Factorization Bandits for Online Influence Maximization

Qingyun Wu
University of Virginia
85 Engineer's Way
Charlottesville, VA, USA 22904
qw2ky@virginia.edu

Zhige Li
Shanghai Jiao Tong University
Shanghai, China
l-zhige@outlook.com

Huazheng Wang
University of Virginia
85 Engineer's Way
Charlottesville, VA, USA 22904
hw7ww@virginia.edu

Wei Chen
Microsoft Research
Beijing, China
weic@microsoft.com

Hongning Wang
University of Virginia
Charlottesville, VA, USA
hw5x@virginia.edu

ABSTRACT

In this paper, we study the problem of *online influence maximization* in social networks. In this problem, a learner aims to identify the set of "best influencers" in a network by interacting with the network, i.e., repeatedly selecting seed nodes and observing activation feedback in the network. We capitalize on an important property of the influence maximization problem named **network assortativity**, which is ignored by most existing works in online influence maximization. To realize network assortativity, we factorize the activation probability on the edges into latent factors on the corresponding nodes, including influence factor on the giving nodes and susceptibility factor on the receiving nodes. We propose an upper confidence bound based online learning solution to estimate the latent factors, and therefore the activation probabilities. Considerable regret reduction is achieved by our factorization based online influence maximization algorithm. Extensive empirical evaluations on two real-world networks showed the effectiveness of our proposed solution.

CCS CONCEPTS

•**Information systems** → **Social advertising**; •**Theory of computation** → **Online learning algorithms**; **Social networks**;

KEYWORDS

Online influence maximization; Factorization bandit; Network assortativity; Regret analysis

1 INTRODUCTION

Online social networks play a vital role in the spread of information, ideas, and influence among people in their modern life [19, 20]. They have been actively utilized as a dissemination and marketing platform. For instance, in viral marketing, a marketer tries to select a set of customers with great influence for a new product promotion. With a fixed budget on the number of selections, a marketer aims to maximize the spread of this influence, which is referred to as the influence maximization problem [6, 16, 17]. In typical solutions for influence maximization, a social network is modeled as a graph with nodes representing users and edges, associated with the activation probability, representing the connections or relationship between users. Influence is propagated through the network under a specific diffusion model, such as independent cascade model and linear threshold model [16].

Most existing influence maximization solutions assume the activation probability is known beforehand. However, in many real-world social networks, this information is not observable. Solutions have been proposed to estimate the activation probabilities from a set of cascades which consist of logged actions by the network users in the past [4, 13, 23, 27]. However, as the observations are independently collected from the learning algorithms, bias might be introduced by the logging mechanism [11, 21]. To combat with biases in offline influence estimation, Aral and Walker [1] proposed randomized online experiments for offline data collection. But it is usually very expensive to carry out such experiments. This motivates the studies of online influence maximization [9, 10, 32, 33, 38], in which seed nodes are purposely selected by a learning agent to improve its quality in influence estimation and influencer selection on the fly. The foundation of this line of solutions is the combinatorial bandits [9], in which a set of arms are pulled together at each round, and the outcome is only revealed as a whole over the set of pulled arms. Mapping it back to the influence maximization problem, each node in the network is considered as an arm, and at each round the received reward on the selected set of seed nodes is the number of their activated nodes.

Most of existing online influence maximization solutions [9, 10] model the activation probability on the edges independently, which unfortunately cannot capture how social influence forms in real networks [2, 24]. To clarify, here we are referring to the independence/dependence in the estimation of influence across network edges instead of how influence diffuses. First, this independence assumption prevents the model from realizing the dependency among

the influence pattern on related nodes, i.e., the so-called assortative mixing [24]. For example, in "influentials hypothesis" [37] the tendency that a user is likely to influence his/her neighbors is defined as the *influence* of this node and the tendency that he/she is likely to be influenced is defined as the node's *susceptibility*. In Aral and Walker's work [2], via randomized experiments, they found that a node's influence should be separated from its susceptibility: influential individuals are less susceptible to be influenced than non-influential individuals. This directly suggests the activation probability on an edge should be modeled by both its end nodes' influence and susceptibility, and thus edges that share the same set of giving or receiving nodes are no longer independent from each other. Second, the distribution of influence and susceptibility over network nodes is heterogeneous. It is also reported in [2] that influential individuals cluster in the network while susceptible individuals do not. Hence, a joint modeling of nodes' influence and susceptibility is necessary. In a recent work, Aral and Dhillon [1] showed that failing to differentiate influence and susceptibility across network nodes caused an influence maximization algorithm to underestimate influence propagation by 21.7% on average, for a fixed seed set size. To overcome the issue, some online influence maximization algorithms [15, 32, 38] introduced edge-level features to help the modeling of influence probability. However, as revealed in [2], the factors affecting users' influence and susceptibility may include age, gender, marital status and many other sensitive attributes, which can hardly be exhausted and are often prohibited under privacy constraints. Third, edge-level estimation of activation probability costs an algorithm both high computational complexity and sample complexity [10], given the number of edges in a network is usually significantly larger than the number of nodes. Utilizing the dependency in the influence probability across edges is expected to reduce the complexity.

In this paper, we propose to model the dependency of activation probabilities on the edges for online influence maximization. Specifically, we assume each network node's influence and susceptibility are distinct and individually specified; and the activation probability on an edge is jointly determined by the giving node's influence and receiving node's susceptibility. This makes the activation probability matrix for the network low-rank. Then with such a low-rank structure, we propose a factorization based bandit solution to learn the latent *influence factors* and *susceptibility factors* on the nodes from the interactions with the environment. There are three important advantages of our factorization based bandit solution for online influence maximization. Firstly, it is able to capture the assortative mixing property of influence distribution in a network. Secondly, by directly learning the node-level parameters, the activation observation from one edge can be readily leveraged to other edges that share the same node. This reduces sample complexity for online learning. Third, comparing to existing online influence maximization solution with linear generation assumptions [15, 32, 38], our solution does not depend on the availability of manually constructed edge- or node-level features; it learns the property of network nodes via factorization. A rigorous theoretical analysis on the upper regret bound of the proposed solution is provided, where we prove considerable regret reduction comparing to solutions that model the activation probability on edges

independently. Extensive empirical evaluations on two large-scale networks confirmed the effectiveness of our proposed solution.

2 RELATED WORK

The problem of influence maximization has been extensively studied in offline settings [7, 8, 16], where the main focus is on computational efficiency in optimization. These solutions simply assume the influence model, i.e., the activation probabilities, can be specified by the network's properties (e.g., node degree), or by a transmission parameter that is specified as constant, random or drawn from a uniform distribution, or by estimations from logged propagation data [4, 13, 23, 27]. However, such an overly simplified influence model ignores many important properties of real network influence patterns, e.g., assortativity [24]. To conquer these limitations, online influence maximization has been studied under different assumptions and settings. Chen et al. [9] and Wang and Chen [36] formulated it as a combinatorial bandit problem and proposed an upper confidence bound [?] based algorithm to estimate activation probabilities on a per-edge basis. Lei et al. [18] studied a different objective of maximizing the number of unique activated nodes across multiple rounds of activation. In [33], a bandit learning algorithm with both node-level (which nodes are eventually activated) and edge-level (who activated those nodes) feedback was developed. Methods similar to [13, 23, 27] were used to map the node-level feedback back to edge-level activation probability estimation. Wen et al. [38] assumed the activation probability is a linear combination of edge-level features and an unknown global activation parameter, and proposed a linear contextual bandit solution. However, in practice it is very difficult to exhaustively specify the features for influence modeling on every edge, and many of those features are prohibited under privacy constraints, such as age, gender and martial status [2].

Our solution falls into another line of online influence maximization research that estimates influence parameters at the node level. Carpentier and Valko [5] proposed a minimax optimal algorithm under a local model of the influence spread, where a source node can only activate its neighbors without considering influence cascade. Vaswani et al. [32] proposed a diffusion model independent algorithm to estimate pairwise reachability between all pairs of nodes in a network. Olkhovskaya et al. [26] developed an algorithm for node-level feedback with a theoretical guarantee. But their algorithm is only designed for graphs with specific structures. In general, previous works in this line of research do not explicitly separate influence and susceptibility of nodes, which causes seriously degraded estimation of influence propagation, as reported in [1]. To the best of our knowledge, ours is the first to model the structural dependency in edge activation probabilities for online influence maximization. We assume the activation probability of an edge is jointly determined by the influence of its giving node and the susceptibility of its receiving node. We estimate the influence and susceptibility parameters of nodes via edge-level activation feedback, without the need of edge features. This directly leads to reduced computational complexity in model estimation and reduced regret in influence maximization.

3 METHOD

In this section, we discuss our developed factorization-based bandit solution for online influence maximization. We first illustrate the problem setup, then introduce the details of our solution, and conclude this section by comparing our solution with existing algorithms to demonstrate its unique advantages.

3.1 Problem Setup

3.1.1 Influence maximization. In an influence maximization problem, an input social network is modeled as a probabilistic directed graph $G = (\mathcal{V}; \mathcal{E}; \mathbf{P})$ with nodes $\mathcal{V}$ over a set of users, edges $\mathcal{E}$ for the set of directed connections between users, and activation probabilities $\mathbf{P} = (p_{e_1}, p_{e_2}, ..., p_{e_{|\mathcal{E}|}})$ where $p_e \in [0,1]$ represents the probability that the receiving node of edge e will be activated by the giving node on e. We describe how influence propagates from nodes to their network neighbors with a stochastic diffusion model D. The influence spread of the selected seed node set $\mathcal{S}$ with activation probability $\mathbf{P}$ is the expected number of nodes activated by $\mathcal{S}$ under the diffusion model D, denoted by $f_{D,\mathbf{P}}(\mathcal{S})$. Given the graph G, the IM problem aims to find a particular set $\mathcal{S}$ under a cardinality constraint K, which maximizes the influence spread $\mathcal{S}^{opt} = \arg\max_{|\mathcal{S}| \leq K} f_{D,\mathbf{P}}(\mathcal{S})$. This problem is known to be NP-hard [16, 22], but it can be efficiently solved with approximations [25, 29, 30]. In this paper, such influence maximization algorithms are treated as an oracle, which takes a given graph, seed set size and activation probability as input, and outputs an appropriate set of seeds. Define $\mathcal{S}^* = ORACLE(G, K, \mathbf{P})$ as the solution from the oracle. It serves as an (α, γ)-approximation of $\mathcal{S}^{opt}$, where $f_{D,\mathbf{P}}(\mathcal{S}^*) \geq \gamma f_{D,\mathbf{P}}(\mathcal{S}^{opt})$ with probability at least α [9]. For example, we have $\gamma = 1 - \frac{1}{e} - \varepsilon$ and $\alpha = 1 - 1/|\mathcal{V}|$ for most of influence maximization solutions [7, 8, 16], where e is the base of natural logarithm and ε depends on the accuracy of their Monte-Carlo estimate.

3.1.2 Online influence maximization with bandit. Traditional works in influence maximization [7, 8, 16] assume that in addition to the network structure, the algorithm either knows the per-edge activation probability or the probability can be specified from past propagation data [13, 23, 27]. However, as we discussed before this assumption imposes various limitations for the practical use of influence maximization solutions; and it is preferred for the learner to estimate the activation probabilities by directly interacting with the network, i.e., online influence maximization. In each round of online influence maximization, the learner needs to choose seeds to 1) maximize influence spread (i.e., exploitation), and 2) improve its knowledge of the activation probabilities via feedback (i.e., exploration). Multi-armed bandit framework is thus a natural choice to handle the exploration-exploitation trade-off for online influence maximization [33, 38].

Formally, at each round t during online influence maximization, the learner first chooses a seed node set $\mathcal{S}_t \in \mathcal{V}$ with cardinality K by a predefined oracle based on its current knowledge from past observations: $\mathcal{S}_t = ORACLE(G, K, \hat{\mathbf{P}}_t)$, where $\hat{\mathbf{P}}_t$ is the learner's current estimate of activation probability $\mathbf{P}$. Influence then diffuses from the nodes in $\mathcal{S}_t$ according to the diffusion model D. For example, under the Independent Cascade (IC) diffusion model, this can be interpreted as the environment generates a binary response $y_{e,t}$ by independently sampling $y_{e,t} \sim \text{Bern}(p_e)$ for each concerned edge e in the resulting cascade. Note the environment is always using ground-truth activation probability $\mathbf{P}$. Then the learner receives the reward $f_{\text{IC},\mathbf{P}}(\mathcal{S}_t)$ which is the number of activated nodes by $\mathcal{S}_t$. We should emphasize that for any edge $e \in \mathcal{E}$, the learner observes the realization of $y_{e,t}$ if and only if its giving node is activated. To clarify, the active nodes deactivate at each time step, which makes it different from the setting in adaptive influence maximization [12]. With such edge-level feedback, the learner updates its estimate about the edge activation probability. The learner's objective is to maximize the expected cumulative reward over a finite T rounds of interactions.

3.2 A Factorization Bandit Solution

In most of existing bandit-based online influence maximization solutions, the learner estimates the activation probabilities on all edges independently [9, 10], which cannot capture how the influence between nodes is formed and thus ignores the underlying structure behind the activation probabilities of edges. As suggested in a recent work [1], modeling nodes' influence and susceptibility is vital for influence maximization. In this work, we explicitly model this influence structure by assuming activation probability $p_e \in [0,1]$ on edge e can be decomposed into two d-dimensional latent factors on the giving node and receiving node, i.e.,

$$p_e = \boldsymbol{\theta}_{g_e}^{\mathsf{T}} \boldsymbol{\beta}_{r_e}, \tag{1}$$

where g_e and r_e denote the giving node and receiving node on edge e respectively. $\boldsymbol{\theta}_v \in \mathbb{R}^d$ is considered as the *influence factor* on node v and $\boldsymbol{\beta}_v \in \mathbb{R}^d$ is the *susceptibility factor* on node v. This essentially imposes a low-rank structure assumption on the activation probability $\mathbf{P}$ and considers the activation probabilities as a reflection of influence and susceptibility properties of the connected nodes. It naturally captures the assortativity of the influence network: users who are more influential, i.e., the nodes that have larger values on $\boldsymbol{\theta}_v$, trend to associate with larger activation probability on their neighboring node. The same for users who are more susceptible (e.g., measured by $\boldsymbol{\beta}_v$).

In this work, we assume the diffusion follows the independent cascade model [16], and the edge-level feedback $y_{e,t}$ is observable to the learner. The estimation of activation probability $\mathbf{P}$ can thus be obtained by an online factorization of the sequential observations about edge activations in the network. As a result, instead of learning the edge activation probabilities directly, the complexity of which is $O(|\mathcal{E}|)$, our solution learns the underlying influence factor $\boldsymbol{\theta}_v$ and susceptibility factor $\boldsymbol{\beta}_v$ for all the nodes. This reduces the model complexity to $O(d|\mathcal{V}|)$, which is considerably smaller than $O(|\mathcal{E}|)$, especially in a large and dense graph. To better illustrate our proposed learning solution, we introduce the following definitions about observed edges and observed nodes.

Definition 3.1 (Observed edge). For any round t, a directed edge e is considered as observed if and only if its start node is activated.

Note that an edge is observed does not necessarily mean the edge is active (i.e., not necessarily $y_{e,t} = 1$). With the observed edges, a set of nodes can be classified as observed, which is defined as follows,

Definition 3.2 (Observed node). A node v is observed if and only if at least one of its giving-neighbor nodes is active.

Based on the above definition of observed edges and observed nodes, we can realize the relationship between observed edges and observed nodes: the giving and receiving nodes on an observed edges are both observed nodes. In this case, once an activation observation $y_{e,t} = 1$ is obtained on edge e at time t, statistics about the influence factor on its giving node g_e and susceptibility factor on its receiving node r_e can be updated. More importantly, all observed edges can be used for model update, which enables observation propagation to unobserved edges. We will discuss the benefit of this aspect with more details in Section 3.3.

Due to the coupling between θ and β imposed in the influence structure assumption in Eq (1), we appeal to a coordinate decent algorithm built on matrix factorization to estimate the latent influence factor θ_v and susceptibility factor β_v for each node $v \in \mathcal{V}$. Specifically, the objective function for our factor estimation can be written as follows,

$$\min_{\{\theta_v, \beta_v\}_{v \in \mathcal{V}}} \sum_{t=1}^{T} \sum_{e \in \tilde{\mathcal{E}}_t} (\theta_{g_e}^{\mathsf{T}} \beta_{r_e} - y_{e,t})^2 + \lambda_1 \sum_{v \in \mathcal{V}} \|\theta_v\|_2 + \lambda_2 \sum_{v \in \mathcal{V}} \|\beta_v\|_2 \tag{2}$$

where $\tilde{\mathcal{E}}_t$ is the set of observed edges at time t, λ_1 and λ_2 are L2 regularization coefficients. The inclusion of L2 regularization term is critical to our solution in two folds. First, it makes the sub-problems in coordinate decent based optimization well-posed, so that we have closed form solutions for θ_v and β_v at each round. Second, it helps to remove the scaling indeterminacy between the estimates of θ_v and β_v, and makes the q-linear convergence rate of parameter estimation achievable [31, 34, 35]. The closed-form estimation of θ_v and β_v with respect to Eq (2) at round $t + 1$ can be obtained by $\hat{\theta}_{v,t+1} = \mathbf{A}_{v,t+1}^{-1} \mathbf{b}_{v,t+1}$ and $\hat{\beta}_{v,t+1} = \mathbf{C}_{v,t+1}^{-1} \mathbf{d}_{v,t+1}$, in which,

$$\mathbf{A}_{v,t+1} = \lambda_2 \mathbf{I} + \sum_{i=1}^{t} \sum_{e \in \tilde{\mathcal{E}}_i^{v,\text{giving}}} \hat{\beta}_{r_e,t} \hat{\beta}_{r_e,t}^{\mathsf{T}} \tag{3}$$

$$\mathbf{b}_{v,t+1} = \sum_{i=1}^{t} \sum_{e \in \tilde{\mathcal{E}}_i^{v,\text{giving}}} \hat{\beta}_{r_e,t} y_{e,i} \tag{4}$$

$$\mathbf{C}_{v,t+1} = \lambda_1 \mathbf{I} + \sum_{i=1}^{t} \sum_{e \in \tilde{\mathcal{E}}_i^{v,\text{receiving}}} \hat{\theta}_{g_e,t} \hat{\theta}_{g_e,t}^{\mathsf{T}} \tag{5}$$

$$\mathbf{d}_{v,t+1} = \sum_{i=1}^{t} \sum_{e \in \tilde{\mathcal{E}}_i^{v,\text{receiving}}} \hat{\theta}_{g_e,t} y_{e,i} \tag{6}$$

$\tilde{\mathcal{E}}_i^{v,\text{giving}}$ is the set of observed edges at round i where node v is the giving node, and accordingly $\tilde{\mathcal{E}}_i^{v,\text{receiving}}$ is the set of observed edges where node v is the receiving node. $\mathbf{I}$ is a $d \times d$ identity matrix.

The estimated influence factors $\{\hat{\theta}_{v,t}\}_{v \in \mathcal{V}}$ and susceptibility factors $\{\hat{\beta}_{v,t}\}_{v \in \mathcal{V}}$ give us an estimate of the edge activation probability by $\hat{p}_{e,t} = \hat{\theta}_{g_e,t}^{\mathsf{T}} \hat{\beta}_{r_e,t}$ at round t. This represents the model's current best knowledge about influence propagation in graph G,

and therefore serves for the exploitation purpose. We also need a term to control exploration in online learning; and we appeal to the upper confidence bound (UCB) principle in this work [3], as the confidence interval of our factor estimation is readily available. In our solution, the uncertainty of activation probability estimation during online update comes from the uncertainty on both latent factors, i.e., $\|\hat{\theta}_{v,t} - \theta_v^*\|$ and $\|\hat{\beta}_{v,t} - \beta_v^*\|$, where θ_v^* and β_v^* are the ground-truth factors. Based on the closed form solution in our coordinate descend estimation, the estimation of activation probability $\hat{p}_{e,t}$ can be analytically bounded with a high probability as shown in the following lemma.

LEMMA 3.3 (CONFIDENCE BOUND OF INFLUENCE PROBABILITY ESTIMATION). *For any node v, and directed edge e, $\delta \in (0, 1), q \in (0, 1)$, with a probability at least $1 - \delta$, we have,*

$$\|\hat{\theta}_{v,t} - \theta_v^*\|_{\mathbf{A}_{v,t}} \leq \alpha_v^{\beta}, \|\hat{\beta}_{v,t} - \beta_v^*\|_{\mathbf{C}_{v,t}} \leq \alpha_v^{\theta}$$

$$|p_e^* - \hat{p}_{e,t}| \leq B_{g_e,t}^{\theta} + B_{r_e,t}^{\beta} + 2q^{2t}$$

where $B_{v,t}^{\theta} = \alpha_v^{\theta} \|\hat{\theta}_{v,t}\|_{C_{v,t}^{-1}}, B_{v,t}^{\beta} = \alpha_v^{\beta} \|\hat{\beta}_{v,t}\|_{\mathbf{A}_{v,t}^{-1}}, \alpha_v^{\theta} = \sqrt{\log \frac{det(\mathbf{C}_{v,T})}{\delta^2 det(\lambda_2 \mathbf{I})}} + \frac{\lambda_2(1-q)+2q}{\sqrt{\lambda_2}(1-q)}, \text{ and } \alpha_v^{\beta} = \sqrt{\log \frac{det(\mathbf{A}_{v,T})}{\delta^2 det(\lambda_1 \mathbf{I})}} + \frac{\lambda_1(1-q)+2q}{\sqrt{\lambda_1}(1-q)}.$

This lemma provides a tight high probability upper bound of the activation probability estimation on each edge, which enables us to perform efficient UCB-based exploration for model update. Denote the resulting upper confidence bound as $\bar{p}'_{e,t}$, with $\bar{p}'_{e,t} = \hat{\theta}_{g_e,t}^{\mathsf{T}} \hat{\beta}_{r_e,t} + \mathrm{CB}_{e,t}$ and $\mathrm{CB}_{e,t} = B_{g_e,t}^{\theta} + B_{r_e,t}^{\beta} + 2q^{2t}$. As this upper bound might not be in a valid value as a probability, we use a projection operation to map it to the range of $[0, 1]$ and get $\bar{p}_{e,t} = \mathrm{Proj}_{[0,1]}(\bar{p}'_{e,t})$. $\mathrm{Proj}_{[0,1]}(x)$ is a function projecting its real value input x to the range of $[0, 1]$. More specifically, if the input x is negative, the output is 0; if x is larger than 1, the output is 1; otherwise x will be returned. Note that $\bar{p}'_{e,t}$ is a high probability upper bound of $p_{e,t}$, i.e., $p_{e,t} \leq \bar{p}'_{e,t}$. After this projection operation we can still guarantee that with a high probability $p_{e,t} \leq \bar{p}_{e,t}$. This is true because: 1). When $\bar{p}'_{e,t} > 1$, $\bar{p}_{e,t} = 1$, which naturally guarantees $p_{e,t} \leq \bar{p}_{e,t}$; 2). When $\bar{p}'_{e,t} < 0$, $\bar{p}_{e,t} = 0$, we have $p_{e,t} \leq \bar{p}'_{e,t} < \bar{p}_{e,t}$. As an upper confidence bound of activation probability $\bar{p}_{e,t}$, it well balances the need of exploration and the need of exploitation [3]. We thus directly feed $\bar{\mathbf{P}}_t = (\bar{p}_{e_1,t}, \bar{p}_{e_2,t}, ..., \bar{p}_{e_{|\mathcal{E}|},t})$ at round t into the ORACLE to solve one round of influence maximization, from which edge-level feedback on the observed edges will be obtained. Based on the observed edges, we get a set of observed nodes, and node-level influence factor θ_v and susceptibility factor β_v and their associated statistics can be updated based on their closed form solutions. We summarize the details of our factorization based online influence maximization algorithm in Algorithm 1, and name it as Influence Maximization with Factorization-Bandits, or IMFB in short.

3.3 Comparison with Existing Solutions

3.3.1 Node-level v.s., edge-level influence model estimation. In our proposed solution, we directly estimate the node-level influence and susceptibility factors by performing online factorization

Algorithm 1 Influence Maximization with Factorization-Bandits (IMFB)

1: **Inputs:** Graph G and oracle **ORACLE**, latent factor dimension d, regularization parameter λ_1 and λ_2, $q \in (0, 1)$
2: **Initialization:** $A_{v,1} = \lambda_1 I, C_{v,1} = \lambda_2 I$, sample $\hat{\theta}_{v,1} \in \mathbf{R}^d$, $\hat{\beta}_{v,1} \in \mathbf{R}^d$ for $v \in \mathcal{V}$
3: **for** $t = 1, 2, \ldots, T$ **do**
4:     Compute $\bar{p}_{e,t} = \text{Proj}_{[0,1]}(\hat{\theta}_{g_e,t}^{\mathsf{T}} \hat{\beta}_{r_e,t} + \text{CB}_{e,t})$ with $\text{CB}_{e,t} = \alpha_{g_e}^{\beta} \|\hat{\beta}_{g_e,t}\|_{A_{g_e,t}^{-1}} + \alpha_{r_e}^{\theta} \|\hat{\theta}_{r_e,t}\|_{C_{r_e,t}^{-1}} + 2q^t$ for $e \in \mathcal{E}$
5:     Choose $\mathcal{S}_t = \text{ORACLE}(G, K, \bar{\mathbf{P}}_t)$ in which $\bar{\mathbf{P}}_t = \{\bar{p}_{e,t}\}_{e \in \mathcal{E}}$
6:     Observe the edge-level feedback $\{y_{e,t}\}_{e \in \tilde{\mathcal{E}}_t}$ in which $\tilde{\mathcal{E}}_t$ is the observed edges.
7:     Update the node-level influence statistics and susceptibility statistics on the observed nodes according to Eq (3),(4),(5),(6)
8: **end for**

(a) CUCB update   (b) IMFB update

Figure 1: Comparison between CUCB and IMFB in influence model update.

based on sequential edge-level observations. In this subsection, we illustrate the advantages of such a node-level parameter estimation by comparing with solutions that directly estimate edge-level parameters. One classic online influence maximization solution with edge-level parameter estimation is the CUCB algorithm [36]. Comparing to CUCB, there are two important advantages in IMFB. First, reduced model complexity. In CUCB the edge activation probabilities are estimated independently across the edges, the complexity of which is $O(|\mathcal{E}|)$ (and it can become as large as $O(|\mathcal{V}|^2)$ in the complete graph case). While the model complexity in IMFB is $O(d|\mathcal{V}|)$, which is considerably smaller, as the dimension of latent space should be much smaller than the number of nodes. Second, reduced sample complexity. Figure 1 shows an example of influence model update in CUCB and IMFB after one round of influence maximization. Suppose node v_0 is activated at this round, according to the definition of observed edges, e_{v_0,v_1} and e_{v_0,v_2} (edges with solid lines) are observed edges, and e_{v_3,v_1}, e_{v_4,v_1}, and e_{v_5,v_1} (edges shown with dash lines) are unobserved edges. In CUCB, shown in Figure 1 (a), its activation probability statistics will only be updated for those observed edges and no information is learned for those unobserved edges. In IMFB, by utilizing the fact that the activation on a particular edge is a reflection of the giving node's influence and the receiving node's susceptibility, observations from the observed edges can be propagated to the unobserved edges, as shown in Figure 1 (b). Specifically, because $\hat{\beta}_{v_1}$ will be updated by this round's observations, the estimation of $\hat{p}_{e_{v_3,v_1}}$, $\hat{p}_{e_{v_4,v_1}}$ and $\hat{p}_{e_{v_5,v_1}}$ can all be improved for the next round of interaction. This directly accelerates the learning of influence structure. This advantage becomes more obvious and significant when the network is denser. Our theoretical analysis in the following section proves that when the network is fully connected, we can reduce the regret of IMFB comparing to CUCB by a factor of $\sqrt{|\mathcal{V}|}$. In addition, as we discussed earlier, independent edge-level estimations cannot guarantee to capture the important properties of influence propagation in real networks, e.g., assortativity. IMFB explicitly encode such structure into model learning, which can better realize the structural dependency in influence propagation.
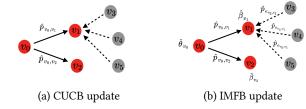
3.3.2 Activation probability v.s., reachability. A recent work, DIL-inUCB [32], also obtains an $O(d|\mathcal{V}|)$ model complexity by performing node-level parameter estimation. But DILinUCB is conceptually very different from our IMFB. Instead of estimating the activation probability on edges from the interactions, DILinUCB directly estimates the reachability between all pairs of nodes (no matter if they are directly connected or not) from edge-level feedback via a linear contextual bandit model. There are two important limitations of it. Firstly, as mentioned in their paper, the proposed reachability based surrogate objective is only an heuristic approximation to the original online influence maximization problem. There is no theoretical guarantee about its approximation quality. Secondly, although its claimed diffusion model independent, their linear reachability assumption may violate the influence diffusing rules in some diffusion models, for example the independent cascading diffusion model.

4 REGRET ANALYSIS

Let $\mathcal{S}^{opt}$ be the optimal solution on an input graph, and $\mathcal{S}^* = \text{ORACLE}(G, K, \mathbf{P}^*)$ be the solution from an oracle. Here we use $\mathbf{P}^*$ to denote the ground-truth activation probability in the network. For any $\alpha, \gamma \in [0, 1]$, we say that ORACLE is an (α, γ)-approximation oracle for a given (G, K) if for any activation probability vector $\mathbf{P}$, $f_{\mathbf{P}}(\mathcal{S}^*) \geq \gamma f_{\mathbf{P}}(\mathcal{S}^{opt})$ with probability at least α. In this section, we simplify the notation of reward function $f_{D,\mathbf{P}}(\mathcal{S})$ to $f_{\mathbf{P}}(\mathcal{S})$, since we are consistently using independent cascade model.

To analyze the performance of online influence maximization algorithms with an approximation/randomized oracle, we look at an approximate performance metric – the scaled cumulative regret: $R^{\alpha\gamma}(T) = \sum_{i=1}^{T} \mathbb{E}[R_i^{\alpha\gamma}]$, in which $\alpha\gamma > 0$ is the scale and $R_t^{\alpha\gamma} = f_{\mathbf{P}^*}(\mathcal{S}^{opt}) - \frac{1}{\alpha\gamma} f_{\mathbf{P}^*}(\mathcal{S}_t)$. Based on this regret definition, we have Theorem 4.1 specifying the regret upper bound of IMFB, which utilized two important properties of the reward function $f_{\mathbf{P}}(\mathcal{S})$ in influence maximization problem: monotonicity and 1-Norm bound smoothness [36].

THEOREM 4.1. *Assuming that the ORACLE is an (α, γ)-approximation algorithm, we have the following upper regret bound in IMFB,*

$$R^{\alpha\gamma}(T) \leq O\left(\frac{dB}{\alpha\gamma}\left(\sqrt{T|\mathcal{V}^{giving}||\mathcal{V}|D^{out}} \log(TD^{out})\right.\right. \tag{7}$$

$$\left.\left. + \sqrt{T|\mathcal{V}^{receiving}||\mathcal{V}|D^{in}} \log(TD^{in})\right)\right)$$

in which B is the bounded smoothness constant and it can be proved that $B \leq |\mathcal{V}|$ in influence maximization problem [36]. D^{out} and D^{in} are the maximum out-degree and in-degree of graph G. $|\mathcal{V}^{giving}|$ and $|\mathcal{V}^{receiving}|$ are the maximum number of observed giving nodes and receiving nodes at any time point during the interactions.

Proof sketch of Theorem 4.1: According to the regret definition, the expected regret at time t can be bounded by,

$$\mathbb{E}[R_t^{\alpha\gamma}] \leq \frac{1}{\alpha\gamma}\mathbb{E}[f_{\mathbf{P}^*}(\mathcal{S}^*) - f_{\mathbf{P}^*}(\mathcal{S}_t)] \leq \mathbb{E}[f_{\bar{\mathbf{P}}_t}(\mathcal{S}_t) - f_{\mathbf{P}^*}(\mathcal{S}_t)]$$

$$\leq B \sum_{e \in \tilde{\mathcal{E}}_t} |\bar{p}_{e,t} - p_e^*|$$

in which the second inequality is based on the monotonicity of the reward function and the ORACLE's seed node selection. The third inequality is based on the 1-Norm bounded smoothness of the reward function and B is the bounded smoothness constant [36]. Then combining the activation probability upper bound provided in Lemma 3.3, the conclusion in Theorem 4.1 can be obtained. Due to space limit, detailed proof of this theorem and related technical lemmas will be provided in the supplement of this paper.

Analysis of Theorem 4.1: Since $|\mathcal{V}^{\text{receiving}}|$, $|\mathcal{V}^{\text{giving}}|$, D^{out}, and D^{in} can all be simply upper bounded by $|\mathcal{V}|$, the worst case upper regret bound of IMFB is $O(d|\mathcal{V}|^{\frac{5}{2}}\sqrt{T})$, which is still better than the upper regret bound most online influence maximization algorithms: in a complete graph, their upper regret bound for IMLinUCB [38] and CUCB [36] are $O(d|\mathcal{V}|^3\sqrt{T})$ and $O(|\mathcal{V}|^3\sqrt{T})$ respectively.

5 EXPERIMENTS

In this section, we performed extensive empirical evaluations on two real-world networks and compare the proposed solution with a set of state-of-the-art online influence maximization baselines.

5.1 Datasets

We evaluate our proposed IMFB algorithm on two real-world datasets: NetHEPT[1] (with 27,770 nodes, and 352,807) and Flickr[2] (with 12,812 nodes and 137,986 edges). On the NetHEPT dataset, the ground-truth influence factors and susceptibility factors on the nodes are sampled from a uniform distribution $U(0, 0.1)$ and then normalized by their L2 norms. The activation probabilities on the edges are generated according to Eq (1). The average edge activation probability on this resulting dataset is 0.053. The Flickr dataset has a very skewed degree distribution, which makes it a very easy task for all the online influence maximization algorithms if the influence factors and susceptibility factors of all the nodes are sampled from the same distribution. In order to make the learning task on this dataset more challenging, we need to make the 'soft degree' distribution on this network more balanced. Here we define a node's 'soft degree' as the summation of its activation probability on all its connected outgoing edges. In order to achieve this, we sample influence factors and susceptibility factors from 10 uniform distributions with different expectations, including $U(0, 0.1), U(0.1, 0.2), ..., U(0.9, 1.0)$, and then normalize them using their L2 norms respectively. We group the nodes into 10 groups according to their 'hard degree' (here a node's original degree is referred as 'hard degree'). We assign the groups of nodes with higher hard degree to the distribution that has a lower expectation, such that we have a much more balanced soft degree distribution on this network. This makes the learning of activation probabilities more challenging. The average edge activation probability on this resulting dataset is 0.065.

[1]https://snap.stanford.edu/data/cit-HepTh.html
[2]https://snap.stanford.edu/data/web-flickr.html

In all following experiments, the dimensionality for the latent influence and susceptibility factor is set to 20, and seed node size K is set to 300 by default.

5.2 Baseline Algorithms

We empirically evaluate our proposed IMFB algorithm against the following state-of-the-art online IM algorithms.

• **CUCB** [9, 36]: it estimates the activation probabilities on the edges independently and uses upper confidence bound of the estimation to balance exploitation and exploration.

• **ϵ-greedy**: it estimates the activation probabilities on the edges independently, and uses ϵ-greedy [28] to balance exploitation and exploration.

• **IMLinUCB** [38]: it learns an edge-level bandit model and estimates the activation probabilities with given edge-level features. In our experiment, we test two variants of IMLinUCB depending on how the edge-level features are constructed: (1). We directly use the ground-truth influence and susceptibility factors θ^* and β^* to generate a set of high-quality features for it. For each edge e, we take outer product on $\theta_{g_e}^*$ and $\beta_{r_e}^*$ and reshape it to a d^2-dimensional vector. We need to emphasize that since $\theta_{g_e}^*$ and $\beta_{r_e}^*$ are directly used to generate the ground-truth activation probability, this feature setting can lead to the best possible version of IMLinUCB, i.e., it only needs to recognize the diagonal terms in the outer product. (2). We follow the setting proposed in [38], where we first use node2vec [14] to create node embedding vectors and use an element-wise product between two nodes of an edge to get the edge features. This variant of IMLinUCB is referred to as IMLinUCB-Node2Vec in the following discussion.

• **DILinUCB** [32]: is a model-independent contextual bandit algorithm for online influence maximization. DILinUCB requires features on the receiving nodes as input to estimate the reachability between node pairs. We provide ground-truth susceptibility factor β_v^* for $v \in \mathcal{V}$ as input of node features to DILinUCB.

For IMFB and all baselines except DILinUCB, the DegreeDiscountIC algorithm proposed in [8] is used as the influence maximization oracle. For DILinUCB, since the original influence maximization objective function is replaced with a heuristic surrogate function, the greedy oracle suggested in their solution [32] is used here. In all our experiments, we run all algorithms over 200 rounds and report the averaged results from 5 independent runs.

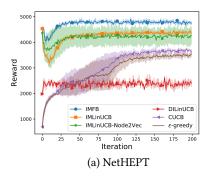## 5.3 Comparison on Influence Maximization

We first compare different algorithms' performance in maximizing influence during online learning. For better visibility, all subfigures in Figure 2 share the same set of legend.
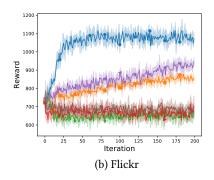
In Figure 2 (a) & (b), we report the real-time average reward and variance (showed by the shaded area) of the collected reward from different algorithms on NetHEPT and Flickr dataset respectively. From this result, we first observe that IMFB shows significant improvement over all the baselines on both datasets. We also observe that on both datasets DILinUCB performs consistently the worst, which is expected because of the heuristic reachability assumption.

On the NetHEPT dataset shown in Figure 2 (a), we can observe that the learning process of CUCB and ϵ-greedy are much slower than the other solutions. This observation verified our claim about

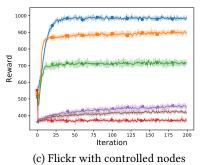(a) NetHEPT   (b) Flickr   (c) Flickr with controlled nodes

Figure 2: Real-time average reward from online influence maximization.

the necessity of explicitly modeling the influence structure: when the edge activation probabilities are independently modeled (e.g., in CUCB and ϵ-greedy), the learner can only update and improve its estimation quality on the observed edges. This makes the learning very slow, especially on this relative denser dataset. Although uncertainty of the estimation can help avoid local optimal, it may also take time to recognize the globally optimal seed nodes, especially when a lot of edges need exploration. While for IMFB and both variants of IMLinUCB, since they can propagate observations through node-level parameter estimation (in IMFB) or shared edge parameter estimation, the learning process is much faster.

On the Flickr dataset shown in Figure 2 (b), IMLinUCB performs even worse than CUCB. The potential reason for this bad performance is that IMLinUCB is over-parameterized, which slows down its learning convergence. For example, among all provided d^2 features, only d of them are effective. The algorithm needs a lot more observations to recognize the relevance of those features. While at the same time, since the dataset is relatively sparser, i.e., the number of observed edges are relatively smaller in each round of interaction, which makes IMLinUCB's learning process even slower. In addition, we can also notice that IMLinUCB-Node2Vec is one of the worst performing algorithms. This indicates IMLinUCB is very sensitive to the quality of manually constructed edge-features.

Based on our observations in Figure 2 (a) & (b), we designed a new setting on the Flickr dataset to better illustrate the importance and advantages of node-level parameter estimation in our proposed IMFB. On the Flickr dataset, we first categorize the nodes into two types: Type 1 includes nodes that have very high hard degrees; and Type 2 includes nodes that have relatively low hard degrees. Then we controlled the generation of influence and susceptibility factors to make the out-going edges from Type 1 nodes associated with extremely low activation probabilities. The purpose of this design is to make the average soft degree in Type 1 nodes small. And for Type 2 nodes, we control the activation probability generation to make their associated edges' activation probability extremely large. And we remove some of the edges (but not all of them) connecting these two types of nodes to further control the degree distribution.

At the beginning of online influence maximization, when a learner's estimation about activation probabilities is not accurate, hard degree plays a dominating role in the ORACLE's decision. Hence, on this manipulated Flickr network, the ORACLE trends to start with Type 1 nodes (because their hard degree is very high) but the optimal seed nodes are more likely to concentrate on Type

2 nodes, where the average activation probability is higher. In this case, if there is no observation prorogation among edges, it would take more rounds of interactions for a learner to realize that Type 2 nodes are actually better. In IMFB, even though at the beginning stage it would also start from Type 1 nodes, because of its node-level parameter estimation, observations from observed edges can be effectively propagated to those unobserved edges. If those unobserved edges are connected to Type 2 nodes (because we did not remove all the edges connecting Type 1 and Type 2 nodes), estimation about the Type 2 nodes and their connected edges can thus be improved. This observation propagation to unobserved edges helps IMFB learn much faster than the other baselines. This benefit is also illustrated in Figure 1 of Section 3.3. Empirical evaluation of average reward under this setting can be found in Figure 2 (c), which verified our claimed benefit on IMFB: On this dataset, IMFB obtained much more significant improvement: it obtained 230% improvement over CUCB and ϵ-greedy.

5.4 Influence Model Estimation Quality

In order to better understand the algorithms' effectiveness in activation probability learning, we compared the estimation quality of IMFB with related baselines on both datasets in Figure 3 with solid lines (using the right-side of y-axis). Specifically, we calculated the absolute difference between the estimated activation probability $\hat{p}_{e,t}$ and ground-truth probability p_e^* over *observed edges*. We omit ϵ−greedy and DILinUCB here since ϵ−greedy's performance is very similar to CUCB, and DILinUCB does not directly estimate the activation probabilities.
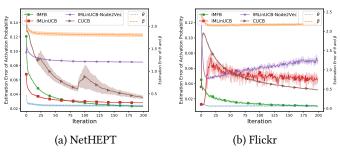

(a) NetHEPT   (b) Flickr

Figure 3: Influence model estimation quality.

From the results on NetHEPT in Figure 3 (a), we can observe that, in addition to IMFB, the estimation error of activation probabilities in IMLinUCB also converge very fast but still worse than IMFB.

Table 1: Robustness to dimensionality misspecification.

	IMFB-1	IMFB-5	IMFB-20	IMFB-60	ϵ-greedy	CUCB	IMLinUCB-Node2Vec	IMLinUCB	DILinUCB
NetHEPT	**4657.80**	**4711.30**	**4731.90**	**4715.80**	2882.10	3022.30	4262.40	4392.50	2368.80
Flickr	**979.80**	**1044.88**	**1054.56**	**1044.17**	706.14	880.37	744.52	800.86	715.89

Table 2: Reward comparison under different environment settings on Flickr dataset.

	IMFB	ϵ-greedy	CUCB	IMLinUCB-Node2Vec	IMLinUCB	DILinUCB
$K = 50$	**203.92**	161.80	194.71	130.47	179.24	131.31
$K = 100$	**405.50**	295.09	352.51	243.68	332.28	257.73
$K = 300$	**1054.56**	706.14	880.37	744.52	800.86	715.89
$K = 1,000$	**2737.86**	2224.10	2508.25	2013.49	2157.98	2198.64
$\eta = 0$	**1054.56**	706.14	880.37	744.52	800.86	715.89
$\eta \sim U(-0.05, 0.05)$	**1455.83**	1120.56	1241.26	1205.89	1267.67	1131.39
$\eta \sim U(-0.1, 0.1)$	**1956.80**	1641.30	1740.00	1856.83	1921.10	1711.80
$\eta \sim U(-0.3, 0.3)$	**2766.07**	2587.10	2690.30	2659.34	2736.78	2590.18
$c = 0.5$	**432.45**	359.19	367.75	383.46	412.45	350.28
$c = 1.0$	**1054.56**	706.14	880.37	744.52	800.86	715.89
$c = 1.5$	**2727.07**	1863.94	2407.70	2035.44	2479.88	2278.18
$c = 2.0$	**3884.68**	3517.8	3776.86	3598.31	3658.57	3526.53

IMLinUCB-Node2Vec has a much worse activation probability estimation than all the other baselines. The unsatisfactory activation probability estimation in CUCB again verified our motivation of joint edge level parameter estimation. We can see that at around round 85, there is a sudden jump of its activation probability estimation quality. This is because at that round CUCB starts to explore edges that were rarely observed before. For those edges, CUCB has to learn the activation probability from scratch, which leads to a sudden drop and a large variance in its estimation. This observation directly verify phenomenon shown in the illustration example in Figure 1, where we showed CUCB can only improve the estimation of activation probabilities on edges that are directly observed.

On the Flickr dataset in Figure 3 (b), we observe similar estimation error pattern in CUCB at round interaction 15. Since this happens much earlier than that on the NetHEPT dataset, the performance of CUCB is not greatly affected after all. And on this dataset, IMFB converges the fastest and converges to an error that is much smaller than any other baselines'. Results on both these two dataset empirically confirmed the reduced sample complexity of IMFB.

In addition, we also visualize the average L2 difference between the estimated factors in IMFB and the ground-truth factors in the environment, i.e., $\|\hat{\boldsymbol{\theta}}_{v,t} - \boldsymbol{\theta}_v^*\|_2$ and $\|\hat{\boldsymbol{\beta}}_{v,t} - \boldsymbol{\beta}_v^*\|_2$, on *observed* nodes v. The results on both datasets are shown in Figure 3 (a) & (b) with dotted lines using the left side of the y-axis. We can find that within 25 iterations, IMFB's estimation error in both $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$ reduces. As a result, its estimated activation probability $\hat{p}_{e,t}$ quickly converges to ground-truth probability, which corresponds to the convergence of the real time reward. The good estimation quality of activation probability in IMFB coincides our analysis in Lemma 3.3 regarding the confidence bound of activation probability estimation.

5.5 Robustness in Influence Model Learning

5.5.1 Dimensionality Misspecification. In our proposed solution, dimensionality of the latent influence factors and susceptibility factors need to be specified as the input. On the tested datasets, the

dimensionality is set to 20 when generating the ground-truth activation probabilities. However, such information would be unknown to a learner in practice. Thus it is important to study the robustness of our proposed IMFB algorithm when dimensionality of latent factor is misspecified. Here we use IMFB-d' to denote IMFB with latent dimension size d'. We vary d' to study IMFB's robustness to dimensionality misspecification. Average reward on NetHEPT and Flickr dataset are reported in Table 1. We also include results from all the other baselines for comparison. From this result we can clearly find that: firstly IMFB is very robust to dimensionality misspecification, specifically when d' is set within a reasonable range. For example when d' is in the range of 5 to 60, given the ground-truth dimension size is 20, the variance in the performance of IMFB is very small. Even when d' is set to 1, IMFB can still maintain competitive performance. Secondly, by comparing IMFB with all other baselines, we can see that even when the dimensionality is misspecified, IMFB can still significantly outperform all the baselines. This suggests specifying the correct influence structure is more important for online influence maximization.

5.5.2 Different Environment Settings. To examine our algorithm's robustness to different environment setups, we performed evaluation on the Flickr dataset by varying several important parameters of the environment, and report the results in Table 2.
• **Number of seed nodes K.** We vary the number of seed nodes K from 50 to 1,000. We observe that for different settings of K, IMFB consistently performs better than other baselines.
• **Noise level η.** Previous works in online influence maximization assumes the activation of edge e is controlled by a fixed ground-truth activation probability p_e. In order to get a sense of our algorithm's robustness to additional noise on the edge activation, at each round of interaction we add an additional noise term η on the ground-truth activation probabilities and then project the resulting noisy activation probabilities to $[0, 1]$. From Table 2 we observe that IMFB is robust to different noise levels and outperforms other baselines even when the noise scale is very large.

• **Activation probability scale coefficient** c. To simulate some practical scenarios where the graph is generally more active or inactive, we re-scale the activation probability p_e in our default setting by a constant coefficient c, i.e., activate edge e with probability $c \cdot p_e$, where c is varied from 0.5 to 2.0. According to the results in Table 2, our proposed IMFB is robust to the choice of c and generally improves over other baselines. We also observe that when $c = 2.0$ the difference between all algorithms becomes much smaller. This is expected: when c is large, the network is very active in every node, which makes the number of observations much larger which reduces necessity of observation propagation.

6 CONCLUSIONS & FUTURE WORK

Motivated by the network assortative mixing property, we decompose the edge activation probability into giving node's influence and receiving node's susceptibility, and therefore perform node-level influence model estimation for better model complexity and generalization in online influence maximization. We provide rigorous theoretical analysis of IMFB, and show considerable regret reduction comparing with several state-of-the-art online maximization solutions. We also empirically evaluate on two real-world network datasets and confirm the effectiveness and robustness of IMFB.

Our factorization based solution provides a flexible framework to introduce more desired properties, e.g., homophily and simultaneity, to influence modeling. Currently, we have assumed the availability of edge-level feedback from influence maximization. But in practice node-level feedback is more prevalent, we can incorporate more sophisticated inference methods based on our factor models to learn from such feedback.

7 ACKNOWLEDGEMENTS

REFERENCES

[1] Sinan Aral and Paramveer S. Dhillon. 2018. Social influence maximization under empirical influence models. *Nature Human Behaviour* 6, 2 (2018), 375–382.
[2] Sinan Aral and Dylan Walker. 2012. Identifying Influential and Susceptible Members of Social Networks. *Science* 337, 6092 (2012), 337–341.
[3] Peter Auer. 2002. Using Confidence Bounds for Exploitation-Exploration Trade-offs. *Journal of Machine Learning Research* 3 (2002), 397–422.
[4] Simon Bourigault, Sylvain Lamprier, and Patrick Gallinari. 2016. Representation learning for information diffusion through social networks: an embedded cascade model. In *Proceedings of the 9th ACM WSDM*. ACM, 573–582.
[5] Alexandra Carpentier and Michal Valko. 2016. Revealing graph bandits for maximizing local influence. In *AISTATS*. 10–18.
[6] Damon Centola and Michael Macy. 2007. Complex contagions and the weakness of long ties. *American journal of Sociology* 113, 3 (2007), 702–734.
[7] Wei Chen, Chi Wang, and Yajun Wang. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD*. ACM, 1029–1038.
[8] Wei Chen, Yajun Wang, and Siyu Yang. 2009. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD*. ACM, 199–208.
[9] Wei Chen, Yajun Wang, and Yang Yuan. 2013. Combinatorial multi-armed bandit: General framework and applications. In *ICML*. 151–159.
[10] Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. 2016. Combinatorial Multi-armed Bandit and Its Extension to Probabilistically Triggered Arms. *J. Mach. Learn. Res.* 17, 1 (Jan. 2016), 1746–1778.
[11] Abdulrahman M El-Sayed, Peter Scarborough, Lars Seemann, and Sandro Galea. 2012. Social network analysis and agent-based modeling in social epidemiology. *Epidemiologic Perspectives & Innovations* 9, 1 (2012), 1.
[12] Daniel Golovin and Andreas Krause. 2011. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. *Journal of Artificial Intelligence Research* 42 (2011), 427–486.
[13] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. 2010. Learning influence probabilities in social networks. In *Proceedings of the third ACM WSDM*. ACM, 241–250.
[14] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD*. ACM, 855–864.
[15] Dimitris Kalimeris, Yaron Singer, Karthik Subbian, and Udi Weinsberg. 2018. Learning Diffusion using Hyperparameters. In *Proceedings of the 35th ICML*, Vol. 80. PMLR, StockholmsmÄssan, Stockholm Sweden, 2420–2428.
[16] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD*. ACM, 137–146.
[17] Maksim Kitsak, Lazaros K Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H Eugene Stanley, and Hernán A Makse. 2010. Identification of influential spreaders in complex networks. *Nature physics* 6, 11 (2010), 888.
[18] Siyu Lei, Silviu Maniu, Luyi Mo, Reynold Cheng, and Pierre Senellart. 2015. Online influence maximization. In *Proceedings of the 21th ACM SIGKDD*. ACM, 645–654.
[19] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.
[20] Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. 2007. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM*. ACM, 29–42.
[21] Stephen L Morgan and Christopher Winship. 2014. *Counterfactuals and causal inference.* Cambridge University Press.
[22] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming* 14, 1 (1978), 265–294.
[23] Praneeth Netrapalli and Sujay Sanghavi. 2012. Learning the Graph of Epidemic Cascades. *SIGMETRICS Perform. Eval. Rev.* 40, 1 (June 2012), 211–222. https://doi.org/10.1145/2318857.2254783
[24] Mark EJ Newman. 2002. Assortative mixing in networks. *Physical review letters* 89, 20 (2002), 208701.
[25] Hung T. Nguyen, My T. Thai, and Thang N. Dinh. 2016. Stop-and-Stare: Optimal Sampling Algorithms for Viral Marketing in Billion-scale Networks. In *Proceedings of the 2016 ACM SIGMOD*. ACM, New York, NY, USA, 695–710. https://doi.org/10.1145/2882903.2915207
[26] Julia Olkhovskaya, Gergely Neu, and GÃ¡bor Lugosi. 2018. Online Influence Maximization with Local Observations. (2018). arXiv:cs.LG/1805.11022
[27] Kazumi Saito, Ryohei Nakano, and Masahiro Kimura. 2008. Prediction of Information Diffusion Probabilities for Independent Cascade Model. In *Proceedings of the 12th KES*. 67–75.
[28] Richard S Sutton, Andrew G Barto, et al. 1998. *Introduction to reinforcement learning.* Vol. 135. MIT press Cambridge.
[29] Youze Tang, Yanchen Shi, and Xiaokui Xiao. 2015. Influence Maximization in Near-Linear Time: A Martingale Approach. In *Proceedings of the 2015 ACM SIGMOD*. ACM, New York, NY, USA, 1539–1554.
[30] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence Maximization: Near-optimal Time Complexity Meets Practical Efficiency. In *Proceedings of the 2014 ACM SIGMOD*. ACM, New York, NY, USA, 75–86.
[31] André Uschmajew. 2012. Local convergence of the alternating least squares algorithm for canonical tensor approximation. *SIAM J. Matrix Anal. Appl.* 33, 2 (2012), 639–652.
[32] Sharan Vaswani, Branislav Kveton, Zheng Wen, Mohammad Ghavamzadeh, Laks VS Lakshmanan, and Mark Schmidt. 2017. Model-independent online learning for influence maximization. In *ICML*. 3530–3539.
[33] Sharan Vaswani, Laks Lakshmanan, Mark Schmidt, et al. 2015. Influence Maximization with Bandits. *arXiv preprint arXiv:1503.00024* (2015).
[34] Huazheng Wang, Qingyun Wu, and Hongning Wang. 2016. Learning hidden features for contextual bandits. In *Proceedings of the 25th ACM CIKM*. ACM, 1633–1642.
[35] Huazheng Wang, Qingyun Wu, and Hongning Wang. 2017. Factorization bandits for interactive recommendation. In *Thirty-First AAAI Conference on Artificial Intelligence*.
[36] Qinshi Wang and Wei Chen. 2017. Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications. In *NIPS*. 1161–1171.
[37] Duncan J Watts and Peter Sheridan Dodds. 2007. Influentials, networks, and public opinion formation. *Journal of consumer research* 34, 4 (2007), 441–458.
[38] Zheng Wen, Branislav Kveton, Michal Valko, and Sharan Vaswani. 2017. Online influence maximization under independent cascade model with semi-bandit feedback. In *NIPS*. 3022–3032.

8 SUPPLEMENT

8.1 Implementation of IMFB

The implementation of our proposed solution IMFB is publicly available at https://github.com/Matrix-Factorization-Bandit/IMFB-KDD2019.

8.2 Proof of Theorem 4.1 and related lemmas

To carry out our analysis, we make the following mild conditions on the expected reward $f_P(S)$. Note that these two conditions are the same as stated in [36].

- **Monotonicity**. The expected reward of playing any super arm $S \in \mathbf{S}$ is monotonically non-decreasing with respect to the expectation vector, i.e., if for all $i \in [m]$ $p_i \le p'_i$, we have $f_P(S) \le f_{P'}(S)$ for all $S \in \mathbf{S}$, in which $\mathcal{S}$ is the set of all candidate super arms.

- **1-Norm Bounded Smoothness**. We say that a CMAB-T (Combinatorial Multi-Armed Bandit with probabilistically Triggered arms) satisfy 1-norm bounded smoothness, if there exists a bounded smoothness constant $B \in \mathbb{R}^+$ such that for any two distributions with expectation vectors $\mathbf{P}$ and $\mathbf{P}'$, and any action $\mathcal{S}$, we have $|f_P(S) - f_{P'}(S)| \le B \sum_{i \in \tilde{S}} |p_i - p'_i|$, where $\tilde{S}$ is the set of arms that are triggered by $\mathcal{S}$.

PROOF OF THEOREM 4.1. According to the definition of scaled regret, we have

$$\mathbb{E}[R_t^{\alpha\gamma}] = f_{P^*}(S^{opt}) - \mathbb{E}[\frac{1}{\alpha\gamma}f_{P^*}(S_t)] \qquad (8)$$

$$\le \frac{1}{\alpha\gamma}\mathbb{E}[f_{P^*}(S^*) - f_{P^*}(S_t)]$$

where the expectation is over the possible randomness of S_t since the oracle might be a randomized algorithm. Notice that the randomness coming from the edge activation is already taken care of in the definition of r function. For any $t \le T$, we define event ξ_t as,

$$\xi_{t-1} = \{|p_e^* - \hat{p}_{e,t}| \le \alpha_{v_{ge}}^{\beta}\|\hat{\boldsymbol{\beta}}_{ge,i-1}\|_{A_{ge,i-1}^{-1}} \qquad (9)$$

$$+ \alpha_{r_e}^{\theta}\|\hat{\boldsymbol{\theta}}_{v_{re},i-1}\|_{C_{re,i-1}^{-1}} + 2q^{2i}, \forall e \in \mathcal{E}, \forall i \le t\}$$

and $\bar{\xi}_{t-1}$ as the complement event of ξ_{t-1}. Hence we have,

$$\mathbb{E}[R_t^{\alpha\gamma}] \le \frac{\mathbb{P}(\xi_{t-1})}{\alpha\gamma}\mathbb{E}[f_{P^*}(S^*) - f_{P^*}(S_t)|\xi_{t-1}] + \mathbb{P}(\bar{\xi}_{t-1})(L-K) \qquad (10)$$

$$\le \frac{\mathbb{P}(\xi_{t-1})}{\alpha\gamma}\mathbb{E}[f_{\bar{P}_t}(S^*) - f_{P^*}(S_t)|\xi_{t-1}] + \mathbb{P}(\bar{\xi}_{t-1})(L-K)$$

$$\le \frac{\mathbb{P}(\xi_{t-1})}{\alpha\gamma}\mathbb{E}[f_{\bar{P}_t}(S_t) - f_{P^*}(S_t)|\xi_{t-1}] + \mathbb{P}(\bar{\xi}_{t-1})(L-K)$$

in which the second inequality is based on the monotonicity of the expected reward function (since we proved that with high probability each element in $\bar{P}_t$ is an upper bound of the coressponding element in P^*.), and the third inequality is based on the oracle's seed node selection strategy.

By the 1-Norm bounded smootheness of the reward function as stated in Condition 2, we have,

$$f_{\bar{P}_t}(S_t) - f_{P^*}(S_t) \le B \sum_{e \in \tilde{\mathcal{E}}_t} |\bar{p}_{e,t} - p_e^*]| \qquad (11)$$

in which $\tilde{\mathcal{E}}_t$ is the set of observed edges at interaction t (i.e. the set of triggered edges by S_t at time t if map it to the 1-Norm Bounded Smoothness condition).

Substituting Eq (11) to Eq (10) and combining the definition of cumulateive regret we have,

$$R^{\alpha\gamma}(T) = \sum_{t=1}^{T}\mathbb{E}[R_i^{\alpha\gamma}] \qquad (12)$$

$$\le \frac{B}{\alpha\gamma}\sum_{t=1}^{T}\sum_{e \in \tilde{\mathcal{E}}_t}|\bar{p}_{e,t} - p_e^*| + \mathbb{P}(\bar{\xi}_{t-1})(L-K)$$

According to a proof similarly as in Lemma 2 of [38], the term $\mathbb{P}(\bar{\xi}_{t-1})(L-K)$ can be bounded by 1. Then combining the conclusion in Lemma 8.1, we have,

$$\mathbf{R}^{\alpha\gamma}(T) \le O(\frac{dB}{\alpha\gamma}(\sqrt{T|\mathcal{V}^{\text{giving}}||\mathcal{V}|D^{\text{out}}\log(TD^{\text{out}})} \qquad (13)$$

$$+ \sqrt{T|\mathcal{V}^{\text{receiving}}||\mathcal{V}|D^{\text{in}}\log(TD^{\text{in}})}))$$

□

LEMMA 8.1. Denote $\mathcal{V}_{S_t}^{receiving}$ as the set of observed giving nodes at time t under seed node set S_t, $\mathcal{V}_{S_t}^{receiving}$ as the set of receiving nodes at time t under seed node set S_t, we have with high probability,

$$\sum_{t=1}^{T}\sum_{e \in \tilde{\mathcal{E}}_t}|\bar{p}_{e,t} - p_e^*| \qquad (14)$$

$$\le O(d\sqrt{T|\mathcal{V}^{giving}||\mathcal{V}|D^{out}\log(TD^{out})}$$

$$+ d\sqrt{T|\mathcal{V}^{receiving}||\mathcal{V}|D^{in}\log(TD^{in})})$$

PROOF OF LEMMA 8.1.

$$\sum_{t=1}^{T}\sum_{e \in \tilde{\mathcal{E}}_t}|\bar{p}_{e,t} - p_e^*| \qquad (15)$$

$$\le \sum_{t=1}^{T}\sum_{e \in \tilde{\mathcal{E}}_t}(\alpha_{v_{ge}}^{\beta}\|\hat{\boldsymbol{\beta}}_{ge,i-1}\|_{A_{ge,i-1}^{-1}} + \alpha_{r_e}^{\theta}\|\hat{\boldsymbol{\theta}}_{v_{re},i-1}\|_{C_{v_{re},i-1}^{-1}} + 2q^t)$$

$$= \sum_{t=1}^{T}\sum_{u \in \mathcal{V}_{S_t,}^{\text{giving}}}\alpha_u^{\beta}\|\hat{\boldsymbol{\beta}}_{u,t}\|_{A_{u,t}^{-1}} + \sum_{t=1}^{T}\sum_{v \in \mathcal{V}_{S_t,}^{\text{receiving}}}\alpha_v^{\theta}\|\hat{\boldsymbol{\theta}}_{v,i-1}\|_{C_{v,i-1}^{-1}} + \sum_{t=1}^{T}|\tilde{\mathcal{E}}_t|2q^t$$

$$(16)$$

$$\le \alpha^{\beta}\sum_{t=1}^{T}\sum_{u \in \mathcal{V}_{S_t,}^{\text{giving}}}\|\hat{\boldsymbol{\beta}}_{u,t}\|_{A_{u,t}^{-1}} + \alpha^{\theta}\sum_{t=1}^{T}\sum_{v \in \mathcal{V}_{S_t,}^{\text{receiving}}}\|\hat{\boldsymbol{\theta}}_{v,i-1}\|_{C_{v,i-1}^{-1}} + \sum_{t=1}^{T}|\tilde{\mathcal{E}}_t|2q^t$$

in which $\alpha^{\boldsymbol{\beta}} = \max_{v \in \mathcal{V}}\alpha_v^{\boldsymbol{\beta}}$ and $\alpha^{\boldsymbol{\theta}} = \max_{v \in \mathcal{V}}\alpha_v^{\boldsymbol{\theta}}$.

$$\sum_{t=1}^{T} \sum_{u \in \mathcal{V}_{S_t,}^{\text{giving}}} \|\hat{\boldsymbol{\beta}}_{u,t}\|_{A_{u,t}^{-1}} \leq \sqrt{T|\mathcal{V}^{\text{giving}}| \sum_{t=1}^{T} \sum_{u \in \mathcal{V}_{S_t,}^{\text{giving}}} \|\hat{\boldsymbol{\beta}}_{u,t}\|_{A_{u,t}^{-1}}^2} \tag{17}$$

$$\leq \sqrt{T|\mathcal{V}^{\text{giving}}||\mathcal{V}|d \frac{D^{\text{out}} \log(1 + \frac{TD^{\text{out}}}{d\delta^2})}{\log(1 + 1/\delta^2)}}$$

in which $|\mathcal{V}^{\text{giving}}|$ is the maximum number of observed giving nodes at any time point, and the last inequality is based on Lemma 8.2. and similarly we have,

$$\sum_{t=1}^{T} \sum_{v \in \mathcal{V}_{S_t,}^{\text{receiving}}} \|\hat{\boldsymbol{\theta}}_{v,t}\|_{C_{v,t}^{-1}} \leq \sqrt{T|\mathcal{V}^{\text{receiving}}| \sum_{t=1}^{T} \sum_{v \in \mathcal{V}_{S_t,}^{\text{receiving}}} \|\hat{\boldsymbol{\theta}}_{v,t}\|_{C_{v,t}^{-1}}^2} \tag{18}$$

$$\leq \sqrt{T|\mathcal{V}^{\text{receiving}}||\mathcal{V}|d \frac{D^{\text{in}} \log(1 + \frac{TD^{\text{in}}}{d\delta^2})}{\log(1 + 1/\delta^2)}}$$

in which $|\mathcal{V}^{\text{receiving}}|$ is the maximum number of observed receiving nodes at any time point, and the last inequality is based on Lemma 8.2.

Then we obtain that,

$$\sum_{t=1}^{T} \sum_{e \in \mathcal{E}_{S_t}} |\bar{p}_{e,t} - p_e^*| \tag{19}$$

$$\leq \alpha^{\boldsymbol{\beta}} \sqrt{T|\mathcal{V}^{\text{giving}}||\mathcal{V}|d \frac{D^{\text{out}} \log(1 + \frac{TD^{\text{out}}}{d\delta^2})}{\log(1 + 1/\delta^2)}}$$

$$+ \alpha^{\boldsymbol{\theta}} \sqrt{T|\mathcal{V}^{\text{receiving}}||\mathcal{V}|d \frac{D^{\text{out}} \log(1 + \frac{TD^{\text{in}}}{d\delta^2})}{\log(1 + 1/\delta^2)}} + |\mathcal{E}_O| \frac{2q(1 - q^T)}{1 - q}$$

$$\leq O(d\sqrt{T|\mathcal{V}^{\text{giving}}||\mathcal{V}|D^{\text{out}} \log(TD^{\text{in}})}$$

$$+ d\sqrt{T|\mathcal{V}^{\text{receiving}}||\mathcal{V}|D^{\text{in}} \log(TD^{\text{in}})})$$

in which $\alpha^{\boldsymbol{\beta}} = \sqrt{d \log(\frac{d+TD^{\text{out}}}{d\delta^2 \lambda_1})} + \frac{\lambda_1(1-q)+2q}{\sqrt{\lambda_1}(1-q)}$, and $\alpha^{\boldsymbol{\theta}} = \sqrt{d \log(\frac{d+TD^{\text{in}}}{d\delta^2 \lambda_2})} + \frac{\lambda_2(1-q)+2q}{\sqrt{\lambda_2}(1-q)}$ and $|\mathcal{E}_O|$ is the maximum number of observed edges at any interaction up to time T.

$\square$

LEMMA 8.2. *Denote $\mathcal{V}_{S_t}^{giving}$ as the set of observed giving nodes at time t under seed node set S_t, $\mathcal{V}_{S_t}^{receiving}$ as the set of receiving nodes at time t under seed node set S_t, we have*

$$\sum_{t=1}^{T} \sum_{u \in \mathcal{V}_{S_t,}^{giving}} \|\hat{\boldsymbol{\beta}}_{u,t}\|_{A_{u,t}^{-1}}^2 \leq |\mathcal{V}| \frac{dD^{out} \log(1 + \frac{TD^{out}}{d\delta^2})}{\log(1 + 1/\delta^2)} \tag{20}$$

and

$$\sum_{t=1}^{T} \sum_{v \in \mathcal{V}_{S_t,}^{receiving}} \|\hat{\boldsymbol{\theta}}_{v,t}\|_{C_{v,t}^{-1}}^2 \leq |\mathcal{V}| \frac{dD^{in} \log(1 + \frac{TD^{in}}{d\delta^2})}{\log(1 + 1/\delta^2)} \tag{21}$$

PROOF OF LEMMA 8.2. The proof of this lemma is similar as Lemma 1 in [38]. $\square$

PROOF OF LEMMA 3.3. The proof of inequalities $\|\hat{\theta}_{v,t} - \theta_v^*\|_{A_{v,t}} \leq \alpha_v^{\beta}$ and $\|\hat{\beta}_{v,t} - \beta_v^*\|_{C_{v,t}} \leq \alpha_v^{\theta}$ are technically similar as the proof in Lemma 1 in [34], and thus omitted here. For the simplicity of notations, we denote the head node and tail node of edge e as u and v respectively.

$$p_e^* - \hat{p}_{e,t} \tag{22}$$

$$= \boldsymbol{\theta}_u^{*\mathsf{T}} \boldsymbol{\beta}_v^* - \hat{\boldsymbol{\theta}}_{u,t}^{\mathsf{T}} \hat{\boldsymbol{\beta}}_{v,t}$$

$$= \boldsymbol{\theta}_u^{*\mathsf{T}} \boldsymbol{\beta}_v^* - \boldsymbol{\theta}_u^{*\mathsf{T}} \hat{\boldsymbol{\beta}}_{v,t} + \boldsymbol{\theta}_u^{*\mathsf{T}} \hat{\boldsymbol{\beta}}_{v,t} - \hat{\boldsymbol{\theta}}_{u,t}^{\mathsf{T}} \hat{\boldsymbol{\beta}}_{v,t}$$

$$= \boldsymbol{\theta}_u^{*\mathsf{T}}(\boldsymbol{\beta}_v^* - \hat{\boldsymbol{\beta}}_{v,t}) + (\boldsymbol{\theta}_u^* - \hat{\boldsymbol{\theta}}_{u,t})^{\mathsf{T}} \hat{\boldsymbol{\beta}}_{v,t}$$

$$= (\boldsymbol{\theta}_u^* - \hat{\boldsymbol{\theta}}_{u,t})^{\mathsf{T}}(\boldsymbol{\beta}_v^* - \hat{\boldsymbol{\beta}}_{v,t}) + \hat{\boldsymbol{\theta}}_u^{\mathsf{T}}(\boldsymbol{\beta}_v^* - \hat{\boldsymbol{\beta}}_{v,t}) + (\boldsymbol{\theta}_u^* - \hat{\boldsymbol{\theta}}_{u,t})^{\mathsf{T}} \hat{\boldsymbol{\beta}}_{v,t}$$

$$\leq \|\boldsymbol{\theta}_u^* - \hat{\boldsymbol{\theta}}_{u,t}\|\|\boldsymbol{\beta}_v^* - \hat{\boldsymbol{\beta}}_{v,t}\| + \|\hat{\boldsymbol{\theta}}_{u,t}\|\|\boldsymbol{\beta}_v^* - \hat{\boldsymbol{\beta}}_{v,t}\| + \|\boldsymbol{\theta}_u^* - \hat{\boldsymbol{\theta}}_{u,t}\|\|\hat{\boldsymbol{\beta}}_{v,t}\|$$

$$\leq \|\boldsymbol{\theta}_u^* - \hat{\boldsymbol{\theta}}_{u,0}\|\|\boldsymbol{\beta}_v^* - \hat{\boldsymbol{\beta}}_{v,0}\|q^{2t}$$

$$+ \|\boldsymbol{\beta}_v^* - \hat{\boldsymbol{\beta}}_{v,t}\|_{C_{u,t}} \|\hat{\boldsymbol{\theta}}_{u,t}\|_{C_{u,t}^{-1}} + \|\boldsymbol{\theta}_u^* - \hat{\boldsymbol{\theta}}_{u,t}\|_{A_{u,t}} \|\hat{\boldsymbol{\beta}}_{v,t}\|_{A_{u,t}^{-1}}$$

$$\leq 2q^{2t} + \alpha_u^{\theta} \|\hat{\boldsymbol{\theta}}_{u,t}\|_{C_{u,t}^{-1}} + \alpha_v^{\beta} \|\hat{\boldsymbol{\beta}}_{v,t}\|_{A_{u,t}^{-1}}$$

in which q is a constant in the range of $(0, 1)$, and the second inequality is based on the q-linear convergence rate of parameter estimation [31, 34]. $\square$