Sequential Learning Algorithms for Contextual Model-Free Influence Maximization

Alexandra Iacob
Université Paris-Saclay, CNRS, LISN
Gif-sur-Yvette, France
alexandra.s.iacob@gmail.com

Bogdan Cautis
Université Paris-Saclay & CNRS IPAL
Singapore
bogdan.cautis@universite-paris-saclay.fr

Silviu Maniu
Université Paris-Saclay, CNRS, LISN
Gif-sur-Yvette, France
silviu.maniu@lisn.upsaclay.fr

ABSTRACT

The *online influence maximization* (OIM) problem aims to learn sequentially an optimal policy for selecting seed nodes which maximize the cumulative spread of information (influence) in a diffusion medium, throughout a multi-round diffusion campaign. We consider the sub-class of OIM problems where (i) the reward of a given round of the ongoing campaign consists of only the *new activations* (not observed at previous rounds), and (ii) the round's context and the historical data from previous rounds can be exploited to learn the best policy. This problem is directly motivated by the real-world scenarios of information diffusion in *influencer marketing*, where (i) only a target user's *first* / unique activation is of interest (and this activation will *persist* as an acquired, latent one throughout the campaign), and (ii) valuable side-information is available to the learning agent. We call this OIM formulation *Episodic Contextual Influence Maximization with Persistence* (in short, ECIMP). We propose the algorithm `LSVI-GT-UCB`, which implements the *optimism in the face of uncertainty* principle for episodic reinforcement learning with linear approximation. The learning agent estimates for each seed node its remaining potential with a Good-Turing estimator, modified by an estimated Q-function. The algorithm is empirically proven to perform better than state-of-the-art methods on two real-world datasets and a synthetically generated one.

CCS CONCEPTS

• **Computing methodologies** → *Q-learning*; *Online learning settings*.

KEYWORDS

information diffusion, influencer marketing, MDP, Good-Turing estimator

ACM Reference Format:
Alexandra Iacob, Bogdan Cautis, and Silviu Maniu. 2023. Sequential Learning Algorithms for Contextual Model-Free Influence Maximization. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '23), August 6–10, 2023, Long Beach, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3580305.3599498

KDD '23, August 6–10, 2023, Long Beach, CA, USA

ACM ISBN 979-8-4007-0103-0/23/08…$15.00
https://doi.org/10.1145/3580305.3599498

1 INTRODUCTION

Information diffusion by stochastic epidemic models has been widely studied in recent research, for diverse domains of application, from social networks [19] to electricity grids [28] to IoT and communication networks [16].

When information diffusion occurs in online media (e.g., social networks), the generic problem of *Influence (i.e., spread) Maximization* (IM) has been introduced by [19], and is one of the most studied problems in the literature, due to its applicability to viral marketing [8], ad placement [30], or personalized recommendation [14, 29].

From a simple and generic perspective, given a diffusion network represented as a directed, weighted (probabilistic) graph $G = (V, E, p)$, the IM problem to solve is that of finding L seed nodes (influencers), from which to initiate an information diffusion process, with the objective of maximizing the number of influenced (activated) nodes, i.e. *the reward*. The seminal work of Kempe et al. [19] proposed two models for the information diffusion process, the Independent Cascade (IC) model and the Linear Threshold (LT) one. Under the former model, the process develops in discrete steps, starting with the selection of L seed nodes, where at each step the newly activated nodes attempt to influence and activate their neighbors, succeeding with a probability $p_{ij}, \forall (i, j) \in E$. In the LT model, at any step in the diffusion process, a node becomes active if the sum of the weights of its active incoming neighbors is above that node's own activation threshold. Under both diffusion models, the IM problem has been shown to be NP-hard, reducing to the Set-Cover problem and the Vertex-Cover problem respectively [18].

Many instances of the IM problem have been considered in recent years, for diverse problem settings, application scenarios, or performance objectives. E.g., the diffusion network may be a bipartite graph, modeling any-path diffusion from influencers to target nodes [2], diffusions may be topic-aware [3, 7], the seeds may be selected by an online approach [23] instead of the classical select-then-spread offline one, or the diffusion process may be repeated over multiple rounds (a diffusion campaign), with the objective being the cumulative reward [21].

In particular, online influence maximization (OIM) over multiple rounds [15, 21, 33] allows to deal with problem settings having (partially) unknown diffusion specifications (i.e., network and / or diffusion model). Therein, starting from a known base of few influential nodes, one can discover and *learn* the diffusion environment while maximizing spread, over multiple rounds of a learn & spread campaign. Consequently, the IM objective shifts from a per-diffusion (round) one to a per-campaign one, e.g., by maximizing the total number of activations or, alternatively, the total number of *distinct* activations.

We study an online IM problem setting that connects many of the recently considered IM assumptions for practical purposes, namely (i) an unknown diffusion medium and an influencer-target node bipartite graph abstraction thereof, (ii) diffusion model-independent spread over multiple rounds of an influence campaign, (iii) topic / context dependent diffusions, and finally (iv) a spread objective given by the campaign's total number of *distinct* activations.

Our problem setting is directly motivated by (but not limited to) the real-world scenarios of information diffusion in *influencer marketing*, where (i) the diffusion medium is highly uncertain and only a few influencer nodes may be known in advance, (ii) only a target user's *first* activation is of interest and, once acquired, it will *persist*; e.g., as in political endorsements or subscriptions to a media service, and (iii) valuable side-information may be available to the learning agent.

Recent research has considered topic-aware online IM in uncertain environments, by sequential learning approaches of Contextual Multi-Armed Bandits (MABs) [1, 9, 15, 22]. Contextual MABs are rather versatile approaches, providing both formal guarantees and effective approximation algorithms. However, most often their performance is only theoretically established. Moreover, they fail to capture many realistic scenarios, where actions are influenced by other factors than the context and the previously sampled i.i.d reward random variables. In particular, for information diffusion with persistent activations – i.e., where only the new activations are rewarded – repeatedly selecting the same seemingly optimal seed node(s) may lead to non-optimal policies for seed selection. Instead, this important aspect may be captured by *reinforcement learning* states, thus preserving the i.i.d requirement for the rewards.

To enable decision-making based on the seed's number of selections, we propose instead a sequential learning method based on *episodic reinforcement learning* (RL), called *Episodic Contextual Influence Maximization with Persistence* (in short, ECIMP).

In our approach, not only each influencer's number of selections can inform the decision making process, but also the historical data from previous campaigns can be exploited for the learnt policy. Therefore, the focus moves from one of sequentially learning during a single campaign with multiple rounds to one of learning from *multiple campaigns* with multiple rounds. In the RL terminology, a diffusion *episode* will be the equivalent of a diffusion campaign, and its *horizon* will be the equivalent of the number of rounds.

Due to the contextual information provided by the environment at the beginning of each round, the state space may be extremely large. For such cases, the recent works of [17, 31, 32] have successfully used (generalized) linear function approximations to estimate the value function or the policy, a direction we also adopt here.

An optimal policy would be the one finding at each round the seed nodes with the most yet-to-be-activated target nodes in that round's context. This untapped resource is seen a seed's *remaining potential*, and it has been previously successfully computed with Good-Turing estimators [15, 21]. If we assign to each potential seed node its own episodic MDP, then the value function in each state represents the respective seed's remaining potential. Assuming that an activation's novelty is relative to one campaign, and knowing that the Good-Turing estimator is just an average of new activations, we use the historical data from previous campaigns for computing the average GT estimator for the round over the episodes.

Contribution. The main contributions of this paper are as follows:

- We formally describe the ECIMP problem, jointly drawing motivation from several up-to-now disjoint and practically relevant recent studies in the area of online Influence Maximization, for learning seed selection policies in unknown diffusion environments.
- We propose the novel algorithm `LSVI-GT-UCB`, which implements the *optimism in the face of uncertainty* principle for episodic reinforcement learning with linear approximation.
- As a key aspect of `LSVI-GT-UCB`, we describe how the learning agent can estimate for each seed node its remaining potential with a Good-Turing estimator, modified by an estimated Q-function.
- We evaluate empirically the performance of `LSVI-GT-UCB` on two real-world datasets and a synthetically generated one, comparing with state-of-the-art methods.

2 MAIN RELATED WORK

The work of [25] studies the OIM problem in social networks under the assumptions of the LT model and node-level feedback. The proposed algorithmic solution is called `LT-LinUCB`. It exploit the linearity of node activations in the LT model, obtaining an $O(\text{poly}(m)\sqrt{T}\log T)$ regret, where m is the number of edges and T is the number of rounds. They propose also the model-free `OIM-ETC` algorithm, with an $O(\text{poly}(m)T^{\frac{2}{3}})$ regret bound. In [35], the authors propose an $\tilde{O}(\sqrt{T})$ algorithm, called `IC-UCB`, for the OIM problem, assuming the IC model and node-level feedback. `IC-UCB` uses a standard offline IM oracle to find the best seed set, and estimates the IC model's edge parameters p by transforming them into another parameter; this leads to an instance of the generalized linear model problem [11], which is solved with MLE. The recent work of [15] focuses on the OIM problem formulation where the reward consists of only the new activations. Their contextual MABs solution, called `GLM-GT-UCB`, estimates the number of remaining inactivated target nodes for each seed node with a Good-Turing estimator, which is modified by a function of the side-information (context) provided by the environment. Their Good-Turing estimator has guarantees for its confidence bound, and it is experimentally shown to be effective.

In [20], the authors designed a generic framework for learning graph heuristics and finding approximate solutions for NP-hard combinatorial optimization problems. The framework is a greedy meta-algorithm, `Q-learning for the Greedy Algorithm`, learned over multiple episodes of RL over different problem instances sampled from a given graph distribution. For each problem type, one must provide specific helper / cost functions, as well as termination criteria for the meta-learning algorithm to solve the problem. E.g., for the Set-Cover problem, the helper function would be the identity one, as there is no need for a combinatorial structure on the partial solution, the cost function would the size of the partial solution, and the termination criteria would be either when all nodes are covered or the budget is spent. At each step within the episode's horizon, `Q-learning for the Greedy Algorithm` chooses a seed node either randomly with probability ϵ, or the one which maximizes the estimated Q-function $\hat{Q}(h(S_t), v; \Theta), \forall v \in V, G = (V, E)$. For each node $v \in V$ in the graph, the algorithm uses *structure2vec* (S2V) [10] to encode its neighborhood, given the current partial

solution S_t. This framework is quite versatile and applicable on a wide-range of combinatorial optimization problems on graphs. However, it requires knowledge about the graph's topology, as the embedding of the potential new seed node and the pooled embedding over the entire graph are combined to provide the estimated Q-function. The approximator's parameters are updated in batches, which allows delayed rewards. The algorithm is experimentally proven to be successful for various types of NP-hard problems, graph types, and graph sizes. Recently, [24] has adapted this approach, in the PIANO framework for influence maximization.

The work of [17] considers RL with a very large number of states, by incorporating function approximation in the learning process. The proposed algorithm, called *LSVI-UCB*, models the problem as an episodic Markov Decision Process (MDP), with the assumptions that the transition dynamics and the reward function are linear. It is proven that the action-value function is consequently linear as well, and the algorithm is designed to approximate well this quantity. Inspired by the linear bandits literature, the algorithm implements the *"optimism in the face of uncertainty"* principle – it encourages exploration by adding an UCB bonus. It achieves $\tilde{O}(\sqrt{d^3H^3T})$ regret, where d is the ambient dimension of the feature space, H denotes the horizon (i.e., length of each episode), and T is the total number of steps. LSVI-UCB runs in polynomial time ($O(d^2AKT)$), where A is the size of the action space and K is the number of episodes. The algorithm benefits from the sample complexity guarantees, in the sense that with constant probability it can learn an $\epsilon-$optimal policy π which satisfies $V^*(x_1) - V^\pi(x_1) \leq \epsilon$, using $\tilde{O}(d^3H^4/\epsilon^2)$ samples, when the initial state x_1 is fixed for all episodes. The algorithm is shown to be robust to small model variations, under the condition of using a different hyper-parameter β from the UCB in different episodes. The main drawback of [17] is that their solution remains limited to "almost" linear MDPs.

Under weaker assumptions than [17], [32] proposes an efficient least-squares dynamic programming algorithm for episodic RL, also called *LSVI-UCB*, which approximates the Q-function with a Generalized Linear Model. The approximator overestimates the optimal Q-function, implementing the "optimism in face of uncertainty" principle. The statistical efficiency is theoretically proven with a regret bound of $\tilde{O}(\sqrt{d^3T})$, where d is the feature dimension, and T is the number of episodes. For these results, an optimistic closure with respect to the Bellman operator assumption is made. The assumption is shown to be weaker than the one of linearity, by providing an MDP which meets the former but not the latter. Optimistic closure also implies realizability, which is typical for contextual multi-armed bandits, where the horizon is $H = 1$.

3 PROBLEM FORMULATION

The IM problem addressed in this paper is aimed at information diffusion scenarios – e.g., in a social network, but generally in any medium that may exhibit stochastic / epidemic diffusion – where multiple attempts of spreading the information are made, and the new activations make up the reward. The diffusion network can be naturally represented as a graph $G = (V, E)$ where V are the nodes (users, profiles) and E are the edges (relationships). In our setting, this topology is assumed to be unknown.

Table 1: Summary of notations.

T	total number of campaigns / episodes
H	total number of rounds in a campaign
K	total number of available influencers
d	the ambient dimension of the feature space
$Y_{t,h}$	the d-dimensional context at round h of episode t
$I_{t,h}$	the set of L influencers selected at round h of episode t
A_k	set of basic nodes reachable by influencer k
$S(I_{t,h}, Y_{t,h})$	the spread given by the environment at round h of episode t
$p_{k,j}(t,h)$	the probability of influencer k activating the basic node j at round h of episode t
$\theta_{k,j}$	feature vector that explains the probability of influencer k to activate basic node j in the round's context Y_t
$n_{k,t,h}$	the history of number of selections of influencer k at round h of episode t
$p(j)$	the intrinsic probability of activating itself of a basic node j
$r_{k,t,h}$	the reward for influencer k at the end of round h in episode t
$R_{k,t,h}$	the influencer k's remaining potential (i.e. the feasible reward) at round h of episode t
$\hat{R}_{k,t,h}$	Good-Turing estimator of the remaining potential for influencer k at round h of episode t
$\hat{Q}^{GT}_{k,t,h}$	Q-function estimated with a Good-Turing estimator for influencer k at round h of episode t
$\hat{Q}^{LSVI}_{k,t,h}(\cdot,\cdot)$	Q-function estimated with Least-Squares Value Iteration for influencer k at round h of episode t

Instead, a set of K potential influencers (among the nodes in V) is assumed to be known, and the influence process can only start from them, with the effect of activating certain nodes among those from an unknown overall set of *basic* (influenced) nodes. While we make no assumptions on the diffusion model that leads to activations, we assume to get *semi-bandit feedback* after each *round* that spreads a given "message", as the set influenced nodes (a set of node *Ids*).

Over a *campaign*, consisting of a number of H rounds, the *reward* is defined as the number of *new activations*.

The message that is to be spread at each round $h \in [H]$ is encoded as a vector $Y_h \in \mathbb{R}^d$ and the probability that each target (or basic) node j adopts it – or gets influenced / activated by it – depends linearly on j's hidden profile relative to the influencer k seeded at that round, denoted $\theta_{k,j}$, and the message (plus some noise). So the response of a target node j is given by $\langle \theta_{k,j}, Y_h \rangle + \epsilon$.

This response of node j, along with the number of times the influencer k was seeded to send a message in the campaign, denoted n_k, are used in a generalized linear function α, called the *external factor*. The role of the external factor α is to modulate the default (inherent) propensity of node j to activations, denoted as $p(j)$.

The following single-campaign problem, formulated in [15], can be used as an intermediary step to introduce the more general ECIMP setting we study in this paper:

Problem 1 (Contextual Influence Maximization [15]). *Given a set of influencers $[K] = 1, \ldots, K$, a budget of H 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:*

$$\underset{I_h \subseteq [K], |I_h| = L, \forall 1 \leq h \leq H}{\operatorname{argmax}} \mathbb{E}\Big| \bigcup_{1 \leq h \leq H} S(I_h, Y_h)\Big|, \tag{1}$$

where $S(I_h, Y_h)$ is the spread of the chosen set of influencers for round h, and the probability that influencer k activates basic node j depends

on the round's context Y_h and the number of k's selections $n_k(h)$:

$$p_{k,j}(h) = \alpha(\langle \theta_{k,j}, Y_h \rangle, n_k(h))p(j). \quad (2)$$

The solution to Problem 1 relies on upper-confidence bound approaches (UCB), which need to estimate at each step the *remaining potential* of influencers. In the bandit literature, an estimator that has been used successfully for such problems is the Good-Turing (GT) estimator [6], where the remaining potential can be estimated via the *hapaxes*, a notion from linguistics that describes here the nodes that have been activated only once. Applying this estimator, coupled with a theoretically derived upper-confidence bound, has been shown to work well in practice when the distribution on the number of newly activated nodes follows a Poisson distribution (Algorithm `GLM-GT-UCB` of [15]).

We describe next how the Problem 1 can be extended to the *episodic* case of multiple campaigns, each having multiple rounds, with the objective of learning between campaigns.

3.1 Reinforcement learning setting

In the episodic, i.e., multi-campaign setting, the problem becomes:

Problem 2 (Episodic Contextual Influence Maximization with Persistence (ECIMP)). *Given a set of influencers $[K] = 1, \ldots, K$, a budget of T campaigns, each consisting of of H rounds, 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:*

$$\underset{I_{t,h} \subseteq [K], |I_{t,h}| = L, \forall 1 \leq h \leq H, \forall 1 \leq t \leq T}{\operatorname{argmax}} \mathbb{E}\Big| \bigcup_{1 \leq h \leq H, 1 \leq t \leq T} S(I_{t,h}, Y_{t,h}) \Big|, \quad (3)$$

where $S(I_{t,h}, Y_{t,h})$ is the spread of the chosen set of influencers for round h in campaign t, and the probability that an influencer k activates some basic node j depends on the round's context $Y_{t,h}$ and the number of k's selections $n_{k,t,h}$ in campaign t:

$$p_{k,j}(t,h) = \alpha(\langle \theta_{k,j}, Y_{t,h} \rangle, n_{k,t,h})p(j). \quad (4)$$

(In what follows, we will use the terms campaign and episode interchangeably. The former is closer to the terminology of the application scenario, the latter is common from episodic RL.)

The problem can be naturally modeled as K episodic Markov decision processes, one for each influencer $k \in [K]$, namely $\text{MDP}(\mathcal{S}_k, \mathcal{A}_k, H_k, \mathbb{P}_k, r_k)$, where $\mathcal{S}_k$ is the state space, $\mathcal{A}_k$ is the set of possible actions, H_k is the horizon within each episode, $\mathbb{P}_k = \{\mathbb{P}^H_{k,h=1}\}$ is the set of state transition probability measures, and $r_k = \{r^H_{k,h=1}\}$ is the set of reward functions. The state space of such an MDP can be very large, possibly infinite. The action spaces, given that an MDP is maintained for each influencer, is the binary set for each influencer being selected or not, i.e., $\mathcal{A}_k = \{0, 1\}$ and $\mathcal{A} = \{0, 1\}^K$. The reward is assumed to be uniquely defined by $r_{k,h}(s_k, a_k, s_k^{'}), \forall s_k, s_k^{'} \in \mathcal{S}_k, a_k \in \mathcal{A}_k$, which can further be bounded and simplified to $r_{k,h}(s_k, a_k)$. Recall the reward is the count of new activations, which is naturally bounded by the total number of users.

As in the RL literature in general, at the beginning of each episode, the initial states $s_{k,1}, \forall k \in [K]$ are given and thereon the learning agent interacts with the episodic MDPs. It observes the states $s_{k,h} \in \mathcal{S}_k$ at each step $h \in [H]$ and proceeds to take an overall action $a_h \in \mathcal{A}$. The MDPs of the selected influencers are transitioning according to their transition dynamics $\mathbb{P}_k$ to the new states $s_{k,h+1}$. When an influencer is not selected, i.e., $a_{k,h} = 0$, the state remains the same, $\mathbb{P}_{k,h}(s|s, 0) = 1$. The state transition dynamics are stochastic and unknown upon selecting an influencer. The final state of each MDP is $s_{k,H+1}$, where no action can be taken anymore and the reward is consequently zero. Obviously, in this setup, the MDPs may not reach their final state before the total campaign budget H is spent.

The goal is to maximize the number of distinct activations at the end of a campaign, as defined in Problem 2, leveraging information from previous rounds *and previous campaigns*. This optimization problem expressed in terms of episodic MDPs has as solution the optimal policy $\pi^* \in \operatorname{argmax}_{\pi \in \Pi} V^\pi(s), \forall s \in \mathcal{S}$, where Π is the policy set, and $\mathcal{S} = \bigcup_{k \in [K]} \mathcal{S}_k$. The agent aims to learn the optimal policy $\pi^* : \mathcal{S} \times [H] \to \mathcal{A}$.

The policies are evaluated by their corresponding value functions or their action-value functions. The Bellman equations for these values are the following:

$$V^\pi_{k,h}(s_k) = \mathbb{E}\left[\sum_{\tau=h}^{H-1} r_{k,\tau}(s_{k,\tau}, \pi(s_{k,\tau}, \tau)) | s_{k,h} = s_k\right], \forall s_k \in \mathcal{S}_k, \forall h \in [H].$$

$$Q^\pi_{k,h}(s_k, a_k) = r_{k,h}(s_k, a_k) + \mathbb{E}\left[\sum_{\tau=h}^{H-1} r_{k,\tau}(s_{k,\tau}, \pi(s_{k,\tau}, \tau)) | s_{k,h} = s_k, a_{k,h} = a_k\right], \forall (s_k, a_k) \in \mathcal{S}_k \times \mathcal{A}_k, \forall h \in [H].$$

The optimal value function is

$$V^*_h(s) = \max_{a_k \in \mathcal{A}_k, k \in [K]} Q^*_{k,h}(s, a_k), \ \forall s \in \mathcal{S}.$$

So the learning agent chooses its actions according to the greedy policy with respect to the estimated action-value functions:

$$\pi(h, s) = \operatorname*{argmax}_{a_k \in \mathcal{A}_k, k \in [K]} Q^\pi_{k,h}(s, a_k).$$

The flow of this learning process hence depends on how the Q function is estimated, as detailed next.

4 RL WITH AVERAGE GT ESTIMATORS AND LSVI LEARNED MODIFIERS

The Multi-Armed Bandit problem, and many of its well-known variants such as stochastic bandits, contextual bandits [22], contextual bandits with linear rewards [9], with generalized linear rewards [12], or with Good-Turing reward estimators [21] [15], restrict the reward random variables to be independent and identically distributed, i.e. independent of the previous action choices and rewards. However, the choice of actions may alter the state of the environment. In more general Reinforcement Learning problems, theoretical guarantees for the estimators can be obtained without ignoring the state of the environment.

4.1 The Good-Turing estimator

Good [13] has proposed an estimator – Good-Turing (GT) – for the missing mass in a sample, i.e., the probability mass function of the new species from a population to be discovered in the next

sample – estimated as the proportion of species encountered only once (a.k.a., *hapaxes*). The work of [26] studied the convergence rate for the GT estimator for the missing mass. In [6], the authors have used the GT estimator for estimating the expert's probability of identifying new interesting items from an underlying population. In a similar vein, [21] used the GT estimator for the potential new activations to be seen in online influence maximization campaigns. In this work, the influencer's diminishing reward is modelled as a function of the number of selections $\gamma(n_{k,t})$, and applied it as a modifier to the GT estimator (denoted Fat-GT, for Fatigue-aware Transformation of the Good-Turing estimator):

$$\hat{R}_{k,h} = \frac{1}{n_{k,h}} \sum_{j \in A_k} U^{\gamma}_{n_{k,h}}(j) \tag{5}$$

where

$$U^{\gamma}_{n_{k,h}}(j) = \sum_{1 \le s \le n_{k,h}} \mathbb{I}\{X_{k,1}(j) = \cdots = X_{k,s-1}(j) = X_{k,s+1}(j) = \cdots = X_{k,n_{k,h}}(j) = 0, X_{k,s}(j) = 1\} \times \frac{\gamma(n_{k,h}+1)}{\gamma(s)},$$

while $n_{k,h}$ is the number of selections of influencer k at round h, A_k is the set of basic nodes reachable by influencer k, $\gamma(n_{k,h})$ is the fatigue function (e.g., $\frac{1}{n_{k,h}}$), $X_{k,s}(j)$ is the i.i.d. random variable equal to 1 if influencer k activates basic node j at round s.

In short, the influencer k's estimated remaining potential $\hat{R}_{k,h}$ at round h is the average number of discounted hapaxes.

In [15] adapted the Good-Turing estimator to new activations, in a scenario where context information would be available at the beginning of each round in the campaign. Namely, the modifier to the GT estimator is replaced by a function of both the influencer's number of selections and the round's context: $\gamma(\langle \hat{\theta}_{k,t}, Y_t \rangle, n_{k,t})$. The influencer's potential within a given context is assumed to be well-represented by the scalar product of the round's context Y_t and an estimated unknown quantity $\hat{\theta}_{k,t}$ for that influencer.

4.2 LSVI-GT-UCB

Motivated by the potential gain from using available historical information when choosing actions, we propose the novel algorithm `LSVI-GT-UCB`[1].

The state of each influencer's MDP is composed by concatenating the context given by the environment at the beginning of each step in the horizon $Y_{t,h} \in \mathbb{R}^d$, and the reward received by the respective influencer upon its previous selection within the current episode $r_{k,t,n_{k,t,h}}$.

At a high level, `LSVI-GT-UCB` combines the LSVI algorithm of [17], based on linear regression estimation, to which we add the Good-Turing estimator approach. For each influencer we have an MDP $(\mathcal{S}_k, \mathcal{A}_k, H, \mathbb{P}_k, r_k)$, assumed to be linear via a feature map $\phi_k : \mathcal{S}_k \times \mathcal{A}_k \to \mathbb{R}^d$ [5, 17, 27]. Since each influencer has their own MDP, their action set is binary, i.e. $\mathcal{A}_k = \{0, 1\}$.

The linear regression data is created for each influencer, based on its historical selections and rewards:

$$y_{k,\tau,h} = r_{k,\tau,n_{k,\tau,h}} + \hat{V}^{LSVI}_{k,t,h+1}(s_{k,\tau,h+1}), \tau \in [1, t], \tag{6}$$

[1] https://github.com/AlexandraIacob/lsvi_gt_ucb

where $\hat{V}^{LSVI}_{k,t,h+1}(\cdot) = \max_{a \in \mathcal{A}_k} \hat{Q}^{LSVI}_{k,t,h+1}(\cdot, a)$. The data is used in the linear regression estimator to which an UCB bound is added:

$$\hat{Q}^{LSVI}_{k,t,h}(\cdot,\cdot) = \langle \phi_k(\cdot,\cdot), \hat{\theta}_{k,t,h} \rangle + \zeta \sqrt{\phi_k(\cdot,\cdot)^T \Sigma^{-1}_{k,t,h} \phi_k(\cdot,\cdot)}, \text{where}$$
$$\hat{\theta}_{k,t,h} = \Sigma^{-1}_{k,t,h} \sum_{\tau=1}^{t-1} \phi_k(s_{k,\tau,h}, a_{k,\tau,h}) y_{k,\tau,h},$$
$$\Sigma_{k,t,h} = \eta \cdot I_d + \sum_{\tau=1}^{t-1} \phi_k(s_{k,\tau,h}, a_{k,\tau,h}) \phi_k(s_{k,\tau,h}, a_{k,\tau,h})^T, \tag{7}$$

where $\zeta = cdH\sqrt{\log(\frac{2dTH}{\delta})}$ as in [17][Theorem 3.1], with an absolute constant $c > 0$, ensures with probability $1 - \delta$ a total regret of $O\left(\sqrt{d^3H^4T\log^2(\frac{2dTH}{\delta})}\right)$, and the penalty factor η ensures a unique minimizer for the regularized least-squares estimator.

In parallel, the Good-Turing estimator for the remaining potential is computed for each step in each influencer's MDP as well. The Fat-GT estimator for the remaining potential, previously introduced in Equation (5), is adapted to Problem 2 by maintaining an independent estimator for each episode, as follows:

$$\hat{R}_{k,t,h} = \frac{1}{n_{k,t,h}} \sum_{j \in A_k} U^{\gamma}_{n_{k,t,h}}(j), \text{where}$$
$$U^{\gamma}_{n_{k,t,h}}(j) = \sum_{1 \le i \le n_{k,t,h}} \mathbb{I}\{X_{k,t,1}(j) = \cdots = X_{k,t,h}(j) = 0, X_{k,t,i}(j) = 1\} \frac{\gamma(n_{k,t,h}+1)}{\gamma(i)}, \tag{8}$$

with its respective confidence bound index given by [21][Th. C.2]:

$$\beta_{k,t,h} = (1+\sqrt{2})\sqrt{\frac{\hat{\lambda}_{k,t,h}\log 4h}{n_{k,t,h}}} + \frac{\log 4h}{3n_{k,t,h}}, \text{where}$$
$$\hat{\lambda}_{k,t,h} = \frac{\gamma(n_{k,t,h}+1)}{n_{k,t,h}} \sum_{s=1}^{n_{k,t,h}} \frac{r_{k,t,s}}{\gamma(s)}. \tag{9}$$

Furthermore, in order to learn from historical data, an average of the Fat-GT estimators for the given step over the previous and current episodes is computed, as follows:

$$\hat{Q}^{GT}_{k,t,h} = \frac{1}{t} \sum_{\tau=1}^{t} \hat{R}_{k,\tau,h}, \tag{10}$$

implementing our interpretation that the state-action value function (Q-function) in an MDP is the influencer's remaining potential:

$$Q^{GT}_{k,t,h} = R_{k,t,h}$$
$$= \sum_{j \in A_k} \mathbb{I}\left\{ j \notin \bigcup_{i=1}^{h} S(I_{t,i}, Y_{t,i}) \right\} \gamma(n_{k,t,h}+1) p_{k,j}(t,h). \tag{11}$$

We derive the optimism bonus for the Q-function estimator in the following theorem:

Theorem 4.1. *With probability at least $1-\delta$, for $\lambda_{k,t,h} = \gamma(n_{k,t,h}) \sum_{j \in A_k} p(j)$ and $\beta_{k,t,h} = (1+\sqrt{2})\sqrt{\frac{\lambda_{k,t,h+1}\log 4/\delta}{n_{k,t,h}}} + \frac{1}{3n_{k,t,h}}\log\frac{4}{\delta}$, the*

following holds:

$$\frac{-1}{t}\sum_{\tau=1}^{t}\left(\beta_{k,\tau,h}+\frac{(n_{k,\tau,h}+1)\lambda_{k,\tau,h}}{n_{k,\tau,h}}\right)\leq Q^{GT}_{k,t,h}-\hat{Q}^{GT}_{k,t,h}$$
$$\leq \lambda_{k,t,h}+\frac{1}{t}\sum_{\tau=1}^{t}\beta_{k,\tau,h}.$$

Proof. Estimating the Q-function with the averaged Fat-GT estimators as in Eq. (10), the estimator's confidence interval is:

$$Q^{GT}_{k,t,h}-\hat{Q}^{GT}_{k,t,h}=R_{k,t,h}-\frac{1}{t}\sum_{\tau=1}^{t}\hat{R}_{k,\tau,h}.$$

We know from [21][Theorem C.2] that:

$$-\beta_{k,\tau,h}-\frac{\lambda_{k,\tau,h}}{n_{k,\tau,h}}\leq R_{k,\tau,h}-\hat{R}_{k,\tau,h}\leq\beta_{k,\tau,h},\forall\tau\in[1,t].$$

Aggregating the confidence bounds from all episodes, we obtain:

$$\frac{-1}{t}\sum_{\tau=1}^{t}\left(\beta_{k,\tau,h}+\frac{\lambda_{k,\tau,h}}{n_{k,\tau,h}}\right)\leq\frac{1}{t}\sum_{\tau=1}^{t}\left(R_{k,\tau,h}-\hat{R}_{k,\tau,h}\right)\leq\frac{1}{t}\sum_{\tau=1}^{t}\beta_{k,\tau,h}$$
$$\Leftrightarrow\frac{-1}{t}\sum_{\tau=1}^{t}\left(\beta_{k,\tau,h}+\frac{\lambda_{k,\tau,h}}{n_{k,\tau,h}}\right)+R_{k,t,h}-\frac{1}{t}\sum_{\tau=1}^{t}R_{k,\tau,h}$$
$$\leq R_{k,t,h}-\frac{1}{t}\sum_{\tau=1}^{t}\hat{R}_{k,\tau,h}\leq R_{k,t,h}-\frac{1}{t}\sum_{\tau=1}^{t}R_{k,\tau,h}+\frac{1}{t}\sum_{\tau=1}^{t}\beta_{k,\tau,h}.$$

Having the remaining potential of influencer k at round h of episode t defined as in Equation (11), we can obtain that:

$$0\leq R_{k,t,h}\leq\lambda_{k,t,h}$$
$$\Leftrightarrow\frac{-1}{t}\sum_{\tau=1}^{t}\lambda_{k,\tau,h}\leq R_{k,t,h}-\frac{1}{t}\sum_{\tau=1}^{t}R_{k,\tau,h}\leq\lambda_{k,t,h}.$$

Therefore,

$$\frac{-1}{t}\sum_{\tau=1}^{t}\left(\beta_{k,\tau,h}+\frac{(n_{k,\tau,h}+1)\lambda_{k,\tau,h}}{n_{k,\tau,h}}\right)\leq Q^{GT}_{k,t,h}-\hat{Q}^{GT}_{k,t,h}$$
$$\leq\lambda_{k,t,h}+\frac{1}{t}\sum_{\tau=1}^{t}\beta_{k,\tau,h}.$$

which concludes our proof. □

The Q-function is finally estimated by optimistically choosing from the linear regression-based Q estimator and the GT estimators for each potential influencer:

$$\hat{Q}_{k,t,h}(\cdot,\cdot)=\max\left\{\hat{Q}^{LSVI}_{k,t,h}(\cdot,\cdot),\hat{Q}^{GT}_{k,t,h}+\lambda_{k,t,h}+\frac{1}{t}\sum_{\tau=1}^{t}\beta_{k,\tau,h}\right\}. \tag{12}$$

The L influencers having the highest value of $\hat{Q}_{k,t,h}(\cdot,\cdot)$ are then chosen. This is outlined in Algorithm 1, as described next.

We assume that the environment provides a context at the beginning of each step, and each MDP's state is computed by concatenating (i) the influencer's number of selections, and (ii) the reward resulting from playing that influencer at its last selection. `LSVI-GT-UCB` starts the first episode by playing each influencer once in order to gather initial information. Then, for the following steps, it proceeds by computing for each influencer k the two Q-function estimators: $\hat{Q}^{LSVI}_{k,t,h}$ and $\hat{Q}^{GT}_{k,t,h}$. The former is computed using regularized least squares, as in [17]. The latter is computed with the formula from Equation (10), with its optimism bonus given by Theorem 4.1. Finally, the learning agent chooses to play the L influencers with the highest estimated Q-functions, observes the reward, and updates the statistics.

`LSVI-GT-UCB` learns *in parallel* the linear and the GT estimators from feedback collected by either of them, and chooses its action with optimism not only from the estimator's UCB, but also from the highest estimated remaining potential with either method.

Algorithm 1 LSVI-GT-UCB

Input: Number of influencers L per round, penalty factor η, the ambient dimension d of the feature space, feature maps ϕ_k.
Initialize $\hat{Q}_{k,1,h}(s_{k,t,h},a_{k,t,h})=0,\forall(s_{k,t,h},a_{k,t,h})\in\mathcal{S}_k\times\mathcal{A}_k$.
for episodes $t=1,\ldots,T$ **do**
  if first episode $t=1$ **then**
    for step $k=1,\ldots,K$ **do**
      Receive the arbitrary context $Y_{t,h}$,and create the state $s_{k,t,h}=[Y_{t,h}|0|0]$.
      Play influencer k.
      Observe rewards $r_{t,h}$, and next state $s_{k,t,h+1}$.
      Update $n_{k,t,h}=1,\Sigma_{k,t,h}=\eta\cdot I_d+\phi_k(s_{k,t,h},1)\phi_k(s_{k,t,h},1)^T$.
    end for
  else
    for step $h=H,\ldots,1$ (or until K, for the first episode) **do**
      Receive the arbitrary context $Y_{t,h}$.
      for influencer $k=1,\ldots,K$ **do**
        Create the state $s_{k,t,h}=[Y_{t,h}|n_{k,t,h}|r_{k,t,n_{k,t,h}}]$, and set $\hat{\theta}_{k,t,H+1}=\mathbf{0}$,
        $\hat{Q}^{LSVI}_{k,t,H+1}(s_{k,t,h},a_{k,t,h})=\langle\phi_k(s_{k,t,h},a_{k,t,h}),\hat{\theta}_{k,t,H+1}\rangle=0$.
        Compute the linear regression data $y_{k,\tau,h},\forall\tau\in[1,t]$ (as in Equation (6)).
        Calculate the regularized least-squares estimator (as in Equation (7)).
      end for
    end for
    for step $h=1,\ldots,H$ **do**
      for influencer $k=1,\ldots,K$ **do**
        Compute Fat-GT estimator $\hat{Q}^{GT}_{k,t,h}$ as in Equation (10).
        Compute the influencer k's Q-function estimator (as in Equation (12)).
      end for
      Play action $a_{t,h}$ made of the L influencers with the highest estimated Q-functions $\hat{Q}_{k,t,h}$.
      Update $n_{k,t,h}=n_{k,t,h-1}+1,\forall a_{t,h}[k]=1$.
      Observe reward $r_{t,h}$, and next states $s_{k,t,h+1}$.
    end for
  end if
end for

Regret. The performance guarantee of our algorithm is directly linked to the regret bound of `LSVI-UCB` given by Th. 3.1 in [17], i.e. $O(\sqrt{d^3H^4T\log^2\left(\frac{2dTH}{\delta}\right)})$ - where we replaced the original t by the total number of rounds TH from our setting. Given that this bound

Table 2: Data statistics.

Net.	#Users	#Edges	#Orig.-messages	#Retweets
Weibo	1.8M	308M	300K	23.8M
Twitter	11.6M	309M	242M	341.8M

is super-linear in H, it dominates the estimator bound that we find in our paper for the Good-Turing estimation. Hence the regret we have is comparable to the one found for `LSVI-UCB` in [17].

5 EXPERIMENTS

The algorithm `LSVI-GT-UCB` – which solves Problem 2 – is tested on three datasets: one consisting of synthetically generated data, and two consisting of real-world data. Its performance is compared to `LSVI-UCB` [17], and `Fat-GT-UCB` [21]. `LSVI-UCB` is originally designed to solve use upper confidence bounds for linear function approximation MDPs, and can be adapted to Problem 2 by equating an episode with a campaign. `Fat-GT-UCB`, on the othe hand, is run independently between episodes. As a comparison metric, we use the total sum of cumulative rewards over all campaigns. In addition to these two state-of-the-art solutions, we can also design two other baselines: `RL-Fat-GT-UCB` is created by adapting `Fat-GT-UCB` to learn from the previous episodes by averaging the GT estimators over the episode, for each step in the horizon; and `LSVI-UCB -separate thetas`, created by adapting `LSVI-UCB` to learn an estimator per influencer. This modification enables the combination of the estimator of the Q-function with the GT estimator in order to estimate an influencer's remaining potential.

5.1 Synthetic data

Experiments are run on a synthetic dataset in the following way. First, a graph is generated using the Albert-Barabàsi model [4] for 30,000 nodes, and each influencer is chosen using their degree, i.e., the K highest degrees in the graph. Then, the activation probability of each node attached to these influencers is computed using a sigmoid function of the scalar product of the randomly generated context and a randomly chosen feature vector, specific to the node. For each dimension, the feature vectors are sampled from a normal distribution $\mathcal{N}(1,3)$, and the contexts are sampled from another normal distribution $\mathcal{N}(1,0.1)$. The variance in the distribution of user profiles is greater than that in the distribution of contexts to mark the greater difference that can appear between users compared to differences between messages from a campaign. The diffusion model is assumed to be the Independent Cascade [19].

5.2 Real-world datasets

Experiments are also run on real-world datasets from the two major micro-blogging applications, Twitter [15] and Sina Weibo [34]. The datasets' main statistics are presented in Table 2.

The *Weibo* dataset containts a log of posts (equivalent to tweets) with each posts's text being encoded as a distribution over 100 topics computed using Latent Dirichlet Allocation. The dataset contains the topic distribution for each post, the reposting logs containing the list of unique users which had reposted the post, and information about the original author. We processed and merged this information in order to obtain a file containing the original post, its author / the influencer's Id, the set of basic user Ids which reposted that message, and the topic distribution for the message. The set of influencers is found by taking the ones having the highest number of reposts, and all the tweets of the other influencers are filtered out. During the experiments, random contexts, i.e. topic distributions, are chosen for each round from all the available contexts in the dataset. At the beginning of each round the context is provided by the environment, an algorithm chooses the influencer(s), and a post for the pair (influencer Id, context) is sampled from the log. The *new activations* are given by discounting the previously seen basic user Ids from the sampled post's set of user Ids.

The *Twitter* dataset is created from a crawled dataset from Twitter with tweets from August 2012. We have extracted, using K-means, 24 centroids from the *glove-twitter-200*[2] vocabulary, and each tweet's text was processed by encoding each word and replacing it with its nearest centroid. The original tweet text is then replaced by a distribution over 24 centroids. Each tweet contains a set of node Ids representing the users who retweeted it. The set of influencers, as in the case of Sina Weibo, are chosen as the ones containing the highest number of retweeting users. The experimental simulation process is the same as the one described for Sina Weibo.

5.3 Results

The results of the experiments are averaged over 50 runs, and the algorithms are run for 50 episodes each with a horizon of 30 rounds, i.e., 1500 rounds in the end. For all the LSVI-based algorithms the exploration factor is $\beta = c \cdot dH\sqrt{\log(\frac{2dTH}{\delta})}$, where $c > 0$ is an absolute constant, $T = 50$ is the number of episodes, $H = 30$ is the number of steps, d is the dimension of the feature space. For Sina Weibo and the synthetic dataset, an absolute constant $c = 1$ performs well. However, for the Twitter dataset, we had to choose a much smaller absolute constant, $c = 0.0005$, to have an exploration factor suitable for the scale of the reward. The dimension of the feature space d depends on the how the state is constructed for each algorithm. This follows the theoretical results of Theorem 3.1 in [17], with probability set as $1 - \delta = 0.99$.

On the synthetic dataset (Figure 1), we see that our algorithm outperforms the baseline methods for $L = 1$ and $L = 2$. For $L = 5$ the LSVI-based algorithm with an estimator for each influencer is stronger than the Good-Turing-based estimator. We witness a saturation of the rewards when more influencers are chosen, which explains the periodical flattening in the graphs of the reward functions. The large final cumulative rewards are possible due to the reset of counting the new activations at the start of each episode.

The results of Sina Weibo – Figure 2 – show that, in terms of cumulative reward, `LSVI-GT-UCB` outperforms the other methods for $L = 2$ and $L = 5$ especially. For $L = 1$ it is competitive with `RL-Fat-GT-UCB`, but it exhibits much lower variance in the rewards, making it a more reasonable choice in practice.

On the other hand, in the Twitter dataset (Figure 3), our algorithm clearly outperforms the other algorithms for $L = 1$, which from a theoretical point of view remains the main case (one decision per round). For $L = 2$ and $L = 5$, the results of several algorithms

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

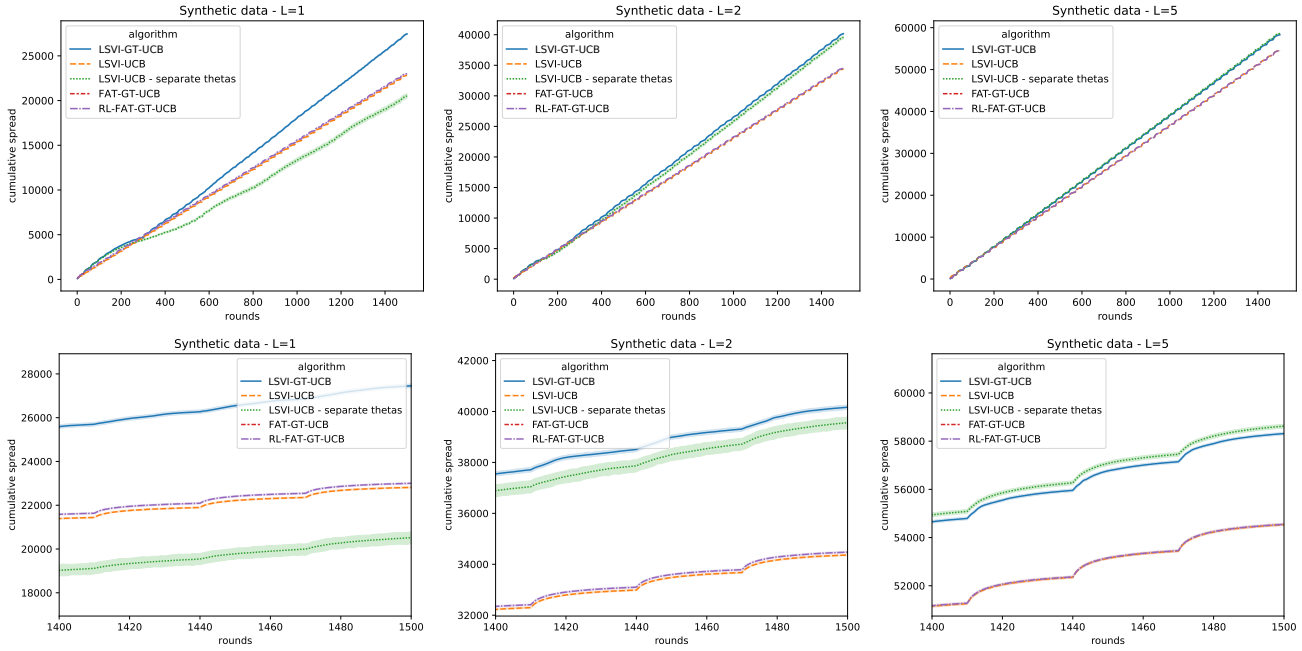

Figure 1: Synthetic - Cumulative rewards, full plot (top row), plot zoomed to last 100 rounds (bottom row).

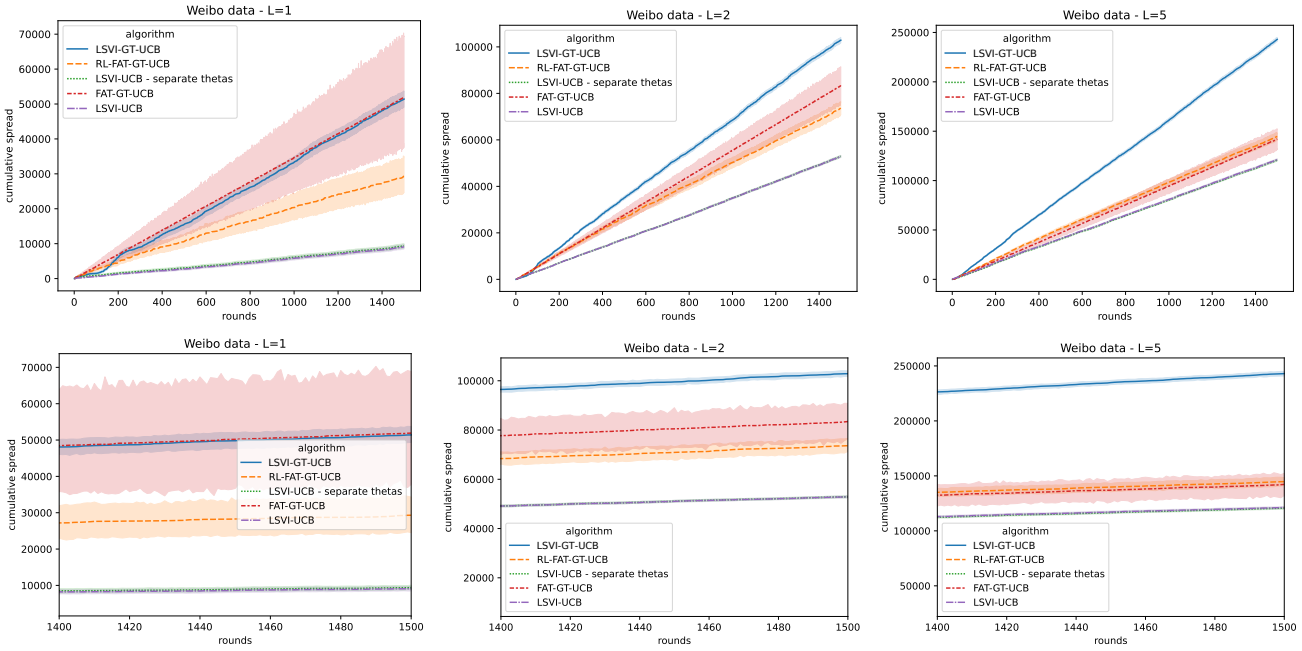

Figure 2: Sina Weibo - Cumulative rewards, full plot (top row), plot zoomed to last 100 rounds (bottom row).

– including ours – are very close, with low variance. This may indicate that the dataset characteristics lead to a saturation of the rewards when more influencers are chosen at each round. The algorithms `Fat-GT-UCB` and `RL-Fat-GT-UCB` perform similarly.

Running time. We measured the running time for all the algorithms. `LSVI-GT-UCB` and `LSVI-UCB - separate thetas` take approximately 16 minutes / round with a standard deviation of about 3 seconds / round. `LSVI-UCB` is on average K times faster than the previous two approaches: about 2 minutes / round with a standard

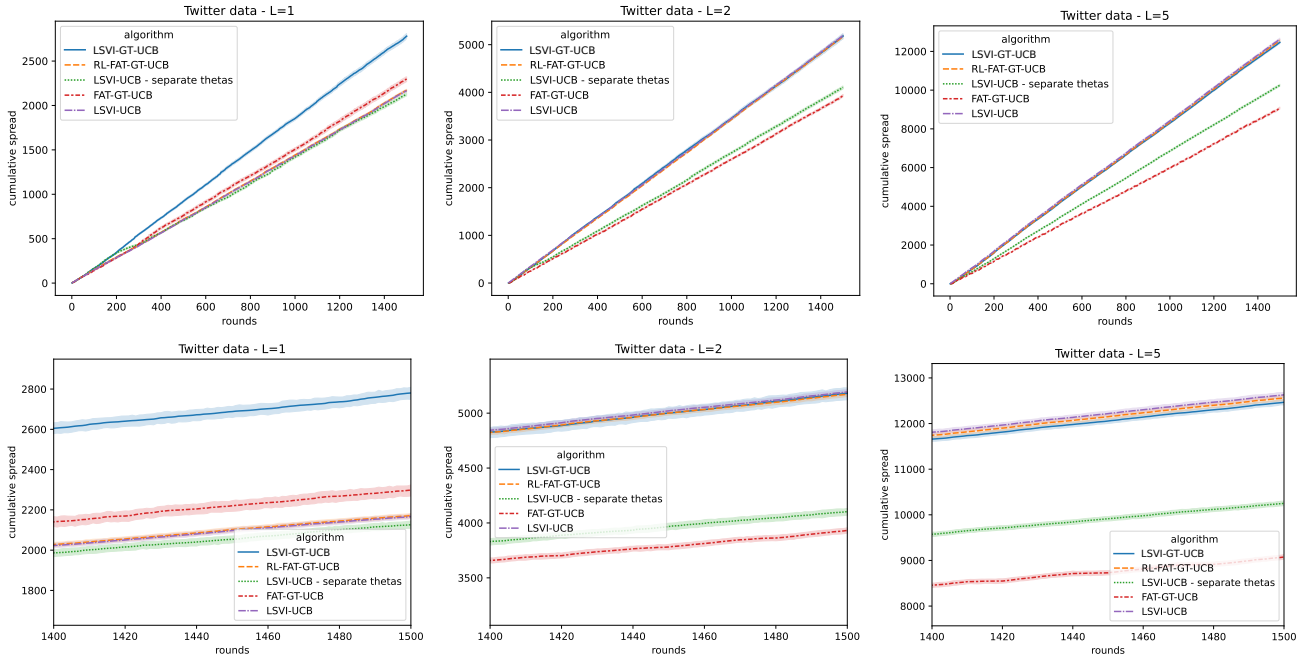

Figure 3: Twitter - Cumulative rewards, full plot (top row), plot zoomed to last 100 rounds (bottom row).

deviation of 2 seconds / round. This is because `LSVI-GT-UCB` and `LSVI-UCB - separate thetas` estimate a Q-function per influencer (lines 15-22 in Alg. 1), instead of a single, global one. All the Good-Turing estimations have negligible execution cost (at most 10 secondes / round) – updating the average of hapaxes takes at most a few seconds. The main bottleneck of LSVI-based approaches is the regression steps occurring $T \times H$ times for each of the K influencers/arms, for a total cost of the order of $O(Kd^2(TH + d)$ if we use matrix inversions. The linear dependence on the TH steps and the K influencers is unavoidable, and the main issue here is the dimension d of the feature map. As a side note, whenever the dimension d of the feature map may lead to unacceptable computation time for the application scenario, one way to reduce it would be via embedding approaches. However, we believe running time is somewhat inconsequential in practice for the approaches considered here, since the feedback delay – in our case, gathering the influence lists – should dominate the computation step.

6 CONCLUSION

We introduce and study in this paper a novel online influence maximization problem (ECIMP), which is directly motivated by the real-world scenarios of information diffusion where (i) the diffusion medium is highly uncertain and only a few influencer nodes may be known in advance, (ii) only a target user's *first* activation is of interest (e.g., for political endorsements or subscriptions to a media service), and (iii) valuable side-information may be available, allowing the agent to learn effectively *within* and *across* campaigns. By its focus, we believe our work reduces further the gap between theoretical studies and practical deployments. Indeed, we connect some of the main working assumptions that have been considered for practical purposes in the recent IM literature, such as an unknown diffusion medium and a bipartite influencer-influencee graph abstraction thereof, spread over multiple rounds in a campaign, context-dependent diffusion, and the number of distinct activations as the objective function. Furthermore, the ability to exploit context and correlations across campaigns is of obvious interest, and justifies our reinforcement learning solution.

For this problem, we presented a novel algorithm, `LSVI-GT-UCB`, which brings together for the first time the LSVI approach for linear function approximation and the Good-Turing estimator used in multi-armed bandits for estimating missing mass – with an application to contextual influence maximization over multiple influence campaigns. `LSVI-GT-UCB` runs the two estimators in parallel, and makes the optimistic choice between them when deciding which influencers to select at each step of the campaign – thus implementing the *optimism in the face of uncertainty* principle. The experimental study, performed on two real-world datasets – Sina Weibo and Twitter – and a synthetically generated one, shows that `LSVI-GT-UCB` is competitive to state-of-the-art baselines, and is less susceptible to noise while allowing learning over multiple campaigns.

ACKNOWLEDGEMENTS

DesCartes: this research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) program.

REFERENCES

[1] Alekh Agarwal, Daniel Hsu, Satyen Kale, John Langford, Lihong Li, and Robert Schapire. 2014. Taming the Monster: A Fast and Simple Algorithm for Contextual Bandits. In *Proceedings of the 31st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 32)*. 1638–1646.

[2] Noga Alon, Iftah Gamzu, and Moshe Tennenholtz. 2012. Optimizing budget allocation among channels and influencers. In *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*. 381–388. https://doi.org/10.1145/2187836.2187888

[3] Cigdem Aslay, Nicola Barbieri, Francesco Bonchi, and Ricardo Baeza-Yates. 2014. Online Topic-aware Influence Maximization Queries.. In *EDBT*. 295–306.

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

[5] Steven J Bradtke and Andrew G Barto. 1996. Linear least-squares algorithms for temporal difference learning. *Machine learning* 22, 1 (1996), 33–57.

[6] Sébastien Bubeck, Damien Ernst, and Aurélien Garivier. 2013. Optimal Discovery with Probabilistic Expert Advice: Finite Time Analysis and Macroscopic Optimality. *J. Mach. Learn. Res.* 1 (2013), 601–623.

[7] Shuo Chen, Ju Fan, Guoliang Li, Jianhua Feng, Kian-lee Tan, and Jinhui Tang. 2015. Online topic-aware influence maximization. *Proceedings of the VLDB Endowment* 8, 6 (2015), 666–677.

[8] Wei Chen, Chi Wang, and Yajun Wang. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1029–1038.

[9] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. 2011. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. JMLR Workshop and Conference Proceedings, 208–214.

[10] Hanjun Dai, Bo Dai, and Le Song. 2016. Discriminative embeddings of latent variable models for structured data. In *International conference on machine learning*. PMLR, 2702–2711.

[11] Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. 2010. Parametric bandits: The generalized linear case. *Advances in Neural Information Processing Systems* 23 (2010).

[12] Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. 2010. Parametric Bandits: The Generalized Linear Case. In *Advances in Neural Information Processing Systems 23*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta (Eds.).

[13] Irving J Good. 1953. The population frequencies of species and the estimation of population parameters. *Biometrika* (1953).

[14] Jing Guo, Peng Zhang, Chuan Zhou, Yanan Cao, and Li Guo. 2013. Personalized influence maximization on social networks. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*. 199–208.

[15] Alexandra Iacob, Bogdan Cautis, and Silviu Maniu. 2022. Contextual Bandits for Advertising Campaigns: A Diffusion-Model Independent Approach. In *Proceedings of the 2022 SIAM International Conference on Data Mining, SDM 2022, Alexandria, VA, USA, April 28-30, 2022*. 513–521.

[16] Mohammad Towhidul Islam, Mursalin Akon, Atef Lotfy Abdrabou, and Xuemin Shen. 2011. Modeling Epidemic Data Diffusion for Wireless Mobile Networks. In *2011 IEEE Global Telecommunications Conference - GLOBECOM 2011*. 1–5. https://doi.org/10.1109/GLOCOM.2011.6134272

[17] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. 2020. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory (CoLT)*. PMLR, 2137–2143.

[18] Richard M Karp. 1972. Reducibility among combinatorial problems. In *Complexity of computer computations*. Springer, 85–103.

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

[20] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. 2017. Learning combinatorial optimization algorithms over graphs. *Advances in neural information processing systems* 30 (2017).

[21] Paul Lagrée, Olivier Cappé, Bogdan Cautis, and Silviu Maniu. 2018. Algorithms for online influencer marketing. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 13, 1 (2018), 1–30.

[22] Tor Lattimore and Csaba Szepesvári. 2020. *Bandit algorithms*. Cambridge University Press.

[23] Siyu Lei, Silviu Maniu, Luyi Mo, Reynold Cheng, and Pierre Senellart. 2015. Online Influence Maximization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*. 645–654. https://doi.org/10.1145/2783258.2783271

[24] Hui Li, Mengting Xu, Sourav S. Bhowmick, Joty Shafiq Rayhan, Changsheng Sun, and Jiangtao Cui. 2022. PIANO: Influence Maximization Meets Deep Reinforcement Learning. *IEEE Transactions on Computational Social Systems* (2022).

[25] Shuai Li, Fang Kong, Kejie Tang, Qizhi Li, and Wei Chen. 2020. Online influence maximization under linear threshold model. *Advances in Neural Information Processing Systems* 33 (2020), 1192–1204.

[26] David A McAllester and Robert E Schapire. 2000. On the Convergence Rate of Good-Turing Estimators.. In *COLT*. 1–6.

[27] Francisco S Melo and M Isabel Ribeiro. 2007. Q-learning with linear function approximation. In *International Conference on Computational Learning Theory*. Springer, 308–322.

[28] Miguel Romero and Luis Gallego. 2017. Analysis of voltage sags propagation in distribution grids using a SI epidemic model. 1–6. https://doi.org/10.1109/PEPQA.2017.7981685

[29] Xiaodan Song, Belle L Tseng, Ching-Yung Lin, and Ming-Ting Sun. 2006. Personalized recommendation driven by information flow. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. 509–516.

[30] Shaojie Tang and Jing Yuan. 2016. Optimizing ad allocation in social advertising. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 1383–1392.

[31] Ruosong Wang, Simon S Du, Lin Yang, and Russ R Salakhutdinov. 2020. On reward-free reinforcement learning with linear function approximation. *Advances in neural information processing systems* 33 (2020), 17816–17826.

[32] Yining Wang, Ruosong Wang, Simon S Du, and Akshay Krishnamurthy. 2019. Optimism in reinforcement learning with generalized linear function approximation. *arXiv preprint arXiv:1912.04136* (2019).

[33] Zheng Wen, Branislav Kveton, Michal Valko, and Sharan Vaswani. 2017. Online Influence Maximization under Independent Cascade Model with Semi-Bandit Feedback. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/7137debd45ae4d0ab9aa953017286b20-Paper.pdf

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

[35] Zhijie Zhang, Wei Chen, Xiaoming Sun, and Jialin Zhang. 2022. Online influence maximization with node-level feedback using standard offline oracles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 9153–9161.