

Online Influence Maximization under Decreasing Cascade Model

Fang Kong
Shanghai Jiao Tong University
Shanghai, China
fangkong@sjtu.edu.cn

Jize Xie
Shanghai Jiao Tong University
Shanghai, China
xjzzjl@sjtu.edu.cn

Baoxiang Wang
The Chinese University of Hong
Kong, Shenzhen
Shenzhen, China
bxiangwang@cuhk.edu.cn

Tao Yao^{*}
Alibaba, Inc.
Seattle, USA
tao.yao@alibaba-inc.com

Shuai Li[‡]
Shanghai Jiao Tong University
Shanghai, China
shuaili8@sjtu.edu.cn

ABSTRACT

We study online influence maximization (OIM) under a new model of decreasing cascade (DC). This model is a generalization of the independent cascade (IC) model by considering the common phenomenon of market saturation. In DC, the chance of an influence attempt being successful reduces with previous failures. The effect is neglected by previous OIM works under IC and linear threshold models. We propose the DC-UCB algorithm to solve this problem, which achieves a regret bound of the same order as the state-of-the-art works on the IC model. Extensive experiments on both synthetic and real datasets show the effectiveness of our algorithm.

KEYWORDS

Multi-armed bandits; Social network; Influence maximization; Decreasing cascade model

ACM Reference Format:

Fang Kong, Jize Xie, Baoxiang Wang, Tao Yao^{*}, and Shuai Li[‡]. 2023. Online Influence Maximization under Decreasing Cascade Model. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 8 pages.

1 INTRODUCTION

The study of information diffusion on social networks has received increasing attention from the community of machine learning, data mining, and graph algorithms. A term to be diffused has many forms, including the spread of news and opinions, adoption of products, and broadcast of alarms. To characterize this process, many influence propagation models have been proposed. Among them, the independent cascade (IC) model and the linear threshold (LT) model are widely adopted [2, 5, 8, 12, 28]. The common ground of all these models is to use a weighted graph to represent a social network, where the weights denote influence abilities between nodes. For example, the IC model assumes that the term transmits through each edge independently with a probability equal to the weight; under LT, the transmission happens when the cumulative

weight of in-neighbors exceeds a certain threshold. Despite the popularity, both IC and LT neglect the decay characterization, which is common in many real applications [13, 20, 28, 32]. This decay property reflects the phenomenon of *market saturation* where more failed influence attempts would turn the market to be more saturated, making subsequent influence trials less likely to succeed. The decreasing cascade (DC) model [13] is a generalization of IC that depicts these observations for better feasibility.

When the term is the alarm of an emergency or a broadcast of an important notification, the relevant party has a crucial responsibility to spread it to the possible extent under limited resources. Similarly, the marketing team of a company advertises products on social networks, aiming to attract as many users as possible [12, 28]. These real tasks motivate the problem of influence maximization (IM) [5, 12, 21, 24]. Given a graph and its underlying influence propagation model together with the model parameters (the graph weights), it desires to find an initial set of adopted users (the seed set) to maximize the influence spread. The IM problem has been widely studied under IC [11, 24], LT [5, 7] and the DC model [13].

A major concern of IM is that in real applications, the parameters of the influence propagation models are usually unknown. For example in advertisement placing, a company might not know the actual influence probabilities before they place the advertisement. A heuristic to address this issue is to estimate the unknown parameters from the collected past observations [6, 19]. However, there might not exist sufficient logs, and even if they exist, the logs have biases. Also, such estimates cannot adapt to any change in the social network. Online influence maximization (OIM), instead, learns the unknown parameters through the iterative interactions with the social network and progressively finds the optimal seed set [4, 15, 17, 22, 25, 27, 30]. Previous theoretical OIM studies mainly focus on the IC and LT models [3, 4, 17, 23, 25, 27, 30]. Despite their importance, these model assumptions fail to characterize the common damping phenomenon of market saturation in influence spreading [13, 20, 28, 32]. Thus, how to design efficient algorithms on the more general DC setting remains an open problem.

We are the first to formulate the OIM problem under the DC model. In this paper, we propose an upper confidence bound (UCB)-based algorithm, DC-UCB, to solve this problem. The algorithm meticulously readjusts the UCB indices of activation probabilities to maintain the decreasing property of DC. Through careful analysis

^{*} Work done at Alibaba Group, and now affiliated with the Chinese University of Hong Kong, Shenzhen.

[‡] Corresponding author.

of the information diffusion process under DC, we prove a DC-based triggering probability modulated (TPM) bounded smoothness condition, as an analogy to that under IC [25, 27]. We can then provide rigorous theoretical guarantees on the regret of DC-UCB. The regret upper bound of DC-UCB achieves the same order as the state-of-the-art results under IC. Extensive experiments on both synthetic and real-world networks show the effectiveness and efficiency of our algorithm.

2 RELATED WORK

The (offline) IM problem was formulated by the work [12]. It conducts discrete optimization on the seed set to maximize the influence spread on graphs when given parameters of the underlying diffusion model. Since then, this problem has attracted a lot of attention [5, 13, 21, 24, 32] focusing on different underlying diffusion models and different solving techniques. The adaptive influence maximization (AIM) problem is a variant of IM where the agent can adaptively select seed nodes after it observes the propagation results of previously selected seeds [? ?]. Both problems assume that the diffusion parameters are known beforehand.

When the parameters of the diffusion model are unknown, the problem can be solved through online IM (OIM), aiming to learn the parameters through the interactions with the social network. The framework of OIM can be formulated as a problem of combinatorial multi-armed bandits (CMAB) [3, 4, 25] - a T -round game between the learning agent and the environment to maximize the cumulative reward. In each round, the learning agent executes a combination of base arms, defined as a super arm, and observes the feedback (i.e. the influence propagation in OIM). Based on the collected feedback, it then updates its knowledge for the unknown environmental parameters and improves the subsequent choices.

Chen *et al.* are the first to use the CMAB framework with probabilistically triggered arms to study the OIM problem under the IC model with edge-level feedback [3, 4, 25]. In this formulation, each edge is regarded as a base arm and all outgoing edges from the seed set are regarded as a super arm. The line of studies proposes a CUCB algorithm based on the canonical upper confidence bound (UCB) algorithm [1] and derives rigorous guarantees of it [25]. To generalize it to large-scale real applications, Wen *et al.* [27] present a linear variant for the activation probabilities and propose the IMLinUCB algorithm. Wu *et al.* [30] consider the network assortativity and assume the activation probability of each edge can be decomposed by the influence factor of the source node and the susceptibility factor of the destination node to reduce the sample complexity. Node-level feedback, which needs less information and is more realistic than the edge-level feedback, has also been considered under the IC model [23]. It assumes that the identities of influenced nodes (instead of transmitted edges) can be observed, and provides a bound on estimation gap of the activation probabilities between node-level feedback and edge-level feedback. Recently, Zhang *et al.*, [31] give a regret upper bound for this challenging feedback.

A few OIM works consider different diffusion models. Until recently, the OIM problem under the LT model was solved by the work [17]. This work assumes the full node-level feedback, the influence status of each node in each diffusion step, can be observed and gives the first regret upper bound under this model. Another

work [22] considers a pairwise feedback scheme, where the agent can directly observe the influence status between each node and each seed node. Though the setting can be applied to many diffusion models like IC, LT, and DC, there are no optimality guarantees for their heuristically proposed objective function. Our work is the first one to study the OIM problem under the DC model with rigorous theoretical guarantees.

3 SETTING

In this section, we formulate the OIM problem under the DC model. The social network is the basic structure of this problem, which is usually represented by a directed graph $G = (V, E)$ with the node set V and the edge set E denoting the set of users and the set of relationships between users, respectively. An edge $e = (u, v) \in E$, for example in Twitter, can correspond to the relationship of user v following user u and the information transmitting from u to v . Let $n = |V|$ and $m = |E|$ be the number of nodes and edges, respectively. For each node $v \in V$, denote $N(v) := N^{\text{in}}(v)$ as the set of all incoming neighbors of v , abbreviated as in-neighbors.

The IC and LT models [12] are two of the most common and widely used influence propagation models in characterizing the information diffusion on social networks [5, 17, 24]. Under the IC model, each node will try to activate all its inactive out-neighbors independently right after it is activated. The success probability of activation attempt between every such pair is equivalent to the weight of this edge. Under LT, a node is activated if the sum of edge weights from its active in-neighbors exceeds a certain threshold.

However, the information diffusion can be very complicated in real applications. One of the main considerations is the famous effect of *market saturation* in real life [13, 14, 16, 20, 28, 32]. As the information spreads more around the neighborhood, users usually become more saturated with the marketing and their in-neighbors will have diminishing influence effects on them. For example in the diffusion of a news story, the story would become more redundant and unattractive every time the user sees it from the broadcasts of the neighbors but expresses no interest, making the adoption probability decrease with the time of exposures [10, 18, 29]. Such characterization of influence diffusion is known as the decay property [20, 28, 32], which is not fully covered in the common IC and LT models but can be captured in the DC model [13].

The information diffusion process of the DC model starting from the seed set S is described as follows. Define S_τ as the set of influenced nodes until the end of time step τ . In the beginning when $\tau = 0$, only nodes in S are influenced, that is, $S_0 = S$. Then after time step $\tau \geq 0$ for each inactive node $v \in V \setminus S_\tau$, all of its active in-neighbors who are influenced at the last time step, i.e. elements in $N(v) \cap (S_\tau \setminus S_{\tau-1})$, will make an attempt to activate v in an arbitrary order (denote $S_{-1} = \emptyset$ for consistency). Specifically, each node $u \in N(v) \cap (S_\tau \setminus S_{\tau-1})$ tries to activate v with probability $p_v(u, S')$, where S' is the set of nodes that have already tried but failed to activate v in all previous steps ($< \tau$) and the current step ($= \tau$). If there exists an u who successfully activates v at $\tau + 1$, then v becomes active, or equivalently $v \in S_{\tau+1} \setminus S_\tau$; otherwise if all the nodes in $N(v) \cap (S_\tau \setminus S_{\tau-1})$ fail to activate v , then v is still inactive at $\tau + 1$, or equivalently $v \notin S_{\tau+1}$. Such an information diffusion process ends when no node is activated at a new step. The

influence spread $r(S, p)$ is defined as the expected number of total influenced nodes under the seed set S and the activation probabilities $p = (p_v(u, S'))_{v \in V, u \in N(v), S' \subseteq N(v) \setminus \{u\}}$. Here the expectation is taken over the randomness in the diffusion process, specifically the success or failure events of all activation attempts.

The activation probabilities p under the DC model satisfy the following two mild but important properties.

Decreasing. The activation probability of $u \in N(v)$ on v decreases with more previous failed attempts. Specifically, if $S' \subseteq S'' \subseteq N(v) \setminus \{u\}$, then $p_v(u, S') \geq p_v(u, S'')$.

Order-independence. The probability that v is eventually influenced by the set $S' = \{u_1, u_2, \dots, u_\ell\} \subset N(v)$ does not depend on the order of these nodes' activation attempts. That is, the probabilities that S' successfully activate v in order of u_1, u_2, \dots, u_ℓ and $u_\ell, u_{\ell-1}, \dots, u_1$ are the same.

With seed set cardinality K , denote $\mathcal{S} = \{S \subset V : |S| \leq K\}$ as the action set which consists of all feasible seed sets with size smaller than K . When the activation probability vector p is known, the (offline) IM problem aims to find an $S \in \mathcal{S}$ with the maximum influence spread $\text{argmax}_{S \in \mathcal{S}} r(S, p)$. This problem under the DC model is NP-hard but can be approximately solved with a greedy algorithm, since the influence spread function is monotone and submodular [13]. We use $\text{opt}_p = \max_{S \in \mathcal{S}} r(S, p)$ and $S_p^{\text{opt}} \in \text{argmax}_{S \in \mathcal{S}} r(S, p)$ to denote the maximum influence spread and an optimal seed set, respectively. Let Oracle be an offline IM algorithm that outputs a seed set given the activation probabilities p . For $\alpha, \beta \in [0, 1]$, we say Oracle is an (α, β) -approximation if its output $S^* = \text{Oracle}(p)$ satisfies $\mathbb{P}(r(S^*, p) \geq \alpha \cdot \text{opt}_p) \geq \beta$ for any input p .

It is worth pointing out that the IC model is a special case of the DC model since it satisfies that $p_v(u, S') = p_{u,v}$ for any edge $(u, v) \in E$ and the two properties of the DC model can be verified easily. In this paper, we consider another special case of DC where $p_v(u, S')$ only depends on the size of S' and node v , not on u and the elements in S' . It characterizes that the probability of a node being influenced depends on the susceptibility of the node itself [26] and the previous failed attempts. This setting keeps the most important decay property to describe the real-world phenomena of market saturation. Specifically, each node v is associated with a decreasing probability sequence of size $|N(v)|$, denoted as $p_v := [p_v(1), p_v(2), \dots, p_v(|N(v)|)]$. Then the probability that u successfully activates v after the attempts of the nodes in S' is $p_v(u, S') = p_v(|S'| + 1)$. The activation probability vector can thus be written as $p := (p_v(i))_{v \in V, i \in [|N(v)|]}$ and it is immediate to verify that two properties of the DC model hold. Note that if for each node v , its related activation probabilities are the same, or $p_v(i) \equiv p'_v$ for all $i \in [|N(v)|]$, the diffusion process under this specific DC model is the same with that under IC.

In the online version where the activation probability vector p is unknown, the problem aims to learn those unknown probabilities from the interactions with the social network and to gradually identify the optimal seed set. In each round t , the learning agent selects a seed set $S_t \in \mathcal{S}$. Then the diffusion process originating from S_t could reveal some influence propagation, based on which the agent could get some information about the unknown parameters. Similar to most OIM works, we consider the (partial) edge-level feedback where an edge is observed only when its start node is active and the

end node is inactive. Recall that the (full) edge-level feedback assumes an edge to be observed if its start node is active [4, 25, 27, 30]. Our (partial) edge-level feedback requires less information since each node is activated at most once. It is more reasonable that the following activation attempts on a node after it is activated are not supposed to be observed.

With an (α, β) -approximation oracle, the objective of the learning agent is to maximize the T -round cumulative influence spread, or equivalently to minimize the cumulative $\alpha\beta$ -scaled regret [4, 25, 27, 30] over T rounds

$$R(T) = \mathbb{E} \left[\sum_{t=1}^T r(t) \right] = \mathbb{E} \left[\sum_{t=1}^T \left(\alpha\beta \cdot \text{opt}_p - r(S_t, p) \right) \right], \quad (1)$$

where $r(t)$ is the regret at t and the expectation is taken over the randomness in diffusion processes and the adopted oracle.

4 THE DC-UCB ALGORITHM

In this section, we introduce DC-UCB (Algorithm 1), a UCB-type algorithm, to solve the OIM problem under the DC model with (partial) edge-level feedback.

The DC-UCB algorithm takes the graph $G = (V, E)$, the seed set cardinality K as well as an offline Oracle as input. For each $p_v(i)$ that represents the success probability of the i -th activation attempt on node v , the algorithm maintains its empirical mean $\hat{p}_v(i)$ and the number of observations $T_v(i)$.

Algorithm 1 DC-UCB

- 1: **Input:** Graph $G = (V, E)$; seed set size K ; Oracle
 - 2: **Initialize:** $\hat{p}_v(i) = 0, T_v(i) = 0$, for $v \in V, i \in [|N(v)|]$
 - 3: **for** $t = 1, 2, \dots$ **do**
 - 4: **for** each node $v \in V, i = 1, 2, 3, \dots, |N(v)|$ **do**
 - 5: $\bar{p}_v(i) = \text{Proj}_{[0,1]} \left(\hat{p}_v(i) + \sqrt{\frac{3 \log t}{2T_v(i)}} \right)$
 - 6: $\bar{p}'_v(i) = \min \{ \bar{p}'_v(i-1), \bar{p}_v(i) \}$ ($\bar{p}'_v(0) = 1$)
 - 7: **end for**
 - 8: Choose $S_t = \text{Oracle}(G, K, \bar{p}')$ and observe feedback
 - 9: **for** each status $Y_v(i)$ of all attempts **do**
 - 10: $\hat{p}_v(i) = \frac{T_v(i) \cdot \hat{p}_v(i) + Y_v(i)}{T_v(i)+1}$; $T_v(i) = T_v(i) + 1$
 - 11: **end for**
 - 12: **end for**
-

In each round t , the learning agent first computes the UCB $\bar{p}_v(i)$ for each activation probability $p_v(i)$ based on collected observations (line 5). The computation of the UCBs are based on the Chernoff-Hoeffding inequality [9] to guarantee the value is an upper bound of the true value with high probability, which is applicable here since observations on the same term $p_v(i)$ in different rounds are independent. The operating $\text{Proj}_{[0,1]}(\cdot)$ projects a real number into interval $[0, 1]$ to ensure the UCBs of probabilities still fall into this interval. Specially, if $T_v(i) = 0$ for i -th activation probability of node v , we simply set $\bar{p}_v(i) = 1$. Since the decreasing property of the DC model guarantees that the i -th real probability always larger than the $i+1$ -th real probability for any node v , these UCB indices are then capped to maintain the decreasing property (line 6).

With the capped UCBs \bar{p}' , graph G and seed set size K as input, the offline Oracle computes a seed set S_t (line 8). The returned

solution automatically balance the exploitation and exploration: If all activation probabilities are observed enough, then their UCBs are roughly the empirical means and the Oracle will return a solution that is approximately optimal under the estimated weights, whose value is close to the one under true probabilities since the influence spread is continuous in vector p ; if some activation probabilities are not observed enough, their confidence interval would be wide and their UCBs would be high, making the graph lean towards these less-explored parts and thus forcing the exploration.

Then the influence spreads from the selected seed set S_t . And the agent can observe a binary variable $Y_v(i)$ if there is the i -th attempt to activate v , where 1 represents the successful activation and 0 represents failure. With the $Y_v(i)$, the corresponding $\hat{p}_v(i)$ is updated (line 10).

4.1 Regret Bounds

The following theorem shows the problem-independent regret bound for our algorithm DC-UCB.

THEOREM 1. *The (α, β) -scaled regret of DC-UCB satisfies*

$$R(T) \leq O\left(nm\sqrt{T\log T}\right). \quad (2)$$

This is the first theoretical result of the OIM problem under DC, which does not require the strong independence assumption in the IC model and considers the common market-saturation phenomenon in real life. Compared with the regret bound of the IC model, our regret achieves the same order in the graph parameters n, m and time horizon T [25], though strictly speaking the regret bounds under two different models are not directly comparable. Recall that if the activation probabilities satisfy $p_v(i) \equiv p'_v$, the diffusion process would be the same under the IC model and the regret bounds under this case can be directly comparable.

Besides problem-independent bound (Theorem 1), we also provide the problem-dependent bound for DC-UCB.

To get this, define the gap between the influence spread of a seed set S and the α -scaled optimal influence spread as

$$\Delta_S = \max\left\{0, \alpha \cdot \text{opt}_p - r(S, p)\right\},$$

where α is the approximation factor of the offline Oracle. And for each entry (v, i) with $v \in V, i \in [|N(v)|]$, define $p_{v,i}^S$ as the probability that node v 's i -th attempt can be observed in the diffusion process starting from S under activation probability vector p (here we omit the dependence on p for simplicity). Then we can define the arm gap with the aid of such observation probability

$$\Delta_{\min}^{v,i} = \inf_{S \in \mathcal{S}: \Delta_S > 0, p_{v,i}^S > 0} \Delta_S$$

and take the minimum over all entries

$$\Delta_{\min} = \min_{v \in V, i \in [|N(v)|]} \Delta_{\min}^{v,i}.$$

Similarly to the IC model, let

$$\tilde{V} = \max_{u \in V} \sum_{v \in V} \mathbb{1}\{\text{there is a path from } u \text{ to } v\} \quad (3)$$

be the maximum number of nodes that a node can reach in G . With these notations, the problem-dependent regret bound is provided in the next theorem.

THEOREM 2. *The (α, β) -scaled regret of the DC-UCB algorithm can be bounded as*

$$\begin{aligned} R(T) &\leq \sum_{v \in V, i \in [|N(v)|]} \frac{576\tilde{V}^2 m \ln T}{\Delta_{\min}^{v,i}} \\ &\quad + \frac{\pi^2 \Delta_{\max}}{6} \sum_{v \in V, i \in [|N(v)|]} \left(2 + \log \frac{4m\tilde{V}}{\Delta_{\min}^{v,i}}\right) + 4m\tilde{V} \quad (4) \\ &= O\left(\frac{m^2 n^2}{\Delta_{\min}} \ln T\right). \quad (5) \end{aligned}$$

Due to the space limit, the detailed proof of Theorem 1 and 2 are provided in Appendix. By carefully analyzing the information diffusion process under DC, we prove our DC-based TPM condition, similar to that under IC [25, 27]. Such TPM condition bounds the difference between influence spread under different activation probabilities, which is crucial to acquire the above theoretical guarantees.

4.2 The TPM Condition under the DC model

To bound the influence spread difference $r(S_t, \bar{p}_t) - r(S_t, p)$ under two activation probabilities, we need the following key theorem of the triggering probability modulated (TPM) condition. Such a condition is crucial in deriving the final regret bound, similarly to that under the IC model [25, 27]. Denote $V_{S,v}$ as the set of vertices who are on any path from S to v for any seed set S .

THEOREM 3. *For any two activation probability vectors p and \bar{p} satisfying $p_v(i) \leq \bar{p}_v(i)$ for any $v \in V, i \in [|N(v)|]$, the difference between the influence spread of any seed set S under these two activation probability vectors is at most*

$$\begin{aligned} &r(S, \bar{p}) - r(S, p) \\ &\leq \mathbb{E}\left[\sum_{v \in V \setminus S} \sum_{u \in V_{S,v}} \sum_{i \in [|N(u)|]} \mathbb{1}\{O_u(i)\} \cdot [\bar{p}_u(i) - p_u(i)]\right] \quad (6) \\ &= \sum_{v \in V \setminus S} \sum_{u \in V_{S,v}} \sum_{i \in [|N(u)|]} p_{u,i}^S \cdot [\bar{p}_u(i) - p_u(i)], \quad (7) \end{aligned}$$

where $O_u(i)$ denotes the event that the i -th attempt to activate u under p can be observed.

PROOF OF THEOREM 3. Recall that $r(S, p, v)$ is the probability that v is finally influenced in the diffusion process starting from S under p . We can decompose the influence spread difference under two activation probability vectors as

$$\begin{aligned} r(S, \bar{p}) - r(S, p) &= \sum_{v \in V \setminus S} r(S, \bar{p}, v) - r(S, p, v) \\ &= \mathbb{E}\left[\sum_{v \in V \setminus S} \mathbb{1}\{v \text{ is influenced under } \bar{p}\} \right. \\ &\quad \left. - \mathbb{1}\{v \text{ is influenced under } p\}\right]. \end{aligned}$$

According to the monotonicity of the influence spread,

$$\mathbb{1}\{v \text{ is influenced under } \bar{p}\} - \mathbb{1}\{v \text{ is influenced under } p\} \in \{0, 1\}.$$

When $\mathbb{1}\{v \text{ is influenced under } \bar{p}\} - \mathbb{1}\{v \text{ is influenced under } p\} = 1$, which means

$$\begin{aligned}\mathbb{1}\{v \text{ is influenced under } \bar{p}\} &= 1, \\ \mathbb{1}\{v \text{ is influenced under } p\} &= 0,\end{aligned}$$

v is influenced under \bar{p} but not influenced under p .

Since the influence status of v under two activation probabilities are different, there must exist a step $\tau > 0$ in the diffusion process such that starting from τ , the influence statuses of the nodes related to v are different. To be specific, there must exist some time step τ and node $u \in V_{S,v}$ such that the active in-neighbors of u under p and \bar{p} are the same until the end of τ , but the influence status of u under p and \bar{p} are not the same at $\tau + 1$.

Let $\ell_{u,\tau-1}, \ell_{u,\tau}$ be the number of active in-neighbors at step $\tau - 1$ and τ , respectively. Then at step τ , there are totally $(\ell_{u,\tau} - \ell_{u,\tau-1} - 1)$ attempts to activate u under both p and \bar{p} . We denote them as the $(\ell_{u,\tau-1} + 1)$ -th, $(\ell_{u,\tau} + 2)$ -th, \dots , $\ell_{u,\tau}$ -th attempt. Also for any $\ell_{u,\tau-1} + 1 \leq j \leq \ell_{u,\tau}$, let $Y_u(j)$ and $\bar{Y}_u(j)$ be the status of the j -th attempt under p and \bar{p} , respectively. It holds that $\mathbb{E}[Y_u(j)] = p_u(j)$, $\mathbb{E}[\bar{Y}_u(j)] = \bar{p}_u(j)$.

Then based on the above analysis, there must exist an i -th attempt, where $\ell_{u,\tau-1} + 1 \leq i \leq \ell_{u,\tau}$ and all attempts to activate u before i fail under both p and \bar{p} and the i -th attempt succeeds under \bar{p} but fails under p . That is,

$$\begin{aligned}\forall j < i, X_u(j) &= 0, \bar{X}_u(j) = 0, \\ X_u(i) &= 0, \bar{X}_u(i) = 1.\end{aligned}$$

This event happens with probability $\prod_{j < i} (1 - \bar{p}_u(j)) \cdot [\bar{p}_u(i) - p_u(i)]$. By the union bound, we conclude that

$$r(S, \bar{p}) - r(S, p) \quad (8)$$

$$= \mathbb{E} \left[\sum_{v \in V \setminus S} \mathbb{1}\{v \text{ is influenced under } \bar{p}\} - \mathbb{1}\{v \text{ is influenced under } p\} \right] \quad (9)$$

$$\begin{aligned}&\leq \mathbb{E} \left[\sum_{v \in V \setminus S} \sum_{u \in V_{S,v}} \sum_{\ell_{u,\tau-1} + 1 \leq i \leq \ell_{u,\tau}} \prod_{j < i} (1 - \bar{p}_u(j)) \cdot [\bar{p}_u(i) - p_u(i)] \right] \\ &\leq \mathbb{E} \left[\sum_{v \in V \setminus S} \sum_{u \in V_{S,v}} \sum_{i \in [|N(u)|]} \mathbb{1}\{A_u(i)\} \prod_{j < i} (1 - \bar{p}_u(j)) \cdot [\bar{p}_u(i) - p_u(i)] \right],\end{aligned} \quad (10)$$

where (10) is derived by the definition of event $A_u(i)$, which denotes that there are more than i active in-neighbors of node u under p .

Also based on the feedback scheme of the DC-UCB algorithm, all status of attempts to activate u before u is finally influenced can be observed, which means that

$$\begin{aligned}\mathbb{E} \left[\mathbb{1}\{A_u(i)\} \prod_{j < i} (1 - \bar{p}_u(j)) \right] &\leq \mathbb{E} \left[\mathbb{1}\{A_u(i)\} \prod_{j < i} (1 - p_u(j)) \right] \\ &= \mathbb{E} [\mathbb{1}\{O_u(i)\}] = P_{u,i}^S.\end{aligned}$$

Thus we conclude that

$$r(S, \bar{p}) - r(S, p)$$

$$\begin{aligned}&\leq \mathbb{E} \left[\sum_{v \in V \setminus S} \sum_{u \in V_{S,v}} \sum_{i \in [|N(u)|]} \mathbb{1}\{O_u(i)\} \cdot [\bar{p}_u(i) - p_u(i)] \right] \\ &= \sum_{v \in V \setminus S} \sum_{u \in V_{S,v}} \sum_{i \in [|N(u)|]} P_{u,i}^S \cdot [\bar{p}_u(i) - p_u(i)]. \quad \square\end{aligned}$$

5 EXPERIMENTS

In this section, we compare our DC-UCB algorithm with related baselines in both synthetic and real-world networks¹. Since this work is the first to study the OIM problem under the DC model with rigorous theoretical guarantees, few directly comparable baselines exist. We exhaust those baselines and also adopt baseline methods that give insight into the performance of our algorithms via indirect comparisons. The following are descriptions.

UCB. The OIM problem under DC can be regarded as a multi-armed bandit problem if we treat each seed set as an arm and the influence spread as its expected reward. Thus, the classical UCB algorithm [1] can be applied to solve this problem. This algorithm maintains a UCB index for each arm and selects the arm enjoying the highest UCB index in each round. It is acceptable when the graph is small but is not feasible when the graph is large as the number of arms grows exponentially with the number of nodes.

CMAB-UCB-average and CMAB-UCB-random. The OIM problem under DC is a CMAB problem, where we can treat each node as a base arm and a seed set as a super arm. The influence spread of a seed set is then the expected reward of the super arm. Note that this reward cannot be written in a linear function form in terms of the utilities of the included base arms (nodes). Since most CMAB works study a linear function reward with semi-bandit feedback, we consider two variant updates here. The first is to divide the received reward by the size of the chosen super arm and assign the quotient to each of base arms in this super arm, denoted as *average*. The second is to randomly select a base arm from the super arm and assign the received reward to this base arm only (other base arms receive a reward 0), denoted as *random*. These two variants both use linear function approximation and adopt different assignments of the received reward.

DILinUCB. This algorithm [22] can also be applied to our DC model, though its heuristic objective has no guaranteed approximation ratio. In this setting, the feedback for each pair of the seed node and a node is 1 if there is an influence path from the seed node to it under our (partial) edge-level feedback. We adopt the greedy oracle designed in [22] as an offline oracle. For the tabular case, we simply use the one-hot representations as features.

CUCB(IC), IMLinUCB(IC), and IMFBI(IC). Recall that if the probabilities of each node are the same, the diffusion process under DC and IC are equivalent. In this special case, we can compare with the state-of-the-art IC-based algorithms CUCB [25], IMLinUCB [27] and IMFBI [30]. For the offline oracle required by these algorithms, we adopt the greedy algorithm [12] as is in our DC-UCB.

¹The code is available at <https://github.com/fangkongx/OIM-DC>.

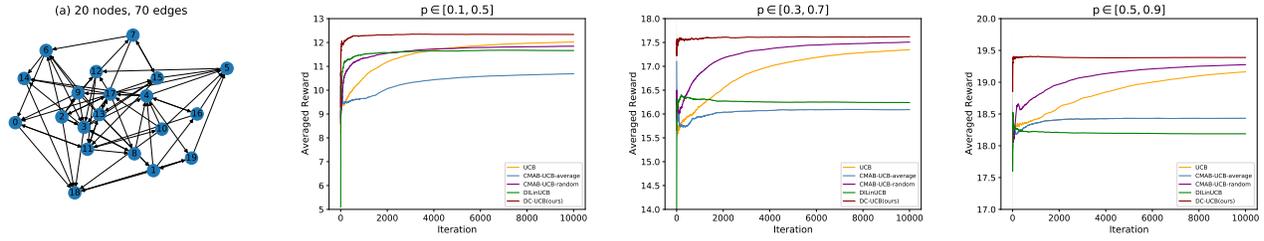


Figure 1: Comparison of DC-UCB with UCB, CMAB-UCB-average, CMAB-UCB-random and DILinUCB on synthetic networks with different activation probabilities. The performance is evaluated in averaged reward (the cumulative reward divides by the number of iterations). The standard error is represented in gray shades and all results are averaged over 10 independent runs.

	NetHEPT [0.1,0.5]	NetHEPT [0.3,0.7]	NetHEPT [0.5,0.9]	Flickr [0.1,0.5]	Flickr [0.3,0.7]	Flickr [0.5,0.9]
DC-UCB(ours)	209.20	228.44	258.01	278.16	295.48	310.44
DILinUCB	179.17	208.27	229.78	263.45	283.36	298.60
CMAB-UCB-average	119.73	146.01	160.35	200.19	220.18	230.93
CMAB-UCB-random	174.47	209.62	227.17	252.03	274.75	284.76

Table 1: Comparison of DC-UCB with DILinUCB, CMAB-UCB-average and CMAB-UCB-random on NetHEPT and Flickr datasets with different activation probabilities. The performance is evaluated in averaged reward. All algorithms run for 10k rounds and all results are averaged over 10 independent runs. The UCB algorithm is not included due to the exponential number of actions.

5.1 Synthetic Network

In this experiment, we compare the performance of our DC-UCB algorithm with other baselines on the synthetic network with different activation probabilities.

For the synthetic network, we randomly generate an Erdős-Rényi graph with $p = 0.2$ for $n = 20$ nodes. The resulting network contains $m = 70$ edges, as shown in Figure 1(a). For this network, we set up three groups of activation probabilities falling into different intervals. Specifically, each activation probability is uniformly sampled from $[0.1, 0.5]$, $[0.3, 0.7]$, $[0.5, 0.9]$ in three groups, respectively. Different values of activation probabilities correspond to different ability levels of the social network to spread information. Thus the performance of algorithms in these settings could represent their learning ability in different environments. To maintain the decreasing property of the DC model, the probability sequence of each node is then sorted in decreasing order. The seed set cardinality is set to $K = 2$ under all three settings.

We compare the performance of our DC-UCB with UCB, CMAB-UCB-average, CMAB-UCB-random and DILinUCB, when solving the OIM problem under DC. All algorithms run for $T = 10k$ rounds. The averaged rewards of those algorithms defined as the cumulative reward divides by the number of iterations are shown in Figure 1(b)(c)(d). All results are averaged over 10 independent runs.

Benefitting from the careful consideration of the decreasing property in DC, our DC-UCB achieves the best performance over all baselines in three settings. The baselines UCB and CMAB-UCB-random also have comparable performance finally, but they converge much slower. This is because these two algorithms need to evaluate the reward of each seed set/node, which costs a lot of rounds to get accurate estimations. Especially the baseline UCB

is not realistic to apply in larger networks due to the exponential number of seed sets to be evaluated. Other two baselines are at least 5.77% (7.83%, 5.21%) lower than ours in group $[0.1, 0.5]$ ($[0.3, 0.7]$, $[0.5, 0.9]$, respectively).

5.2 Real Networks

We then compare the performance of our DC-UCB with other related baselines on two real networks, NetHEPT² and Flickr³.

The original NetHEPT (Flickr) dataset contains 27, 770 (105, 938) nodes and 352, 807 (2, 316, 948, respectively) edges. Since it will be easier for the learning algorithm to identify the optimal seed set in the original sparse graph, here we extract a relatively dense one to make the learning task more challenging. We first select nodes whose degree (*in-degree* plus *out-degree*) is in $[20, 120]$, then randomly select 10 nodes among them and keep all edges that have a start node or end node in these 10 nodes as an intermediate graph. The largest connected subgraph of it forms our final network. The resulting subgraph of NetHEPT is composed of $n = 323$ nodes and $m = 3, 478$ edges, the subgraph of Flickr is composed of $n = 319$ nodes and $m = 5, 904$ edges.

We again set up three different groups with activation probabilities falling into intervals $[0.1, 0.5]$, $[0.3, 0.7]$, $[0.5, 0.9]$. The probability sequence for each node is then sorted in decreasing order to maintain the decreasing property of DC. The seed set cardinality is set to $K = 5$ in both subgraphs of NetHEPT and Flickr under three different settings.

We compare the performance of our DC-UCB with DILinUCB, CMAB-UCB-average and CMAB-UCB-random. Since UCB needs

²<https://snap.stanford.edu/data/cit-HepTh.html>

³<https://snap.stanford.edu/data/web-flickr.html>

	NetHEPT(0.2)	NetHEPT(0.5)	NetHEPT(0.8)	Flickr(0.2)	Flickr(0.5)	Flickr(0.8)
DC-UCB(ours)	137.43	230.96	256.99	234.72	292.02	309.32
CUCB(IC)	132.01	227.40	252.76	223.31	283.82	304.20
IMLinUCB(IC)	130.64	224.25	251.87	231.36	284.55	305.18
IMFB (IC)	89.58	211.47	235.60	178.96	238.24	261.57
DILinUCB	131.75	204.78	232.31	220.98	277.02	288.89
CMAB-UCB-average	68.45	131.51	160.16	157.48	219.34	251.20
CMAB-UCB-random	95.71	192.71	228.35	207.54	268.89	289.17

Table 2: Comparisons of DC-UCB with CUCB(IC), IMLinUCB(IC), IMFB(IC), DILinUCB, CMAB-UCB-average and CMAB-UCB-random under homogeneous activation probabilities. Conducted on the NetHEPT and Flickr datasets with three values of p tested on each dataset. The performance is evaluated in averaged reward and all results are averaged over 10 independent runs.

to enumerate the exponential number of seed sets, here we do not include this baseline. All algorithms run for $T = 10k$ rounds. The averaged rewards of those algorithms are shown in Table 1 and all results are averaged over 10 independent runs.

Our DC-UCB algorithm again shows consistent advantages over other baselines in all six settings, which demonstrates its strong learning ability in different environments. The CMAB-UCB-random performs the second-best in the setting NetHEPT[0.3, 0.7], but is 8.24% lower than ours. In the other settings, DILinUCB performs the second-best but is at least 3.81% lower than DC-UCB. The baseline CMAB-UCB-average performs worse and more than 25.48% lower than ours in all six settings.

5.3 Homogeneous Activation Probabilities

Recall that when $p_v(i) \equiv p'_v$ for any $v \in V$ and $i \in [|N(v)|]$, the influence propagations under DC and IC are equivalent. Thus in this case, we can compare our DC-UCB with the CUCB(IC), IMLinUCB(IC) and IMFB(IC) directly.

In this experiment, we adopt the same subgraphs of real networks as Section 5.2 but with different activation probabilities. We consider the case that all activation probabilities are the same, where the information diffusions are equivalent under IC and DC. Three choices of $p \equiv 0.2, 0.5, 0.8$ are tested on each network and the seed set cardinality is set to $K = 5$ for all choices. Those three values also reflect different abilities of the social network to spread information and thus can well demonstrate the learning abilities of algorithms.

We compare our DC-UCB with CUCB(IC), IMLinUCB(IC), IMFB(IC), DILinUCB, CMAB-UCB-average and CMAB-UCB-random. The dimension is set to $d = 5$ in IMFB(IC). As for IMLinUCB(IC), since its tabular case is equivalent to CUCB(IC) and the huge number of edges could result in high computational complexity, we randomly generate a 5-dimensional feature vector for each edge as input to improve its learning efficiency. All algorithms run for $T = 10k$ rounds. The averaged rewards of those algorithms are shown in Table 2, which are averaged over 10 independent runs. Again UCB is not tested due to the exponential number of seed sets.

Our DC-UCB algorithm performs better than all baselines in six environments. These results indicate that in the setting where IC and DC are equivalent, our DC-based algorithm is more efficient than that of IC-based algorithms. The reason is that under DC, the leading probabilities of each node would receive more updates, since

those probabilities would be always observed once the node has active in-neighbors. And as the leading probabilities are much more important than the tail probabilities, the influence spread would be estimated more accurately. While under IC, each activation probability term is bound to a specific edge, thus the updates are performed in a uniform manner over all incoming edges, making the estimated influence spread less accurate when observing the same propagations with DC.

6 CONCLUSION

This work is the first to study the OIM problem under the DC model, which generalizes IC by removing its edge independence assumption. The DC model is general enough to consider the decay property and the market saturation phenomenon of real information diffusion. We propose the DC-UCB algorithm to solve this problem with rigorous regret bound guarantees. Compared with the regret order of that under IC, our regret bound is at least as good. The algorithm is tested extensively on both synthetic datasets and real datasets of NetHEPT and Flickr against several baselines. Our algorithm consistently outperforms the baselines by a significant margin which validates its practical effectiveness.

An interesting future direction is to consider the influence factor of the start node u when u tries to activate v . This remains an instance of DC but is more general than the assumption in our work. The generalization is very important as the activation probabilities between users are likely to depend on the influence ability of the start node, apart from the end node, in real diffusion problems. It will be more challenging to derive a general formulation incorporating such a factor with the decreasing and order-independence requirement of the DC model.

ACKNOWLEDGMENTS

Baoxiang Wang is partially supported by National Natural Science Foundation of China (62106213, 72150002) and Shenzhen Science and Technology Program (RCBS20210609104356063, JCYJ20210324120011032). The corresponding author Shuai Li is supported by National Key Research and Development Program of China No. 2020AAA0107600, National Natural Science Foundation of China No. 62006151 and Shanghai Sailing Program.

REFERENCES

- [1] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine learning* 47, 2-3 (2002), 235–256.
- [2] Wei Chen, Carlos Castillo, and Laks V. S. Lakshmanan. 2013. *Information and influence propagation in social networks*. Morgan & Claypool Publishers.
- [3] Wei Chen, Yajun Wang, and Yang Yuan. 2013. Combinatorial multi-armed bandit: General framework, results and applications. In *Proceedings of the 30th International Conference on Machine Learning*, 151–159.
- [4] Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. 2016. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *The Journal of Machine Learning Research* 17, 1 (2016), 1746–1778.
- [5] Wei Chen, Yifei Yuan, and Li Zhang. 2010. Scalable influence maximization in social networks under the linear threshold model. In *Proceedings of the 10th International Conference on Data Mining*, 88–97.
- [6] Amit Goyal, Francesco Bonchi, and Laks V. S. Lakshmanan. 2010. Learning influence probabilities in social networks. In *Proceedings of the 3rd ACM international conference on Web search and data mining*, 241–250.
- [7] Amit Goyal, Wei Lu, and Laks V. S. Lakshmanan. 2011. SIMPATH: An efficient algorithm for influence maximization under the linear threshold model. In *Proceedings of the 11th International Conference on Data Mining*, 211–220.
- [8] Daniel Gruhl, Ramanathan Guha, David Liben-Nowell, and Andrew Tomkins. 2004. Information diffusion through blogspace. In *Proceedings of the 13th international conference on World Wide Web*, 491–501.
- [9] Wassily Hoeffding. 1963. Probability inequalities for sums of bounded random variables. *J. Amer. Statist. Assoc.* 58, 301 (1963), 13–30.
- [10] Tad Hogg and Kristina Lerman. 2009. Stochastic models of user-contributory web sites. *arXiv preprint arXiv:0904.0016* (2009).
- [11] Kyomin Jung, Wooram Heo, and Wei Chen. 2012. IRIE: Scalable and robust influence maximization in social networks. In *Proceedings of the 12th International Conference on Data Mining*. IEEE, 918–923.
- [12] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 137–146.
- [13] David Kempe, Jon Kleinberg, and Éva Tardos. 2005. Influential nodes in a diffusion model for social networks. In *International Colloquium on Automata, Languages, and Programming*. Springer, 1127–1138.
- [14] Gueorgi Kossinets and Duncan J Watts. 2006. Empirical analysis of an evolving social network. *Science* 311, 5757 (2006), 88–90.
- [15] 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 & Data Mining*, 645–654.
- [16] Jure Leskovec, Lada A. Adamic, and Bernardo A. Huberman. 2007. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)* 1, 1 (2007), 5–es.
- [17] Shuai Li, Fang Kong, Kejie Tang, Qizhi Li, and Wei Chen. 2020. Online Influence Maximization under Linear Threshold Model. In *Advances in Neural Information Processing Systems*.
- [18] Seth A. Myers, Chenguang Zhu, and Jure Leskovec. 2012. Information diffusion and external influence in networks. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 33–41.
- [19] Praneeth Netrapalli and Sujay Sanghavi. 2012. Learning the graph of epidemic cascades. *ACM SIGMETRICS Performance Evaluation Review* 40, 1 (2012), 211–222.
- [20] Greg Ver Steeg, Rumi Ghosh, and Kristina Lerman. 2011. What stops social epidemics?. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*.
- [21] Youze Tang, Yan Chen Shi, and Xiaokui Xiao. 2015. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 1539–1554.
- [22] Sharan Vaswani, Branislav Kveton, Zheng Wen, Mohammad Ghavamzadeh, Laks V. S. Lakshmanan, and Mark Schmidt. 2017. Model-independent online learning for influence maximization. In *Proceedings of the 34th International Conference on Machine Learning*. JMLR.org, 3530–3539.
- [23] Sharan Vaswani, Laks V. S. Lakshmanan, Mark Schmidt, et al. 2015. Influence maximization with bandits. *arXiv preprint arXiv:1503.00024* (2015).
- [24] Chi Wang, Wei Chen, and Yajun Wang. 2012. Scalable influence maximization for independent cascade model in large-scale social networks. *Data Mining and Knowledge Discovery* 25, 3 (2012), 545–576.
- [25] Qinshi Wang and Wei Chen. 2017. Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications. In *Advances in Neural Information Processing Systems*, 1161–1171.
- [26] Duncan J Watts and Peter Sheridan Dodds. 2007. Influentials, networks, and public opinion formation. *Journal of consumer research* 34, 4 (2007), 441–458.
- [27] 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*, 3022–3032.
- [28] Jennifer Wortman. 2008. Viral marketing and the diffusion of trends on social networks. *Technical Reports (CIS)* (2008), 880.
- [29] Fang Wu and Bernardo A. Huberman. 2007. Novelty and collective attention. *Proceedings of the National Academy of Sciences* 104, 45 (2007), 17599–17601.
- [30] Qingyun Wu, Zhige Li, Huazheng Wang, Wei Chen, and Hongning Wang. 2019. Factorization bandits for online influence maximization. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 636–646.
- [31] 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.
- [32] Zhijian Zhang, Hong Wu, Kun Yue, Jin Li, and Weiyi Liu. 2016. Influence maximization for cascade model with diffusion decay in social networks. In *International Conference of Pioneering Computer Scientists, Engineers and Educators*, 418–427.