Contextual Bandits for Advertising Campaigns: A Diffusion-Model Independent Approach

Alexandra Iacob, Bogdan Cautis, Silviu Maniu

Contextual Bandits for Advertising Campaigns: A Diffusion-Model Independent Approach

Alexandra Iacob * Bogdan Cautis † Silviu Maniu ‡

Abstract

Motivated by scenarios of information diffusion and advertising in social media, we study an *influence maximization* problem in which little is assumed to be known about the diffusion network or about the model that determines how information may propagate. In such a highly uncertain environment, one can focus on *multi-round diffusion campaigns*, with the objective to maximize the number of distinct users that are influenced or activated, starting from a known base of few influential nodes. During a campaign, spread seeds are selected sequentially at consecutive rounds, and feedback is collected in the form of the activated nodes at each round. A round's impact (reward) is then quantified as the number of *newly activated nodes*. Overall, one must maximize the campaign's total spread, as the sum of rounds' rewards. In this setting, an explore-exploit approach could be used to learn the key underlying diffusion parameters, while running the campaign. We describe and compare two methods of *contextual multi-armed bandits*, with *upper-confidence bounds* on the remaining potential of influencers, one using a generalized linear model and the Good-Turing estimator for remaining potential (`GLM-GT-UCB`), and another one that directly adapts the LinUCB algorithm to our setting (`LogNorm-LinUCB`). We show that they outperform baseline methods using state-of-the-art ideas, on synthetic and real-world data, while at the same time exhibiting different and complementary behavior, depending on the scenarios in which they are deployed.

1 Introduction

Social media advertising is a booming domain, gradually replacing advertising over traditional channels. It is enabled by the highly effective word-of-mouth mechanisms that are embedded in social applications, such as likes, shares, reposts, or notifications. Social networking applications are therefore an unprecedented medium for advertising, be it with a commercial intent or not, as products, news, ideas, political manifests, etc., can propagate easily to a large yet well-targeted audience.

Motivated by advertising in social media, the class of algorithmic problems under the generic name of *influence maximization* (IM) [18] encompasses all scenarios that aim to maximize the spread of information in a diffusion network, by identifying the most influential nodes from which the diffusion of specific message should start. IM mirrors an increasingly used and highly effective form of marketing in social media, targeting a sub-population of *influential people*, instead of all users of interest, known as *influencer marketing* [6].

IM usually has as objective the *expected spread* under a stochastic diffusion model, which describes a diffusion as a probabilistic process. The seminal work of [18] introduced two such models – Linear Threshold (LT) and Independent Cascade (IC) – which have been adopted by most of the literature (see the survey of [24]). Such models rely on diffusion graphs with edges weighted by a spread probability.

As selecting the seed nodes maximizing the expected spread is NP-hard under such diffusion models, approximation algorithms that exploit the objective's monotonicity and sub-modularity have been studied extensively, yet scaling IM to realistic graphs remains difficult. While most of the IM literature focuses on improving efficiency and scalability (see benchmarks [2, 1]), other major obstacles have limited the practical impact of this research. First, it is hard to obtain meaningful influence probabilities, as it is hard and data-intensive to learn them from past diffusions. [16, 13, 11]. Also, the effectiveness of most IM algorithms depends on diffusion models and their key parameters – whether known or learned in online manner – aspects which are most often hard to align with real-life diffusion dynamics. It is commonly agreed that such parametric diffusion models represent elegant yet coarse interpretations of a reality that is complex and uncertain.

For these reasons, the focus of the IM literature has shifted recently towards *online* and *diffusion-model independent* methods [19, 28, 20] where, during a *multi-round influence campaign*, a learning agent sequentially selects at each round seeds from which a new diffusion

*iacob@lisn.fr, CNRS LISN, U. Paris-Saclay
†cautis@lisn.fr, CNRS LISN & IPAL Singapore; U. Paris-Saclay
‡maniu@lisn.fr, CNRS LISN, U. Paris-Saclay

of the campaign's message is initiated and observed in the network. A round's feedback is then used to update the agent's knowledge. To balance between exploration (of uncertain aspects of the diffusion environment) and exploitation (e.g., focusing on the most promising seeds), such methods rely on *multi-armed bandits*.

Our work follows in this path, as we study an IM problem in which the diffusion topology, the influence probabilities, and the model that determines how information may spread are all assumed to be unknown. Instead, what is known are the potential spread initiators, a set of few influential nodes called hereafter the *influencers*. In such a highly uncertain environment, under *budget limitations* (number of seedings and number of rounds), the campaign aims to maximize the number of *distinct* users that are influenced or activated, starting from the influencers. Seeds are selected sequentially (at each round) among the influencers, and an influencer may be re-seeded, i.e., selected at multiple rounds. After a round's diffusion, the assumed feedback are all the activated nodes from that round, i.e., only the diffusion's effects are observed (the *who*), but not their causes (the *why*). Generically, this feedback is used to refine the estimations for the influencers' remaining spread potential, which will guide future seeding decisions. Matching the overall objective, a round's *reward* is the number of *newly activated nodes*, i.e., those that were not already activated at previous rounds. The campaign's objective is to maximize the sum of rounds' rewards.

Our problem setting is directly inspired by influencer marketing scenarios where marketers have only access to a few influencers who can spread information, where the only feedback that can be realistically gathered are activations (e.g., who purchased or subscribed), and where the goal is to maximize the number of distinct activated users (instead of the number of activations).

We follow a *contextual multi-armed bandits* [27] approach, assuming that contextual information is known and exploitable in the sequential learning process, as features of influencers or of the information being diffused. The intuition is that within a campaign, whose overall goal is to get a specific "message" to as many users as possible, the ways in which that message may be formulated, presented, or diffused may vary from round to round, and such contextual variations will lead to different propagation dynamics. E.g., the campaign's message may be a political manifesto, while the to-be-diffused items may pertain to different aspects thereof, to connections with societal issues, may be framed in news, op-eds, data analysis, multimedia content, etc.

Contribution. We propose two UCB-like algorithms, `GLM-GT-UCB` and `LogNorm-LinUCB`, for the problem of selecting influencers in advertising campaigns, where newly activated nodes make up the reward. They follow *optimism in the face of uncertainty* in sequential learning [8], deriving an upper-confidence bound on the estimator of the remaining spread potential of each influencer. This enables to alternate in a principled way between explore and exploit steps when taking seeding decisions at the campaign's rounds. Our solutions are diffusion-model agnostic and follow different assumptions on the rewards distribution: Poisson for `GLM-GT-UCB`, log-normal for `LogNorm-LinUCB`. `GLM-GT-UCB` uses a Good-Turing estimator [14, 7] for new activations, to which it applies an external factor function modeling an influencer's fatigue (diminishing returns) and potential in a given context. The parameter of the external factor is assumed to be a linear combination of the context and an unknown feature vector learned through linear regression. `LogNorm-LinUCB` assumes a linear structure for the scale of rewards, estimated by the inner product of the context and the influencer's learned feature vector. We experimented with synthetic and real-world data, comparing with state-of-the-art solutions adapted to our problem. The experiments show that our methods successfully learn from the available side-information and achieve higher cumulative rewards. These results are complemented by theoretical *regret* guarantees for a `LogNorm-LinUCB` variant that learns from independent samples.

2 Main Related Work

The work of [23] proposed a solution for the generalized linear contextual bandit problem, earlier considered also in a practical scenario of news recommendation [22]. The solution is based on the work of [12] – considering non-linear rewards for the MAB problem – and it improves it by adapting the algorithm of [3] to use MLE for estimating the unknown parameters, and uses the same approach to create the independent samples.

In [29], the authors proposed an UCB-based algorithm, *IMLinUCB* for the online influence maximization problem in social networks. They assume that the diffusion of information follows the independent cascade (IC) edge semi-bandit model. The algorithm selects multiple influencers per round without suffering from an exponential increase in the combinatorial action space due to the cardinality of the source node set. Efficiency is obtained through the linear generalization of a probability weight function that yields the activation probabilities.

The cumulative regret bounds for *IMLinUCB* are topology dependent; this is also confirmed by the experiments performed on different types of graph topologies.

In [21], the authors consider the weariness of an influencer's effectiveness over time and introduce the so called *rotting bandits*. It assumes that the expected

reward decays as a function of the number of times an arm has been selected, thus the optimal policy being one of choosing different arms. Our problem bears similarities to the non-parametric rotting bandit problem of [21], as we also do not make assumptions about the structure of the reward, but only about its non-increasing nature in the number of selections. To this end, [21] proposed the Sliding-Window Average (*SWA*) algorithm. In the initialization phase, each arm is chosen for a fixed number of times, and for the rest of the "campaign" their empirical average reward is adjusted by a given quantity. SWA is thus able to detect early the significantly sub-optimal arms, while preserving theoretical guarantees.

The work that is most related to ours is [19]. Placed in a similar setting, it focuses on Online Influence Maximization with Persistence (OIMP). [19] has a similar objective formulation, and proposes an algorithm called GT-UCB (for Good-Turing Upper Confidence Bound). The approach is inspired by the work of [7], which used the Good-Turing estimator in a setting where a learning agent needs to sequentially select experts that only sample one of their potential elements at each round. Similar to rotting bandits, an adaptation of GT-UCB (called FAT-GT-UCB) is considered for scenarios where influencers may experience fatigue, i.e., a diminishing tendency to activate their user base as they are re-seeded during a campaign. The key aspect that distinguishes our study from the one of [19] is that *we assume contextual information is known and exploitable in the sequential learning process*, as features of the influencers or of the information being diffused. In doing so, we provide solutions that are no longer agnostic to the information being diffused nor to the profiles of influencers, as was the case in [19]. The contextual assumption leads to entirely different theoretical and algorithmic constructions, and is supported by our empirical evaluation. FAT-GT-UCB is one of our experimental baselines.

Finally, we stress that in our bandit approach the parameters to be estimated throughout a campaign must capture how good an influencer still is (its *remaining potential*). Hence a key difference with other multi-armed bandit studies for IM ([30, 29, 9, 28]) is that they look for a constant optimal seed set, while in our setting a round's best action (choice of seeds) depends on the number of previous rounds and on the context.

3 Problem Statement

We formalize the IM problem, set in a discrete-time campaign consisting of T rounds, with K influencers among which the algorithm chooses seeds at each round.

We model each influencer k as having access to A_k basic nodes, each one being influenced by k with a probability $p_{k,j}(t), \forall j \in [1, \dots, A_k]$. We assume that $p_{k,j}(t)$ depends on each basic node's inner probability $p(j)$ of activating itself, on some d-dimensional profile $\theta_{k,j}$, and on the round's context. In each round, a d-dimensional context $Y_t \in [0,1]^d$ is provided by the environment, in a similar manner to the contextual multi-armed bandit setting [27]. Considering that in our setting the reward is the number of *newly* activated nodes, we assume also the impact of the number of selections of the influencer up to round t, $n_k(t)$, on the probability $p_{k,j}(t)$. Therefore, the probability of a basic node j to be influenced by influencer k is well-approximated by a function $\alpha(\langle \theta_{k,j}, Y_t \rangle, n_k(t))$ applied as a modifier to the basic node's inner activation probability $p(j)$. The modifier α is a function of the relation between the influencer, the basic node, and the round's context. Formally, the problem we study in this paper is defined as follows:

Table 1: Summary of notations.

T	total number of rounds in a campaign
K	total number of available influencers
Y_t	the context in round t
I_t	the set of L influencers selected in round t
A_k	set of basic nodes reachable by influencer k
$S(I_t, Y_t)$	the spread given by the environment in round t
$p_{k,j}(t)$	the probability of influencer k to activate basic node j in round t
$\theta_{k,j}$	feature vector that explains the probability of influencer k to activate basic node j in round's context Y_t
$n_k(t)$	the history of number of selections of influencer k in round t
$p(j)$	the basic node's j intrinsic probability of activating itself
α	the external factor function which adjusts the basic node's activation probability; e.g. defined as in Equation 4.17.
F_t	the set of IDs of the activated basic nodes at the end of round t
r_t	the reward at the end of round t
$r'_k(t)$	the reward for the external factor's linear regression problem
$C_j(t)$	the cumulative Poisson count of activations for node j in round t
θ_k	the influencer k's feature vector
$\hat{\theta}_k(t)$	the estimator of the influencer k's feature vector in round t
λ_j	the Poisson intensity of activations for basic node j
λ_k	the Poisson intensity of activations due to influencer k
$R_k(t)$	the influencer k's remaining potential (i.e. the feasible reward) in round t
$G_k(t)$	Good-Turing estimator of the remaining potential for influencer k
$V_k(t)$	design matrix updated by the context vectors in rounds when influencer k is played
$s_k(t)$	the rewards history factor for linear regression
γ	the regularization factor for linear regression
$b_k(t)$	the UCB computed for influencer k in round t

PROBLEM 1. *[Contextual Influence Maximization] Given a set of influencers $[K] = 1, \dots, K$, a budget of N rounds (or trials), and a number $1 \leq L \leq K$ of influencers to be activated at each round, the objective is to solve the following optimization problem:*

$$\underset{I_t \subseteq [K], |I_t| = L, \forall 1 \leq t \leq N}{\operatorname{argmax}} \mathbb{E}\Big| \bigcup_{1 \leq t \leq N} S(I_t, Y_t)\Big|, \tag{3.1}$$

where $S(I_t, Y_t)$ is the spread of the chosen set of influencers for round t, and the probability that influencer k activates basic node j depends on the round's context Y_t and the number of k's selections $n_k(t)$:

$$p_{k,j}(t) = \alpha(\langle \theta_{k,j}, Y_t \rangle, n_k(t)) p(j). \tag{3.2}$$

A similar variant of this problem, which does not use contexts, was proven to be NP-hard in [19], and this hardness result immediately transfers to our problem (e.g., with a constant context for all rounds).

We now formulate the problem in a contextual bandit setting. We assume a semi-bandit feedback at the end of each round, denoted F_t, consisting of the set of IDs of the activated basic nodes. The *reward* for the round is the number of new activations:

$$r_t = \sum_{j=1}^{\bigcup_{k\in I_t} A_k} \mathbb{I}\{C_j(t) > 0\} - r_{t-1}; r_0 = 0, \tag{3.3}$$

where $C_j(t) = \sum_{s=1}^{t} \mathbb{I}\{j \in F_s\}$ denotes for each basic node the number of times it has been activated.

Given that the reward in each round is the number of *newly* activated basic nodes, Problem 1 exhibits a *diminishing returns property*: for each influencer, the expected number of new basic nodes it can activate decreases with each of its selections.

For each basic node j, its cumulative count of activations $C_j(t)$ up to round t is a random quantity depending on the node's probability $p_{k,j}(t)$ of being activated by the played influencer; these activation probabilities are assumed to be unknown. As estimating all user profiles $\theta_{k,j}$ is computationally expensive, our goal will be instead – given that the objective is to select the best influencer(s) at each round – to directly estimate the influencers' potential based on the context at each round, as proxy for the probabilities of individual nodes.

To achieve this, we propose two algorithms that both assume a generalization θ_k of the unknown parameters $\theta_{k,j}$, and two different assumptions on the distribution of new activations for each influencer. More precisely, we assume that activations follow either (i) a Poisson distribution, given that they are counts of nodes, or (ii) a log-normal distribution, assuming that the scales of the rewards are normally distributed (in line with observations on the distribution of real-world social phenomena [25]). In Section 4 we present the UCB-based solution that uses the Poisson distribution assumption, and in Section 5 we present the LinUCB-based solution that assumes a log-normal distribution.

4 GLM-GT-UCB Algorithm

The main idea behind the `GLM-GT-UCB` algorithm is to estimate the potential of each influencer, at each round, by some proxy measure. Here, by an influencer's *potential* we understand the number of nodes that it can *still* activate (i.e., the reward); more formally, each influencer's remaining potential of activating new basic nodes in round t is:

$$R_k(t) = A_k - \sum_{j=1}^{A_k} \mathbb{I}\{C_j(t-1) > 0\} \tag{4.4}$$

The stochasticity of $C_j(t-1)$ means that the remaining potential is a random variable too. While this has been analyzed in the non-contextual case [19], the challenge here is to account for the contextual dimension. The proxy we choose is the Good-Turing estimator [14], estimating the proportion of unseen items in a random process as the fraction of items seen only once (*hapaxes*).

There are two main technical challenges to modeling the remaining potential using Good-Turing estimators: (i) we are counting only *new* activations, so a *fatigue* factor needs to be added to the estimator, and (2) the contextual case forces us to make an assumption on the model – in our case, we opted for a generalized linear model using a Poisson distribution.

4.1 Good-Turing with Poisson and External Factor An influencer's remaining potential is an unknown random variable. The Good-Turing estimator [14], adjusted with a fatigue function, was shown to successfully model an influencer's fatigue [19]. The fatigue function, non-increasing in the number of influencer's selections, does not explicitly model an influencer's potential w.r.t. the diffused content. We thus propose a Good-Turing estimator adjusted by a function of the diffused content.

For each basic node j, its activation probability $p_{k,j}(t)$ is a function of (a) the linear combination of the node's feature vector $\theta_{k,j}$ and the round's context, and (b) the number of influencer's selections $n_k(s)$. The assumption we make is that the underlying distribution of each node's cumulative count of activations $C_j(t)$ is Poisson with intensities $\lambda_j \sum_{s=1}^{t} \sum_{k\in I_s} \alpha(\langle \theta_{k,j}, Y_s\rangle, n_k(s))$. Our approach is then to assume that the underlying distribution for the entire remaining potential of an influencer is Poisson with intensities $\alpha(\langle \theta_k, Y_t\rangle, n_k(t))\lambda_k, k \in \{1, \ldots, K\}$, where the individual user response probabilities are small: $\lambda_k \geq \sum_{j=1}^{A_k} \lambda_j \ll A_k$. Recall the true feature vector θ_k is initially unknown , so its estimation becomes a subproblem of our problem. The classical solution is to use the regularized least-squares estimator:

$$\hat{\theta}_k(t) = \underset{\theta\in\mathbb{R}^d}{\operatorname{argmin}} \sum_{s=1}^{t-1} (r_k^{'}(s) - \langle \theta, Y_t\rangle)^2 + \gamma \|\theta\|_2^2), \tag{4.5}$$

where $r_k^{'}(s)$ is the round's reward (adapted for the subproblem) and γ is the penalty factor that ensures the solution's uniqueness; more details are given in Sec. 4.2.

After t rounds, we observe the cumulative Poisson counts $C_j(t)$ of activations of each basic node $j \in \{1, \ldots, A_k\}$ by the corresponding influencer. The cumulative counts are distributed with rate

$$\lambda_j \sum_{s=1}^{t} \sum_{k\in I_s} \alpha(\langle \theta_{k,j}, Y_s\rangle, n_k(s)), \tag{4.6}$$

and in estimation with rate

$$\lambda_j \sum_{s=1}^{t} \sum_{k \in I_s} \alpha(\langle \theta_k, Y_s \rangle, n_k(s)). \tag{4.7}$$

Thus, the remaining potential can be expressed as the conditional expectation of cumulative counts of new basic nodes that would be influenced in round t:

$$R_k(t) = \sum_{j=1}^{A_k} \lambda_j \alpha(\langle \theta_k, Y_t \rangle, n_k(t)) \mathbb{I}\{C_j(t-1) = 0\} \tag{4.8}$$

The expectation of k's remaining potential in round t is

$$\mathbb{E}[R_k(t)] = \alpha(\langle \theta_k, Y_t \rangle, n_k(t)) \cdot \sum_{j=1}^{A_k} \lambda_j e^{-\lambda_j \sum_{s=1}^{t-1} \sum_{k' \in I_s} \alpha(\langle \theta_{k'}, Y_s \rangle, n_{k'}(s))} \tag{4.9}$$

`GLM-GT-UCB` estimates k's remaining potential by:

$$G_k(t) = \alpha(\langle \hat{\theta}_k(t), Y_t \rangle, n_k(t)) \frac{1}{n_k(t)} \sum_{j=1}^{A_k} \sum_{s=1}^{t-1} \cdot \frac{\mathbb{I}\{X_{s,j,k} = 1, \{X_{s,j,k'} = 0\}_{k' \in I_s \setminus \{k\}}, \{X_{l,j,k'} = 0\}_{l \neq s, k' \in I_l}\}}{\alpha(\langle \hat{\theta}_k(s), Y_s \rangle, n_k(s))} \tag{4.10}$$

where $n_k(t)$ is the number of selections of influencer k up to round t, $X_{s,j,k}$ is a binary random variable equal to 1 when j is activated in round s by influencer k, $l \in \{1, 2, \ldots, t-1\}, k^{'} \in I_l \subseteq [K]$. We discuss next the external factor estimated through regular linear regression, used to regulate the proportion of hapaxes in the cascades generated by influencer k.

4.2 The External Factor α. The external factor, as stated before, is a sub-problem of Problem 1. The remaining potential of an influencer is modelled by the combination of the external factor and the average count of hapaxes from the Good-Turing estimator. Under the assumption of a Poisson distribution for the rewards, and their property of diminishing returns, the external factor can be chosen as an adaptation of the inverse link function (mean function) for the Poisson distribution:

$$\alpha(\langle \theta_k, Y_t \rangle, n_k(t)) = e^{f(n_k(t))(\langle \theta_k, Y_t \rangle)} \tag{4.11}$$

The $f(n_k(t))$ function is assumed to be non-increasing, models the influencer's fatigue, and depends on the influencer's selections. By combining the two estimators, the predicted values for this sub-problem are:

$$r'_k(t) = \frac{\ln\left(\frac{r_t n_k(t)}{\sum_{j=1}^{A_k} \sum_{s=1}^{t-1} \frac{\text{hapax}_{s,j,k}}{\alpha(\langle \hat{\theta}_k(s), Y_s \rangle, n_k(s))}}\right)}{f(n_k(t))}, \text{where} \tag{4.12}$$

$$\text{hapax}_{s,j,k} = \mathbb{I}\{X_{s,j,k} = 1, \{X_{s,j,k'} = 0\}_{k' \in I_s \setminus k}, \{X_{l,j,k'} = 0\}_{l \neq s, k' \in I_l}\} \tag{4.13}$$

The argument of the external factor function is a random variable $r'_k(t) = \langle \theta_k, Y_t \rangle + \eta_t$. The noise η_t is assumed conditionally 1-subgaussian. The regularized least-squares estimator for the feature vector is:

$$\hat{\theta}_k(t) = V_k^{-1}(t) \sum_{s=1}^{t-1} Y_s r_k^{'}(s) \mathbb{I}\{k \in I_s\}, \tag{4.14}$$

where $V_k(t) = \gamma I + \sum_{s=1}^{t-1} Y_s Y_s^T \mathbb{I}\{k \in I_s\}$; $\gamma \geq 0$ is the penalty factor that ensures an unique solution. The design matrix $V_k(t)$ is computed from the contexts of the rounds in which the corresponding influencer was played, adjusted by its number of the selections.

4.3 Upper-Confidence Bound. UCB algorithms provide a disciplined balance between the exploitation of the options that are known as best up to the decision round, and the exploration of the ones for which the learning agent has not acquired enough information yet. The `GLM-GT-UCB` algorithm follows the main lines of an UCB-based algorithm, and its flow is presented in Algorithm 1. It starts with an initialization phase, where each influencer is played once in a random context. The observed rewards are used to initialize the influencer's statistics, necessary for further decisions. For the Good-Turing estimator, we maintain the number of selections $n_k(t)$ and the history of the discounted rewards for computing this estimator, as well as the sample-mean activations for computing the UCB index. For the linear regression of the external factor, we maintain a history of the rewards and the design matrix for each influencer:

$$V_k(t) = \gamma I_d + \sum_{s=1}^{t-1} Y_s Y_s^T \mathbb{I}\{k \in I_s\} \tag{4.15}$$

$$s_k(t) = \sum_{s=1}^{t-1} Y_s r'_k(s) \mathbb{I}\{k \in I_s\}. \tag{4.16}$$

To better use the available contextual information, we add the contextual UCB to the estimated external factor

$$\alpha(\langle \hat{\theta}_k, Y_t \rangle, n_k(t)) = e^{f(n_k(t))\left(\langle \hat{\theta}_k, Y_t \rangle + \gamma \sqrt{Y_t^T V_k^{-1}(t) Y_t}\right)} \tag{4.17}$$

In each subsequent round, the agent gets the context from the environment. It estimates for each influencer its feature vector, by the regularized least-square estimator in the stochastic linear bandit $\hat{\theta}_k(t)$, which is then used to compute the estimator of the remaining potential. The UCB $b_k(t)$ is obtained by adding the confidence factor $\beta_k(t)$. The agent plays the influencers with the highest UCBs, observes and divides the reward equally among them, and updates their statistics.

The UCB index computed on the adapted Good-Turing estimator captures both the confidence in the

unmodified Good-Turing estimator, and the one in the estimator of the influencer's true unknown vector θ_k:

$$b_k(t) = G_k(t) + \beta_k(t) + \hat{\lambda}_k(t)\left(1 - e^{\frac{-2+C_k(t)}{n_k(t)}}\frac{1}{n_k(t)}\sum_{s=1}^{t-1} e^{\frac{-2-C_k(s)}{n_k(s)}}\right), \text{ where} \tag{4.18}$$

$$\beta_k(t) = \sqrt{\frac{2\hat{\lambda}_k(t)e^{\frac{3+2C_k(t)}{n_k(t)}}\sum_{s=1}^{t-1} e^{\frac{2-2C_k(s)}{n_k(s)}}}{n_k^2(t)}\ln\frac{1}{\delta}} + \sqrt{\frac{e^{2/n_k(t)}\hat{\lambda}_k(t)\ln(1/\delta)}{\sum_{s=1}^{t-1}\sum_{k'\in I_s} e^{-1/n_{k'}(s)}}} + \frac{e^{\frac{1+C_k(t)}{n_k(t)}}\sum_{s=1}^{t-1} e^{\frac{1+C_k(s)}{n_k(s)}}}{3n_k(t)}\ln\frac{1}{\delta}, \tag{4.19}$$

$C_k(t) = \gamma\|Y_t\|_{V_k^{-1}(t)}$ is the contextual UCB for the external factor and $\hat{\lambda}_k(t) = \frac{1}{n_k(t)}\sum_{s=1}^{t-1}\frac{|F_s|}{L}\mathbb{I}\{k \in I_s\}$ is the sample-mean number of influencer k's activations.

4.4 Theoretical Analysis The UCB index is chosen as the maximum difference that can occur between the GT estimator and the true remaining potential with some chosen confidence. Theorem 4.1 provides the confidence interval for the estimated remaining potential. Its proof has three steps: the concentration of the true remaining potential, the concentration of the Good-Turing estimator, the bias of the estimator.

THEOREM 4.1. *With probability at least $1-\delta$, having the expected activations $\lambda_k = \sum_{j=1}^{A_k} p_{k,j}(t)$, and $\beta_k(t)$ set as in Equation 4.19 , we have*

$$-\beta_k(t) + \Omega\left(\frac{T\lambda_k(T)}{n_k(T)}e^{\frac{C_k(T)}{n_k(T)}}\right) \leq R_k(t) - G_k(t) \leq \beta_k(t) + \mathcal{O}\left(T\frac{\lambda_k(T)}{n_k(T)}e^{\frac{C_k(T)}{n_k(T)}}\right) \tag{4.23}$$

Proof. See the extended version [17]. ∎

5 Log-normal Distribution

We now consider the second alternative, that the underlying distribution is a log-normal one. Now, the influence maximization problem can be solved by an adapted LinUCB [10]. LinUCB computes the expected reward of each arm by finding a linear combination of the previous rewards of the arm. It estimates the unknown parameter θ_t of the current round as a linear combination of the previously seen feature vectors and rewards, and it estimates the expected reward on the current round by linearly combining it with the current feature vector. The adaptation of LinUCB to our problem consists in maintaining a design matrix per influencer, which is updated by the context of the round in which the influencer has been played. This change implies that a separate parameter is estimated for each influencer, and its linear combination with the current round's context will estimate the reward at logarithmic scale. Note that the linear combination estimates the scale of the reward since we assume that the rewards are log-normally distributed. The main flow is presented in Algorithm 2. It is similar to the one of LinUCB [10], in that at each step it chooses the best reward in terms of the linear combination of the context and the learned profile plus a confidence bound. In general linear models – of which `LogNorm-LinUCB` is part of – this bound is based on a design matrix V_k and the given context Y_t.

Algorithm 1 `GLM-GT-UCB`

Input: influencers K, rounds budget T, external factor function α, regularization factor γ, fatigue function f, number of selections L
Initialization: play each influencer $k \in [K]$ once in given random contexts Y_t, observe the reward $r_t, t \in [K]$, and update the statistics $n_k(1) = 1, \hat{\lambda}_k(1) = |F_k|$ for the Good-Turing estimator, and $V_k(1) = \gamma I + Y_t Y_t^T, s_k(1) = Y_t r'_k(1)$ [1] for the external factor.
for $t = K+1, \ldots, T$ **do**
  Get the context Y_t
  for $k \in [K]$ **do**
    Estimate the unknown vector:

$$\hat{\theta}_k(t) = V_k^{-1}(t)s_k(t) \tag{4.20}$$

    Compute UCB for remaining potential estimator

$$b_k(t) = G_k(t) + \beta_k(t), \tag{4.21}$$

$$G_k(t) = \alpha(\langle\hat{\theta}_k(t), Y_t\rangle, n_k(t))\cdot \frac{1}{n_k(t)}\sum_{j=1}^{A_k}\sum_{s=1}^{t-1}\frac{\text{hapax}_{s,j,k}}{\alpha(\langle\hat{\theta}_k(s), Y_s\rangle, n_k(s))} \tag{4.22}$$

    and $\beta_k(t)$ is given by the confidence interval.
  end for
  Choose set I_t of L influencers with largest UCB
  Play the chosen influencers, observe spread, divide it equally among influencers, update their statistics:
  for $k' \in I_t$ **do**
    Update $r_{k'}(t)$ by Eq. (4.12).; $n_{k'}(t+1) = n_{k'}(t) + 1; V_{k'}(t+1) = V_{k'}(t) + Y_t Y_t^T; s_{k'}(t+1) = s_{k'}(t) + Y_t r'_{k'}(t)$
  end for
end for

[1] $r'_k(1) = \frac{1}{f(1)}\ln\left(\frac{r_t}{\sum_{j=1}^{A_k}\sum_{s=1}^{t-1}\frac{\text{hapax}_{s,j,k}}{\alpha(\langle\hat{\theta}_k(s), Y_s\rangle, 1)}}\right)$.

Algorithm 2 `LogNorm-LinUCB`

Input: influencers K, selections L, $\gamma \in \mathbb{R}_+$, $d \in \mathbb{N}$
$V_k(1) = I_d, \forall k \in [K]$ and $s_k(1) = \mathbf{0}_d, \forall k \in [K]$
for t=1,. . . ,T **do**
  Get context Y_t
  for $k \in [K]$ **do**
    $\hat{\theta}_k(t) = V_k^{-1}(t) s_k(t)$
    $b_k(t) = \langle \hat{\theta}_k(t), Y_t \rangle + \gamma \sqrt{Y_t^T V_k^{-1}(t) Y_t}$
  end for
  Choose set I_t of L influencers with largest UCB $b_k(t)$
  Observe spread, compute reward r by discounting previously activated basic nodes and dividing by L.
  for $k^{'} \in [I_t]$ **do**
    $V_{k^{'}(t)}(t+1) = V_{k^{'}(t)}(t) + Y_t Y_t^T$
    $s_{k^{'}(t)}(t+1) = s_{k^{'}(t)}(t) + \ln(r) Y_t$
  end for
end for

5.1 Regret analysis. The regret analysis is performed at logarithmic scale; this restriction stems from having the logarithm of the new activations being normally distributed. In [10], theoretical guarantees for LinUCB were challenging, due to the lack of independence of the random variables for the rounds' rewards. The solution was to use a supporting algorithm, SupLinUCB, estimating the unknown parameter only from the feature vectors and rewards from the rounds in which the agent performs random exploration. Each round is split into levels, and each level maintains an index set used for learning, comprising the indices of the rounds with independent rewards. When exploring, the round is added to the index set of the corresponding level.

We designed similarly IM-SupLinUCB and its subroutine IM-BaseLinUCB, preserving the steps of SupLinUCB [10] and SupLinRel [3]. Each influencer's UCB is computed for the scale of the reward – new activations. We skip the analysis of IM-BaseLinUCB and IM-SupLinUCB's, as it is similar to [10, 3]. Regret for stochastic linear bandits is generally defined as:

$$\hat{\mathcal{R}}_t = \sum_{s=1}^{t} \max_{k \in [K]} \langle \theta_k, Y_t \rangle - \sum_{s=1}^{t} r_s \tag{5.24}$$

$$\mathcal{R}_t = \mathbb{E}[\hat{\mathcal{R}}_t] = \mathbb{E}\left[\sum_{s=1}^{t} \max_{k \in [K]} \langle \theta_k, Y_t \rangle - \sum_{s=1}^{t} r_s \right] \tag{5.25}$$

We have the following $\tilde{\mathcal{O}}(\sqrt{T})$ regret bound for the supporting algorithm on logarithms of rewards:

THEOREM 5.1. *If IM-SupLinUCB uses parameter* $\gamma = \sqrt{\frac{1}{2} \ln(\frac{2TK}{\delta})}$*, with probability* $1 - \delta$ *the regret of* `LogNorm-LinUCB` *at logarithmic scale is*

$$\hat{\mathcal{R}}_t \leq 2\sqrt{T} + 44K \left(1 + \ln\left(2TK \ln(T)/\delta\right)/2\right)^{\frac{3}{2}} \sqrt{Td} \tag{5.26}$$

Proof. Proof similar to that of [3, Theorem 6.]. □

6 Experiments

We tested `GLM-GT-UCB` and `LogNorm-LinUCB` on synthetically generated data, on data we collected from Twitter, and on a publicly available dataset from Sina Weibo [31]. All the results are averaged over 100 runs.

Synthetic data experiments. The synthetic data is generated starting from the premise that each basic node's activation probability is known. Therefore, all the edges and nodes are assumed to be known as well. The synthetic graph is randomly generated following the Barabási-Albert preferential attachment model [5]. The model's parameters are chosen as follows: 30,000 nodes and, at each step, one new edge to be attached from new nodes to existing ones. Then, the 10 nodes having the maximum degrees are chosen to be the influencers.

Activation probabilities are computed as a sigmoid function of the inner product of the context and the basic node's feature vector plus some random small noise. This is preferable in order to project the results intro probability thresholds, i.e., the value over which the node is considered activated - 0.999 in our experiments. The inner product captures the linear relationship between context and hidden profile. For each node, its feature vector is randomly generated from a normal distribution. Then, the context of a campaign's round is generated from another normal distribution. A given round is chosen to be viral with a 50% probability, i.e., the distribution from which the context is drawn is chosen such that its inner product with most of the basic user feature vectors results in higher values for the activation function. For these rounds, only $L+1$ influencers are chosen to use the viral context. The diffusion model is assumed to be IC [18], a campaign consist of 500 rounds, and results are averaged over 100 runs. γ is set to $\sqrt{1/2 \ln(\sqrt{2TK/\delta})}$ everywhere.

Baselines methods. We compare against Random, UCB1 [4], LinUCB [10], and FAT-GT-UCB [19]. The random policy chooses a random influencer in each round. UCB1 is a well-known algorithm in the bandit literature, one which does not model contexts. The FAT-GT-UCB algorithm models the influencer's fatigue in a context-free setting. The results (Fig. 1, top row) show that `GLM-GT-UCB` and `LogNorm-LinUCB` are both capable of learning the remaining potential of influencers from their performance in different contexts. Making decisions based on the available information about the round's context has a clear added value, compared to only considering the time-based fatigue of approaches such as FAT-GT-UCB.

Twitter dataset. We extracted from Twitter logs a collection of retweets. These can be viewed as belonging to basic nodes, representing successful activa-

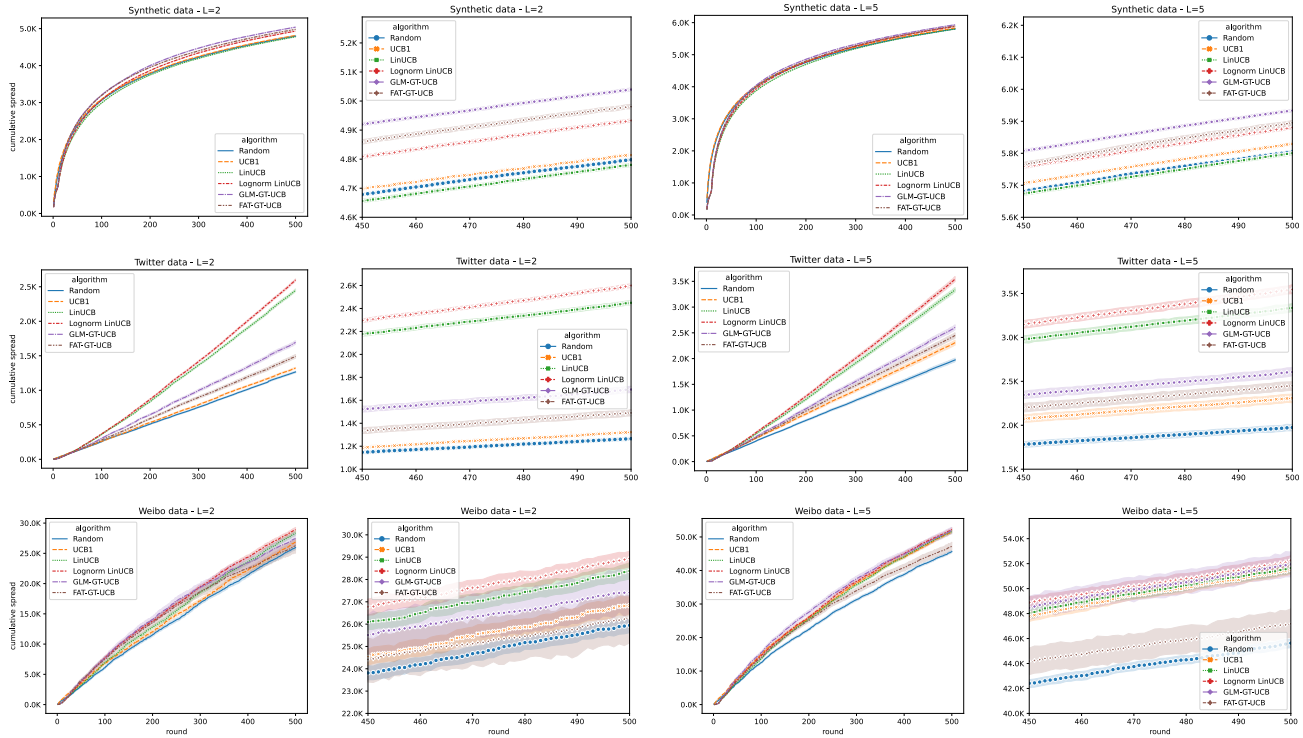

Figure 1: Cumulative rewards – L=2, L=5 – normal plot (odd rows) & plot zoomed to last 50 rounds (even rows).

tions of the original tweets from influencers. To test the capability of the algorithms to choose the right influencers for a given context, we extracted from tweet the round's context. As in [26], a tweet is encoded into a multi-dimensional vector. The encoding represents the distribution of the tweets' words over a predefined number of centroids (24 in our experiments). The centroids are obtained via clustering (K-means) on the public vocabulary *glove-twitter-200*[2] from the word embedding open-source library Gensim[3]. Each word is assigned to its closest centroid, thus obtaining the distribution. The largest cluster is split into 5 smaller clusters.

In Twitter and Weibo, we improve the learning rate of `GLM-GT-UCB` by adding $10/L$ activations only when learning the external factor via linear regression. The plotted results are with the true values of activations.

The campaigns are created by randomly choosing the context for each round to be one of the available centroid distributions in the dataset. We chose the set of influencers to be the users with the highest degrees. In each round, each algorithm chooses which influencers it wants to play. Due to the sparsity in the data, we implemented the bandit to sample with replacement from the set of all tweets with the round's context matching their centroid distribution and the algorithm's chosen influencer as the original user id. If there is no log for this tuple, we consider that no basic node has been activated. The reward is computed by discounting previously activated basic nodes.

The results are in Fig. 1 (middle row), for either the entire campaign of 500 rounds or zoomed on rounds 450 to 500; the shaded areas represent the uncertainty.

Weibo dataset. Using a public dataset from the popular Chinese microblogging platform, we designed the experiment as in the Twitter scenario. The topic distributions created by [31] are used as contexts. There are 100 topics, and for each post the distribution of topics is computed by using Latent Dirichlet Allocation [15]. Once again, in Fig. 1 (bottom row) we can see that our methods manage to perform better by using the round's context information when selecting influencers. The relative performance can depend on time: `GLM-GT-UCB` seems to initially learn faster.

From both experiments on real-world datasets, we can conclude that our approaches – especially `LogNorm-LinUCB` – are capable of learning viral cascades in different datasets and cascade settings, which increases their potential in spread maximization (visible in the "steps" of the plots); this is not the case with other approaches, which seem to work best when the cascades have fewer outliers in terms of size; hence, they do not learn quickly enough to adapt.

[2] `https://nlp.stanford.edu/projects/glove/`

[3] `https://github.com/RaRe-Technologies/gensim-data`

7 Conclusion

We presented in this paper the problem of designing advertising campaigns from the point of view of contextual influence maximization, when the exact diffusion model is not fully exploitable. By adapting approaches from the contextual bandit literature, we designed algorithms `GLM-GT-UCB` and `LogNorm-LinUCB`, using different assumptions on the underlying distributions of the number of influenced nodes: Poisson and log-normal respectively. We showed both theoretically and experimentally that our approaches have the potential to learn the influencers' potential, leading to improved IM campaigns compared to other state-of-the-art methods.

Acknowledgments

We thank Olivier Cappé and Yoan Russac for early discussions and ideas on modeling the distribution of rewards. This work was also supported by the Singapore NRF DesCartes research grant.

References

[1] Akhil Arora, Sainyam Galhotra, and Sayan Ranu. Debunking the myths of influence maximization: An in-depth benchmarking study. In *SIGMOD*, 2017.

[2] Akhil Arora, Sainyam Galhotra, and Sayan Ranu. Influence maximization revisited: The state of the art and the gaps that remain. In *EDBT*, 2019.

[3] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *JMLR*, 2002.

[4] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 2002.

[5] Albert-László Barabási et al. *Network science*. Cambridge university press, 2016.

[6] Danny Brown and Sam Fiorella. *Influence Marketing: How to Create, Manage, and Measure Brand Influencers in Social Media Marketing*. Que Pub, 2013.

[7] S. Bubeck, D. Ernst, and A. Garivier. Optimal discovery with probabilistic expert advice: finite time analysis and macroscopic optimality. *JMLR*, 2013.

[8] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Found. Trends Mach. Learn.*, 2012.

[9] Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *JMLR*, 2016.

[10] W. Chu, L. Li, L. Reyzin, and R. Schapire. Contextual bandits with linear payoff functions. In *AISTATS'11*.

[11] N. Du, L. Song, H. Woo, and H. Zha. Uncover topic-sensitive information diffusion networks. In *AISTATS'13*.

[12] Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *NIPS*, 2010.

[13] M. Gomez-Rodriguez, J. Leskovec, and A. Krause. Inferring networks of diffusion and influence. *ACM TKDD'12*.

[14] Irving J Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 1953.

[15] Gregor Heinrich. Parameter estimation for text analysis. Technical report, 2005.

[16] Shoubo Hu, Bogdan Cautis, Zhitang Chen, Laiwan Chan, Yanhui Geng, and Xiuqiang He. Model-free inference of diffusion networks using RKHS embeddings. *Data Min. Knowl. Discov.*, 2019.

[17] Alexandra Iacob, Bogdan Cautis, and Silviu Maniu. Contextual bandits for advertising campaigns: A diffusion-model independent approach (extended version). *arXiv*, abs/2201.05231, 2022.

[18] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *KDD'03*.

[19] P. Lagrée, O. Cappé, B. Cautis, and S. Maniu. Algorithms for online influencer marketing. *ACM TKDD'18*.

[20] S. Lei, S. Maniu, L. Mo, R. Cheng, and P. Senellart. Online influence maximization. In *SIGKDD'15*.

[21] Nir Levine, Koby Crammer, and Shie Mannor. Rotting bandits. *arXiv preprint arXiv:1702.07274*, 2017.

[22] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW*, 2010.

[23] Lihong Li, Yu Lu, and Dengyong Zhou. Provably optimal algorithms for generalized linear contextual bandits. In *ICML*, 2017.

[24] Y. Li, J. Fan, Y. Wang, and K. Tan. Influence maximization on social graphs: A survey. *TKDE'18*.

[25] Alessandra Sala, Haitao Zheng, Ben Y. Zhao, Sabrina Gaito, and Gian Paolo Rossi. Brief announcement: revisiting the power-law degree distribution for social graph analysis. In *PODC*, 2010.

[26] D. Shin, S. Cetintas, K-C. Lee, and I. S. Dhillon. TUMBLR blog recommendation with boosted inductive matrix completion. In *CIKM'15*.

[27] Aleksandrs Slivkins. Introduction to multi-armed bandits. *Found. Trends Mach. Learn.*, 12(1-2):1–286, 2019.

[28] Sharan Vaswani, Branislav Kveton, Zheng Wen, Mohammad Ghavamzadeh, Laks V. S. Lakshmanan, and Mark Schmidt. Model-independent online learning for influence maximization. In *ICML*, 2017.

[29] Z. Wen, B. Kveton, M. Valko, and S. Vaswani. Online influence maximization under independent cascade model with semi-bandit feedback. In *NIPS*, 2017.

[30] Qingyun Wu, Zhige Li, Huazheng Wang, Wei Chen, and Hongning Wang. Factorization bandits for online influence maximization. In *KDD*, 2019.

[31] Jing Zhang, Biao Liu, Jie Tang, Ting Chen, and Juanzi Li. Social influence locality for modeling retweeting behaviors. In *IJCAI*, 2013.