Online Influence Maximization with Node-level Feedback Using Standard Offline Oracles

Zhijie Zhang, [1,2] **Wei Chen,** [3] **Xiaoming Sun,** [1,2] **Jialin Zhang** [1,2]
[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
[3] Microsoft Research Asia, Beijing, China
{zhangzhijie,sunxiaoming,zhangjialin}@ict.ac.cn, weic@microsoft.com

Abstract

We study the online influence maximization (OIM) problem in social networks, where in multiple rounds the learner repeatedly chooses seed nodes to generate cascades, observes the cascade feedback, and gradually learns the best seeds that generate the largest cascade. We focus on two major challenges in this paper. First, we work with *node-level feedback* instead of *edge-level feedback*. The edge-level feedback reveals all edges that pass through information in a cascade, where the node-level feedback only reveals the activated nodes with timestamps. The node-level feedback is arguably more realistic since in practice it is relatively easy to observe who is influenced but very difficult to observe from which relationship (edge) the influence comes from. Second, we use *standard offline oracle* instead of *offline pair-oracle*. To compute a good seed set for the next round, an offline pair-oracle finds the best seed set and the best parameters within the confidence region simultaneously, and such an oracle is difficult to compute due to the combinatorial core of OIM problem. So we focus on how to use the standard offline influence maximization oracle which finds the best seed set given the edge parameters as input. In this paper, we resolve these challenges for the two most popular diffusion models, the independent cascade (IC) and the linear threshold (LT) model. For the IC model, the past research only achieves edge-level feedback, while we present the first $\widetilde{O}(\sqrt{T})$-regret algorithm for the node-level feedback. For the first challenge above, we apply a novel adaptation of the maximum likelihood estimation (MLE) approach to learn the graph parameters and its confidence region (a confidence ellipsoid). For the second challenge, we adjust the update procedure to dissect the confidence ellipsoid into confidence intervals on each parameter, so that the standard offline influence maximization oracle is enough. For the LT model, a recent study only provides an OIM solution that meets the first challenge but still requires a pair-oracle. In this paper, we apply a similar technique as in the IC model to replace the pair-oracle with a standard oracle while maintaining $\widetilde{O}(\sqrt{T})$-regret.

Introduction

Social networks have gained great attention in the past decades as a model for describing relationships between humans. Typically, researchers show great interest in how information, ideas, news, influence, etc spread over social networks, starting from a small set of nodes called *seeds*. To this end, a variety of diffusion models are proposed to formulate the propagation in reality, and the most well-known ones are the *independent cascade* (IC) model and the *linear threshold* (LT) model (Kempe, Kleinberg, and Tardos 2003). A corresponding optimization problem, known as *influence maximization* (IM), asks how to maximize the influence spread, under a specific diffusion model, by selecting a limited number of "good" seeds. The problem has found enormous applications, including advertising, viral marketing, news transmission, etc.

In the canonical setting, the IM problem takes as input a social network, which is formulated as an edge-weighted directed graph. The problem is NP-hard but can be well-approximated (Kempe, Kleinberg, and Tardos 2003). For the past decade, more efficient and effective algorithms have been designed (Borgs et al. 2014; Tang, Xiao, and Shi 2014; Tang, Shi, and Xiao 2015), leading to an almost complete resolution of the problem. However, the canonical IM is sometimes difficult to apply in practice, as edge parameters of the network are often *unknown* in many scenarios. A possible way to circumvent such difficulty is to learn the edge parameters from past observed diffusion cascades, and then maximize the influence based on the learned parameters. The learning task is referred to as *network inference*, and has been extensively studied in the literature (Gomez-Rodriguez, Leskovec, and Krause 2010; Myers and Leskovec 2010; Gomez-Rodriguez, Balduzzi, and Schölkopf 2011; Du et al. 2012; Netrapalli and Sanghavi 2012; Abrahao et al. 2013; Daneshmand et al. 2014; Du et al. 2013, 2014; Narasimhan, Parkes, and Singer 2015; Pouget-Abadie and Horel 2015; He et al. 2016; Chen et al. 2021). However, this approach does not take into account the cost of the learning process and fails to balance between exploration and exploitation when future diffusion cascades come. This motivates the study of *online influence maximization (OIM)* problem considered in this paper.

In OIM, the learner faces an unknown social network and runs T rounds in total. At each round, the learner chooses a seed set to generate cascades, observes the cascade feedback, and receives the influence value as a reward. The goal is to maximize the influence values received over T rounds, or equivalently, to minimize the cumulative regret compared with the optimal seed set that generates the largest influence. The most widely studied feedback in the literature is the *edge-level* feedback (Chen, Wang, and Yuan 2013; Chen et al. 2016; Wang and Chen 2017; Wen et al. 2017; Wu et al. 2019), where the learner can observe whether an edge passes through the information received by its start point. The *node-level* feedback was only investigated very recently in (Vaswani, Lakshmanan, and Schmidt 2016; Li et al. 2020), where the learner can only observe which nodes receive the information at each time step during a diffusion process. In practice, the node-level feedback is more realistic than the edge-level feedback, not only because it reveals less information, but also because it is usually easy to observe who is influenced but very difficult to observe from which edge the influence comes from. For example, the social network platform is easy to learn whether and when a user buys some specific product or service but is difficult to learn based on whose recommendations or comments the user makes such a decision.

Besides the feedback type, one more challenge about the oracle model also emerges from the work (Li et al. 2020), and many other works in online learning. In (Li et al. 2020), the solution requires an offline pair-oracle, which takes the estimated edge parameters and their confidence regions as the input, and outputs the best seed set and parameters that maximize the social influence. However, though numerous works show that the offline influence maximization problem, which corresponds to the standard offline oracle, can be well approximated, to the best of our knowledge, we do not know how to efficiently compute such pair-oracle, even approximately. We may choose several parameters by either sampling the parameters or enumerating the parameters in a mesh grid, and compute the maximum influence value among such chosen parameters by the standard offline oracle. But such a method is quite time-consuming where the running time is typically exponential to the dimension of the confidence region, which is the number of parameters. Besides, we do not know how to guarantee the approximation ratio for such a pair-oracle. This difficulty in computation might be due to the complex combinatorial structure in the influence maximization problem. Therefore, in this paper, we focus on the weaker oracle model: the standard offline oracle which takes the edge-level parameters as the input and finds the best seed set accordingly.

Our contribution. We resolve the aforementioned challenges for both the IC and LT models. For the IC model, we present the first $\widetilde{O}(\mathrm{poly}(|G|)\sqrt{T})$-regret algorithm for the node-level feedback using standard offline oracles. For the LT model, based on the study (Li et al. 2020), we apply a similar technique as our results in the IC model to replace the pair-oracle with a standard oracle, so that we finally present an $\widetilde{O}(\mathrm{poly}(|G|)\sqrt{T})$-regret algorithm for the node-level feedback using a standard offline oracle.

In the technical part, our main contribution is a novel adaptation of the maximum likelihood estimation (MLE) approach which can learn the edge-level parameters and their confidence ellipsoids based on the node-level feedback. Further, we adjust the update procedure to dissect the confidence ellipsoid into confidence intervals on each parameter, so that we can apply a standard offline influence maximization oracle instead of the pair-oracle.

Related work. The (offline) influence maximization problem has received great attention in the past two decades. We refer interested readers to the surveys of (Chen, Lakshmanan, and Castillo 2013; Li et al. 2018) for an overall understanding.

The online influence maximization problem falls into the field of multi-armed bandits (MAB), a prosperous research area that dates back to 1933 (Thompson 1933). In the classical multi-armed *stochastic bandits* (Robbins 1952; Lai and Robbins 1985), there is a set of n arms, each of which is associated with a reward specified by some *unknown* distribution. At each round t, the learner chooses an arm and receives a reward sampled from the corresponding distribution. The goal is to maximize the total expected rewards received over T rounds. The model was later generalized to the multi-armed *stochastic linear bandits* (Auer, Cesa-Bianchi, and Fischer 2002), where each arm is associated with a characteristic vector and its reward is given by the inner product of the vector and an unknown parameter vector. This model was extensively studied in the literature (Dani, Hayes, and Kakade 2008; Li et al. 2010; Rusmevichientong and Tsitsiklis 2010; Abbasi-Yadkori, Pál, and Szepesvári 2011). Further generalizations include *combinatorial* multi-armed bandits (CMAB) and CMAB with probabilistically triggered arms (CMAB-T) (Chen, Wang, and Yuan 2013; Chen et al. 2016; Wang and Chen 2017), where a subset of arms, called the *super-arm*, can be chosen, and the reward is defined over super-arms and may be non-linear. Besides, the arms beyond the chosen super-arm may also be triggered and observed. CMAB-T is a quite general bandits framework and indeed contains OIM with edge-level feedback as a special case. However, OIM with node-level feedback does not fit into the CMAB-T framework.

OIM has been studied extensively in the literature. For edge-level feedback, existing work (Chen, Wang, and Yuan 2013; Lei et al. 2015; Chen et al. 2016; Wang and Chen 2017; Wen et al. 2017; Wu et al. 2019) present both theoretical and heuristic results. The node-level feedback was first proposed in (Vaswani, Lakshmanan, and Schmidt 2016). However, only heuristic algorithms were presented. Very recently, an $\widetilde{O}(\sqrt{T})$-regret algorithm was presented for the LT model with node-level feedback in (Li et al. 2020). We will compare the regret bounds obtained in this paper with the previous results in the main text.

Preliminaries

Notations

Given a vector $x \in \mathbb{R}^d$, its transpose is denoted by $x^\top$. The Euclidean norm of x is denoted by $\|x\|$. For a positive definite matrix $M \in \mathbb{R}^{d\times d}$, the weighted Euclidean norm of x is defined as $\|x\|_M = \sqrt{x^\top M x}$. The minimum eigenvalue of M is denoted by $\lambda_{\min}(M)$, and its determinant and trace are denoted by $\det[M]$ and $\mathrm{tr}[M]$, respectively. For a real-valued function $\mu : \mathbb{R} \to \mathbb{R}$, its first and second derivatives are denoted by $\dot{\mu}$ and $\ddot{\mu}$, respectively.

Social Network

A social network is a weighted directed graph $G = (V, E)$ with a node set V of $n = |V|$ nodes and an edge set E of $m = |E|$ edges. Each edge $e \in E$ is associated with a weight or probability $p(e) \in [0, 1]$. The edge probability vector is then denoted by $p = (p(e))_{e \in E}$, which describes the graph completely. For a node $v \in V$, let $N(v) = N^{in}(v)$ be the set of in-neighbors of v and $d_v = |N(v)|$ be its in-degree. The maximum in-degree of the graph is denoted by $D = \max_{v \in V} d_v$. In this paper, we use E_v to denote the set of incoming edges of v and $p_v = (p(e))_{e \in E_v} \in [0,1]^{d_v}$ to denote the probability vector corresponding to these edges. The e-th entry of p_v is denoted by $p_v(e)$. Thus, $p(e)$ and $p_v(e)$ refers to the same edge probability and we will use them interchangeably throughout the paper. For an edge $e = (u, v) \in E_v$, we use e_{uv} to explicitly indicate e's endpoints. Let $\chi(e_{uv}) \in \{0,1\}^{d_v}$ be the characteristic vector of e_{uv} over E_v such that all entries of $\chi(e_{uv})$ are 0 except that the entry corresponding to e_{uv} is 1. The characteristic vector of a subset $E' \subseteq E_v$ is then defined as $\chi(E') \coloneqq \sum_{e \in E'} \chi(e) \in \{0,1\}^{d_v}$. For simplicity, we define $x_e \coloneqq \chi(e)$.

Offline and Online Influence Maximization

In this subsection, we introduce the influence maximization (IM) problem in both the offline and online setting. The input of the offline problem is a social network, over which the information spreads. A node $v \in V$ is called *active* if it receives the information and *inactive* otherwise. We first describe the independent cascade (IC) and linear threshold (LT) diffusion models.

In the IC model, the diffusion proceeds in discrete time steps $\tau = 0, 1, 2, \cdots$. At the beginning of the diffusion ($\tau = 0$), there is an initially active set S_0 of nodes called *seeds*. For $\tau \geq 1$, the active node set S_τ after time τ is generated as follows. First, let $S_\tau = S_{\tau-1}$. Next, for each $v \in V \setminus S_{\tau-1}$, every node $u \in N(v) \cap (S_{\tau-1} \setminus S_{\tau-2})$ will try to *activate* v *independently* with probability $p(e_{uv})$ (let $S_{-1} = \emptyset$). Hence, v will be activated with probability $1 - \prod_{u \in N(v) \cap (S_{\tau-1} \setminus S_{\tau-2})} (1 - p(e_{uv}))$ and be added into S_τ once being activated. The diffusion terminates if $S_\tau = S_{\tau-1}$ for some τ and therefore it proceeds in at most n time steps. Let $(S_0, S_1, \cdots, S_{n-1})$ be the sequence of the active node sets during the diffusion process, where S_τ denotes the active node set after time τ. The *influence spread* of S_0 is defined as $\sigma(S_0) = \mathbf{E}[|S_{n-1}|]$, i.e. the expected number of active nodes when the diffusion starting from S_0 terminates. Here, $\sigma : 2^V \to \mathbb{R}_+$ is called the *influence spread function*. In this paper, we also use $\sigma(S, p)$ to state the edge probability vector p explicitly. The influence maximization problem takes as input the social network G and an integer $K \in \mathbb{N}_+$, and requires to find the seed set S^{opt} that gives the maximum influence spread with at most K seeds, i.e. $S^{\text{opt}} \in \operatorname{argmax}_{S \subseteq V, |S| \leq K} \sigma(S)$.

In the LT model, following the convention, we use w to denote the non-negative edge weight vector. It is assumed that $\sum_{e \in E_v} w(e) \leq 1$ for each node $v \in V$ in this model. Besides, each $v \in V$ is associated with a threshold r_v, which is sampled independently and uniformly from $[0, 1]$ before the diffusion starts. The diffusion also proceeds in discrete time step. At time τ, $S_\tau = S_{\tau-1}$ initially. For each node $v \in V \setminus S_{\tau-1}$, it will be added into S_τ if and only if $\chi(S_{\tau-1} \cap N(v))^\top w_v \geq r_v$. The diffusion also terminates if $S_\tau = S_{\tau-1}$ for some τ. Similarly, one can also define the influence spread function and IM problem under the LT model.

It is well-known that the IM problem admits a $(1 - 1/e - \epsilon)$ approximation under both IC and LT models (Kempe, Kleinberg, and Tardos 2003), which is tight for the IC model assuming P $\neq$ NP (Feige 1998). Let `ORACLE` be an (offline) oracle of the IM problem. Under IC model, let $\widetilde{S} = \texttt{ORACLE}(G, K, p)$ be its output and $S^{\text{opt}} \in \operatorname{argmax}_{S:|S| \leq K} \sigma(S, p)$ be the optimal seed set. For $\alpha, \beta \in [0, 1]$, we say `ORACLE` is an (α, β)-oracle if $\Pr[\sigma(\widetilde{S}, p) \geq \alpha \cdot \sigma(S^{\text{opt}}, p)] \geq \beta$, where the probability is taking from the possible randomness of the algorithm `ORACLE`. By replacing parameter p with parameter w, an (α, β)-oracle under LT model can be similarly defined. Existing work (Borgs et al. 2014; Tang, Xiao, and Shi 2014; Tang, Shi, and Xiao 2015) shows that there exists $(1 - 1/e - \epsilon, 1 - o(1))$-oracle of the IM problem under both IC and LT models.

In the online influence maximization problem (OIM) considered in this paper, there is an underlying social network $G = (V, E)$, whose edge parameter vector p^* or w^* is unknown initially. At each round t of total T rounds, the learner chooses a seed set S_t with cardinality at most K, observes the cascade feedback, and updates her knowledge about the parameter p^* or w^* for later selections. The feedback considered in this paper is node-level feedback, which means that the learner observes the realization of the sequence of active nodes $(S_{t,0}, S_{t,1}, \cdots, S_{t,n-1})$ after selecting $S_{t,0} = S_t$. Equipped with an (α, β)-oracle, the objective of OIM is to minimize the cumulative $(\alpha\beta)$-scaled regret over T rounds:

$$R(T) = \mathbf{E}\left[\sum_{t=1}^{T} R_t\right] = \mathbf{E}\left[T\alpha\beta \cdot \sigma(S^{\text{opt}}, p^*) - \sum_{t=1}^{T} |S_{t,n-1}|\right].$$

Due to the additivity of expectation, it is equal to

$$R(T) = \mathbf{E}\left[T\alpha\beta \cdot \sigma(S^{\text{opt}}, p^*) - \sum_{t=1}^{T} \sigma(S_t, p^*)\right].$$

OIM under the IC Model

In this section, we present an algorithm for OIM under the IC model with node-level feedback (Algorithm 1). Our algorithm adopts the canonical *upper confidence bound* (UCB) framework in the bandits problem. Under the UCB framework, at each round t, we first compute an estimate $\hat{p}_{t-1}$ of p^* and a corresponding confidence region (often in the shape of an ellipsoid) based on the feedback before round t. Then, a seed set S_t is selected by invoking the offline oracle with an appropriate edge probability vector within the

Algorithm 1: IC-UCB

Input: Graph $G = (V, E)$, seed set cardinality $K \in \mathbb{N}$, offline oracle `ORACLE`, parameter $\gamma \in (0, 1)$ in Assumption 1.

1: Initialize $M_{0,v} \leftarrow \mathbf{0} \in \mathbb{R}^{d_v \times d_v}$ for all $v \in V$, $\delta \leftarrow 1/(3n\sqrt{T})$, $R \leftarrow \left\lceil \frac{512D}{\gamma^4}\left(D^2 + \ln(1/\delta)\right)\right\rceil$, $T_0 \leftarrow nR$ and $\rho \leftarrow \frac{3}{\gamma}\sqrt{\ln(1/\delta)}$.
2: **for** each $u \in V$ **do**
3: Choose $\{u\}$ as the seed set for R rounds.
4: **end for**
5: **for** $t = T_0 + 1, T_0 + 2, \cdots, T$ **do**
6: $\{\hat{p}_{t-1,v}, M_{t-1,v}\}_{v \in V} =$ Estimate($(S_{k,0}, S_{k,1}, \cdots, S_{k,n-1})_{1 \le k \le t-1}$) (see Algorithm 2).
7: Construct $\tilde{p}_t$ such that $\tilde{p}_{t,v}(e) = \hat{p}_{t-1,v}(e) + \rho \cdot \|x_e\|_{M_{t-1,v}^{-1}}$ for each $e \in E_v$ and each $v \in V$.
8: Choose $S_t \in$ `ORACLE`$(G, K, \tilde{p}_t)$ and observe node-level feedback $(S_{t,0}, S_{t,1}, \cdots, S_{t,n-1})$.
9: **end for**

confidence ellipsoid. The difficulty of applying this framework lies in (1) how to obtain an estimate with a good confidence ellipsoid, and (2) how to dissect the confidence ellipsoid for the edge probability vector into confidence intervals for each edge probability, making the standard offline oracle applicable. We successfully resolve these two issues simultaneously by applying a novel *adaptation* of the classical maximum likelihood estimation approach fed by *carefully handled data* extracted from the observed feedback.

The key part of the algorithm is how to update the estimation of p^* when the algorithm collects a set of node-level feedback in all previous rounds (line 6). We first explain how to construct data pairs $\in \{0,1\}^{d_v} \times \{0,1\}$ to extract information about p_v^* from the feedback $(S_{t,0}, S_{t,1}, \cdots, S_{t,n-1})$ in some round t. On the one hand, for any node $u \in N(v)$, if u is activated in some time step τ while node v keeps inactive in time step $\tau + 1$, we know that the edge e_{uv} is not activated in this cascade process. Thus, we construct an data pair $(\chi(e_{uv}), 0)$ in this case. On the other hand, if the inactive node v is activated in time step τ, then all nodes activated in time step $\tau - 1$ is possible to activate node v in this cascade process. More formally, let $E' := \{e_{uv} \in E_v \mid u \in (S_{t,\tau-1} \setminus S_{t,\tau-2}) \cap N(v)\}$ be the set of incoming edges to v from these nodes. We then construct a data pair $(\chi(E'), 1)$ to indicate that one of the edges in E' passes through the information. Assume that in this way $J_{t,v}$ pairs are constructed for round t for node v in total. We denote them by $(X_{t,j,v}, Y_{t,j,v})$ for $1 \le j \le J_{t,v}$. Note that if v is not activated in this cascade process, no pair has the form $(\cdot, 1)$; while if v is activated in this cascade process, there exists exactly one pair of the form $(\cdot, 1)$ and we assume this is the last pair so that $Y_{t,J_{t,v},v} = 1$. For the initial regularization phase where $t \le T_0$, the process to extract information is slightly different where only the first step activation is taken into account. More formally, let node u be chosen as the seed in round t. In the case $u \in N(v)$, we have $J_{t,v} = 1$ and construct data pair $(\chi(e_{uv}), 1)$ if $v \in S_{t,1}$, or data pair $(\chi(e_{uv}), 0)$ if $v \notin S_{t,1}$. In the case $u \notin N(v)$, no data pair is constructed.

Algorithm 2: Estimate. Note that the code is written as a computation from scratch in each round to accommodate the initialization period of Algorithm 1, and it can be easily adapted to the incremental computation form.

Input: All observations $(S_{k,0}, S_{k,1}, \cdots, S_{k,n-1})_{1 \le k \le t}$ until round t.

1: Construct data pairs $(X_{k,j,v}, Y_{k,j,v})_{1 \le j \le J_{k,v}, 1 \le k \le t, v \in V}$ from observations $(S_{k,0}, S_{k,1}, \cdots, S_{k,n-1})_{1 \le k \le t}$ (see the text in this section for details).
2: $L_{t,v}(\theta_v) \leftarrow \sum_{k=1}^{t}\sum_{j=1}^{J_{k,v}}[-\exp(-X_{k,j,v}^\top \theta_v) - (1 - Y_{k,j,v})X_{k,j,v}^\top \theta_v]$.
3: $\hat{\theta}_{t,v} \leftarrow \mathrm{argmax}_{\theta_v} L_{t,v}(\theta_v)$.
4: $\hat{p}_{t,v}(e) \leftarrow 1 - \exp(-\hat{\theta}_{t,v}(e))$ for each $e \in E_v$
5: $M_{t,v} \leftarrow \sum_{k=1}^{t}\sum_{j=1}^{J_{k,v}} X_{k,j,v} X_{k,j,v}^\top$.

Algorithm 2 provides the estimate process (line 6 in Algorithm 1) in detail based on the data pairs $\{(X_{k,j,v}, Y_{k,j,v})\}_{1 \le k \le t}$. Before giving the formal analysis of the regret, we explain our intuitive in the algorithm design from the following four points.

Transformation of edge parameter p into parameter θ:

By the diffusion rule of the IC model, for each $v \in V$, given $X \in \{0,1\}^{d_v}$, let $Y \in \{0,1\}$ indicates whether v is activated in *one time step*. Then, $\mathbf{E}[Y \mid X] = 1 - \Pi_{e:X(e)=1}(1 - p(e))$ which is a complex relationship of parameter $p(e)$. We therefore consider a transformation of edge parameter p into a new parameter θ where

$$\theta(e) = -\ln(1 - p(e)) \text{ for each } e \in E. \tag{1}$$

Thus, we have $\mathbf{E}[Y \mid X] = \mu(X^\top \theta_v)$ where the *link function* $\mu : \mathbb{R} \to \mathbb{R}$ is defined as $\mu(x) := 1 - \exp(-x)$.

This indeed forms an instance of the *generalized linear bandit* problem studied in (Filippi et al. 2010; Li, Lu, and Zhou 2017), where the MLE approach was adopted and analyzed. Besides, any confidence region of parameter θ also implies the confidence region of parameter p due to the following lemma:

Lemma 1. *For any two vectors $\tilde{p}, p \in [0,1]^m$ and $\tilde{\theta}, \theta$ as defined in Eq.* (1)*, for each $e \in E$,*

$$|\tilde{p}(e) - p(e)| \le |\tilde{\theta}(e) - \theta(e)|.$$

Pseudo log-likelihood function $L_{t,v}$:

In work (Filippi et al. 2010; Li, Lu, and Zhou 2017), a standard log-likelihood function $\mathcal{L}_{t,v}^{std}(\theta_v) := \sum_{k=1}^{t}\sum_{j=1}^{J_{k,v}}[Y_{k,j,v}\ln\mu(X_{k,j,v}^\top \theta_v) + (1 - Y_{k,j,v})\ln(1 - \mu(X_{k,j,v}^\top \theta_v))$ is used in the update process. However, the analysis in their work requires that the gradient of the log-likelihood function has the form $\sum_{k=1}^{t}\sum_{j=1}^{J_{k,v}}[Y_{k,j,v} - \mu(X_{k,j,v}^\top \theta_v)]X_{k,j,v}$. We remark that such requirement is met in (Filippi et al. 2010; Li, Lu, and Zhou 2017) by assuming the distribution of Y conditioned on X falls into some sub-class of the exponential family of distributions, which is not satisfied in our case. Here we present an alternative way to overcome such technical difficulty. The pseudo log-likelihood function $L_{t,v}$ defined in line 2 in Algorithm 2 is constructed by "integrating" the gradient so that we guarantee the gradient of $L_{t,v}$ has the specific form we need. In the regret analysis part, we show this pseudo log-likelihood function can successfully take the place of the standard log-likelihood function. Such an approach is of great independent interest and we leave it as an open problem to find a more intuitive explanation for it.

Special construction of data pairs $\{(X_{t,j,v}, Y_{t,j,v})\}$:

In the construction of data pairs, we treat pair type $(\cdot, 0)$ and $(\cdot, 1)$ differently. Suppose at some time step, a set of nodes V' fails to activate target node v. Let $E' = \{e_{uv} \mid u \in V' \cap N(v)\}$. Instead of constructing data pair $(\chi(E'), 0)$ like the way we treat for the activated case, we construct data pairs $(\chi(e), 0)$ for all edges $e \in E'$. This does not make any difference in the pseudo log-likelihood function $L_{t,v}$, but will make the update of $M_{t,v}$ different. Intuitively, our choice of data pairs reveals more information of the diffusion process; while technically, such choice makes $M_{t,v}$ more similar to the diagonal matrix and enables us to upper bound $\|x\|_{M_{t,v}^{-1}}$ appears in Theorem 1 in the analysis.

Initial regularization step (line 2- 4 in Algorithm 1):

In this part, the algorithm chooses each node $u \in V$ as the seed set for R rounds, and then observes the activation of u's all out-neighbors in order to gather information about its outgoing edges. By the regularization step, each edge will be observed exactly R times. Intuitively, this step leads to a coarse estimate of each individual probability $p(e)$ for $e \in E$ before MLE starts. Technically, this step guarantees a lower bound of the minimum eigenvalue of the Gram matrix $M_{t,v}$, which ensures the correctness of condition (2) in Theorem 1 in the analysis.

Regret Analysis

We now give an analysis of the regret of Algorithm 1. First, we need to show that for each $v \in V$, the estimate $\hat{\theta}_{t,v}$ is close to the true parameter θ_v^*. To ensure this, we require Assumption 1 below. Similar or even stronger assumptions are adopted in all previous approaches for network inference (Netrapalli and Sanghavi 2012; Narasimhan, Parkes, and Singer 2015; Pouget-Abadie and Horel 2015; Chen et al. 2021). Assumption 1 means that node $v \in V$ will remain inactive with probability at least γ even if all of its in-neighbors are simultaneously activated. It reflects the stubbornness of the agent (node). That is, the behavior of a node is partially determined by its intrinsic motivation, not by its neighbors. So, even when all its neighbors adopt a new behavior, there is a nontrivial probability that the node will still not adopt the new behavior. Intuitively, this allows us to observe the state of each incoming edge individually.

Assumption 1. *There exists a parameter $\gamma \in (0, 1)$ such that $\prod_{u \in N(v)} (1 - p^*(e_{uv})) \geq \gamma$ for all $v \in V$.*

Under Assumption 1, it is possible to show that $\hat{\theta}_{t,v}$ and θ_v^* are close to each other in all directions, as Theorem 1 stated. Most of its proof follows directly from Theorem 1 in (Li, Lu, and Zhou 2017). For completeness, we include the proof in the appendix.

Theorem 1. *Suppose that Assumption 1 holds. For each $v \in V$, $\hat{\theta}_{t,v}$ and $M_{t,v}$ are computed according to Algorithm 2. Given $\delta \in (0, 1)$, if*

$$\lambda_{\min}(M_{t,v}) \geq \frac{512 d_v}{\gamma^4} \left(d_v^2 + \ln \frac{1}{\delta} \right). \tag{2}$$

Then, with probability at least $1 - 3\delta$, for any $x \in \mathbb{R}^{d_v}$, we have

$$|x^\top (\hat{\theta}_{t,v} - \theta_v^*)| \leq \frac{3}{\gamma} \sqrt{\ln(1/\delta)} \cdot \|x\|_{M_{t,v}^{-1}}.$$

For each $e \in E_v$, by plugging $x = x_e$ into Theorem 1 and by Lemma 1, we obtain a confidence interval for each individual probability parameter $p^*(e)$, which is upper bounded by $\|x_e\|_{M_{t,v}^{-1}}$ times a factor.

The key difficulty of applying Theorem 1 lies in how to get an upper bound for $\|x_e\|_{M_{t,v}^{-1}}$. To gain some intuitions, let us first consider an ideal case where at each round $k < t$ and each time step τ before v becomes active, there is at most one newly active in-neighbor u of v. We can therefore observe whether v is activated by u through edge e_{uv} in the next time step. In this case, every $X_{k,j,v}$ equals to $x_{e'}$ for some edge $e' \in E_v$, and $M_{t,v}$ is a diagonal matrix where the e-th diagonal entry $M_{t,v}(e, e)$ records the number of times e is observed. Therefore,

$$\|x_e\|_{M_{t,v}^{-1}} = \sqrt{M_{t,v}^{-1}(e, e)} = 1 \Big/ \sqrt{M_{t,v}(e, e)}.$$

This bound coincides with the $\sqrt{1/N}$-accuracy of estimating a biased coin by tossing it N times. In general, however, $M_{t,v}$ is not a diagonal matrix, and it is very difficult to compute $\|x_e\|_{M_{t,v}^{-1}}$ from $M_{t,v}$. Luckily, the construction of data pairs $(X_{k,j,v}, Y_{k,j,v})$ makes $M_{t,v}$ as close as possible to some diagonal matrix. Specifically, define

$$\widetilde{M}_{t,v} := \sum_{k=1}^{t} \sum_{j: Y_{k,j,v} = 0} X_{k,j,v} X_{k,j,v}^\top.$$

Then, $\widetilde{M}_{t,v}$ is a diagonal matrix since in each round k, all data pairs with $Y_{k,j,v} = 0$ have the form $(\chi(e'), \cdot)$ for some edge e'. Besides, we know $\|x_e\|_{M_{t,v}^{-1}} \leq \|x_e\|_{\widetilde{M}_{t,v}^{-1}}$ since $M_{t,v} - \widetilde{M}_{t,v}$ is a positive semidefinite matrix. Thus, we only need to give an upper bound for $\|x_e\|_{\widetilde{M}_{t,v}^{-1}}$. Let $e = e_{uv}$. Consider the case when at time step τ, node u becomes active while node v keeps inactive, due to Assumption 1, node v will remains inactive in time step $\tau + 1$ with probability at least γ. Therefore, $\widetilde{M}_{t,v}(e, e)$ is at least $\gamma M_{t,v}(e, e)$ in expectation and we have

$$\|x_e\|_{M_{t,v}^{-1}} \leq 1 \Big/ \sqrt{\gamma M_{t,v}(e, e)}.$$

This bound still coincides with the previous bound, up to a $\sqrt{1/\gamma}$ factor. The formal statements about these bounds are presented in the appendix.

The last ingredient in our regret analysis is a *group observation modulated* (GOM) bounded smoothness condition for the IC model. The condition is inspired by the GOM condition for the LT model (Li et al. 2020), which is used to handle node-level feedback. We remark that for edge-level feedback, there is a related *triggering probability modulated* (TPM) bounded smoothness condition (Wang and Chen 2017; Wen et al. 2017). However, the TPM condition does not suffice for node-level feedback.

We now state the GOM condition formally. Given a seed set $S \subseteq V$ and a node $v \in V \setminus S$, we say node $u \in V \setminus S$ is *relevant* to node v if there is a path P from S to v such that $u \in P$. Let $V[S, v] \subseteq V$ be the set of nodes relevant to v given seed set S. Given diffusion cascade $(S_0 = S, S_1, \cdots, S_{n-1})$, let E_v^o be the set of incoming edges of v that have a chance to activate v before v becomes active, i.e., $E_v^o = \{e = (u, v) \in E_v \mid u \in S_{\tau-1} \text{ if } v \in S_\tau \setminus S_{\tau-1}; \text{ or } u \in S_{n-1} \text{ if } v \notin S_{n-1}\}$. Note that E_v^o is exactly the set of edges used for constructing the data pairs in the estimate procedure (Algorithm 2), and this links the algorithm with the following important condition. We have the following GOM condition for the IC model, whose proof is presented in the appendix.

Lemma 2 (GOM bounded smoothness for the IC model)**.** *For any seed set $S \subseteq V$, and any two edge-probability vector $\tilde{p}$, $p^* \in [0,1]^{|E|}$ such that $\tilde{p}(e) \geq p^*(e)$ for each $e \in E$, it satisfies that*

$$\sigma(S, \tilde{p}) - \sigma(S, p^*) \leq \sum_{v \in V \setminus S} \sum_{u \in V[S,v]} \mathbf{E}\left[\sum_{e \in E_u^o} (\tilde{p}(e) - p^*(e))\right],$$

where the expectation is taken over the randomness of the diffusion cascade $(S_0, S_1, \cdots, S_{n-1})$, which is generated with respect to parameter p^*.

Given a seed set $S \subseteq V$ and a node $u \in V \setminus S$, define $n_{S,u} := \sum_{v \in V \setminus S} \mathbf{1}\{u \in V[S, v]\}$ to be the number of nodes that u is relevant to. Further, define

$$\zeta(G) := \max_{S:|S| \leq K} \sqrt{\sum_{u \in V} n_{S,u}^2} \leq O(n^{3/2}).$$

We show the following regret of Algorithm 1, which is presented in Theorem 2. Its proof is presented in the appendix.

Theorem 2. *When we use a standard (α, β)-oracle in Algorithm 1, under Assumption 1, the $\alpha\beta$-scaled regret of Algorithm 1 satisfies that*

$$R(T) = \widetilde{O}\left(\frac{\zeta(G)\sqrt{mDT}}{\gamma^2}\right). = \widetilde{O}(n^3\sqrt{T}/\gamma^2).$$

We remark that the worst-case regret for the IC model with edge-level feedback is $\widetilde{O}(n^4\sqrt{T})$ in (Wen et al. 2017) and $\widetilde{O}(n^3\sqrt{T})$ in (Wang and Chen 2017). Thus, our regret bound under node-level feedback matches the previous ones under edge-level feedback in the worst case, up to a $1/\gamma^2$ factor.

Algorithm 3: LT-UCB

Input: Graph $G = (V, E)$, seed set cardinality K, offline oracle `ORACLE`.

Initialize $M_{0,v} \leftarrow I \in \mathbb{R}^{|N(v)| \times |N(v)|}$, $b_{0,v} \leftarrow \mathbf{0} \in \mathbb{R}^{|N(v)|}$, $\delta \leftarrow 1/(n\sqrt{T})$, and $\rho_{t,v} \leftarrow \lceil \sqrt{d_v \ln(1+t) + 2\ln(1/\delta)} \rceil$ for each $v \in V$.
for $t = 1, 2, \cdots, T$ **do**
  for each $v \in V$ **do**
    Let $\hat{w}_{t-1,v} \leftarrow M_{t-1,v}^{-1} b_{t,v}$.
    Construct $\tilde{w}_{t,v}$ such that $\tilde{w}_{t,v}(e) = \hat{w}_{t-1,v}(e) + \rho_{t-1,v} \cdot \|x_e\|_{M_{t-1,v}^{-1}}$ for each $e \in E_v$.
  end for
  Choose $S_t \in$ `ORACLE`$(G, K, \tilde{w}_{t-1})$ and observe node-level feedback $(S_{t,0}, S_{t,1}, \cdots, S_{t,n-1})$.
  for each $v \in V$ **do**
    Pick τ randomly from $\{\tau'_{t,v}, \tau'_{t,v}+1, \cdots, \tau_{t,v}-1\}$ and let $E' := \{e_{uv} \in E_v \mid u \in S_{t,\tau} \cap N(v)\}$.
    if v is activated in time $\tau + 1$ **then**
      $M_{t,v} \leftarrow M_{t-1,v} + \chi(E')\chi(E')^\top$ and $b_{t,v} \leftarrow b_{t-1,v} + \chi(E')$.
    else
      $M_{t,v} \leftarrow M_{t-1,v} + \sum_{e \in E'} x_e x_e^\top$ and $b_{t,v} \leftarrow b_{t-1,v}$.
    end if
  end for
end for

To get an intuition about γ's value, assume that each edge probability $\leq 1 - c$ for some constant $c \in (0, 1)$. Then, $\gamma = O(c^D)$, where D is the maximum in-degree of the graph. Thus, in the worst case, $1/\gamma$ is exponential in n. But when $D = O(\log n)$, $1/\gamma$ is polynomial in n and so is the regret bound. We think $D = O(\log n)$ is reasonable in practice, since a person only has a limited attention and cannot pay attention to too many people in the network. In practical applications, though we often don't know the exact value of γ, we can obtain a rough estimate by checking the maximum in-degree of the network.

OIM under the LT Model

In this section, we present an OIM algorithm with node-level feedback under the LT model using standard offline oracles. The pseudo-code is presented as Algorithm 3. This algorithm also adopts the UCB framework and due to the linearity of LT model, linear regression is used to estimate $\hat{w}_{t-1,v}$ of w_v^* at round t for each node $v \in V$ separately. The key difficulty lies in how to replace the pair-oracle by a standard influence maximization oracle, which left unresolved in (Li et al. 2020). To this end, one needs to compute a confidence interval for each edge parameter instead of a general confidence ellipsoid for the edge parameter vector.

We now describe Algorithm 3 in detail. First, observe that at time step τ of round t, for each inactive node $v \in V$, the active set $(S_{t,\tau-1}) \cap N(v)$ by the end of time $\tau - 1$ will try to activate v and succeeds with probability $\chi(E')^\top w_v^*$ by the diffusion rule of LT model, where $E' := \{e_{uv} \in E_v \mid u \in$

$S_{t,\tau-1} \cap N(v)\}$. We can therefore extract information about w_v^* by constructing corresponding data pairs. Specifically, if v is actually activated at time τ, the data pair $(\chi(E'), 1)$ is constructed. Otherwise, the data pair $(\chi(E'), 0)$ will be constructed. This is exactly what is done in (Li et al. 2020). However, we will modify the construction in a way similar to the IC model. Specifically, if v is actually activated in time τ, the data pair $(\chi(E'), 1)$ remains unchanged. But if v is not activated, we know that the weight of each edge $e \in E'$ is below the threshold r_v. We therefore construct data pairs $(\chi(e), 0)$ for each $e \in E'$ instead of the single data pair $(\chi(E'), 0)$.

Fix the diffusion cascade $(S_{t,0}, S_{t,1}, \cdots, S_{t,n-1})$ at round t. For each node v, define $\tau'_{t,v}$ as the earliest time when v has active in-neighbors:

$$\tau'_{t,v} := \min_{\tau}\{\tau \mid S_{t,\tau} \cap N(v) \neq \emptyset\}.$$

Particularly, $\tau'_{t,v} = n$ if v has no active in-neighbors throughout the diffusion. Define $\tau_{t,v}$ as the first time when v becomes active:

$$\tau_{t,v} := \min_{\tau}\{\tau \mid v \in S_{t,\tau} \text{ and } v \notin S_{t,\tau'} \text{ for } \tau' < \tau\}.$$

Particularly, $\tau_{t,v} = n$ if v remains inactive throughout the diffusion. Then, in each time τ with $\tau'_{t,v} \leq \tau < \tau_{t,v}$, one can construct data pairs described above in principle. However, for dependency issue, we only pick one of the time step randomly and use the data pairs constructed in that step.

Regret Analysis

We now give an analysis of Algorithm 3. First, it can be shown that the linear regression estimator $\hat{w}_{t,v}$ is close to the true parameter w_v^* in every direction, as Lemma 3 states. Its proof follows directly from the self-normalized bound and the proof of Theorem 2 in (Abbasi-Yadkori, Pál, and Szepesvári 2011).

Lemma 3. *For each node $v \in V$, let $\rho_{t,v} = \lceil \sqrt{d_v \ln(1+t) + 2\ln(1/\delta)} \rceil$. For any $\delta > 0$, with probability at least $1-\delta$, for all $t \geq 0$ and any vector $x \in \mathbb{R}^d$,*

$$|x^\top(\hat{w}_{t,v} - w_v^*)| \leq \rho_{t,v} \cdot \|x\|_{M_{t,v}^{-1}}.$$

To obtain a reasonable upper bound on $\|x\|_{M_{t,v}^{-1}}$, we needs Assumption 2 below, which means that v will remains inactive even if all of its in-neighbors are active. The assumption only slightly strengthens the standard $\sum_{u \in N(v)} w^*(e_{uv}) \leq 1$, and hence still covers most instances in practice.

Assumption 2. *There exists a parameter $\gamma \in (0,1)$ such that $\sum_{u \in N(v)} w^*(e_{uv}) \leq 1 - \gamma$ for each $v \in V$.*

Under Assumption 2, as in the IC model, our construction of data pairs allows us to give an explicit and good enough upper bound on $\|x\|_{M_{t,v}^{-1}}$. The formal statement about this is presented in the appendix.

Next, we need the *group observation modulated* (GOM) bounded smoothness property for the LT model developed in (Li et al. 2020), which is conceptually analogous to the TPM bounded smoothness for the IC model (Wang and Chen 2017; Wen et al. 2017) and inspires the GOM condition for the IC model in Lemma 2. Recall that $V[S,v] \subseteq V$ is defined as the set of nodes relevant to v given seed set S. The GOM condition is stated as Lemma 4 below.

Lemma 4 (GOM bounded smoothness for the LT model, (Li et al. 2020))**.** *For any weight vectors $\tilde{w}, w^* \in [0,1]^m$ with $\sum_{u \in N(v)} w^*(e_{uv}) \leq 1$ for each $v \in V$, and any seed set S, it satisfies that*

$$\begin{aligned} &|\sigma(S,\tilde{w}) - \sigma(S,w^*)| \\ &\leq \sum_{v \in V \setminus S} \sum_{u \in V[S,v]} \mathbf{E}\left[\sum_{\tau=\tau'_{t,u}}^{\tau_{t,u}-1} \left| \sum_{e \in E_{\tau,u}} (\tilde{w}(e) - w^*(e)) \right|\right], \end{aligned}$$

where the expectation is taken over the randomness of the diffusion $(S_0, S_1, \cdots, S_{n-1})$, which is generated with respect to w^*. And $E_{\tau,v} := \{e_{uv} \in E_v \mid u \in S_\tau \cap N(v)\}$.

Equipped with the above two lemmas and Assumption 2, we are able to provide the regret of Algorithm 3 in Theorem 3, whose proof is presented in the appendix.

Theorem 3. *When we use a standard (α,β)-oracle in Algorithm 1, under Assumption 2, the $\alpha\beta$-regret of Algorithm 3 satisfies that*

$$R(T) = \widetilde{O}\left(\zeta(G) n D \sqrt{mT}/\gamma\right) = \widetilde{O}\left(n^{9/2}\sqrt{T}/\gamma\right).$$

We remark that the regret for the LT model with node-level feedback in (Li et al. 2020) is $\widetilde{O}(\zeta(G)n^2\sqrt{mT}) = \widetilde{O}(n^{9/2}\sqrt{T})$. Our result matches the result in (Li et al. 2020), up to a $1/\gamma$ factor, which is the cost of using a standard offline oracle instead of a pair oracle. In practice, it is reasonable to assume that γ is inverse polynomial in n. In this case, the regret bound is still polynomial in n.

Conclusion

In this paper, we investigate the OIM problem with node-level feedback. We presents $\widetilde{O}(\sqrt{T})$-regret OIM algorithms for both the IC and LT models. For the IC model, our algorithm is the first one with node-level feedback that almost matches the optimal regret bound. For both models, our algorithms use standard offline oracles instead of the unrealistic pair oracle.

Our novel adaptation of MLE to fit the generalized linear bandits (GLB) model is of great independent interest, which might be combined with the GLB model to handle rewards generated from a broader classes of distributions. Our technique for dissect confidence ellipsoids into confidence intervals may also be used in other learning problem to gain more accurate estimation.

There still remain some open problems on the node-level feedback setting. An immediate one is to either remove our assumptions for edge weights, or remove the assumption parameter from the regret bound, while still using standard offline oracles. Besides, it is interesting to develop a general bandit framework which includes OIM with node-level feedback as a special case, just like CMAB-T containing OIM with edge-level feedback.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China Grants No. 61832003, 61872334, the Strategic Priority Research Program of Chinese Academy of Sciences under Grant No. XDA27000000.

References

Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved Algorithms for Linear Stochastic Bandits. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011.*, 2312–2320.

Abrahao, B. D.; Chierichetti, F.; Kleinberg, R.; and Panconesi, A. 2013. Trace complexity of network inference. In *the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013*, 491–499. ACM.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time Analysis of the Multiarmed Bandit Problem. *Mach. Learn.*, 47(2-3): 235–256.

Borgs, C.; Brautbar, M.; Chayes, J. T.; and Lucier, B. 2014. Maximizing Social Influence in Nearly Optimal Time. In Chekuri, C., ed., *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014*, 946–957. SIAM.

Chen, K.; Hu, I.; and Ying, Z. 1999. Strong consistency of maximum quasi-likelihood estimators in generalized linear models with fixed and adaptive designs. *The Annals of Statistics*, 27(4): 1155 – 1163.

Chen, W.; Lakshmanan, L. V. S.; and Castillo, C. 2013. *Information and Influence Propagation in Social Networks*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers.

Chen, W.; Sun, X.; Zhang, J.; and Zhang, Z. 2021. Network Inference and Influence Maximization from Samples. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021*, 1707–1716. PMLR.

Chen, W.; Wang, Y.; and Yuan, Y. 2013. Combinatorial Multi-Armed Bandit: General Framework and Applications. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*, 151–159. JMLR.

Chen, W.; Wang, Y.; Yuan, Y.; and Wang, Q. 2016. Combinatorial Multi-Armed Bandit and Its Extension to Probabilistically Triggered Arms. *J. Mach. Learn. Res.*, 17: 50:1–50:33.

Daneshmand, H.; Gomez-Rodriguez, M.; Song, L.; and Schölkopf, B. 2014. Estimating Diffusion Network Structures: Recovery Conditions, Sample Complexity & Soft-thresholding Algorithm. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2014*, 793–801. JMLR.

Dani, V.; Hayes, T. P.; and Kakade, S. M. 2008. Stochastic Linear Optimization under Bandit Feedback. In *the 21st Annual Conference on Learning Theory, COLT 2008*, 355–366. Omnipress.

Du, N.; Liang, Y.; Balcan, M.; and Song, L. 2014. Influence Function Learning in Information Diffusion Networks. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014*, 2016–2024. JMLR.

Du, N.; Song, L.; Gomez-Rodriguez, M.; and Zha, H. 2013. Scalable Influence Estimation in Continuous-Time Diffusion Networks. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, 3147–3155.

Du, N.; Song, L.; Smola, A. J.; and Yuan, M. 2012. Learning Networks of Heterogeneous Influence. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, 2789–2797.

Feige, U. 1998. A Threshold of ln n for Approximating Set Cover. *J. ACM*, 45(4): 634–652.

Filippi, S.; Cappé, O.; Garivier, A.; and Szepesvári, C. 2010. Parametric Bandits: The Generalized Linear Case. In *Advances in Neural Information Processing Systems 23 (NIPS 2010)*, 586–594. Curran Associates, Inc.

Gomez-Rodriguez, M.; Balduzzi, D.; and Schölkopf, B. 2011. Uncovering the Temporal Dynamics of Diffusion Networks. In *Proceedings of the 28th International Conference on Machine Learning, ICML 2011*, 561–568. Omnipress.

Gomez-Rodriguez, M.; Leskovec, J.; and Krause, A. 2010. Inferring networks of diffusion and influence. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2010*, 1019–1028. ACM.

He, X.; Xu, K.; Kempe, D.; and Liu, Y. 2016. Learning Influence Functions from Incomplete Observations. In *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, 2065–2073.

Kempe, D.; Kleinberg, J. M.; and Tardos, É. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2003*, 137–146. ACM.

Lai, T. L.; and Robbins, H. 1985. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1): 4–22.

Lei, S.; Maniu, S.; Mo, L.; Cheng, R.; and Senellart, P. 2015. Online Influence Maximization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2015*, 645–654. ACM.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web, WWW 2010*, 661–670. ACM.

Li, L.; Lu, Y.; and Zhou, D. 2017. Provably Optimal Algorithms for Generalized Linear Contextual Bandits. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, 2071–2080. PMLR.

Li, S.; Kong, F.; Tang, K.; Li, Q.; and Chen, W. 2020. Online Influence Maximization under Linear Threshold Model. In *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*.

Li, Y.; Fan, J.; Wang, Y.; and Tan, K.-L. 2018. Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 30(10): 1852–1872.

Myers, S. A.; and Leskovec, J. 2010. On the Convexity of Latent Social Network Inference. In *Advances in Neural Information Processing Systems 23 (NIPS 2010)*, 1741–1749. Curran Associates, Inc.

Narasimhan, H.; Parkes, D. C.; and Singer, Y. 2015. Learnability of Influence in Networks. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 3186–3194.

Netrapalli, P.; and Sanghavi, S. 2012. Learning the graph of epidemic cascades. In *ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2012*, 211–222. ACM.

Pollard, D. 1990. Empirical Processes: Theory and Applications. *NSF-CBMS Regional Conference Series in Probability and Statistics*, 2: i–86.

Pouget-Abadie, J.; and Horel, T. 2015. Inferring Graphs from Cascades: A Sparse Recovery Framework. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015*, 977–986. JMLR.

Robbins, H. 1952. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5): 527–535.

Rusmevichientong, P.; and Tsitsiklis, J. N. 2010. Linearly Parameterized Bandits. *Math. Oper. Res.*, 35(2): 395–411.

Tang, Y.; Shi, Y.; and Xiao, X. 2015. Influence Maximization in Near-Linear Time: A Martingale Approach. In *International Conference on Management of Data, SIGMOD 2014*, 1539–1554. ACM.

Tang, Y.; Xiao, X.; and Shi, Y. 2014. Influence maximization: near-optimal time complexity meets practical efficiency. In *International Conference on Management of Data, SIGMOD 2014*, 75–86. ACM.

Thompson, W. R. 1933. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4): 285–294.

Vaswani, S.; Lakshmanan, L. V. S.; and Schmidt, M. 2016. Influence Maximization with Bandits. arXiv:1503.00024.

Wang, Q.; and Chen, W. 2017. Improving Regret Bounds for Combinatorial Semi-Bandits with Probabilistically Triggered Arms and Its Applications. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 1161–1171.

Wen, Z.; Kveton, B.; Valko, M.; and Vaswani, S. 2017. Online Influence Maximization under Independent Cascade Model with Semi-Bandit Feedback. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 3022–3032.

Wu, Q.; Li, Z.; Wang, H.; Chen, W.; and Wang, H. 2019. Factorization Bandits for Online Influence Maximization. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019*, 636–646. ACM.