057*, 2014.
- [12] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *Proc. 16th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 1029–1038, 2010.
- [13] J. Kim, S. K. Kim, and H. Yu, "Scalable and parallelizable processing of influence maximization for large-scale social networks?," in *Proc. 29th IEEE Int. Conf. Data Engineering (ICDE)*, pp. 266–277, 2013.
- [14] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *Proc. 25th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 946–957, 2014.
- [15] W. Chen, Y. Wang, Y. Yuan, and Q. Wang, "Combinatorial multi-armed bandit and its extension to probabilistically triggered arms," *The Journal of Machine Learning Research*, vol. 17, no. 50, pp. 1–33, 2016.
- [16] T. Lin, J. Li, and W. Chen, "Stochastic online greedy learning with semi-bandit feedbacks," in *Advances in Neural Information Processing Systems*, pp. 352–360, 2015.
- [17] Y. Bao, X. Wang, Z. Wang, C. Wu, and F. C. Lau, "Online influence maximization in non-stationary social networks," in *arXiv preprint arXiv:1604.07638*, 2016.
- [18] S. Lei, S. Maniu, L. Mo, R. Cheng, and P. Senellart, "Online influence maximization," in *Proc. 21st ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pp. 645–654, 2015.
- [19] Z. Wen, B. Kveton, and M. Valko, "Influence maximization with semi-bandit feedback," *arXiv preprint arXiv:1605.06593*, 2016.
- [20] S. Vaswani, L. V. S. Lakshmanan, and M. Schmidt, "Influence maximization with bandits," *arXiv preprint arXiv:1503.00024*, 2015.