Effective Large-Scale Online Influence Maximization

Paul Lagrée*, Olivier Cappé†, Bogdan Cautis*‡ and Silviu Maniu*

*LRI, Université Paris-Sud, Université Paris-Saclay
†LIMSI, CNRS, Université Paris-Saclay
‡Huawei Noah's Ark Lab, Hong Kong
{paul.lagree, bogdan.cautis, silviu.maniu}@u-psud.fr, cappe@universite-paris-saclay.fr

Abstract—In this paper, we study a highly generic version of influence maximization (IM), one of optimizing influence campaigns by sequentially selecting "spread seeds" from a set of *candidates*, a small subset of the node population, under the hypothesis that, in a given campaign, previously activated nodes remain "persistently" active throughout and thus do not yield further rewards. We call this problem *online influence maximization with persistence*. We introduce an estimator on the candidates' *missing mass* – the expected number of nodes that can still be reached from a given seed candidate – and justify its strength to rapidly estimate the desired value. We then describe a novel algorithm, GT-UCB, relying on upper confidence bounds on the missing mass. We show that our approach leads to high-quality spreads on classic IM datasets, even though it makes almost no assumptions on the diffusion medium. Importantly, it is orders of magnitude faster than state-of-the-art IM methods.

Keywords-Influence maximization; information diffusion; online social networks; online learning; multi-armed bandits

I. INTRODUCTION

Advertising based on word-of-mouth diffusion in social media has become very important in the digital marketing landscape. Nowadays, social value and social influence are arguably the hottest concepts in the area of Web advertising and most companies that advertise in the Web space must have a "social" strategy. For example, on widely used platforms such as Facebook or Twitter, promoted posts are interleaved with normal posts on user feeds. Users interact with these posts by actions such as "likes" (adoption), "shares" or "reposts" (network diffusion). This represents an unprecedented tool in advertising, as products, news, ideas, movies, etc, can propagate easily to a large audience [1].

Motivated by the need for effective viral marketing strategies, *influence estimation* and *influence maximization* (IM) have become important research problems, at the intersection of data mining and social sciences [2]. In short, IM is the problem of selecting a set of nodes from a given diffusion graph, maximizing the expected spread under an underlying diffusion model. This problem was introduced in 2003 by the seminal work of Kempe et al. [3], through two stochastic, discrete-time diffusion models, *Linear Threshold* (LT) and *Independent Cascade* (IC). These models rely on diffusion graphs whose edges are weighted by a score of influence. They show that selecting the set of nodes maximizing the expected spread is NP-hard for both models,

and they propose a greedy algorithm that takes advantage of the sub-modularity property of the influence spread, but does not scale to large graphs. A rich literature followed, focusing on computationally efficient and scalable algorithms to solve IM. The recent benchmarking study of Arora et al. [4] summarizes state-of-the-art techniques and also debunks many IM myths.

Importantly, all the IM studies discussed in [4] have as starting point a specific diffusion model (IC or LT), whose graph topology and parameters – basically the edge weights – are known. In order to *infer* the diffusion parameters or the underlying graph structure, or both, [5], [6], [7], [8] propose *offline, model-specific methods*, which rely on observed information cascades.

There are however many situations where it is unreasonable to assume the existence of relevant historical data in the form of cascades. For such settings, *online approaches*, which can learn the underlying diffusion parameters *while running diffusion campaigns*, have been proposed. Bridging IM and inference, this is done by balancing between exploration steps (of yet uncertain model aspects) and exploitation ones (of the best solution so far), by so called *multi-armed bandits* techniques, where an agent interacts with the network to infer influence probabilities [9], [10], [11], [12]. The learning agent sequentially selects seeds from which diffusion processes are initiated in the network; the obtained feedback is used to update the agent's knowledge of the model.

Nevertheless, all these studies on inferring diffusion networks, whether offline or online, rely on parametric diffusion models, i.e., assume that the actual diffusion dynamics are well captured by such a model (e.g., IC). This maintains significant limitations for practical purposes. First, the more complex the model, the harder to learn in large networks, especially in campaigns that have a relatively short timespan, making model inference and parameter estimation very challenging within a small horizon (typically tens or hundreds of spreads). Second, it is commonly agreed that the aforementioned diffusion models represent elegant yet coarse interpretations of a reality that is much more complex and often hard to observe fully. For examples of insights into this complex reality, the *topical* or *non-topical* nature of an influence campaign, the *popularity* of the piece of information being diffused, or its specific *topic* were all shown to have a significant impact on hashtag diffusions in

IEEE
computer society

Twitter [8], [13], [14].

Aiming to address such limitations, we propose in this paper a *large-scale approach for online and adaptive IM*, in which the underlying assumptions for the diffusion processes are kept to a minimum. We argue that it can represent a versatile tool in many practical scenarios. More precisely, we focus on social media diffusion scenarios in which influence campaigns consist of multiple *consecutive trials* (or *rounds*) spreading the same type of information from an arbitrary domain (be it a product, idea, post, hashtag, etc). The goal of each campaign is to reach (or *activate*) as many distinct users as possible, the objective function being the total spread. In our setting, the campaign selects from a set of *spread seed candidates*, a small subset of a potentially large and unknown population. At each round, the learning agent picks among the candidates those from which a new diffusion process is initiated in the network, gathers some feedback on the activations, and adapts the subsequent steps of the campaign; the agent may "re-seed" certain nodes (we may want to ask a particular node to initiate spreads several times, e.g., if it has a strong *convertion impact*). This perspective on influence campaigns imposes naturally a certain notion of *persistence*, which is given the following interpretation: users that were already activated in the ongoing campaign – e.g., have adopted the product or endorsed the political candidate – remain activated throughout that campaign, and thus will not be accounted for more than once in the objective function.

We call this problem *online influence maximization with persistence* (in short, OIMP). Our solution for it follows the multi-armed bandit idea initially employed in Lei et al. [15], but we adopt instead a *diffusion-independent perspective*, whose only input are the spread seed candidates, while the population and underlying diffusion network – which may actually be the superposition of several networks – remain unknown. In our bandit approach, the parameters to be estimated are the values of the candidates – how good is a specific candidate –, as opposed to the diffusion edge probabilities of a known graph as in [15]. Furthermore, we assume that different campaigns are independent, and make the model's feedback more realistic: after each trial, the agent only gathers the set of activated nodes. The rationale is that oftentimes, for a given "viral" item, we can track in applications only *when* it was adopted by various users, but not *why*. A key difference w.r.t. other multi-armed bandit studies for IM such as [9], [10], [11], [12] is that these look for a *constant* optimal set of seeds, while the difficulty with OIMP is that the seemingly best action at a given trial depends on the activations of the previous trials (and thus the learning agent's past decisions).

The multi-armed bandit algorithm we propose, called GT-UCB, relies on a famous statistical tool known as the *Good-Turing estimator*, first developed during WWII to crack the Enigma machine, and later published by Good in a study on species discovery [16]. Our approach is inspired by the work of Bubeck et al. [17], which proposed the use of the Good-Turing estimator in a context where the learning agent needs to sequentially select experts that only sample one of their potential nodes at each trial. In contrast, in OIMP, when a candidate is selected, it may have a potentially large spread and may activate many nodes at once. Our solution follows the well-known *optimism in the face of uncertainty* principle from the bandit literature by deriving an upper confidence bound on the estimator for the remaining potential for spreading information of each candidate, and by choosing in a principled manner between explore and exploit steps.

Through our approach, we show that efficient and effective influence maximization can be done in a highly uncertain or under-specified social environment.

II. Setting

In this section, we formally define the OIMP framework.

A. Background

Given a graph $G = (V, E)$, the traditional problem of influence maximization (IM) is to select a set of seed nodes $I \subseteq V$, under a cardinality constraint $|I| = L$, such that the expected *spread* of an influence cascade starting from I (or the expected number of activated nodes) is maximized. Formally, denoting by the random variable $S(I)$ the spread initiated by the seed set I, IM aims to solve the following optimization problem:

$$\arg\max_{I \subseteq V, |I| = L} \mathbb{E}[S(I)].$$

As mentioned before, a plethora of algorithms have been proposed to solve the IM problem, under specific diffusion models. These algorithms can be viewed as *full-information* and *offline* approaches: they choose all the seeds at once, in one step, and they have the complete diffusion configuration, i.e., the graph topology and the influence probabilities.

In the *online* case, during a sequence of N (what we call hereafter the *budget*) consecutive trials, L seed nodes are selected at each trial, and *feedback* on the achieved spread from these seeds is collected.

B. Influence maximization via candidates

The short timespan of campaigns makes parameter estimation very challenging within small horizons. In other cases, the topology – or even the existence – of a graph is too strong of an assumption. In contrast to [15], we do not try to estimate edge probabilities in some graph, but, instead, we assume the existence of several users – in the following referred to as the *spread seed candidates* (in short, *candidates*) – who are the only access to the medium of diffusion. Formally, let $[K] := \{1, \ldots, K\}$ be a set of candidates for selection; each candidate is connected to an unknown and potentially large base (the candidate's *support*) of basic nodes, each with an unknown activation probability. For illustration, we give in Figure 1 an example of this setting, with 3 candidates connected to 4, 5, and 4 basic nodes, respectively.
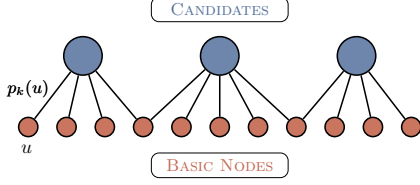
Figure 1: 3 candidates and associated activation probabilities $p_k(u)$.

Now, the problem boils down to estimating the value of the K candidates, for a K which is typically much smaller than the number of parameters of the diffusion model. The medium over which diffusion operates may remain a general diffusion graph, just like in the literature, but we make no assumption on that: the diffusion may happen in a completely unknown environment. Finally, note that by choosing $K = |V|$ candidates, the classic IM problem can be seen as a special instance of our setting.

We complete the formal setting by assuming the existence of K sets $A_k \subseteq V$ of basic nodes such that each candidate $k \in [K]$ is connected to each node in A_k, with $p_k(u)$ the probability for candidate k to activate the child node $u \in A_k$.

Definition 1 (Influence process)**.** *When a candidate $k \in [K]$ is selected, each basic node $u \in A_k$ is* sampled *for activation, according to its probability $p_k(u)$. The* feedback *(or* spread*) for k's selection consists of all the activated nodes, while the associated* reward *consists of the* newly activated *ones.*

Remark: Limiting the IM method to working with a small subset of the entire set of nodes may seem overly restrictive, but it allows to rapidly estimate nodes' values. As a motivating example, take marketing firms that may not have knowledge of the entire diffusion graph, only having access to a few influential people that can diffuse information (the candidates in our setting).

C. Online influence maximization with persistence

We are now ready to state our *online influence maximization with persistence*:

Problem 1 (OIMP)**.** *Given a set of candidates $[K] := \{1, \ldots, K\}$, a budget of N trials, and a number $1 \le L \le K$ of candidates to be activated at each trial, the objective of the* online influence maximization with persistence *(OIMP) is to solve the following optimization problem:*

$$\underset{I_n \subseteq [K], |I_n| = L, \forall 1 \le n \le N}{\arg\max} \mathbb{E} \left| \cup_{n=1}^N S(I_n) \right|.$$

Note that, in contrast to persistence-free online influence maximization – considered, e.g., in [9], [11] – the performance criterion used in OIMP displays the so-called *diminishing returns property*: the expected number of nodes activated by successive selections of a given candidate is decreasing, due to the fact that nodes that have already been activated are discounted. We refer to the expected number of nodes remaining to be activated as the *potential* or *missing mass* of a candidate. The diminishing returns property implies

that there is no static best set of candidates to be selected, but that the algorithm must follow an adaptive policy, which can detect that the remaining potential of a candidate is small and switch to another candidate that has been less exploited. Our solution to this problem overcomes challenges on two fronts: (1) it needs to estimate the potential of nodes at each round, without knowing the diffusion model nor the activation probabilities, and (2) it needs to identify the currently best spread candidates, according to their estimated potentials.

Other approaches for the online IM problem rely on estimating diffusion parameters [15], [9], [11] – generally, a distribution over the influence probability of each edge in the graph. However, the assumption that one can estimate accurately the diffusion parameters may be overly ambitious, especially in cases where the number of allowed trials (the budget) is rather limited.

III. ALGORITHM

We describe now our UCB-like algorithm, which relies on the Good-Turing estimator to sequentially select the spread candidates to activate at each round, from the available ones.

A. Missing mass and Good-Turing estimator

Given the K candidates, the OIMP problem boils down to the following: *How should we select a candidate at each step?* More precisely, a good algorithm for OIMP should aim at selecting the candidate k with the largest potential for influencing its children A_k. However, the true potential value of a candidate is *a priori* unknown to the decision maker.

In the following, we index trials by t when referring to the time of the algorithm, and we index trials by n when referring to the number of selections of the candidate. For example, the t-th spread initiated by the algorithm is noted $S(t)$ whereas the n-th spread of candidate k is noted $S_{k,n}$.

Definition 2 (Missing mass $R_k(t)$)**.** *Consider a candidate $k \in [K]$ connected to A_k basic nodes. Let $S(1), \ldots, S(t)$ be the set of nodes that were activated during the first t trials by the seeded candidates. The* missing mass $R_k(t)$ *is the expected number of* new *nodes that would be activated upon starting a $t+1$-th cascade from k:*

$$R_k(t) := \sum_{u \in A_k} \mathbb{1} \left\{ u \notin \cup_{i=1}^t S_k(i) \right\} p_k(u),$$

where $\mathbb{1}\{\cdot\}$ denotes the indicator function.

Definition 2 provides a formal way to obtain the remaining potential of a candidate k at a given time. The optimal policy would simply select the candidate with the largest missing mass at each time step. The difficulty is, however, that the probabilities $p_k(u)$ are unknown. Hence, we have to design a *missing mass estimator* $\hat{R}_k(t)$ instead. It is important to stress that the missing mass is a random quantity, because of the dependency on the spreads $S_k(t), \ldots, S_k(t)$. Furthermore, due to the diminishing returns property, the sequence $(S_{k,n})_{n \ge 1}$ is stochastically decreasing.

Following ideas from [16], [17], we now introduce a version of the Good-Turing statistic, tailored to our problem of rapidly estimating the missing mass. Denoting by $n_k(t)$ the number of times candidate k has been selected after t trials, we let $S_1, \ldots, S_{n_k(t)}$ be the $n_k(t)$ cascades sampled independently from candidate k. We denote by $U_k(u,t)$ the binary function whose value is 1 if node u has been activated *exactly* once by candidate k – such occurrences are called *hapaxes* in linguistics – and $Z_k(u,t)$ the binary function whose value is 1 if node u has never been activated by candidate k. The idea of the Good-Turing estimator is to estimate the missing mass as the proportion of hapaxes in the $n_k(t)$ sampled cascades, as follows:

$$\hat{R}_k(t) := \frac{1}{n_k(t)} \sum_{u \in A_k} U_k(u,t) \prod_{l \neq k} Z_l(u,t).$$

Albeit simple, this estimator turns out to be quite effective in practice. If a candidate is connected to a combination of both nodes having high activation probabilities and nodes having low activation probabilities, then successive traces sampled from this candidate will result in multiple activations of the high-probability nodes and few of the low-probability ones. Hence, after observing a few spreads, the candidate's potential will be low, a fact that will be captured by the low proportion of hapaxes. In contrast, estimators that try to estimate each activation probability independently will require a much larger number of trials to properly estimate the candidate's potential.

B. Upper confidence bounds

Following principles from the bandit literature, the GT-UCB algorithm relies on *optimism in the face of uncertainty*. At each step (trial) t, the algorithm selects the highest upper-confidence bound on the missing mass – denoted by $b_k(t)$ – and activates (plays) the corresponding candidate k. This algorithm achieves robustness against the stochastic nature of the cascades, by ensuring that candidates who "underperformed" with respect to their potential in previous trials may still be selected later on. Algorithm 1 presents the main components of GT-UCB for the case $L = 1$, that is, when a single candidate is chosen at each step.

Algorithm 1 – GT-UCB ($L = 1$)

Require: Set of candidates $[K]$, time budget N
 1: **Initialization:** play each candidate $k \in [K]$ once, observe the spread $S_{k,1}$, set $n_k = 1$
 2: For each $k \in [K]$: update the reward $W = W \cup S_{k,1}$
 3: **for** $t = K + 1, \ldots, N$ **do**
 4:     Compute $b_k(t)$ for every candidate k
 5:     Choose $k(t) = \arg\max_{k \in [K]} b_k(t)$
 6:     Play candidate $k(t)$ and observe spread $S(t)$
 7:     Update cumulative reward: $W = W \cup S(t)$
 8:     Update statistics of candidate $k(t)$: $n_{k(t)}(t+1) = n_{k(t)}(t) + 1$ and $S_{k,n_k(t)} = S(t)$.
 9: **end for**
10: **return** W

The algorithm starts by activating each candidate $k \in [K]$ once, in order to initialize its Good-Turing estimator. The main loop of GT-UCB occurs at lines 3-9. Let $S(t)$ be the observed spread at the trial t, and let $S_{k,s}$ be the result of the s-th diffusion initiated at candidate k. At every step $t > K$, we recompute for each candidate $k \in [K]$ its index $b_k(t)$, representing the upper confidence bound on the expected reward in the next trial. The computation of this index uses the previous samples $S_{k,1}, \ldots, S_{k,n_k(t)}$ and the number of times each candidate k has been activated up to trial t, $n_k(t)$. The index is set as:

$$b_k(t) = \hat{R}_k(t) + \left(1 + \sqrt{2}\right) \sqrt{\frac{\hat{\lambda}_k(t) \log(4t)}{n_k(t)}} + \frac{\log(4t)}{3n_k(t)},$$

where $\hat{R}_k(t)$ is the Good-Turing estimator and $\hat{\lambda}_k(t) := \sum_{s=1}^{n_k(t)} \frac{|S_{k,s}|}{n_k(t)}$ is an estimator for the expected spread from candidate k. The index formula is motivated by the confidence intervals given in Theorem 1[1].

Theorem 1. *With probability at least $1 - \delta$, for $\lambda = \sum_{u \in A} p(u)$ and $\beta_n := \left(1 + \sqrt{2}\right) \sqrt{\frac{\lambda \log(4/\delta)}{n}} + \frac{\log(4/\delta)}{3n}$,*

$$-\beta_n - \frac{\lambda}{n} \leq R_n - \hat{R}_n \leq \beta_n.$$

Then, in line 5, GT-UCB selects the candidate $k(t)$ with the largest index and initiates a cascade from it. The feedback $S(t)$ is observed and is used to update the cumulative reward set W. Note that $S(t)$ provides only the Ids of the nodes that were activated, with no information on *how* this diffusion happened in the hidden diffusion medium. Finally, statistics associated to the chosen candidate $k(t)$ are updated.
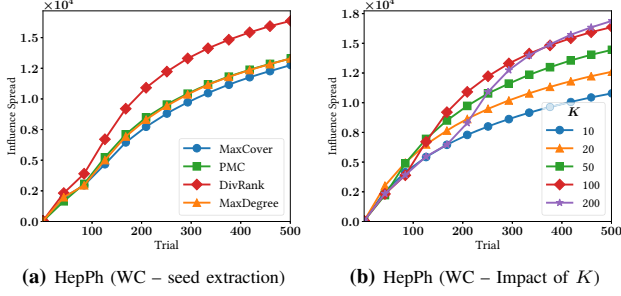
C. Extensions for the case $L > 1$

Algorithm 1 can be easily adapted to select $L > 1$ candidates at each round. Instead of choosing the candidate maximizing the Good-Turing UCB in line 5, we can select those having the L largest indices. Note that $k(t)$ then becomes a *set* of L candidates. A diffusion is initiated from the associated nodes and, at termination, all activations are observed. As in [12], the algorithm would require feedback to include the candidate responsible for the activation of each node, in order to update the statistics accordingly.

IV. EXPERIMENTS

We conducted experiments on two widely-used graphs from the IM literature, including [15]. Namely, we tested our algorithm on HepPh – a graph with $34.5K$ nodes and $422K$ edges – and DBLP – $317K$ nodes and $2.1M$ edges –, both representing publicly available collaboration networks, where undirected edges are drawn between authors who have collaborated on at least one paper. All methods are implemented[2] in C++ and simulations are done on an Ubuntu

---

[1]The proof of Theorem 1 is available at https://arxiv.org/abs/1702.05354
[2]The code is available at https://github.com/smaniu/oim.

(a) HepPh (WC – seed extraction)   **(b)** HepPh (WC – Impact of K)

Figure 2: Impact of GT-UCB parameters on influence spread.

16.04 machine with an Intel Xeon 2.4GHz CPU 20 cores and 98GB of RAM.

A. Extracting candidates from graphs

GT-UCB does not make any assumptions about the topology of the nodes influenced by the candidates. Indeed, in many settings it may be more natural to assume that the set of candidates is given and that the activations at each trial can be observed, while the topology of the underlying graph G remains unknown. In other settings, we may start from an existing social network G, in which case we need to extract a set of K representative candidates from it. Ideally, we should choose candidates that have little intersection in their "scopes of influence" to avoid useless seed selections. While this may be interpreted and performed differently, from one application to another, we discuss next some of the most natural heuristics for selecting candidates which we use in our experiments.

MaxDegree. This method selects the K nodes with the highest out-degrees in G.

Greedy MaxCover. This strategy follows the well-known greedy approximation algorithm for selecting a cover of the graph G. Specifically, the algorithm executes the following steps K times: (i.) select the node with highest out-degree (ii.) remove all out-neighbors of the selected node.

DivRank [18]. DivRank is a PageRank-like method relying on reinforced random walks, with the goal of producing diverse high-ranking nodes, while maintaining the rich-gets-richer paradigm. We adapted the original DivRank procedure by inverting the edge directions. In doing so, we get influential nodes instead of prestigious ones.

IM algorithms. This method assigns uniformly at random a propagation probability to each edge of G, assuming the IC model. Then, an IM algorithm – PMC [19] in our experiments – is executed on G to get the set of K candidates.

B. Results

Diffusion models. In the work closest to ours, Lei et al. [15] compared their solution on the Weighted Cascade instance of IC: every edge (u, v) has weight $1/d_v$ where d_v is the in-degree of node v. In this experimental study, and to illustrate that our approach is diffusion-independent, we added the tri-valency model (TV), which associates

randomly a probability from $\{0.1, 0.01, 0.001\}$ to every edge and follows the IC propagation model.

Baselines. We compare GT-UCB to several baselines. RANDOM chooses a random candidate at each round. MAXDEGREE selects the node with the largest degree at each round, where the degree does not include previously activated nodes. Finally, EG corresponds to [15]'s confidence-bound explore-exploit method with exponentiated gradient update, the state-of-the-art method for the OIMP problem (code provided by the authors). We use this last baseline on WC and TV weighted graphs and tune parameters in accordance to the results of their experiments: Maximum Likelihood Estimation is adopted for graph update and edge priors are set to Beta$(1, 20)$. The baselines are compared to an ORACLE that knows beforehand the diffusion model and probabilities. At each round, it runs an IM approximated algorithm – PMC in our experiments. Note that previously activated nodes are not counted when estimating the value of a node with PMC, thus making ORACLE adaptive.

All experiments are done with a trial horizon $N = 500$, a setting in line with many real-world marketing campaigns.

GT-UCB parameters. We show in Fig. 2a the impact of the candidate extraction criterion on HepPh under WC model. We can observe that the spread is only slightly affected by the extraction criterion: different datasets lead to different optimal criteria. On the HepPh network, DivRank clearly leads to larger influence spreads. We emphasize that on some other graph and model combinations we observed that other extraction routines can perform better than DivRank. In summary, we note that GT-UCB performs consistently as long as the method leads to candidates that are well spread over the graph. In the following, for each graph, we used DivRank as candidate extraction criterion in accordance with these observations.

In Fig. 2b, we measure the impact of the number of candidates K on the influence spread. We can observe that a larger amount of candidates leads to greater influence spreads on HepPh: this network is relatively small ($34.5K$ nodes), and thus half of the nodes are already activated after 400 trials. By having more candidates, we are able to access parts of the network that would not be accessible otherwise.

GT-UCB vs. baselines. We evaluate the execution time of the different algorithms in Fig. 3. As expected, GT-UCB largely outperforms EG (and ORACLE). The two baselines
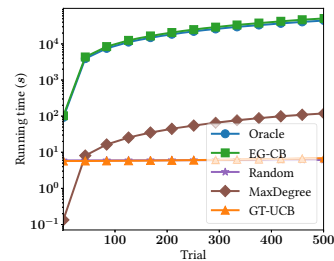
Figure 3: DBLP (WC) – Execution time.

(a) HepPh (WC – $L = 1$) **(b)** DBLP (WC – $L = 10$)

(c) HepPh (TV – $L = 1$) **(d)** DBLP (TV – $L = 10$)
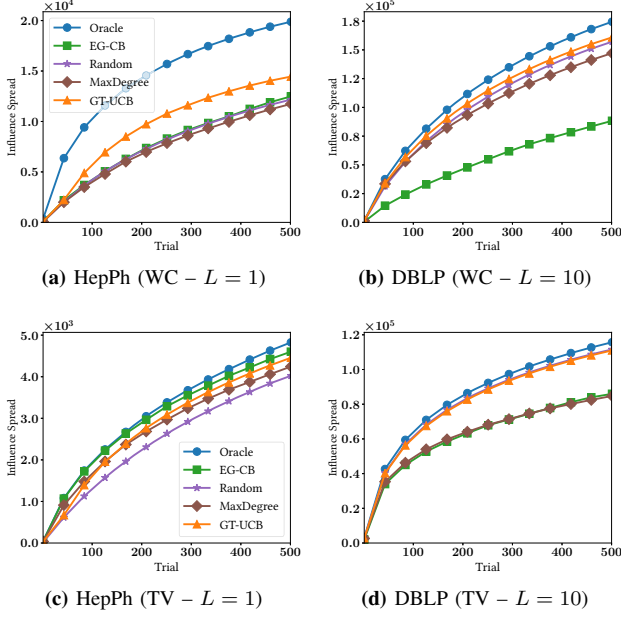
Figure 4: Growth of spreads against the number of rounds.

require the execution of an approximated IM algorithm at each round. GT-UCB is several orders of magnitude faster: it concentrates most its running time on extracting candidates, while statistic updates and UCB computations are negligible.

In Fig. 4, we show the growth of the spread for GT-UCB and baselines. For each experiment, GT-UCB uses $K = 50$ if $L = 1$ and $K = 100$ if $L = 10$. First, we can see that MAXDEGREE is a strong baseline in many cases. GT-UCB results in good quality spreads across every combination of network and diffusion model. Interestingly, on the smaller graph HepPh, we observe an increase in the slope of spread after initialization, particularly visible at $t = 50$ with WC. This corresponds to the step when GT-UCB starts to select candidates maximizing $b_k(t)$ in the main loop. It shows that our strategy adapts well to the previous activations, and chooses good candidates at each iteration. Interestingly, RANDOM performs surprisingly well in many cases, especially under TV weight assignment. However, when certain candidates are significantly better than others, it cannot adapt to select the best candidate unlike GT-UCB. EG performs well on HepPh, especially under TV weight assignment. However, it fails to provide competitive cumulative spreads on DBLP. We believe that EG tries to estimate too many parameters for a horizon $T = 500$. After reaching this time step, less than 10% of all nodes for WC, and 20% for TV, are activated. This implies that we have small confidence regarding many edge probability estimations, as most nodes are located in parts of the graph that have never been explored.

V. CONCLUSION

We proposed a diffusion-independent approach for online and adaptive IM, whose role is to maximize the number of activated nodes in an arbitrary environment, under the OIMP framework. Our method requires as only interfaces with the "real-world" the identification of potential seeds (the candidates) and the spread feedback (i.e., the set of activated nodes) at each trial. Subsequent online iterations are very fast, making it possible to scale to very large graphs.

REFERENCES

[1] D. Watts and P. Dodds, "Influentials, networks, and public opinion formation," *Journal of Consumer Research*, 2007.

[2] D. Easley and J. Kleinberg, *Networks, Crowds, and Markets - Reasoning About a Highly Connected World.* Cambridge University Press, 2010.

[3] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," in *SIGKDD.* ACM, 2003.

[4] A. Arora, S. Galhotra, and S. Ranu, "Debunking the myths of influence maximization: An in-depth benchmarking study," in *SIGMOD.* ACM, 2017.

[5] M. Gomez-Rodriguez, J. Leskovec, and B. Schölkopf, "Structure and dynamics of information pathways in online media," in *WSDM*, 2013.

[6] A. Goyal, F. Bonchi, and L. Lakshmanan, "Learning influence probabilities in social networks," in *WSDM*, 2010.

[7] K. Saito, R. Nakano, and M. Kimura, "Prediction of information diffusion probabilities for independent cascade model," in *KES*, 2008.

[8] N. Du, L. Song, H. Woo, and H. Zha, "Uncover topic-sensitive information diffusion networks," in *AISTATS*, 2013.

[9] S. Vaswani, V. Lakshmanan, and M. Schmidt, "Influence maximization with bandits," in *Workshop NIPS*, 2015.

[10] W. Chen, Y. Wang, Y. Yuan, and Q. Wang, "Combinatorial multi-armed bandit and its extension to probabilistically triggered arms," *JMLR*, vol. 17, no. 1, 2016.

[11] Z. Wen, B. Kveton, and M. Valko, "Influence maximization with semi-bandit feedback," Tech. Rep., 2016.

[12] S. Vaswani, B. Kveton, Z. Wen, M. Ghavamzadeh, L. Lakshmanan, and M. Schmidt, "Diffusion independent semi-bandit influence maximization," in *ICML*, 2017.

[13] P. Grabowicz, N. Ganguly, and K. Gummadi, "Distinguishing between topical and non-topical information diffusion mechanisms in social media," in *ICWSM*, 2016.

[14] D. Romero, B. Meeder, and J. Kleinberg, "Differences in the mechanics of information diffusion across topics: Idioms, political hashtags, and complex contagion on twitter," in *WWW*, 2011.

[15] S. Lei, S. Maniu, L. Mo, R. Cheng, and P. Senellart, "Online influence maximization," in *SIGKDD*, 2015.

[16] I. J. Good, "The population frequencies of species and the estimation of population parameters," *Biometrika*, 1953.

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

[18] Q. Mei, J. Guo, and D. Radev, "Divrank: The interplay of prestige and diversity in information networks," in *SIGKDD*, 2010.

[19] N. Ohsaka, T. Akiba, Y. Yoshida, and K. Kawarabayashi, "Fast and accurate influence maximization on large networks with pruned monte-carlo simulations," in *AAAI*, 2014.