Online Influence Maximization under Linear Threshold Model

Shuai Li[1]*   **Fang Kong**[1]   **Kejie Tang**[1]   **Qizhi Li**[1]   **Wei Chen**[2]

[1]Shanghai Jiao Tong University   [2]Microsoft Research

{shuaili8,fangkong,tangkj00,qizhili}@sjtu.edu.cn   weic@microsoft.com

Abstract

Online influence maximization (OIM) is a popular problem in social networks to learn influence propagation model parameters and maximize the influence spread at the same time. Most previous studies focus on the independent cascade (IC) model under the edge-level feedback. In this paper, we address OIM in the linear threshold (LT) model. Because node activations in the LT model are due to the aggregated effect of all active neighbors, it is more natural to model OIM with the node-level feedback. And this brings new challenge in online learning since we only observe aggregated effect from groups of nodes and the groups are also random. Based on the linear structure in node activations, we incorporate ideas from linear bandits and design an algorithm `LT-LinUCB` that is consistent with the observed feedback. By proving group observation modulated (GOM) bounded smoothness property, a novel result of the influence difference in terms of the random observations, we provide a regret of order $\tilde{O}(\text{poly}(m)\sqrt{T})$, where m is the number of edges and T is the number of rounds. This is the first theoretical result in such order for OIM under the LT model. In the end, we also provide an algorithm `OIM-ETC` with regret bound $O(\text{poly}(m)\, T^{2/3})$, which is model-independent, simple and has less requirement on online feedback and offline computation.

1 Introduction

Social networks play an important role in spreading information in people's life. In viral marketing, companies wish to broadcast their products by making use of the network structure and characteristics of influence propagation. Specifically, they want to provide free products to the selected users (seed nodes), let them advertise through the network and maximize the purchase. There is a budget of the free products and the goal of the companies is to select the optimal seed set to maximize the influence spread. This problem is called influence maximization (IM) [19] and has a wide range of applications including recommendation systems, link prediction and information diffusion.

In the IM problem, the social network is usually modeled as a directed graph with nodes representing users and directed edges representing influence relationship between users. IM studies how to select a seed set under a given influence propagation model to maximize the influence spread when the weights are known. Independent cascade (IC) model and linear threshold (LT) model [19] are two most widely used models to characterize the influence propagation in a social network, and both models use weights on edges as model parameters.

In many real applications, however, the weights are usually unknown in advance. For example, in viral marketing, it is unrealistic to assume that the companies know the influence abilities beforehand. A possible solution is to learn those parameters from the diffusion data collected in the past [6, 36]. But this method lacks the ability of adaptive learning based on the need of influence maximization.

---

*Corresponding author

This motivates the studies on the online influence maximization (OIM) problem [28, 10, 11, 47, 49, 50, 45, 44], where the learner tries to estimate model parameters and maximize influence in an iterative manner.

The studies on OIM are based on the multi-armed bandit (MAB) problem, which is a classical online learning framework and has been well studied in the literature [27]. MAB problem is formulated as a T-round game between a learner and the environment. In each round, the learner needs to decide which action to play and the environment will then reveal a reward according to the chosen action. The objective of the learner is to accumulate as many rewards as possible. An MAB algorithm needs to deal with the tradeoff between exploration and exploitation: whether the learner should try actions that has not been explored well yet (exploration) or focus on the action with the best performance so far (exploitation). Two algorithms, the explore-then-commit (ETC) [15] and the upper confidence bound (UCB) [4], are widely followed in the stochastic MAB setting, where the reward of each action follows an unknown but fixed distribution.

Most existing works in OIM focus on IC model under edge-level feedback [10, 11, 47, 49, 50], where the information propagates independently between pairs of users and the learner can observe the liveness of individual edges as long as its source node is influenced. The independence assumption makes the formulation simple but a bit unrealistic. Often in the real scenarios, the influence propagations are correlated with each other. The LT model is usually used to model the herd behavior that a person is more likely to be influenced if many of her friends are influenced [7, 17, 20]. Thus for the LT model, it is more natural to use the node-level feedback where we only observe the node activations, since it is hard to pinpoint which neighbor or neighbors actually contribute to an activation in a herd behavior.

In this paper, we first formulate the OIM problem under the LT model with the node-level feedback and distill effective information based on the feedback. The main challenge is that only the aggregated group effect on node activations can be observed and the aggregated groups are also random. Based on the linear structure of the LT model, we incorporate the idea of linear bandits and propose the `LT-LinUCB` algorithm, whose update mechanism is consistent with the distilled information. By proving group observation modulated (GOM) bounded smoothness, a key property on the influence spread under two different weight vectors, we can bound the regret. Such a property is similar to the triggering probability modulated (TPM) bounded smoothness condition under the IC model with edge-level feedback [47], but the derivation in our case under the node-level feedback is more difficult. The regret is of order $O(\text{poly}(m)\sqrt{T}\log T)$, where m is the number of edges and T is the number of rounds. Our `LT-LinUCB` is the first OIM algorithm under the LT model that achieves the regret in this order. Finally we give `OIM-ETC` algorithm, applying to both IC and LT with node-level feedback. Though simple, it has less requirement on the observed feedback and the offline computation, and it achieves the regret bound $O(\text{poly}(m)T^{2/3}), O(\text{poly}(m)\log(T)/\Delta^2)$.

Related Work The problem of IM was first proposed as a discrete optimization problem by Kempe *et al.* [19]. Since then, various aspects of IM have been extensively studied (see [9, 31] for surveys in this area). Two most popular models in this field are the IC and LT models. The former assumes that the influence between pairs of users are independent and the latter characterizes the herd behavior. Some works [46, 18, 19, 42] study the IC model and some [12, 16, 19, 42] study the LT model. They all assume the weights on the edges are known and focus on the model properties and approximated solutions. We treat them as the *offline* setting.

When the weight vectors are unknown, Chen *et al.* [11, 47] study the problem in the *online* setting, selecting seed sets as well as learning the parameters. They study the IC model with edge-level feedback, propose CUCB algorithm and show that CUCB achieves the distribution-dependent and distribution-independent regret bounds of $O(\text{poly}(m)\log(T))$ and $O(\text{poly}(m)\sqrt{T})$ respectively. Later Wen *et al.* [49] consider the large-scale setting and assume the edge probability is a linear function of the edge's feature vector. They provide a LinUCB-based algorithm with $O(dmn\sqrt{T}\ln(T))$ worst-case regret, where d is the feature dimension and n is the number of nodes. Wu *et al.* [50] assume that each edge probability can be decomposed as the product of the influence probability of the start node and the susceptibility probability of the end node motivated by network assortativity. All these works study the IC model with edge-level feedback.

Vaswani *et al.* [44] uses a heuristic objective function for OIM and brings up a model-independent algorithm under the pairwise feedback, where a node is influenced by a seed node or not. This applies to both IC and LT and the feedback scheme is relaxed than the edge-level feedback. Unfortunately,

however, the heuristic objective has no theoretical approximation guarantee. Also, Vaswani *et al.* [45] study the IC model with node-level feedback about the estimation gap to that under the edge-level feedback but has no regret analysis. A report [43] studies the LT model with node-level feedback by optimization approaches but without theoretical guarantees. There is another work [26] studying the problem of linear multi-resource allocation, which can be formulated as a bipartite LT model. But they assume every node in the left partition (resources) is selected and the algorithm needs to assign allocations for each pair of left node and right node (tasks) representing the corresponding allocation of resources on tasks. Thus the problem is different from our OIM. The OIM problem under LT has been open for several years. We are the first to provide a reasonable formulation with an algorithm of regret $\tilde{O}(\sqrt{T})$.

OIM is a variant of combinatorial MAB (CMAB) [10, 22], where in each round the learner selects a combination of (base) arms. Most works [25, 24] study stochastic setting with the linear objective and semi-bandit feedback where the learner can observe the selected base arm's reward and the reward of the action is a linear function of these base arms' rewards. CMAB in the stochastic setting with the linear objective and bandit feedback, where only the linear reward of the selected combination can be observed, is a special case of linear bandits. In the linear bandit setting, the learner selects a vector each round and the reward is a linear function of the selected vector action [3]. The most popular method to solve it is to construct confidence ellipsoids [14, 1, 38]. There are also works [8, 13] for CMAB in the adversarial setting and bandit feedback. But OIM is different: its objective function is non-linear and is dependent on unchosen and probabilistically triggered base arms.

OIM is related to the problem of online learning with graph feedback [2] where the learner can observe the feedback of unchosen arms based on the graph structure. Though some of them study random graphs [33, 29, 21] where the set of observed arms is random, the settings are different. Under the graph feedback, the observations of unchosen arms are additional and the reward only depends on the chosen arms, while under the OIM, the additional observations also contribute to the reward. Cascading bandits [23, 30] also consider triggering on any selected list of arms and the triggering is in the order of the lists. Compared with graph feedback and OIM setting, its triggering graph is determined by the learning agent, not the adversary.

As a generalization of graph feedback, partial monitoring [5] is also related to OIM. Most works in this direction, if applied directly to the OIM setting, are inefficient due to the exponentially large action space. Lin *et al.* [32] study a combinatorial version of partial monitoring and their algorithm provides a regret of order $O(\text{poly}(m)T^{2/3}\log T)$ for OIM with LT. Our `OIM-ETC`, however, has regret bounds of $O(\text{poly}(m)T^{2/3})$ (better in the order of T) as well as a problem-dependent bound $O(\text{poly}(m)\log T)$.

2 Setting

This section describes the setting of online influence maximization (OIM) under linear threshold (LT) model. The IM problem characterizes how to choose the seed nodes to maximize the influence spread on a social network. The network is usually represented by a directed graph $G = (V, E)$ where V is the set of users and E is the set of relationships between users. Each edge e is associated with a weight $w(e) \in [0, 1]$. For example, an edge $e = (u, v) =: e_{u,v}$ could represent user v follows user u on Twitter and $w(e)$ represents the 'influence ability' of user u on user v. Denote $w = (w(e))_{e \in E}$ to be the weight vector and $n = |V|, m = |E|, D$ to be node number, edge number and the propagation diameter respectively, where the propagation diameter is defined as the length of the longest simple path in the graph. Let $N(v) = N^{\text{in}}(v)$ be the set of all incoming neighbors of v, shortened as in-neighbors.

Recall that under IC model, each edge is alive with probability equal to the associated weight independently and a node is influenced if there is a directed path connecting from a seed node in the realized graph. Compared to the IC model, the LT model does not require the strong assumption of independence and describes the joint influence of the active in-neighbors on a user, reflecting the herd behavior that often occurs in real life [7, 17, 20].

Now we describe in detail the diffusion process under the LT model. Suppose the seed set is S. In the beginning, each node is assigned with a threshold θ_v, which is independently uniformly drawn from $[0, 1]$ and characterizes the susceptibility level of node v. Denote $\theta = (\theta_v)_{v \in V}$ to be the threshold vector. Let S_τ be the set of activated nodes by the end of time τ. At time $\tau = 0$, only nodes in

the seed set are activated: $S_0 = S$. At time $\tau + 1$ with $\tau \geq 0$, for any node $v \notin S_\tau$ that has not been activated yet, it will be activated if the aggregated influence of its active in-neighbors exceeds its threshold: $\sum_{u \in N(v) \cap S_\tau} w(e_{u,v}) \geq \theta_v$. Such diffusion process will last at most D time steps. The size of the influenced nodes is denoted as $r(S, w, \theta) = |S_D|$. Let $r(S, w) = \mathbb{E}[r(S, w, \theta)]$ be the *influence spread* of seed set S where the expectation is taken over all random variables θ_v's. The IM problem is to find the seed set S with the size at most K under weight vector w to maximize the influence spread, $\max_{S \in \mathcal{A}} r(S, w)$, where $\mathcal{A} = \{S \subset V : |S| \leq K\}$ is the *action set* for the seed nodes. We also adopt the usual assumption that $\sum_{u \in N(v)} w(e_{u,v}) \leq 1$ for any $v \in V$. This assumption makes LT have an equivalent live-edge graph formulation like IC model [19, 9]. The term of graph G and seed size K will be omitted when the context is clear. Here we emphasize that the model parameters are the weights w while the threshold vector θ is not model parameter (which follows uniform distribution).

The (offline) IM is NP-hard under the LT model but it can be approximately solved [19, 42]. For a fixed weight vector w, let S_w^{Opt} be an optimal seed set and Opt_w be its corresponding influence spread: $S_w^{\text{Opt}} \in \arg\max_{S \in \mathcal{A}} r(S, w)$ and $\text{Opt}_w = r(S_w^{\text{Opt}}, w)$. Let `Oracle` be an (offline) oracle that outputs a solution given the weight vector as input. Then for $\alpha, \beta \in [0, 1]$, the `Oracle` is an (α, β)-approximation if $\mathbb{P}(r(S', w) \geq \alpha \cdot \text{Opt}_w) \geq \beta$ where $S' = \texttt{Oracle}(w)$ is a solution returned by the `Oracle` for the weight vector w. Note when $\alpha = \beta = 1$ the oracle is *exact*.

The online version is to maximize the influence spread when the weight vector (or the model parameter) $w = (w(e))_{e \in E}$ is unknown. In each round t, the learner selects a seed set S_t, receives the observations and then updates itself accordingly. For the type of observations, previous works on IC mostly assume the edge-level feedback: the learner can observe the outgoing edges of each active node [11, 49, 50]. But for the LT model, it is not very realistic to assume the learner can observe which in-neighbor influences the target user since the LT model characterizes the aggregate influence of a crowd. So we consider a more realistic *node-level feedback*[2] in this paper: the learner can only observe the influence diffusion process on node sets as $S_{t,0}, \ldots, S_{t,\tau}, \ldots$ in round t.

The objective of the OIM is to minimize the cumulative η-scaled regret [10, 49] over total T rounds:

$$R(T) = \mathbb{E}\left[\sum_{t=1}^{T} R_t\right] = \mathbb{E}\left[\eta \cdot T \cdot \text{Opt}_w - \sum_{t=1}^{T} r(S_t, w)\right], \tag{1}$$

where the expectation is over the randomness on the threshold vector and the output of the adopted offline oracle in each round .

Throughout this paper, we will use 'round' t to denote a step in online learning and use 'time' τ of round t to denote an influence diffusion step of seed set S_t in round t.

3 `LT-LinUCB` Algorithm

In this section, we show how to distill effective information based on the feedback and propose a LinUCB-type algorithm, `LT-LinUCB`, for OIM under LT. For each node $v \in V$, denote $w_v = (w(e_{u,v}))_{u \in N(v)}$ to be the weight vector of its incoming edges. Let $\chi(e_{u,v}) \in \{0, 1\}^{|N(v)|}$ be the one-hot representation of the edge $e_{u,v}$ over all of v's incoming edges $\{e_{u,v} : u \in N(v)\}$, that is its e'-entry is 1 if and only if $e' = e_{u,v}$. Then $w(e_{u,v}) = \chi(e_{u,v})^\top w_v$. For a subset of edges $E' \subseteq \{e_{u,v} : u \in N(v)\}$, we define $\chi(E') := \sum_{e \in E'} \chi(e) \in \{0, 1\}^{|N(v)|}$ to be the vector whose e-entry is 1 if and only if $e \in E'$. Here we abuse the notation that $\chi(\{e\}) = \chi(e)$. By this notation, the weight sum of the edges in E' is simply written as $\chi(E')^\top w_v$. A subset $V' \subset N(v)$ of v's in-neighbors can activate v if the weight sum of associated edges exceeds the threshold, that is $\chi(E')^\top w_v \geq \theta_v$ with $E' = \{e_{u,v} : u \in V'\}$.

Fix a diffusion process $S_0, S_1, \ldots, S_\tau, \ldots$, where the seed set is S_0. For each node v, define

$$\tau_1(v) := \min_\tau \{\tau = 0, \ldots, D : N(v) \cap S_\tau \neq \emptyset\} \tag{2}$$

---

[2]One may think of the node-level feedback as knowing only the set of nodes activated by the end of the diffusion process. We refer to this as (partial) node-level feedback and ours as (full) node-level feedback. This naming comes from [35].

4

as the earliest time step when node v has active in-neighbors. Particularly we set $\tau_1(v) = D + 1$ if node v has no active in-neighbor until the diffusion ends. For any $\tau \geq \tau_1(v)$, further define

$$E_\tau(v) := \{e_{u,v} : u \in N(v) \cap S_\tau\} \tag{3}$$

as the set of incoming edges associated with active in-neighbors of v at time step τ.

Recall that the learner can only observe the aggregated influence ability of a node's active in-neighbors. Let $\tau_2(v)$ represent the time step that node v is influenced ($\tau_2(v) > \tau_1(v)$), which is equivalent to mean that v's active in-neighbors of time $\tau_2(v) - 1$ succeed to influence it but those in time $\tau_2(v) - 2$ fail ($E_{-1} = \emptyset$). Thus the definition of $\tau_2(v)$ can be written as

$$\tau_2(v) := \left\{\tau = 0, \ldots, D : \chi(E_{\tau-2}(v))^\top w_v < \theta_v \leq \chi(E_{\tau-1}(v))^\top w_v\right\}. \tag{4}$$

For consistency, we set $\tau_2(v) = D + 1$ if node v is finally not influenced after the information diffusion ends. Then based on the definition of $\tau_1(v)$ and $\tau_2(v)$, we can obtain that node v is not influenced at time $\tau \in (\tau_1(v), \tau_2(v))$, which means that the set of active in-neighbors of v at time step $\tau - 1$ fails to activate it.

According to the rule of information diffusion under the LT model, an event that $E' \subseteq \{e_{u,v} : u \in N(v)\}$ fails to activate v is equivalent to $\chi(E')^\top w_v < \theta_v$, which happens with probability $1 - \chi(E')^\top w_v$ since θ_v is uniformly drawn from $[0, 1]$. Similarly an event that $E' \subseteq \{e_{u,v} : u \in N(v)\}$ succeeds to activate v is equivalent to $\chi(E')^\top w_v \geq \theta_v$, which happens with probability $\chi(E')^\top w_v$. So for node v who has active in-neighbors, v is not influenced at time step τ ($\tau_1(v) < \tau < \tau_2(v)$) means that the set of v's active in-neighbors by $\tau - 1$ fails to activate it, thus we can use $(\chi(E_{\tau-1}(v)), 0)$ to update our belief on the unknown weight vector w_v; v is influenced at time step $\tau_2(v)$ means that the set of v's active in-neighbors by $\tau_2(v) - 1$ succeeds to activate it, we can thus use $(\chi(E_{\tau_2(v)-1}(v)), 1)$ to update our belief on the unknown weight vector w_v; v is finally not influenced means that all of its active in-neighbors (by time step D) fail to activate it, we can use $(\chi(E_{\tau_2(v)-1}(v)), 0)$ to update w_v since $\tau_2(v)$ is defined as $D + 1$ in this case. Note all of these events are correlated (based on a same θ_v), thus we can only choose at most one of them to update w_v for node v who has active in-neighbors. If v has no active in-neighbors, we have no observation on w_v and could update nothing.
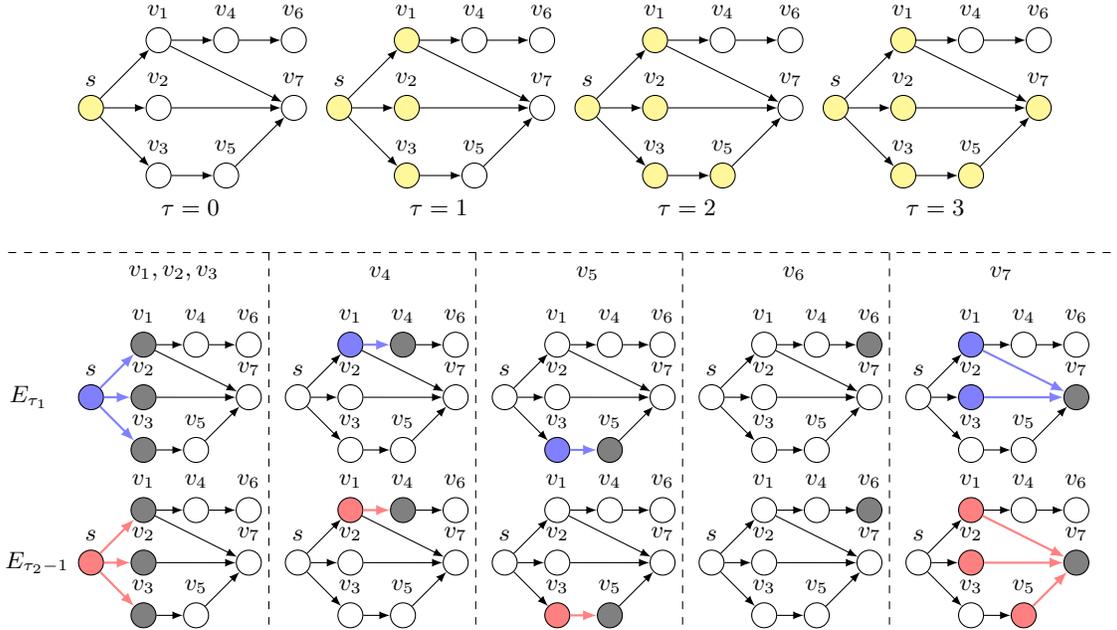

Figure 1: An example of diffusion process starting from $S = \{s\}$ under LT. The upper part describes an influence diffusion process where yellow nodes represent influenced nodes by the current time. The lower part describes what E_{τ_1}, E_{τ_2-1} are where we use blue (red) color to represent the edges and the associated active in-neighbors in E_{τ_1} (E_{τ_2-1}, respectively) for the objective black node.

Figure 1 gives an example of diffusion process and the definitions of edge-sets E_{τ_1} and E_{τ_2-1}. The diffusion process is illustrated by the upper four figures, where the set S_τ of influenced nodes by time τ is yellow colored. The lower five columns represent the sets E_{τ_1}, E_{τ_2-1} for different nodes. For example, node v_7 has active in-neighbors starting from $\tau = 1$, thus $\tau_1(v_7) = 1$ and $E_{\tau_1(v_7)}(v_7) = \{e_{u,v_7} : u \in N(v_7) \cap S_1\} = \{e_{v_1,v_7}, e_{v_2,v_7}\}$. And v_7 is influenced at $\tau = 3$ thus $\tau_2(v_7) = 3$ and $E_{\tau_2(v_7)-1}(v_7) = \{e_{u,v_7} : u \in N(v_7) \cap S_2\} = \{e_{v_1,v_7}, e_{v_2,v_7}, e_{v_5,v_7}\}$. Node v_6 has no active in-neighbors, thus $\tau_1(v_6) = \tau_2(v_6) = D + 1$, both its $E_{\tau_1(v_6)}(v_6)$ and $E_{\tau_2(v_6)-1}(v_6)$ are empty sets.

The above describes how to distill key observations for a diffusion under the LT model and also explains the update rule in the design of the algorithm. Denote τ_1, τ_2, E_τ at round t as $\tau_{t,1}, \tau_{t,2}, E_{t,\tau}$ and the diffusion process at round t as $S_{t,0}, \ldots, S_{t,\tau}, \ldots$. Here we abuse a bit the notation S to represent both the seed set and the spread set in a round when the context is clear.

Our algorithm LT-LinUCB is given in Algorithm 1. It maintains the Gramian matrix M_v and the moment vector b_v of regressand by regressors to store the information for w_v. At each round t, the learner first computes the confidence ellipsoid for w_v based on the current information (line 4) (see the following lemma).

Lemma 1. *Given $\{(A_t, y_t)\}_{t=1}^\infty$ with $A_t \in \{0,1\}^N$ and $y_t \in \{0,1\}$ as a Bernoulli random variable with $\mathbb{E}\left[y_t \mid A_1, y_1, \ldots, A_{t-1}, y_{t-1}, A_t\right] = A_t^\top w_v$, let $M_t = I + \sum_{s=1}^t A_s A_s^\top$ and $\hat{w}_t = M_t^{-1}\left(\sum_{s=1}^t A_s y_s\right)$ be the linear regression estimator. Then with probability at least $1 - \delta$, for all $t \geq 1$, it holds that w_v lies in the confidence set*

$$\tilde{\mathcal{C}}_t := \left\{ w' \in [0,1]^N : \|w' - \hat{w}_t\|_{M_t} \leq \sqrt{N \log(1 + tN) + 2 \log \frac{1}{\delta}} + \sqrt{N} \right\}.$$

Algorithm 1 LT-LinUCB

1: **Input:** Graph $G = (V, E)$; seed set cardinality K; exploration parameter $\rho_{t,v} > 0$ for any t, v; offline oracle PairOracle
2: **Initialize:** $M_{0,v} \leftarrow I \in \mathbb{R}^{|N(v)| \times |N(v)|}, b_{0,v} \leftarrow 0 \in \mathbb{R}^{|N(v)| \times 1}, \hat{w}_{0,v} \leftarrow 0 \in \mathbb{R}^{|N(v)| \times 1}$ for any node $v \in V$
3: **for** $t = 1, 2, 3, \ldots$ **do**
4:     Compute the confidence ellipsoid $\mathcal{C}_{t,v} = \left\{ w'_v \in [0,1]^{|N(v)| \times 1} : \|w'_v - \hat{w}_{t-1,v}\|_{M_{t-1,v}} \leq \rho_{t,v} \right\}$ for any node $v \in V$
5:     Compute the pair (S_t, w_t) by PairOracle with confidence set $\mathcal{C}_t = \{\mathcal{C}_{t,v}\}_{v \in V}$ and seed set cardinality K
6:     Select the seed set S_t and observe the feedback
7:     // Update
8:     **for** node $v \in V$ **do**
9:         Initialize $A_{t,v} \leftarrow 0 \in \mathbb{R}^{|N(v)| \times 1}$, $y_{t,v} \leftarrow 0 \in \mathbb{R}$
10:        Uniformly randomly choose $\tau \in \{\tau' : \tau_{t,1}(v) \leq \tau' \leq \tau_{t,2}(v) - 1\}$
11:        **if** v is influenced and $\tau = \tau_{t,2}(v) - 1$ **then**
12:            $A_{t,v} = \chi\left(E_{t,\tau}(v)\right), \ y_{t,v} = 1$
13:        **else if** $\tau = \tau_1(v), \ldots, \tau_2(v) - 2$ or $\tau = \tau_2(v) - 1$ but v is not influenced **then**
14:            $A_{t,v} = \chi\left(E_{t,\tau}(v)\right), \ y_{t,v} = 0$
15:        **end if**
16:        $M_{t,v} \leftarrow M_{t-1,v} + A_{t,v}A_{t,v}^\top, \ b_{t,v} \leftarrow b_{t-1,v} + y_{t,v}A_{t,v}, \ \hat{w}_{t,v} = M_{t,v}^{-1}b_{t,v}$
17:     **end for**
18: **end for**

This lemma is a direct corollary of [1, Theorem 2] for the concentration property of the weight vector w_v. Thus when $\rho_{t,v} \geq \sqrt{|N(v)| \log(1 + t|N(v)|) + 2 \log \frac{1}{\delta}} + \sqrt{|N(v)|}$, the true weight vector w_v lies in the confidence set $\mathcal{C}_{t,v}$ (line 4) for any t with probability at least $1 - \delta$.

Given the confidence set $\mathcal{C}_v$ for w_v, the algorithm expects to select the seed set by solving the *weight-constrained influence maximization* (WCIM) problem

$$\operatorname{argmax}_{(S,w'):S \in \mathcal{A}, w' \in \mathcal{C}} r(S, w'). \tag{5}$$

This (offline) optimization problem turns out to be highly nontrivial. Since we want to focus more on the online learning solution, we defer the full discussion on the offline optimization, including its general difficulty and our proposed approximate algorithms for certain graph classes such as directed acyclic graphs to Appendix B.

Suppose its best solution is $(S_\mathcal{C}^{\mathrm{POpt}}, w_\mathcal{C}^{\mathrm{POpt}})$ where 'P' stands for 'pair'. Let `PairOracle` be an offline oracle to solve the optimization problem. We say `PairOracle` is an (α, β)-approximation oracle if $\mathbb{P}\left(r(S', w') \geq \alpha \cdot r(S_\mathcal{C}^{\mathrm{POpt}}, w_\mathcal{C}^{\mathrm{POpt}}) \right) \geq \beta$ where (S', w') is an output by the oracle when the confidence set is $\mathcal{C}$. Then the algorithm runs with the seed set output by the `PairOracle` and the confidence set $\mathcal{C}_t = \{\mathcal{C}_{t,v}\}_{v \in V}$ (line 5).

After observing the diffusion process (line 6), For each node v who has active in-neighbors, we randomly choose its active in-neighbors at time step $\tau_1(v), \ldots, \tau_2(v) - 1$ to update (line 10). Specifically, if v is influenced and $\tau = \tau_2(v) - 1$, then it means that the set of active in-neighbors at time step τ succeeds to activate v, thus we use $(\chi(E_{t,\tau}(v)), 1)$ to update (line 12); if $\tau = \tau_1(v), \ldots, \tau_2(v) - 2$ or $\tau = \tau_2(v) - 1$ but node v is not influenced, it means that the set of active in-neighbors at τ fail to activate node v, thus we use $(\chi(E_{t,\tau}(v)), 0)$ to update (line 14). These updates are consistent with the distilled observations we get for nodes who have active in-neighbors. For node v who has no active in-neighbors, we have no obervation on w_v and not update on it since the set $\{\tau' : \tau_1(v) \leq \tau' \leq \tau_2(v) - 1\}$ is an empty set in this case.

For example in Figure 1, node v_7 has active in-neighbors from $\tau_1(v_7) = 1$ and is influenced at $\tau_2(v_7) = 3$. The `LT-LinUCB` will uniformly randomly choose $\tau \in \{1, 2\}$ (line 10). It updates $(A_{v_7} = \chi(E_1(v_7)), y_{v_7} = 0)$ if $\tau = 1$ (line 14) and $(A_{v_7} = \chi(E_2(v_7)), y_{v_7} = 1)$ otherwise (line 12). For nodes v_1, v_2, v_3, they all have $\tau_1 = 0$ and $\tau_2 = 1$. Thus for these three nodes, the algorithm chooses $\tau = 0$ (line 10) and updates $(A_v = \chi(E_0(v)), y_v = 1)$ (line 12). Node v_4 has active in-neighbors from $\tau_1(v_4) = 1$ but is not influenced finally, the algorithm will randomly choose $\tau \in \{1, 2 \ldots, D\}$ and update $(A_{v_4} = \chi(E_\tau(v_4)), y_{v_4} = 0)$ (line 14). Node v_6 has no active in-neighbors, so we have no observation for its weight vector and will not update on it.

3.1 Regret Analysis

We now provide the group observation modulated (GOM) bounded smoothness property for LT model, an important relationship of the influence spreads under two weight vectors. It plays a crucial role in the regret analysis and states that the difference of the influence spread $r(S, w)$ under two weight vectors can be bounded in terms of the weight differences of the distilled observed edge sets under one weight vector. It is conceptually similar to the triggering probability modulate (TPM) bounded smoothness condition under the IC model with edge-level feedback [47], but its derivation and usage are quite different. For the seed set S, define the set of all nodes related to a node v, $V_{S,v}$, to be the set of nodes that are on any path from S to v in graph G.

Theorem 1. *(GOM bounded smoothness) For any two weight vectors $w, w' \in [0, 1]^m$ with $\sum_{u \in N(v)} w(e_{u,v}) \leq 1$, the difference of their influence spread for any seed set S can be bounded as*

$$|r(S, w') - r(S, w)| \leq \mathbb{E}\left[\sum_{v \in V \setminus S} \sum_{u \in V_{S,v}} \sum_{\tau = \tau_1(u)}^{\tau_2(u)-1} \left| \sum_{e \in E_\tau(u)} (w'(e) - w(e)) \right| \right], \qquad (6)$$

where the definitions of $\tau_1(u), \tau_2(u)$ and $E_\tau(u)$ are all under weight vector w, and the expectation is taken over the randomness of the thresholds on nodes.

This theorem connects the reward difference with weight differences on the distilled observations, which are also the information used to update the algorithm (line 7-17). It links the effective observations, updates of the algorithm and the regret analysis. The proof needs to deal with intricate dependency among activation events, and is put in Appendix A.1. due to the space constraint.

For seed set $S \in \mathcal{A}$ and node $u \in V \setminus S$, define $N_{S,u} := \sum_{v \in V \setminus S} \mathbb{1}\{u \in V_{S,v}\} \leq n - K$ to be the number of nodes that u is relevant to. Then for the vector $N_S = (N_{S,u})_{u \in V}$, define the upper bound of its L^2-norm over all feasible seed sets

$$\gamma(G) := \max_{S \in \mathcal{A}} \sqrt{\sum_{u \in V} N_{S,u}^2} \leq (n - K)\sqrt{n} = O(n^{3/2}),$$

Algorithm 2 `OIM-ETC`

1: **Input:** $G = (V, E)$, seed size K, exploration budget k, time horizon T, offline oracle `Oracle`
2: **for** $s \in [k], u \in V$ **do**
3:     Choose $\{u\}$ as the seed set
4:     $X_s(e_{u,v}) := \mathbb{1}\{v$ is activated$\}$ for any $v \in N^{\text{out}}(u)$
5: **end for**
6: Compute $\hat{w}(e) := \frac{1}{k} \sum_{s=1}^{k} X_s(e)$ for any $e \in E$
7: $\hat{S} = \text{Oracle}(\hat{w})$
8: **for** the remaining $T - nk$ rounds **do**
9:     Choose $\hat{S}$ as the seed set
10: **end for**

---

which is a constant related to the graph. Then we have the following regret bound.

Theorem 2. *Suppose the* `LT-LinUCB` *runs with an* (α, β)*-approximation* `PairOracle` *and parameter* $\rho_{t,v} = \rho_t = \sqrt{n \log(1 + tn) + 2 \log \frac{1}{\delta}} + \sqrt{n}$ *for any node* $v \in V$. *Then the* $\alpha\beta$*-scaled regret satisfies*

$$R(T) \le 2\rho_T \gamma(G) D \sqrt{mnT \log(1 + T) / \log(1 + n)} + n\delta \cdot T(n - k). \tag{7}$$

When $\delta = 1/(n\sqrt{T})$, $R(T) \le C \cdot \gamma(G) \, Dn\sqrt{mT} \log(T)$ *for some universal constant* C.

Due to space limits, the proof and the detailed discussions, as well as the values of $\gamma(G)$, are put in Appendix A.

4 The Explore-then-Commit Algorithm

This section presents the explore-then-commit (ETC) algorithm for OIM. Though simple, it is efficient and model independent, applying to both LT and IC model with less requirement on feedback and offline computation.

Recall that under LT model, a node v is activated if the sum of weights from active in-neighbors exceeds the threshold θ_v, which is uniformly drawn from $[0, 1]$. Since the feedback is node-level, if the activated node v has more than one active in-neighbors, then we can only observe the group influence effect of her active in-neighbors instead of each single in-neighbor. A simple way to overcome this limitation and manage to observe directly the single weight $w(e_{u,v})$ is to select a single seed $\{u\}$ and take only the first step influence as feedback, which formulates our `OIM-ETC` algorithm (Algorithm 2), representing the ETC algorithm of the OIM problem.

Our `OIM-ETC` takes the exploration budget k as input parameter such that each node u is selected as the (single) seed for k rounds (line 3). For each round in which u is the seed, each outgoing neighbor (shortened as out-neighbor) $v \in N^{\text{out}}(u)$ will be activated in the first step with probability $\mathbb{P}(w(e_{u,v}) > \theta_v) = w(e_{u,v})$ since the threshold θ_v is independently uniformly drawn from $[0, 1]$. Thus the first-step node-level feedback is actually edge-level feedback and we can observe the independent edges from the first-step feedback (line 4). Since each node is selected k times, we have k observations of Bernoulli random variables with expectation $w(e_{u,v})$ in this *exploration* phase. Then we take the empirical estimate $\hat{w}(e)$ for each $w(e)$ (line 6) after the exploration and run with the seed set output by the offline `Oracle` (line 7) for the remaining $T - nk$ *exploitation* rounds (line 9). We assume the offline `Oracle` is (α, β)-approximation.

Since it only needs the first step of the diffusion process and calls only once of the usual IM oracle, it is efficient and has less requirement. By selecting reasonable k, we can derive good regret bounds. Before that we need two definitions.

Definition 1. *(Bad seed set) A seed set* S *is bad if* $r(S, w) < \alpha \cdot \text{Opt}_w$. *The set of bad seed sets is* $\mathcal{S}_B := \{S \mid r(S, w) < \alpha \cdot \text{Opt}_w\}$.

Definition 2. *(Gaps of bad seed sets) For a bad seed set* $S \in \mathcal{S}_B$, *its gap is defined as* $\Delta_S := \alpha \cdot \text{Opt}_w - r(S, w)$. *The maximum and minimum gap are defined as*

$$\Delta_{\max} := \alpha \cdot \text{Opt}_w - \min\{r(S, w) \mid S \in \mathcal{S}_B\}, \tag{8}$$

$$\Delta_{\min} := \alpha \cdot \text{Opt}_w - \max\{r(S, w) \mid S \in \mathcal{S}_B\}. \tag{9}$$

Theorem 3. *When* $k = \max\left\{1, \frac{2m^2n^2}{\Delta_{\min}^2} \ln\left(\frac{T\Delta_{\min}^2}{mn^3}\right)\right\}$, *the* $\alpha\beta$-*scaled regret bound of our* `OIM-ETC` *algorithm over* T *rounds satisfies*

$$R(T) \leq \min\left\{T\Delta_{\max}, n\Delta_{\max} + \frac{2m^2n^3\Delta_{\max}}{\Delta_{\min}^2}\left(1 + \max\left\{0, \ln\left(\frac{T\Delta_{\min}^2}{mn^3}\right)\right\}\right)\right\}$$

$$= O\left(\frac{m^2n^3\Delta_{\max}}{\Delta_{\min}^2}\ln(T)\right). \tag{10}$$

When $k = 3.9(m^2T/n)^{2/3}$, *the* $\alpha\beta$-*scaled regret bound of* `OIM-ETC` *algorithm over* T *rounds satisfies*

$$R(T) \leq 3.9(mn)^{4/3}T^{2/3} + 1 = O\left((mn)^{4/3}T^{2/3}\right). \tag{11}$$

The proof of the problem-dependent bound follows routine ideas of ETC algorithms but the proof of the problem-independent bound is new. The proofs and discussions are put in Appendix C.

5   Conclusion

In this paper, we formulate the problem of OIM under LT model with node-level feedback and design how to distill effective information from observations. We prove a novel GOM bounded smoothness property for the spread function, which relates the limited observations, algorithm updates and the regret analysis. We propose `LT-LinUCB` algorithm, provide rigorous theoretical analysis and show a competitive regret bound of $O(\text{poly}(m)\sqrt{T}\ln(T))$. Our `LT-LinUCB` is the first algorithm for LT model with such regret order. Besides, we design `OIM-ETC` algorithm with theoretical analysis on its distribution-dependent and distribution-independent regret bounds. The algorithm is efficient, applies to both LT and IC models, and has less requirements on feedback and offline computation.

In studying the OIM with LT model, we encounter an optimization problem of weight-constrained influence maximization (WCIM). Reconsidering an (offline) optimization problem by relaxing some fixed parameter to elements of a convex set is expected to be common in online learning. So we believe this problem could have independent interest. Also the OIM problem under IC model with node-level feedback is an interesting future work. Our regret analysis goes through thanks to the linearity of the LT model. But the local triggering is nonlinear for IC model, and thus we expect more challenges in the design and analysis of IC model with node-level feedback. Applying Thompson sampling to influence maximization is also an interesting future direction, but it could also be challenging, since it may not work well with offline approximation oracles as pointed out in [48].

Acknowledgement

References

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

[2] Noga Alon, Nicolo Cesa-Bianchi, Ofer Dekel, and Tomer Koren. Online learning with feedback graphs: Beyond bandits. In *Annual Conference on Learning Theory*, volume 40. Microtome Publishing, 2015.

[3] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, 2002.

[4] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multi-armed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[5] Gábor Bartók, Dean P Foster, Dávid Pál, Alexander Rakhlin, and Csaba Szepesvári. Partial monitoring—classification, regret bounds, and algorithms. *Mathematics of Operations Research*, 39(4):967–997, 2014.

[6] Simon Bourigault, Sylvain Lamprier, and Patrick Gallinari. Representation learning for information diffusion through social networks: An embedded cascade model. In *Proceedings of the 9th ACM international conference on Web Search and Data Mining*, pages 573–582, 2016.

[7] Damon Centola and Michael Macy. Complex contagions and the weakness of long ties. *American journal of Sociology*, 113(3):702–734, 2007.

[8] Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

[9] Wei Chen, Laks V. S. Lakshmanan, and Carlos Castillo. *Information and Influence Propagation in Social Networks*. Morgan & Claypool Publishers, 2013.

[10] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework, results and applications. In *Proceedings of the 30th International Conference on Machine Learning*, pages 151–159, 2013.

[11] Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *The Journal of Machine Learning Research*, 17(1):1746–1778, 2016.

[12] Wei Chen, Yifei Yuan, and Li Zhang. Scalable influence maximization in social networks under the linear threshold model. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, pages 88–97, 2010.

[13] Richard Combes, Mohammad Sadegh Talebi Mazraeh Shahi, Alexandre Proutiere, et al. Combinatorial bandits revisited. In *Advances in Neural Information Processing Systems*, pages 2116–2124, 2015.

[14] Varsha Dani, Thomas P Hayes, and Sham M Kakade. *Stochastic Linear Optimization under Bandit Feedback*. Citeseer, 2008.

[15] Aurélien Garivier, Tor Lattimore, and Emilie Kaufmann. On explore-then-commit strategies. In *Advances in Neural Information Processing Systems*, pages 784–792, 2016.

[16] Amit Goyal, Wei Lu, and Laks V. S. Lakshmanan. SIMPATH: An efficient algorithm for influence maximization under the linear threshold model. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, pages 211–220, 2011.

[17] Mark Granovetter. Threshold models of collective behavior. *American journal of sociology*, 83(6):1420–1443, 1978.

[18] Kyomin Jung, Wooram Heo, and Wei Chen. IRIE: Scalable and robust influence maximization in social networks. In *Proceedings of the 2012 IEEE 12th International Conference on Data Mining*, pages 918–923, 2012.

[19] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146, 2003.

[20] Elias Boutros Khalil, Bistra Dilkina, and Le Song. Scalable diffusion-aware optimization of network topology. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1226–1235, 2014.

[21] Tomáš Kocák, Gergely Neu, and Michal Valko. Online learning with erdős-rényi side-observation graphs. In *Proceedings of the 32nd Conference on Uncertainty in Artificial Intelligence*, pages 339–346, 2016.

[22] Fang Kong, Qizhi Li, and Shuai Li. Survey on online influence maximization. *Computer Science*, 47(5):7–13, 2020.

[23] Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. Cascading bandits: Learning to rank in the cascade model. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 767–776, 2015.

[24] Branislav Kveton, Zheng Wen, Azin Ashkan, Hoda Eydgahi, and Brian Eriksson. Matroid bandits: Fast combinatorial optimization with learning. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*, pages 420–429, 2014.

[25] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight Regret Bounds for Stochastic Combinatorial Semi-Bandits. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, pages 535–543, 2015.

[26] Tor Lattimore, Koby Crammer, and Csaba Szepesvári. Linear multi-resource allocation with semi-bandit feedback. In *Advances in Neural Information Processing Systems*, pages 964–972, 2015.

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

[28] Siyu Lei, Silviu Maniu, Luyi Mo, Reynold Cheng, and Pierre Senellart. Online influence maximization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 645–654, 2015.

[29] Shuai Li, Wei Chen, Zheng Wen, and Kwong-Sak Leung. Stochastic online learning with probabilistic graph feedback. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2020.

[30] Shuai Li, Baoxiang Wang, Shengyu Zhang, and Wei Chen. Contextual combinatorial cascading bandits. In *Proceedings of the 33rd International Conference on Machine Learning*, pages 1245–1253, 2016.

[31] Yuchen Li, Ju Fan, Yanhao Wang, and Kian-Lee Tan. Influence maximization on social graphs: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1852–1872, 2018.

[32] Tian Lin, Bruno Abrahao, Robert Kleinberg, John Lui, and Wei Chen. Combinatorial partial monitoring game with linear feedback and its applications. In *Proceedings of the 31st International Conference on Machine Learning*, pages 901–909, 2014.

[33] Fang Liu, Swapna Buccapatnam, and Ness Shroff. Information directed sampling for stochastic bandits with graph feedback. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, 2018.

[34] Elchanan Mossel and Sebastien Roch. Submodularity of influence in social networks: From local to global. *SIAM Journal on Computing*, 39(6):2176–2188, 2010.

[35] Harikrishna Narasimhan, David C Parkes, and Yaron Singer. Learnability of influence in networks. In *Advances in Neural Information Processing Systems*, pages 3186–3194, 2015.

[36] Praneeth Netrapalli and Sujay Sanghavi. Learning the graph of epidemic cascades. In *Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems*, pages 211–222, 2012.

[37] Parikshit Ram and Alexander G Gray. Maximum inner-product search using cone trees. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 931–939, 2012.

[38] Paat Rusmevichientong and John N Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

[39] Fumin Shen, Wei Liu, Shaoting Zhang, Yang Yang, and Heng Tao Shen. Learning binary codes for maximum inner product search. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4148–4156, 2015.

[40] Anshumali Shrivastava and Ping Li. Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips). In *Advances in Neural Information Processing Systems*, pages 2321–2329, 2014.

[41] Anthony Man-Cho So, Yinyu Ye, and Jiawei Zhang. A unified theorem on sdp rank reduction. *Mathematics of Operations Research*, 33(4):910–920, 2008.

[42] Youze Tang, Yanchen Shi, and Xiaokui Xiao. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1539–1554, 2015.

[43] Sharan Vaswani and Nayantara Duttachoudhury. Learning influence diffusion probabilities under the linear threshold model. *Github pages*, 2013. https://vaswanis.github.io/social_networks_report.pdf.

[44] Sharan Vaswani, Branislav Kveton, Zheng Wen, Mohammad Ghavamzadeh, Laks V. S. Lakshmanan, and Mark Schmidt. Model-independent online learning for influence maximization. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3530–3539, 2017.

[45] Sharan Vaswani, Laks V. S. Lakshmanan, Mark Schmidt, et al. Influence maximization with bandits. *arXiv preprint arXiv:1503.00024*, 2015.

[46] Chi Wang, Wei Chen, and Yajun Wang. Scalable influence maximization for independent cascade model in large-scale social networks. *Data Mining and Knowledge Discovery*, 25(3):545–576, 2012.

[47] Qinshi Wang and Wei Chen. Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications. In *Advances in Neural Information Processing Systems*, pages 1161–1171, 2017.

[48] Siwei Wang and Wei Chen. Thompson sampling for combinatorial semi-bandits. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5114–5122, 2018.

[49] Zheng Wen, Branislav Kveton, Michal Valko, and Sharan Vaswani. Online influence maximization under independent cascade model with semi-bandit feedback. In *Advances in Neural Information Processing Systems*, pages 3022–3032, 2017.

[50] Qingyun Wu, Zhige Li, Huazheng Wang, Wei Chen, and Hongning Wang. Factorization bandits for online influence maximization. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 636–646, 2019.

A Analysis and Discussions of `LT-LinUCB`

A.1 Proof of Theorem 1

Let $r_S^v(w)$ be the probability that node v will be influenced under the weight vector w when the seed set is S. Then

$$|r(S, w') - r(S, w)|$$
$$\leq \sum_{v \in V \setminus S} |r_S^v(w') - r_S^v(w)|$$
$$= \sum_{v \in V \setminus S} \mathbb{E}_{\theta \sim (\mathcal{U}[0,1])^n} \left[\mathbb{1}\{v \text{ is influenced under } w', \theta\} \neq \mathbb{1}\{v \text{ is influenced under } w, \theta\} \right],$$

where we use $\mathcal{U}[0,1]$ to denote the uniform distribution on the interval $[0, 1]$. The reason that the activation of v is different under w and w' must be that during the propagation from S to v, at some step τ and some node $u \in V_{S,v}$, the activation of u is different. We enumerate $u \in V_{S,v}$ and enumerate τ from 1 to D to bound the above probability. Recall that D is the propagation diameter. Henceforth in this section, parameters w, w', S, and v are all fixed. All the randomness comes from $\theta \sim (\mathcal{U}[0,1])^n$, and once θ is determined, the diffusion process is determined. Thus, we could assume that every event is a subset of $[0, 1]^n$. Define the following event, given the seed set S and target node v:

$$\mathcal{E}_0 = \{\theta \mid \mathbb{1}\{v \text{ is influenced under } w', \theta\} \neq \mathbb{1}\{v \text{ is influenced under } w, \theta\}\}.$$

Thus

$$|r(S, w') - r(S, w)| \leq \sum_{v \in V \setminus S} \Pr_{\theta \sim (\mathcal{U}[0,1])^n} \{\mathcal{E}_0\}. \tag{12}$$

Let $\Phi(w, \theta) = (S_0 = S, S_1, \dots, S_D)$ be the sequence of activation sets given weight factor w and threshold factor θ. Let $\Phi_i(w, \theta) = S_i$ be the set of nodes activated by time step i. For every node $u \in V_{S,v}$, we define the event that u is the first node that has different activation under w and w'.

$$\mathcal{E}_1(u) = \{\theta \mid \exists \tau \in [D], \forall \tau' < \tau, \Phi_{\tau'}(w, \theta) = \Phi_{\tau'}(w', \theta),$$
$$u \in (\Phi_\tau(w, \theta) \setminus \Phi_\tau(w', \theta)) \cup (\Phi_\tau(w', \theta) \setminus \Phi_\tau(w, \theta))\}.$$

It is clear that

$$\mathcal{E}_0 \subseteq \bigcup_{u \in V_{S,v}} \mathcal{E}_1(u). \tag{13}$$

Note that for each node $u \in V_{S,v}$, u may be activated at different time steps from different paths, or not activated at all. Thus, the fact that u is not activated at one time step may have implications on u's activations at other time steps, and thus we need to carefully classify the activation of u in order to bound the probability of $\mathcal{E}_1(u)$. Define the following events for each $\tau \in [D]$:

$$\mathcal{E}_{2,0}(u, \tau) = \{\theta \mid \forall \tau' < \tau, \Phi_{\tau'}(w, \theta) = \Phi_{\tau'}(w', \theta), u \notin \Phi_{\tau-1}(w, \theta)\},$$
$$\mathcal{E}_{2,1}(u, \tau) = \{\theta \mid \forall \tau' < \tau, \Phi_{\tau'}(w, \theta) = \Phi_{\tau'}(w', \theta), u \in \Phi_\tau(w, \theta) \setminus \Phi_\tau(w', \theta)\},$$
$$\mathcal{E}_{2,2}(u, \tau) = \{\theta \mid \forall \tau' < \tau, \Phi_{\tau'}(w, \theta) = \Phi_{\tau'}(w', \theta), u \in \Phi_\tau(w', \theta) \setminus \Phi_\tau(w, \theta)\},$$
$$\mathcal{E}_{3,1}(u, \tau) = \{\theta \mid u \in \Phi_\tau(w, \theta) \setminus \Phi_\tau(w', \theta)\},$$
$$\mathcal{E}_{3,2}(u, \tau) = \{\theta \mid u \in \Phi_\tau(w', \theta) \setminus \Phi_\tau(w, \theta)\}.$$

Note that all the events $\mathcal{E}_{2,1}(u, \tau), \mathcal{E}_{2,2}(u, \tau)$ for $\tau \in [D]$ are mutually exclusive. Therefore,

$$\Pr_{\theta \sim (\mathcal{U}[0,1])^n} \{\mathcal{E}_1(u)\} = \sum_{\tau=1}^{D} \Pr_{\theta \sim (\mathcal{U}[0,1])^n} \{\mathcal{E}_{2,1}(u, \tau)\} + \sum_{\tau=1}^{D} \Pr_{\theta \sim (\mathcal{U}[0,1])^n} \{\mathcal{E}_{2,2}(u, \tau)\}. \tag{14}$$

We first bound $\Pr_{\theta \sim (\mathcal{U}[0,1])^n} \{\mathcal{E}_{2,1}(u, \tau)\}$. Now fix all entries of θ vector except θ_u, denoted as θ_{-u}, and the corresponding subevent of $\mathcal{E}_{2,1}(u, \tau)$ is defined as $\mathcal{E}_{2,1}(u, \tau, \theta_{-u}) \subseteq \mathcal{E}_{2,1}(u, \tau)$. Similarly $\mathcal{E}_{2,0}(u, \tau, \theta_{-u}) \subseteq \mathcal{E}_{2,0}(u, \tau)$ and $\mathcal{E}_{3,1}(u, \tau, \theta_{-u}) \subseteq \mathcal{E}_{3,1}(u, \tau)$ are defined. Also $\mathcal{E}_{2,1}(u, \tau, \theta_{-u}) = \mathcal{E}_{2,0}(u, \tau, \theta_{-u}) \cap \mathcal{E}_{3,1}(u, \tau, \theta_{-u})$.

Note that $\mathcal{E}_{2,1}(u, \tau) = \mathcal{E}_{2,0}(u, \tau) \cap \mathcal{E}_{3,1}(u, \tau)$, and $\mathcal{E}_{2,2}(u, \tau) = \mathcal{E}_{2,0}(u, \tau) \cap \mathcal{E}_{3,2}(u, \tau)$. Thus

$$\Pr_{\theta \sim (\mathcal{U}[0,1])^n} \{\mathcal{E}_{2,1}(u, \tau)\} = \Pr_{\theta \sim (\mathcal{U}[0,1])^n} \{\mathcal{E}_{2,0}(u, \tau)\} \cdot \Pr_{\theta \sim (\mathcal{U}[0,1])^n} \{\mathcal{E}_{3,1}(u, \tau) \mid \mathcal{E}_{2,0}(u, \tau)\} . \quad (15)$$

Then

$$\Pr_{\theta_u \sim \mathcal{U}[0,1]} \{\mathcal{E}_{2,1}(u, \tau, \theta_{-u})\} = \Pr_{\theta_u \sim \mathcal{U}[0,1]} \{\mathcal{E}_{2,0}(u, \tau, \theta_{-u})\} \cdot \Pr_{\theta_u \sim \mathcal{U}[0,1]} \{\mathcal{E}_{3,1}(u, \tau, \theta_{-u}) \mid \mathcal{E}_{2,0}(u, \tau, \theta_{-u})\}. \quad (16)$$

Symmetric equations also hold for $\mathcal{E}_{2,2}(u, \tau, \theta_{-u})$.

In the event $\mathcal{E}_{2,0}(u, \tau, \theta_{-u})$, all entries in θ vector is fixed except for θ_u. It is easy to check that if $(\theta_{-u}, \theta_u) \in \mathcal{E}_{2,0}(u, \tau, \theta_{-u})$, then for all $\theta'_u \geq \theta_u$, $(\theta_{-u}, \theta'_u) \in \mathcal{E}_{2,0}(u, \tau, \theta_{-u})$. This is because the θ_{-u} is fixed, so the activations of all nodes other than u have the same conditions, while for u it is even harder to activate u with larger θ_u. Therefore, in $\mathcal{E}_{2,0}(u, \tau, \theta_{-u})$, the entry on θ_u must be an interval from some lowest value to 1. Let $\theta_{u,2,0}(\tau, \theta_{-u})$ be the left point of this interval. That is $\mathcal{E}_{2,0}(u, \tau, \theta_{-u}) = \{(\theta_{-u}, \theta_u) \mid \theta_u > \theta_{u,2,0}(\tau, \theta_{-u})\}$. Then we have

$$\Pr_{\theta_u \sim \mathcal{U}[0,1]} \{\mathcal{E}_{2,0}(u, \tau, \theta_{-u})\} = 1 - \theta_{u,2,0}(\tau, \theta_{-u}). \quad (17)$$

For now, let's first assume that $\mathcal{E}_{2,0}(u, \tau, \theta_{-u}) \neq \emptyset$, that is, $\theta_{u,2,0}(\tau, \theta_{-u}) < 1$. In the event $\mathcal{E}_{2,0}(u, \tau, \theta_{-u})$, we know that the set of activated nodes until $\tau - 1$ are the same under both w and w' and u is not activated by time $\tau - 1$, and since θ_{-u} is fixed, the set of activated nodes by time $\tau - 1$ are all fixed. We denote the set of nodes activated by time step i under event $\mathcal{E}_{2,0}(u, \tau, \theta_{-u})$ as $\Phi_i(\mathcal{E}_{2,0}(u, \tau, \theta_{-u}))$.

Now conditioned on the event $\mathcal{E}_{2,0}(u, \tau, \theta_{-u})$, we consider event $\mathcal{E}_{3,1}(u, \tau, \theta_{-u}) \cup \mathcal{E}_{3,2}(u, \tau, \theta_{-u})$. This means that conditioned on $\theta_u > \theta_{u,2,0}(\tau, \theta_{-u})$ and a fixed activated set $\Phi_{\tau-1}(\mathcal{E}_{2,0}(u, \tau, \theta_{-u}))$ by time $\tau - 1$, u is activated at step τ under one of w and w' but not both. According to the information diffsuion under the LT model, this means either the following inequality holds,

$$\sum_{u' \in \Phi_{\tau-1}(\mathcal{E}_{2,0}(u,\tau,\theta_{-u})) \cap N(u)} w'(e_{u',u}) < \theta_u \leq \sum_{u' \in \Phi_{\tau-1}(\mathcal{E}_{2,0}(u,\tau,\theta_{-u})) \cap N(u)} w(e_{u',u}) .$$

or the following holds

$$\sum_{u' \in \Phi_{\tau-1}(\mathcal{E}_{2,0}(u,\tau,\theta_{-u})) \cap N(u)} w(e_{u',u}) < \theta_u \leq \sum_{u' \in \Phi_{\tau-1}(\mathcal{E}_{2,0}(u,\tau,\theta_{-u})) \cap N(u)} w'(e_{u',u}) .$$

This in turn implies that

$$\Pr_{\theta_u \sim \mathcal{U}[0,1]} \{\mathcal{E}_{3,1}(u, \tau, \theta_{-u}) \cup \mathcal{E}_{3,2}(u, \tau, \theta_{-u}) \mid \mathcal{E}_{2,0}(u, \tau, \theta_{-u})\}$$

$$= \frac{\left| \sum_{u' \in \Phi_{\tau-1}(\mathcal{E}_{2,0}(u,\tau,\theta_{-u})) \cap N(u)} w(e_{u',u}) - \sum_{u' \in \Phi_{\tau-1}(\mathcal{E}_{2,0}(u,\tau,\theta_{-u})) \cap N(u)} w'(e_{u',u}) \right|}{1 - \theta_{u,2,0}(\tau, \theta_{-u})} .$$

Plugging the above equality and Eq.(17) into Eq.(16), and use the fact that $\mathcal{E}_{3,1}(u, \tau, \theta_{-u})$ and $\mathcal{E}_{3,2}(u, \tau, \theta_{-u})$ are mutually exclusive, we have

$$\Pr_{\theta_u \sim \mathcal{U}[0,1]} \{\mathcal{E}_{2,1}(u, \tau, \theta_{-u}) \cup \mathcal{E}_{2,2}(u, \tau, \theta_{-u})\}$$

$$= \left| \sum_{u' \in \Phi_{\tau-1}(\mathcal{E}_{2,0}(u,\tau,\theta_{-u})) \cap N(u)} (w(e_{u',u}) - w'(e_{u',u})) \right| . \quad (18)$$

Note that when $\mathcal{E}_{2,0}(u, \tau, \theta_{-u}) = \emptyset$, both the LHS and the RHS of the above equality is zero, so this equality holds in general.

We now need to relax event $\mathcal{E}_{2,0}(u, \tau, \theta_{-u})$, since it depends on both w and w'. We define a new event to detach it from w',

$$\mathcal{E}_{4,0}(u, \tau, \theta_{-u}) = \{\theta = (\theta_{-u}, \theta_u) \mid u \notin \Phi_{\tau-1}(w, \theta)\} .$$

14

It is clear that $\mathcal{E}_{2,0}(u,\tau,\theta_{-u}) \subseteq \mathcal{E}_{4,0}(u,\tau,\theta_{-u})$. Moreover, when $\mathcal{E}_{2,0}(u,\tau,\theta_{-u}) \neq \emptyset$, we see that both events $\mathcal{E}_{2,0}(u,\tau,\theta_{-u})$ and $\mathcal{E}_{4,0}(u,\tau,\theta_{-u})$ have fixed θ_{-u} and dictate that u is not activated by time $\tau - 1$ under w. This implies that they have the same set of nodes activated by time step i for $i \leq \tau - 1$. Denote $\Phi_i(\mathcal{E}_{4,0}(u,\tau,\theta_{-u}))$ be this set. The above means that for all $i \leq \tau - 1$, $\Phi_i(\mathcal{E}_{2,0}(u,\tau,\theta_{-u})) = \Phi_i(\mathcal{E}_{4,0}(u,\tau,\theta_{-u}))$. Therefore, we can relax Eq.(18) to get the following.

$$\Pr_{\theta_u \sim \mathcal{U}[0,1]}\{\mathcal{E}_{2,1}(u,\tau,\theta_{-u}) \cup \mathcal{E}_{2,2}(u,\tau,\theta_{-u})\}$$

$$\leq \left| \sum_{u' \in \Phi_{\tau-1}(\mathcal{E}_{4,0}(u,\tau,\theta_{-u})) \cap N(u)} (w(e_{u',u}) - w'(e_{u',u})) \right|.$$

Note that when $\mathcal{E}_{2,0}(u,\tau,\theta_{-u}) = \emptyset$, the LHS of above is zero, so the inequality still holds.

Combining the above with Eq.(14), we have

$$\Pr_{\theta \sim (\mathcal{U}[0,1])^n}\{\mathcal{E}_1(u)\}$$

$$= \int_{\theta_{-u} \in [0,1]^{n-1}} \sum_{\tau=1}^{D} \Pr_{\theta_u \sim \mathcal{U}[0,1]}\{\mathcal{E}_{2,1}(u,\tau,\theta_{-u}) \cup \mathcal{E}_{2,2}(u,\tau,\theta_{-u})\}\, d\theta_{-u}$$

$$= \sum_{\tau=1}^{D} \int_{\theta_{-u} \in [0,1]^{n-1}} \Pr_{\theta_u \sim \mathcal{U}[0,1]}\{\mathcal{E}_{2,1}(u,\tau,\theta_{-u}) \cup \mathcal{E}_{2,2}(u,\tau,\theta_{-u})\}\, d\theta_{-u}$$

$$\leq \sum_{\tau=1}^{D} \int_{\theta_{-u} \in [0,1]^{n-1}} \left| \sum_{u' \in \Phi_{\tau-1}(\mathcal{E}_{4,0}(u,\tau,\theta_{-u})) \cap N(u)} (w(e_{u',u}) - w'(e_{u',u})) \right| d\theta_{-u}$$

$$= \sum_{\tau=1}^{D} \mathbb{E}_{\theta_{-u} \sim (\mathcal{U}[0,1])^{n-1}}\left[\left| \sum_{u' \in \Phi_{\tau-1}(\mathcal{E}_{4,0}(u,\tau,\theta_{-u})) \cap N(u)} (w(e_{u',u}) - w'(e_{u',u})) \right| \right].$$

Combining the above with Eq.(12) and Eq.(13), we have

$$|r(S,w') - r(S,w)|$$

$$\leq \sum_{v \in V \setminus S} \sum_{u \in V_{S,v}} \sum_{\tau=1}^{D} \mathbb{E}_{\theta_{-u} \sim (\mathcal{U}[0,1])^{n-1}}\left[\left| \sum_{u' \in \Phi_{\tau-1}(\mathcal{E}_{4,0}(u,\tau,\theta_{-u})) \cap N(u)} (w(e_{u',u}) - w'(e_{u',u})) \right| \right]$$

$$= \mathbb{E}\left[\sum_{v \in V \setminus S} \sum_{u \in V_{S,v}} \sum_{\tau=\tau_1(u)}^{\tau_2(u)-1} \left| \sum_{e \in E_\tau(u)} (w(e) - w'(e)) \right| \right],$$

where the last equality comes from the definition of $\tau_1(u), \tau_2(u)$ and $E_\tau(u)$ under weight vector w and the expectation is taken over the randomness of the thresholds on nodes, specifically the value of $\tau_1(u), \tau_2(u)$ and $E_\tau(u)$ for each time step τ. Thus we get the desired result.

A.2   Proof of the Regret

The key Theorem 1 describes the difference of the influence spread under two weight vectors in terms of the (expected) weight differences of some edge sets, which coincides with the possible observations under LT model. So this theorem justifies why we distill the information and design the updates of the algorithm `LT-LinUCB` in this way. Next lemma further states the rationality explicitly. Recall that w is the (unknown) true weight vector.

Lemma 2. *Let S, w' be the seed set and the weight vector output at line 5 of* `PairOracle` *in a round t. Then for each fixed threshold $\theta \in [0,1]^n$,*

$$\sum_{\tau=\tau_1(u)}^{\tau_2(u)-1} \left| \sum_{e \in E_\tau(u)} (w'(e) - w(e)) \right| \leq D \cdot \mathbb{E}\left[\left| A_u^\top (w'_u - w_u) \right| \right],$$

15

where the definitions of $\tau_1(u), \tau_2(u)$ and $E_\tau(u)$ are defined under weight vector w, A_u is the value of $A_{t,u}$ updated in lines 11–15 in round t, which is the distilled edge set chosen by LT-LinUCB *to update for node u, and the expectation is taken over the randomness of τ (line 10) in determining A_u when u has active in-neighbors.*

Proof. Let $D_u = \tau_2(u) - \tau_1(u)$. If u has active in-neighbors, then according to line 7-17 of the Algorithm 1, $A_u = \chi(E_\tau(u))$ where $\tau = \tau_1(u), \ldots, \tau_2(u) - 1$ with probability $1/D_u$ respectively. Thus,

$$\mathbb{E}\left[\left|A_u^\top(w_u' - w_u)\right|\right] = \frac{1}{D_u} \sum_{\tau=\tau_1(u)}^{\tau_2(u)-1} \left|\chi(E_\tau(u))^\top(w_u' - w_u)\right|$$

$$= \frac{1}{D_u}\left(\sum_{\tau=\tau_1(u)}^{\tau_2(u)-1}\left|\sum_{e \in E_\tau(u)}(w'(e) - w(e))\right|\right).$$

Since the diffusion process lasts for at most D steps, it is straightforward that $D_u \leq D$, thus we get the inequality holds.

If u has no active in-neighbors, then by definition the values of both LHS and RHS are 0, thus the inequality still holds. □

Now we are ready to prove the regret bound.

Proof of Theorem 2. Define the failure event

$$\mathcal{F} = \left\{\exists t \leq T, v \in V : \|w_v - \hat{w}_{t-1,v}\|_{M_{t-1,v}} > \rho_{t,v}\right\} \tag{19}$$

to represent the true weight vector w_v does not lie in the confidence ellipsoid $\mathcal{C}_{t,v}$ for some round t and node v. Then by Lemma 1, when $\rho_{t,v} = \rho_t = \sqrt{n\log(1 + tn) + 2\log\frac{1}{\delta}} + \sqrt{n}$, $\mathcal{F}^c$ holds with probability at least $1 - n\delta$. Next we bound the regret conditioned on the event $\mathcal{F}^c$.

Recall that PairOracle is an (α, β)-approximation oracle, adopted in LT-LinUCB. Then the (α, β)-scaled regret of round t satisfies

$$\mathbb{E}[R_t] = \mathbb{E}[\alpha\beta \cdot \mathrm{Opt}_w - r(S_t, w)] \leq \mathbb{E}\left[\alpha\beta \cdot r(S_{\mathcal{C}_t}^{\mathrm{POpt}}, w_{\mathcal{C}_t}^{\mathrm{POpt}}) - r(S_t, w)\right]$$

$$\leq \mathbb{E}[r(S_t, w_t) - r(S_t, w)],$$

where the last inequality is by the property that PairOracle is (α, β)-approximation, and the expectation is over the randomness of the oracle and the randomness in the influence spread.

Then by Theorem 1 and Lemma 2,

$$\mathbb{E}[r(S_t, w_t) - r(S_t, w)] \leq D \cdot \mathbb{E}\left[\sum_{v \in V \setminus S_t} \sum_{u \in V_{S_t,v}} \left|A_{t,u}^\top(w_{t,u} - w_u)\right|\right]$$

$$\leq D \cdot \mathbb{E}\left[\sum_{v \in V \setminus S_t} \sum_{u \in V_{S_t,v}} \|A_{t,u}\|_{M_{t-1,u}^{-1}} \|w_{t,u} - w_u\|_{M_{t-1,u}}\right]$$

$$\leq D \cdot \mathbb{E}\left[\sum_{v \in V \setminus S_t} \sum_{u \in V_{S_t,v}} 2\rho_t \|A_{t,u}\|_{M_{t-1,u}^{-1}}\right],$$

since $w_{t,u}, w_u$ are both in the confidence set. Thus

$$R(T) = \mathbb{E}\left[\sum_{t=1}^{T} R_t\right] \leq 2\rho_T D \cdot \mathbb{E}\left[\sum_{t=1}^{T} \sum_{v \in V \setminus S_t} \sum_{u \in V_{S_t,v}} \|A_{t,u}\|_{M_{t-1,u}^{-1}}\right]$$

16

$$\leq 2\rho_T D \cdot \mathbb{E}\left[\sqrt{\left(\sum_{t=1}^{T}\sum_{u\in V} N_{S_t,u}^2\right)\left(\sum_{t=1}^{T}\sum_{u\in V}\|A_{t,u}\|_{M_{t-1,u}^{-1}}^2\right)}\right]$$

$$\leq 2\rho_T D \cdot \mathbb{E}\left[\sqrt{T}\gamma(G)\cdot\sqrt{\left(\sum_{t=1}^{T}\sum_{u\in V}\|A_{t,u}\|_{M_{t-1,u}^{-1}}^2\right)}\right]$$

where the second line is by Cauchy-Schwartz inequality.

Note that $M_{t,u} = M_{t-1,u} + A_{t,u}A_{t,u}^\top$ and

$$\begin{aligned}
\det(M_{t,u}) &= \det\left(M_{t-1,u} + A_{t,u}A_{t,u}^\top\right)\\
&= \det\left(M_{t-1,u}^{1/2}\left(I + M_{t-1,u}^{-1/2}A_{t,u}A_{t,u}^\top M_{t-1,u}^{-1/2}\right)M_{t-1,u}^{1/2}\right)\\
&= \det(M_{t-1,u})\det\left(I + M_{t-1,u}^{-1/2}A_{t,u}A_{t,u}^\top M_{t-1,u}^{-1/2}\right)\\
&= \det(M_{t-1,u})\left(1 + \|A_{t,u}\|_{M_{t-1,u}^{-1}}^2\right)
\end{aligned}$$

where the last inequality holds because the determinant of a matrix is the product of its eigenvalues and the matrix $I + xx^\top$ has eigenvalues 1 and $1 + \|x\|_2^2$. And here $\left\|M_{t-1}^{-1/2}A_{t,u}\right\| = \|A_{t,u}\|_{M_{t-1}^{-1}}$. Then

$$\sum_{t=1}^{T}\sum_{u\in V}\|A_{t,u}\|_{M_{t-1,u}^{-1}}^2 \leq \sum_{t=1}^{T}\sum_{u\in V}\frac{n}{\log(1+n)}\cdot\log\left(1 + \|A_{t,u}\|_{M_{t-1,u}^{-1}}^2\right) \tag{20}$$

$$\leq \sum_{u\in V}\frac{n}{\log(1+n)}\log\frac{\det(M_{T,u})}{\det(I)}$$

$$\leq \sum_{u\in V}\frac{n\,|N(u)|}{\log(1+n)}\log(\mathrm{trace}(M_{T,u})/\,|N(u)|) \tag{21}$$

$$\leq \sum_{u\in V}\frac{n\,|N(u)|}{\log(1+n)}\log\left(1 + \sum_{t=1}^{T}\|A_{t,u}\|_2^2/\,|N(u)|\right)$$

$$\leq \sum_{u\in V}\frac{n\,|N(u)|}{\log(1+n)}\log(1+T)$$

$$= \frac{n}{\log(1+n)}\log(1+T)\cdot\sum_{u\in V}|N(u)|$$

$$= \frac{nm}{\log(1+n)}\log(1+T), \tag{22}$$

where (20) is by the inequality that $u \leq \frac{a}{\log(1+a)}\log(1+u)$ for $u \in [0,a]$ and $\|A_{t,u}\|_2^2 \leq |N(u)| \leq n$; (21) is by the inequality that $\det(M_{T,u}) \leq (\mathrm{trace}(M_{T,u})/\,|N(u)|)^{|N(u)|}$ and (22) holds obviously since the sum of the number of in-neighbors of all nodes is just the number m of edges in the graph.

Therefore the $\alpha\beta$-scaled regret satisfies

$$\begin{aligned}
R(T) &\leq 2\rho_T\gamma(G)D\sqrt{mnT\log(1+T)/\log(1+n)} + n\delta\cdot T(n-k)\\
&\leq C\cdot\gamma(G)Dnm^{1/2}\sqrt{T}\log(T),
\end{aligned}$$

for some universal constant C. $\qquad\square$

A.3 Discussions

Comparisons of regret bounds   We compute the $\gamma(G)$ and D for some special graphs and compare our regret bound with the IMLinUCB algorithm [49] and CUCB algorithm [47], where these two are under IC model and edge-level feedback. The results are listed in Table 1 where we use

the same examples as in [49, Figure 1]. For general graphs, our `LT-LinUCB` has regret bound $O(\gamma(G)Dnm^{1/2}\sqrt{T}\ln(T)) = O(n^{7/2}m^{1/2}\sqrt{T}\ln(T))$, the IMLinUCB algorithm has regret bound (in the tabular case) $O(C_G m\sqrt{T}\ln(T)) = O(nm^{3/2}\sqrt{T}\ln(T))$ and the CUCB algorithm has regret bound $O(B_G\sqrt{mK'T\ln(T)}) = O(mn\sqrt{T\ln(T)})$. So ours is at most $O(n^{5/2}/m)$ worse than IMLinUCB and $O(n^{5/2}\sqrt{\ln(T)}/\sqrt{m})$ worse than CUCB. Note that the freedom degree of LT model is $O(n)$ as there are only n random variable $((\theta_v)_{v\in V})$ while the freedom degree of IC model is $O(m)$. Also we assume only node-level feedback is observed while edge-level feedback can be observed in their work on IC model.

Graphs	D	$\gamma(G)$	`LT-LinUCB` (ours)	IMLinUCB	CUCB
bar graph	$O(1)$	$O(\sqrt{K})$	$O(n^{3/2}\sqrt{KT}\ln(T))$	$O(n\sqrt{KT}\ln(T))$	$O(\sqrt{nKT\ln(T)})$
star graph	$O(1)$	$O(n\sqrt{K})$	$O(n^{5/2}\sqrt{KT}\ln(T))$	$O(n^2\sqrt{KT}\ln(T))$	$O(n^2\sqrt{T\ln(T)})$
ray graph	$O(\sqrt{n})$	$O(n^{5/4}\sqrt{K})$	$O(n^{13/4}\sqrt{KT}\ln(T))$	$O(n^{9/4}\sqrt{KT}\ln(T))$	$O(n^2\sqrt{T\ln(T)})$
tree graph	$O(\log n)$	$O(n^{3/2})$	$O(n^3\log n\sqrt{T}\ln(T))$	$O(n^{5/2}\sqrt{T}\ln(T))$	$O(n^2\sqrt{T\ln(T)})$
grid graph	$O(n)$	$O(n^{3/2})$	$O(n^4\sqrt{T}\ln(T))$	$O(n^{5/2}\sqrt{T}\ln(T))$	$O(n^2\sqrt{T\ln(T)})$
complete graph	$O(n)$	$O(n^{3/2})$	$O(n^{9/2}\sqrt{T}\ln(T))$	$O(n^4\sqrt{T}\ln(T))$	$O(n^3\sqrt{T\ln(T)})$

Table 1: The values of $\gamma(G), D$ and regret bound comparisons of `LT-LinUCB`, IMLinUCB [49] and CUCB [47] for special graphs.

If we represent each edge by a d-dimensional feature vector, then we can generalize our `LT-LinUCB` for the large-scale case. The regret bound would become

$$O(\rho\gamma(G)D\sqrt{dmnT\log(1+Tn^2/d)}) = O(\gamma(G)Dd\sqrt{mnT}\ln(T))$$
$$= O(dn^3\sqrt{mT}\ln(T))$$

where $\rho = O(\sqrt{d\log(1+Tn^2/d)+2\log(1/\delta)})$. We have used $\|A_{t,u}\|^2 \le |N(u)|^2$ by assuming the feature vector all have L2-norm at most 1. The regret bound of IMLinUCB [49] under IC model with edge-level feedback is $O(C_G d\sqrt{mT}\ln(T)) = O(dmn\sqrt{T}\ln(T))$, which achieves $\sqrt{m}/n^2$ better order than ours.

GOM property Our GOM property (Theorem 1) plays a key role to bound the regret, similar to the TPM condition [47] in the IC model with edge-level feedback. Their proofs [47, 49] can be simplified by coupling the influence spread under weight w and w' to reduce the proof length significantly (see Appendix E). Under their setting, it is sufficient to prove the key property for monotone case $w \le w'$ since the confidence is estimated for each edge (base arm). The coupling technique can be designed so that the realized graph of w is always a subgraph of w'. Then by comparing the connectivity difference in a subgraph, it is easy to derive the desired result.

Situations are different in our setting of node-level feedback. Since only group effect can be observed, we can not guarantee that the representative weight w' is always larger than w (see Section B for more discussions). Even though we can prove similar property for monotone $w \le w'$ and hope to generalize it to arbitrary w, w' by leveraging $w \wedge w', w \vee w'$, it does not work. By leveraging $w \wedge w', w \vee w'$, the absolute function would be added to the edge-level (compared with the result formula of Theorem 1), while we can not observe single edges in group effect. Only the absolute functions on the differences of the weight sum are suitable for node-level feedback.

B   The Optimization Problem of Weight-Constrained IM

Recall that we have a confidence ellipsoid $\mathcal{C} = \{\mathcal{C}_v\}_{v\in V}$ with $\mathcal{C}_v = \{w'_v \in [0,1]^{|N(v)|} : \|w'_v - \hat{w}_v\|_{M_v} \le \rho_v\}$ and want to consider the optimization problem of weight-constrained influence maximization (WCIM):

$$\text{argmax}_{(S,w'):S\in\mathcal{A},w'\in\mathcal{C}}\ r(S,w'). \tag{23}$$

Let $(S_{\mathcal{C}}^{\mathrm{POpt}}, w_{\mathcal{C}}^{\mathrm{POpt}})$ be the best solution. We want to find an (α, β)-approximation oracle that outputs (S', w') with $\mathbb{P}\left(r(S', w') \geq \alpha \cdot r(S_{\mathcal{C}}^{\mathrm{POpt}}, w_{\mathcal{C}}^{\mathrm{POpt}})\right) \geq \beta$ for some $\alpha, \beta > 0$.

In the following, we first discuss the general difficulty, then give a general solution and later provide efficient methods for some special graph classes.

B.1 General Difficulties

The UCB-type method does not directly apply here Under the edge-level feedback of the IC model, the learner can update the information of each single edge if it is observed; then the confidence set of the unknown weight vector is just the direct product of the confidence interval over the edges:

$$\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_e \times \cdots \times \mathcal{C}_m,$$

where $\mathcal{C}_e = [L(e), U(e)]$ is 1-dimensional confidence interval of weight $w(e)$. Thus if we take the upper bound $U(e)$ of $\mathcal{C}_e$ for each edge e, the resulting vector $U = (U(e))_{e \in E}$ still lie in the confidence set $\mathcal{C}$ and any weight vector $w' \in \mathcal{C}$ satisfies $w' \leq U$. Since the reward function $r(S, w)$ is monotone increasing in weight vector w (Lemma 9), the influence spread of any seed set S under U will be larger than w'. Hence U would be the optimal weight vector for the WCIM problem (23). Then if we take the output S_U from an usual (α, β)-approximation Oracle for the IM with weight vector U, the pair (S_U, U) is an (α, β)-approximation solution for the problem WCIM. In such derivations, we have described a design of an (α, β)-approximation PairOracle. This also explains why the designs in [49, 47] work.
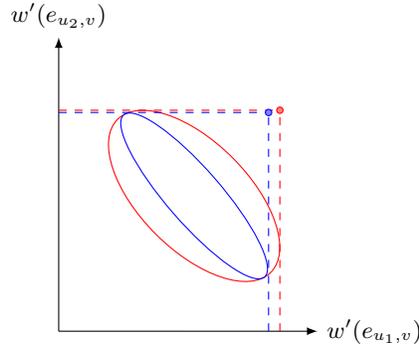

Figure 2: An example that upper bound vector fails to lie in the confidence set.

But things are different in the node-level feedback of the LT model. In the node-level feedback, the learner can only observe group effects of edges instead of single edges, so the confidence set is high-dimensional ellipsoid instead of nice cuboid. If we take the upper bounds of each edge, which is equivalent to find the upper confidence bound of the vector $\chi(e)$ with respect to the confidence set $\mathcal{C}$, the resulting vector might jump out of the confidence set $\mathcal{C}$. Specifically, since LT-LinUCB updates the information of each single node if it has active in-neighbors, the confidence set $\mathcal{C}$ of the unknown weight vector w is actually the direct product of the confidence set over the nodes:

$$\mathcal{C} = \mathcal{C}_1 \times \cdots \times \mathcal{C}_v \times \cdots \times \mathcal{C}_n,$$

where $\mathcal{C}_v$ is a $N(v)$-dimensional confidence set and is related to the edges with ending node v. Note that the confidence set $\mathcal{C}_v$ is different from the above $\mathcal{C}_e$ and we reuse the notation. For an example of 2-dimensional case (see Figure 2), there are two in-neighbors u_1, u_2 of v and suppose a confidence ellipse has such a shape (the red ellipse). The vector of largest $w(e'_{u_1}, v), w(e'_{u_2}, v)$ (the red point) is not in the confidence set, and actually is far away from the confidence set. When more observations are collected, the red ellipse may shrink to the blue ellipse, but the vector of largest $w(e'_{u_1}, v), w(e'_{u_2}, v)$ (the blue point) just moves a little and its relative distance to the confidence set is even farther.

Mixed integer optimization problem in bipartite graphs Consider the special bipartite graphs. The node set V can be divided into V_1 and V_2 and each edge is from V_1 to V_2. Without loss of

generality, assume $|V_1| \geq K$, then a good solution S must satisfy $S \subset V_1$. So the WCIM problem can be reformulated as

$$
\begin{aligned}
\max_{\alpha, w} \quad & \sum_{u \in V_1, v \in N^{\mathrm{out}}(u)} \alpha(u)\, w(u,v) \\
s.t. \quad & \alpha(u) \in \{0,1\} \ \text{ for any } u \in V_1 \\
& \sum_{u \in V_1} \alpha(u) \leq K \\
& w(u,v) \in [0,1] \ \text{ for any } u \in V_1 \text{ and } v \in N^{\mathrm{out}}(u) \\
& w(\cdot, v)^\top M_v\, w(\cdot, v) \leq \rho_v^2 \ \text{ for any } v \in V_2
\end{aligned}
\tag{24}
$$

where $w(\cdot, v) \in [0,1]^{|N(v)|}$, $M_v \in \mathbb{R}^{|N(v)| \times |N(v)|}$ is some positive-definite matrix and ρ_v is some constant.

This is a mixed integer optimization problem. Even if we relax the constraint of $\alpha(u) \in \{0,1\}$ to $\alpha(u) \in [0,1]$ to make the constraints convex, the objective is bilinear but not convex (or concave), making the problem hard to solve. This mixed integer programming is known to be difficult in the optimization field [41]. Some techniques of semidefinite programming (SDP) relaxations might be useful. We conjecture the approximation ratio, if solvable, is not constant and is $O(1/\ln(n))$ since there are roughly n constraints for w, as also motivated by the greedy method for the problem of max vertex cover. We leave this as interesting future work.

If we write the vector α in a nice vector form, we can see the problem is a special maximum inner product [40, 37, 39]. This is an interesting direction but there are still many cases unexplored.

B.2 ϵ-net Method

The usual oracle for IM problem is to compute the seed set for a given weight vector. Now the confidence set $\mathcal{C}$ is a continuous set. A method is to discretize it. We can first find an ϵ-net cover, compute the seed set by any usual oracle for each representative, and select the best pair. The complete method is provided in Algorithm 3. Recall that an ϵ-net $\mathcal{N}$ for a set $\mathcal{C}$ is for any $w' \in \mathcal{C}$, there exists a $\pi(w') \in \mathcal{N}$ such that $\|w' - \pi(w')\|_2 \leq \epsilon$. The minimal size of possible $\mathcal{N}$ is denoted as $N_{\mathcal{C}, \epsilon}$.

Algorithm 3 ϵ-net `PairOracle`

1: **Input:** Confidence ellipsoid $\mathcal{C}$; offline IM `Oracle`; seed set cardinality K; parameter ϵ
2: Find an optimal ϵ-net $\mathcal{N}$ for $\mathcal{C}$ with size $N_{\mathcal{C}, \epsilon}$
3: **for** $\pi \in \mathcal{N}$ **do**
4:     Compute the seed set S_π and $r(S_\pi, \pi)$ by `Oracle` with π and K
5: **end for**
6: **Output:** $(S', w') = \mathrm{argmax}_{(S_\pi, \pi): \pi \in \mathcal{N}}\, r(S_\pi, \pi)$

Then we have the following approximation guarantee for the ϵ-net method.

Lemma 3. *The Algorithm 3 runs with confidence ellipsoid $\mathcal{C}$, seed set cardinality K, parameter ϵ and an (α', β')-approximation* `Oracle`. *Then its output satisfies*

$$
\mathbb{P}\left(r(S', w') \geq \alpha \cdot r(S_{\mathcal{C}}^{\mathrm{POpt}}, w_{\mathcal{C}}^{\mathrm{POpt}}) \right) \geq \beta\,,
$$

where $\alpha = \alpha'\left(1 - \frac{mn \cdot \epsilon}{K}\right)$ *and* $\beta = \beta'$. *Thus the ϵ-net* `PairOracle` *(Algorithm 3) is (α, β)-approximation.*

Proof. For any $w' \in \mathcal{C}$, let $\pi(w') \in \mathcal{N}$ be its representative such that $\|w' - \pi(w')\|_2 \leq \epsilon$. Let $S_{w'}^*$ denote the output of `Oracle` with input w', then $\mathbb{P}\left(r(S_{w'}^*, w') \geq \alpha' \cdot \mathrm{Opt}_{w'} \right) \geq \beta'$. Thus

$$
\begin{aligned}
r\left(S_{\mathcal{C}}^{\mathrm{POpt}}, w_{\mathcal{C}}^{\mathrm{POpt}} \right) &\leq r\left(S_{\mathcal{C}}^{\mathrm{POpt}}, \pi\left(w_{\mathcal{C}}^{\mathrm{POpt}} \right) \right) + mn \cdot \epsilon \\
&\leq \mathrm{Opt}_{\pi\left(w_{\mathcal{C}}^{\mathrm{POpt}} \right)} + mn \cdot \epsilon
\end{aligned}
$$

20

$$\leq \frac{1}{\alpha'} r \left(S^*_{\pi(w_{\mathcal{C}}^{\text{POpt}})}, \pi \left(w_{\mathcal{C}}^{\text{POpt}} \right) \right) + mn \cdot \epsilon$$

$$\leq \frac{1}{\alpha'} r(S', w') + mn \cdot \epsilon,$$

where the first inequality is by Lipschitz continuity of r (Lemma 10) and $\left\| w_{\mathcal{C}}^{\text{POpt}} - \pi(w_{\mathcal{C}}^{\text{POpt}}) \right\|_2 \leq \epsilon$, the third inequality is by the definition of `Oracle` and holds with probability at least β' and the last inequality is by the rule of Algorithm 3.

Hence with probability at least β',

$$r(S', w') \geq \alpha' \cdot \left(r(S_{\mathcal{C}}^{\text{POpt}}, w_{\mathcal{C}}^{\text{POpt}}) - mn \cdot \epsilon \right) \geq \alpha' \left(1 - \frac{mn \cdot \epsilon}{K} \right) \cdot r(S_{\mathcal{C}}^{\text{POpt}}, w_{\mathcal{C}}^{\text{POpt}}),$$

where the second inequality is by $r(S_{\mathcal{C}}^{\text{POpt}}, w_{\mathcal{C}}^{\text{POpt}}) \geq K$. $\qquad\square$

The minimal size $N_{\mathcal{C},\epsilon}$ of the ϵ-net for the m-dimensional ellipsoid $\mathcal{C}$ has order $\Theta((1/\epsilon)^m)$, which is exponential in ϵ. So this method, though accurate, is not very efficient.

B.3 Graphs with In-degree at Most 1

We discuss the method to solve the case of graphs that any node has at most one incoming edge. This includes examples in Figure 3. For such graphs, the node-level feedback is actually edge-level feedback. More specifically, our `LT-LinUCB` will update the information of each single edge if its start node is active. Thus the confidence set $\mathcal{C}$ is the direct product of the confidence intervals of each edge, similar to IC model with edge-level feedback.
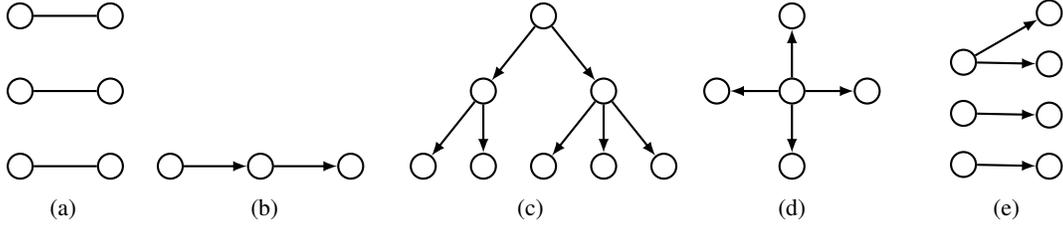

Figure 3: Examples of graphs with in-degree at most 1. (a) bar graph. (b) chain graph. (c) out-arborescence graph. (d) out-star graph. (e) certain bipartite graph. Each undirected edge represents a pair of directed edges pointing to opposite directions.

Hence we can just perform as [11, 49]. As mentioned above, we first take the upper bound for each edge and formulate U, then use an (α, β)-approximation `Oracle` to compute S_U such that

$$\mathbb{P}\left(r(S_U, U) \geq \alpha \cdot r(S_{\mathcal{C}}^{\text{POpt}}, w_{\mathcal{C}}^{\text{POpt}}) \right) \geq \beta. \tag{25}$$

So we get an efficient (α, β)-approximation `PairOracle` for these special graphs.

B.4 Bipartite Graphs

We consider the special case of bipartite graphs here where there are two node sets V_1 and V_2 and each edge is from V_1 to V_2 (see Figure 4 for examples). This is a popular influence spread formulation for one step and is a generalization of vertex cover.

Recall that the objective is to solve

$$\max_{S \in \mathcal{A}, w' \in \mathcal{C}} r(S, w') = \max_{S \in \mathcal{A}} \max_{w' \in \mathcal{C}} r(S, w').$$

Note that $r(S, w) = \sum_{u \in S, v \in N^{\text{out}}(u)} w(e_{u,v})$ is linear in w for the bipartite graphs. Let

$$r(S) := \max_{w' \in \mathcal{C}} r(S, w') = \max_{w' \in \mathcal{C}} \sum_{u \in S, v \in N^{\text{out}}(u)} w(e_{u,v}).$$
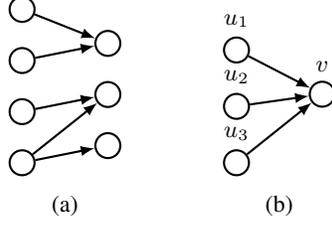
21

(a)   (b)

Figure 4: Examples of special bipartite graphs. (a) bipartite graph with in-degree at most 2. (b) bipartite graph with in-degree 3.

Recall that $\mathcal{C} = \{w' \in \mathbb{R}^m : \|w' - \hat{w}\|_M \leq \rho\}$ for some positive-definite matrix M and a constant $\rho \geq 0$. So the computation of $r(S)$ is quadratic constrained linear programming and can be solved efficiently. We have the following properties for $r(S)$. The first one is about its monotonicity for any graph.

Lemma 4. *For any graph, given the confidence set $\mathcal{C}$, the function $r(S) = \max_{w' \in \mathcal{C}} r(S, w')$ is monotone increasing in S. That is, $r(S) \leq r(S')$ if $S \subseteq S'$.*

Proof. Let $r(S) = r(S, w_S), r(S') = r(S', w_{S'})$. Then

$$r(S) = r(S, w_S) \leq r(S', w_S) \leq r(S', w_{S'}) = r(S') \,.$$

□

The next one states the submodularity of $r(S)$ for bipartite graphs with in-degree at most 2 (for example Figure 4(a)).

Lemma 5. *In bipartite graphs with in-degree at most 2, the function $r(S) = \max_{w' \in \mathcal{C}} r(S, w')$ satisfies submodularity. That is, for arbitrary set $S \subseteq S'$ and node $u \notin S'$, there is*

$$r(S \cup \{u\}) - r(S) \geq r(S' \cup \{u\}) - r(S') \,. \tag{26}$$

Proof. As discussed in Section B.1, the confidence set $\mathcal{C}$ for unknown weight vector w is actually the direct product of confidence set over nodes, that is

$$\mathcal{C} = \Pi_{v \in V_2} \, \mathcal{C}_v \,,$$

where $\mathcal{C}_v$ is the confidence set for in-coming edges of node v. The edges in each $\mathcal{C}_v$ are disjoint with each other, so

$$r(S) = \max_{w' \in \mathcal{C}} \sum_{u \in S, v \in N^{\text{out}}(u)} w'(e_{u,v}) = \max_{w' \in \mathcal{C}} \sum_{v \in V_2} \sum_{u \in S, u \in N^{\text{in}}(v)} w'(e_{u,v})$$

$$= \sum_{v \in V_2} \max_{w'_v \in \mathcal{C}_v} \sum_{u \in S, u \in N^{\text{in}}(v)} w'_v(e_{u,v}) \,,$$

where $w'_v = w'(e_{u,v})_{u \in N^{\text{in}}(v)}$. So to maximize over $\mathcal{C}$, it suffices to maximize the weights of incoming edges for each $v \in V_2$.

Since each node has at most two in-coming edges, if $e_{u,v} \in E$ for some v then it must hold that there is at most one in-neighbor of v from S'. For node v such that $e_{u,v} \in E$ but there is no edge from S' to v, the contribution of v's part to S, S' are the same.

For node v such that $e_{u,v}, e_{u',v} \in E$ for some $u' \in S' \setminus S$, it suffices to prove that

$$\max_{w'_v \in \mathcal{C}_v} w'_v(e_{u,v}) \geq \max_{w'_v \in \mathcal{C}_v} \{w'_v(e_{u,v}) + w'_v(e_{u',v})\} - \max_{w'_v \in \mathcal{C}_v} w'_v(e_{u',v}) \,,$$

which is obviously true.

For node v such that $e_{u,v}, e_{u',v} \in E$ for some $u' \in S \subseteq S'$, the contribution of v's part to S, S' are the same. □

22

Algorithm 4 Greedy `PairOracle`

1: **Input:** Graph $G = (V, E)$, confidence set $\mathcal{C}$, seed set cardinality K
2: **Initialize:** $S = \emptyset$
3: **for** $i \in [K]$ **do**
4: $\quad v = \text{argmax}_{u \in V \setminus S} \, r(S \cup \{u\}) - r(S)$
5: $\quad S = S \cup \{v\}$
6: **end for**
7: Output S

With the submodularity property, we can get the approximation result by designing a greedy policy (Algorithm 4).

Lemma 6. *Recall that $S_{\mathcal{C}}^{\text{POpt}} = \text{argmax}_{S \in \mathcal{A}} \max_{w' \in \mathcal{C}} r(S, w') = \text{argmax}_{S \in \mathcal{A}} r(S)$ is the optimal seed set given confidence set $\mathcal{C}$. Let S' be the solution returned by Greedy `PairOracle` (Algorithm 4). Then for bipartite graphs with in-degree at most 2,*

$$r(S') \geq \left(1 - \frac{1}{e}\right) r(S_{\mathcal{C}}^{\text{POpt}}). \tag{27}$$

The proof is a direct application of [19, Theorem 2.1] by noting that the function $r(\cdot)$ satisfies monotonicity (Lemma 4) and submodularity (Lemma 5) in such graphs.

A counterexample of in-degree 3   Here we show an example of bipartite graphs with in-degree 3 but the $r(\cdot)$ does not have the submodularity property.

Let $V_1 = \{u_1, u_2, u_3\}, |V_2| = 1$ and there are only 3 edges (see Figure 4(b) for example). The confidence set $\mathcal{C} = \{w' \in \mathbb{R}^3 : \|w'\|_M \leq 1\}$ with

$$M = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 2 \end{bmatrix}.$$

Note $M = I + (1, 1, 0)^\top (1, 1, 0) + (0, 1, 1)^\top (0, 1, 1)$ can happen for our algorithm `LT-LinUCB`. We solve the optimization problem and get $r(\{u_1, u_2\}) \approx 0.791, r(\{u_2, u_3\}) \approx 0.791, r(\{u_2\}) \approx 0.707, r(\{u_1, u_2, u_3\}) \approx 1.000$. Thus let $u = u_1, S = \{u_2\}, S' = \{u_2, u_3\}$, we have

$$r(S \cup \{u\}) - r(S) < 0.09 < 0.2 < r(S' \cup \{u\}) - r(S'),$$

which violates the definition of submodularity.

B.5   Directed Acyclic Graphs

Recall that $r(S) = \max_{w' \in \mathcal{C}} r(S, w')$ and $S_{\mathcal{C}}^{\text{POpt}} = \text{argmax}_{S \in \mathcal{A}} r(S)$ is the optimal solution. Let S' be the output of Greedy `PairOracle` (Algorithm 4). Then we have the following $1/K$-approximation result.

Lemma 7. *For general graphs, suppose we can compute $r(S)$ for any S. Then*

$$r(S') \geq \frac{1}{K} \cdot r(S_{\mathcal{C}}^{\text{POpt}}). \tag{28}$$

Proof. Denote $S_{\mathcal{C}}^{\text{POpt}} = \{s_1^*, s_2^*, ..., s_K^*\}$. Assume the Greedy `PairOracle` first chooses s'. Then $s' \in S'$ and $s' = \text{argmax}_{u \in V} r(\{u\})$ or equivalently $r(\{s'\}) \geq r(\{u\})$ for any $u \in V$. By monotonicity of r (Lemma 4),

$$r(S') \geq r(\{s'\}) \geq \frac{1}{K} \cdot (r(\{s_1^*\}) + r(\{s_2^*\}) + ... + r(\{s_K^*\})).$$

It suffices to prove that r satisfies the subadditivity. It is well known that the reward function $r(\cdot, w')$ satisfies submodularity in LT model [19]. Then for any $S \subseteq S''$,

$$r(S, w') + r(S'', w') \geq r(S \cup S'', w') + r(S \cap S'', w') \geq r(S \cup S'', w').$$

Recall that $r(S_{\mathcal{C}}^{\text{POpt}}) = r(S_{\mathcal{C}}^{\text{POpt}}, w_{\mathcal{C}}^{\text{POpt}})$. Then

$$r(\{s_1^*\}) + r(\{s_2^*\}) + \ldots + r(\{s_K^*\})$$
$$\geq r(\{s_1^*\}, w_{\mathcal{C}}^{\text{POpt}}) + r(\{s_2^*\}, w_{\mathcal{C}}^{\text{POpt}}) + \ldots + r(\{s_K^*\}, w_{\mathcal{C}}^{\text{POpt}})$$
$$\geq r(S_{\mathcal{C}}^{\text{POpt}}, w_{\mathcal{C}}^{\text{POpt}}) = r(S_{\mathcal{C}}^{\text{POpt}})$$

and the result follows. $\qquad\square$

Next we show that for directed acyclic graphs (DAGs), there is an efficient method to compute $r(S) = \max_{w' \in \mathcal{C}} r(S, w')$.

Algorithm 5 Compute $r(S)$ in DAGs

1: **Input:** DAG $G = (V, E)$; seed set S;
           the set of confidence ellipsoids $(\mathcal{C}_v)_{v \in V}$ with $\mathcal{C}_v = \{w_v' : (w_v')^\top M_v w_v' \leq \rho_v^2\}$
2: **Initialize:** Delete all in-edges to nodes in $S \subseteq V$
3: Use topological ranking to form 'layers' of nodes $L_0, \ldots, L_\ell, \ldots, L_{n-1}$ satisfies any edge $e \in E$
       points from L_i to L_j for some $i < j$
4: $r_S^u = 1$ for $u \in S$; $r_S^u = 0$ for $u \in L_0 \setminus S$
5: **for** $\ell = 1, 2, \ldots$ **do**
6:     **for** $v \in L_\ell$ **do**
7:         Solve $r_S^v = \max_{w' \in \mathcal{C}_v} \sum_{u \in N(v)} r_S^u \cdot w'(e_{u,v})$
8:     **end for**
9: **end for**
10: **Output:** $r(S) = \sum_{v \in V} r_S^v$

For seed set S, delete all in-coming edges to S. Take all nodes with in-degree 0 and form a set $L_0 \supseteq S$. Then consider the reduced subgraph for remaining nodes $V \setminus L_0$, take all nodes in the subgraph with in-degree 0 and form a set L_1. Note subgraphs of DAGs are still DAGs and in DAGs there are nodes with in-degree 0, otherwise we could find a cycle by adaptively adding in-neighbors. Then the procedure can continue until no node is left. Such process is just topological ranking to form 'layers' of nodes. For any node $u \in L_\ell$, its incoming edges are all from previous layers (except seed nodes), or equivalently nodes in $L_0 \cup L_1 \cup \cdots \cup L_{\ell-1}$. There are at most n layers.

Let E_ℓ' to denote the edges that has end node in layer ℓ and $E_{\ell:\ell'}' = E_\ell' \cup E_{\ell+1}' \cup \cdots \cup E_{\ell'}'$. Then $E_\ell' \bigcap E_{\ell'}' = \emptyset$ if $\ell \neq \ell'$.

Let $r_S^u(w')$ be the probability that node u will be influenced under the weight vector w' when the seed set is S and $r_S^u = \max_{w' \in \mathcal{C}} r_S^u(w')$. For seed node $u \in S$, it is activated with probability 1, or $r_S^u(w') \equiv 1$. For node $u \in L_0 \setminus S$, there is no directed path connecting from seed node S, so its activation probability is always 0, or $r_S^u(w') \equiv 0$. So we have computed r_S^u for $u \in L_0$.

Let $\ell = 1$. For node $u \in L_\ell$, its incoming edges all come from former layers $< \ell$. Note that r_S^u has been defined for any layer $< \ell$ and $r(S, w')$ can be decomposed as

$$\sum_{v \in L_\ell} \left(\sum_{u \in N^{\text{in}}(v)} r_S^u \cdot w'(e_{u,v}) \right) \cdot f_v(w', E_{\ell+1:n}'), \tag{29}$$

where $f_v(w', E_{\ell+1:n}')$ is the expected influenced nodes by node v for *later layers* and it only relates with the edges ending in later layers. Recall that the constraints are added to the edges with the same ending node. The edge $e_{u,v}$ for $v \in L_\ell$ ends in L_ℓ, so it is independent with $E_{\ell'}'$ for $\ell' > \ell$. Note that $f_v(w', E_{\ell+1:n}') \geq 1 > 0$ since node v at least influences itself. So to maximize $r(S, w')$ over $w' \in \mathcal{C}$, the weights related with edges in E_ℓ' can be maximized separately. Specifically, we can solve the maximization problem for each $v \in L_\ell$:

$$\max \sum_{u \in N^{\text{in}}(v)} r_S^u \cdot w'(e_{u,v}) \tag{30}$$
$$s.t. \quad (w_v')^\top M_v w_v' \leq \rho_v^2$$

24

where the $w'_v = (w'(e_{u,v}))_{u \in N(v)}$, M_v is some positive-definite matrix and ρ_v is some constant. This optimization problem is linear programming with quadratic convex constraints and can be solved efficiently. The resulting maximum value is actually r_S^v. So we can compute r_S for layer ℓ. Then we can compute $r(S) = \sum_{v \in V} r_S^v$ by repeating steps (29) (30) with induction on ℓ. The process is presented in Algorithm 5.

The key point to make this through for DAGs is based on the linearity of LT. Then we can decompose the objective functions to *isolated* parts and use common optimization methods to solve each part step by step.

C Analysis of `OIM-ETC` Algorithm

We first provide the regret bound of `OIM-ETC` Algorithm under both IC and LT models and then give discussions about it.

C.1 Proof of Theorem 3

Recall that $\hat{w}$ is the empirical estimate of weight vector w (line 6 of Algorithm 2) and $\hat{S}$ is the output of the (α, β)-approximation `Oracle` under estimated weight vector $\hat{w}$ (line 7 of Algorithm 2). Define event

$$\mathcal{F} = \left\{ r(\hat{S}, \hat{w}) < \alpha \cdot \mathrm{Opt}_{\hat{w}} \right\}.$$

Then $\mathbb{P}(\mathcal{F}) < 1 - \beta$ since the `Oracle` is (α, β)-approximation.

We first decompose the regret

$$
\begin{aligned}
R(T) &= \mathbb{E}\left[\sum_{t=1}^{T} (\alpha\beta \cdot \mathrm{Opt}_w - r(S_t, w)) \right] \\
&= \mathbb{E}\left[\sum_{t=1}^{nk} (\alpha\beta \cdot \mathrm{Opt}_w - r(S_t, w)) \right] + \mathbb{E}\left[\sum_{t=nk+1}^{T} (\alpha\beta \cdot \mathrm{Opt}_w - r(S_t, w)) \right] \\
&\leq nk\Delta_{\max} + (T - nk)\mathbb{E}\left[\alpha\beta \cdot \mathrm{Opt}_w - r(\hat{S}, w) \right] \\
&\leq nk\Delta_{\max} + (T - nk)\beta \cdot \mathbb{E}\left[\alpha \cdot \mathrm{Opt}_w - r(\hat{S}, w) \Big| \mathcal{F}^c \right]
\end{aligned}
\tag{31}
$$

where the last inequality is by

$$\mathbb{E}\left[r(\hat{S}, w) \right] = \mathbb{E}\left[r(\hat{S}, w) \Big| \mathcal{F} \right] \mathbb{P}(\mathcal{F}) + \mathbb{E}\left[r(\hat{S}, w) \Big| \mathcal{F}^c \right] \mathbb{P}(\mathcal{F}^c) \geq \beta \cdot \mathbb{E}\left[r(\hat{S}, w) \Big| \mathcal{F}^c \right].$$

Note under $\mathcal{F}^c$,

$$
\begin{aligned}
\alpha \cdot \mathrm{Opt}_w &= \alpha \cdot r(S_w^{\mathrm{Opt}}, w) \\
&\leq \alpha \cdot r(S_w^{\mathrm{Opt}}, \hat{w}) + \alpha \cdot mn \cdot \max_{e \in E} |\hat{w}(e) - w(e)| \\
&\leq \alpha \cdot r(S_{\hat{w}}^{\mathrm{Opt}}, \hat{w}) + \alpha \cdot mn \cdot \max_{e \in E} |\hat{w}(e) - w(e)| \\
&\leq r(\hat{S}, \hat{w}) + \alpha \cdot mn \cdot \max_{e \in E} |\hat{w}(e) - w(e)| \\
&\leq r(\hat{S}, w) + (1 + \alpha) \cdot mn \cdot \max_{e \in E} |\hat{w}(e) - w(e)|.
\end{aligned}
\tag{32}
$$

Then when $\max_{e \in E} |\hat{w}(e) - w(e)| < \frac{\Delta_{\min}}{(1+\alpha)mn} =: \epsilon_0$, $\hat{S} \notin \mathcal{S}_B$. So the regret becomes

$$
\begin{aligned}
R(T) &\leq nk\Delta_{\max} + (T - nk) \cdot 2m \exp(-2k\epsilon_0^2)\Delta_{\max} \\
&\leq \left(nk + 2mT \exp(-2k\epsilon_0^2) \right) \Delta_{\max} \\
&= \frac{n\Delta_{\max}}{2\epsilon_0^2} \ln^+ \frac{4mT\epsilon_0^2}{n} + \frac{n\Delta_{\max}}{2\epsilon_0^2}
\end{aligned}
$$

where the first inequality is to bound the complement of the event $\max_{e \in E} |\hat{w}(e) - w(e)| < \epsilon_0$ by the Chernorff-Hoeffding bound (Lemma 8), the equality is optimized with k satisfying $\exp(2k\epsilon_0^2) = 4mT\epsilon_0^2/n$ and $\ln^+(x) = \max\{0, \ln(x)\}$.

Therefore taking $k = \max\left\{1, \frac{1}{2\epsilon_0^2} \ln \frac{4mT\epsilon_0^2}{n}\right\} = \max\left\{1, \frac{2m^2n^2}{\Delta_{\min}^2} \ln \frac{T\Delta_{\min}^2}{mn^3}\right\}$ together with $R(T) \leq T\Delta_{\max}$, the regret satisfies

$$R(T) \leq \min\left\{T\Delta_{\max}, n\Delta_{\max} + \frac{2m^2n^3\Delta_{\max}}{\Delta_{\min}^2}\left(1 + \ln^+ \frac{T\Delta_{\min}^2}{mn^3}\right)\right\}$$

$$= O\left(\frac{m^2n^3\Delta_{\max}}{\Delta_{\min}^2}\ln(T)\right). \tag{33}$$

Next we prove the problem-independent bound. Following (32) under $\mathcal{F}^c$, with a suitable ϵ to be decided later,

$$\mathbb{E}\left[\alpha \cdot \mathrm{Opt}_w - r(\hat{S}, w)\right] \leq 2mn \cdot \mathbb{E}\left[\max_{e \in E} |\hat{w}(e) - w(e)|\right]$$

$$\leq 2mn\epsilon + 2mn\sum_{s=0}^{\infty} 2^{s+1}\epsilon \cdot \mathbb{P}\left(2^s\epsilon < \max_{e \in E} |\hat{w}(e) - w(e)| \leq 2^{s+1}\epsilon\right)$$

$$\leq 2mn\epsilon + 2mn\sum_{s=0}^{\infty} 2^{s+1}\epsilon \cdot \mathbb{P}\left(\exists e \in E, |\hat{w}(e) - w(e)| > 2^s\epsilon\right)$$

$$\leq 2mn\epsilon + 2mn\sum_{s=0}^{\infty} 2^{s+1}\epsilon \cdot 2m\exp(-2k2^{2s}\epsilon^2)$$

$$= 2mn\epsilon + \frac{8m^2n}{\sqrt{2k}}\sum_{s=0}^{\infty} \sqrt{2k}2^s\epsilon \cdot \exp(-(\sqrt{2k}2^s\epsilon)^2).$$

Let $X_s := \sqrt{2k}2^s\epsilon$. Note that the function $f(x) = xe^{-x^2}$ increases in $[0, 1/\sqrt{2}]$ and decreases in $[1/\sqrt{2}, \infty)$. Let s_0 satisfy

$$2^{s_0} < \frac{1}{2\sqrt{k}\epsilon} \leq 2^{s_0+1},$$

or equivalently $X_s < 1/\sqrt{2}$ for $s \leq s_0$ and $X_s \geq 1/\sqrt{2}$ for $s \geq s_0 + 1$. Then we can divide the sum into three parts

$$\sum_{s=0}^{s_0-1} f(X_s) + \sum_{s=s_0}^{s_0+1} f(X_s) + \sum_{s_0+2}^{\infty} f(X_s).$$

By monotonicity, $\sum_{s=0}^{s_0-1} f(X_s) \leq \int_0^{s_0} f(x)\,dx$ and $\sum_{s_0+2}^{\infty} f(X_s) \leq \int_{s_0+1}^{\infty} f(x)\,dx$. Thus

$$\mathbb{E}\left[\alpha \cdot \mathrm{Opt}_w - r(\hat{S}, w)\right] \leq 2mn\epsilon + \frac{8m^2n}{\sqrt{2k}}\left(\int_0^{\infty} f(x)\,dx + f(X_{s_0}) + f(X_{s_0+1})\right)$$

$$\leq 2mn\epsilon + \frac{8m^2n}{\sqrt{2k}}\left(\frac{1}{2} + 2f(1/\sqrt{2})\right)$$

$$= 2mn\epsilon + \frac{8m^2n}{\sqrt{2k}}\left(\frac{1}{2} + \sqrt{2}\exp(-1/2)\right)$$

$$\leq 2mn\epsilon + 7.69m^2n/\sqrt{k}.$$

By substituting it to (31), the regret is bounded by

$$R(T) \leq nk\Delta_{\max} + T \cdot \left(2mn\epsilon + 7.69m^2n/\sqrt{k}\right)$$

$$\leq n^2k + T \cdot \left(2mn\epsilon + 7.69m^2n/\sqrt{k}\right)$$

$$\leq 3.9(mn)^{4/3}T^{2/3} + 1 \leq 5(mn)^{4/3}T^{2/3}$$
$$= O\left((mn)^{4/3}T^{2/3}\right), \tag{34}$$

where we take $k = 3.9m^{4/3}n^{-2/3}T^{2/3}$ and $\epsilon = 1/(2mnT)$.

C.2 Discussions

As we mentioned, our `OIM-ETC` algorithm is model independent and applies to both LT and IC model with node-level feedback.

Recall that for a typical influence spread under the IC model, each edge e is *live* with the associated probability $w(e) \in [0, 1]$ and a node is activated if there is a (directed) path connecting from the seed set. For the IC model, there are three types of feedback: (1) bandit feedback, where the learner can only observe the number of influenced nodes; (2) edge-level feedback, where the learner can observe the liveness status of each outgoing edge from the activated nodes; (3) node-level feedback, where the learner can only observe the spread propagation but not individual edge liveness. The bandit feedback presents the least information and is most difficult considering the nonlinearity and complexity of the influence reward function. The edge-level feedback gives the most informative feedback and most previous work study this scheme [11, 47, 49, 50].

Since our `OIM-ETC` only selects size-1 seed set in the exploration phase, so the node-level feedback of the first-step triggering is the same with the edge-level feedback. Thus `OIM-ETC` can be applied to both IC and LT model. Though simple, `OIM-ETC` is the first model-independent algorithm for OIM[3]. Furthermore, the computational complexity for `OIM-ETC` is really low, as it only calls once of the offline oracle.

As mentioned in related work, the algorithm for the combinatorial partial monitoring [32] can be applied in OIM for both LT and IC models with node-level feedback. However, the second best solution used in their algorithm could not be directly computed in the offline IM setting. Hence only their the second stop-exploration condition applies and a regret bound of $O(nm^{3/2}T^{2/3}\ln(T))$ is obtained. Our `OIM-ETC` is better in $O(\ln(T))$ term and a bit worse in $O((n/\sqrt{m})^{1/3})$ term. Also our `OIM-ETC` has a problem-dependent regret bound.

Comparing with `LT-LinUCB` we see that `OIM-ETC` only requires the first-step node feedback, not the full diffusion process feedback of $S_{t,0}, S_{t,1}, \ldots, S_{t,\tau}, \ldots$. Moreover, it only requires the offline oracle to solve the maximization problem using the empirical mean as the fixed weight vector. The objective function in this case is known to be monotone and submodular [19, 34], and thus a greedy algorithm [19] or IMM algorithm [42] could achieve $1 - 1/e - \varepsilon$ approximation (for any small $\varepsilon > 0$) with probability at least $1 - 1/n$. That is, (α, β)-approximation `Oracle` with $\alpha = 1 - 1/e - \varepsilon$ and $\beta = 1 - 1/n$ has an efficient implementation. This is also easier than the `PairOracle`, which has the confidence ellipsoid as the constraint on weight vectors.

D Technical Lemmas

Lemma 8. *(Chernoff-Hoeffding bound) Let $X_1, X_2, \ldots, X_n$ be independent random variables with common support $[0, 1]$. Let $S_n = X_1 + X_2 + \cdots + X_n$ and $\mu = \mathbb{E}[S_n]$. Then for any $\epsilon \geq 0$,*

$$\mathbb{P}(|S_n - \mu| \geq n\epsilon) \leq 2\exp\left(-2n\epsilon^2\right).$$

Next is a property of the reward function on the weight vector under the LT model. Note that the similar property also holds for IC model [11, Lemma 6].

Lemma 9. *Under the LT model, the reward function $r(S, w)$ is monotone increasing in w. And for any seed set S and any two weight vectors $w, w' \in [0, 1]^m$, there is*

$$|r(S, w) - r(S, w')| \leq mn \cdot \max_{e \in E} |w(e) - w'(e)|. \tag{35}$$

---

[3]Note that the work [44] presents a model-free algorithm for an approximated reward function without approximation ratio while we do not relax the spread objective.

Proof. We first prove the monotonicity. Suppose $w(e) \le w'(e)$ for all $e \in E$. For any fixed thresholds θ_v's, the instance of influence graph under weight vector w is always a subgraph of w' since any activated node v under w is always activated under w'. Thus $r(S, w) \le r(S, w')$.

For (35), it is enough to prove the case $w \le w'$; otherwise we can prove it first for $w \wedge w'$ and $w \vee w'$ and then conclude the result since $r(S, w \wedge w') \le \{r(S, w), r(S, w')\} \le r(S, w \vee w')$.

Now assume w, w' only differ on one edge e: $w'(e) > w(e)$ and $w'(e') = w(e')$ for any $e' \ne e$. For any fixed thresholds θ_v's, consider the two diffusion process under w, w'. If the spreads are different, then the starting node that the diffusion processes starts to become different must be the end node of edge e. Then this difference would cause at most n nodes differences. Such an event happens when the difference of $w'(e) - w(e)$ contributes to the activation of end node of edge e, which has probability at most $w'(e) - w(e)$. Thus $r(S, w') - r(S, w) \le n \cdot (w'(e) - w(e))$.

Then for vectors $w \le w'$, we can construct at most $|E| = m$ vector pairs from w to w' with each pair only differing on one edge. By summing them up, we get $r(S, w') - r(S, w) \le mn \cdot \max_{e \in E} |w(e) - w'(e)|$. $\qquad \square$

Lemma 10. *For any seed set S and any two weight vectors $w, w' \in [0, 1]^m$, there is*

$$|r(S, w) - r(S, w')| \le mn \cdot \|w - w'\|_2 .$$

Proof. Lemma 10 can be concluded directly from Lemma 9 since it is obvious that $\max_{e \in E} |w(e) - w'(e)| \le \|w - w'\|_2$. $\qquad \square$

E   A Simplified Proof for the TPM Condition

We give a simplified proof for the TPM condition under the IC model with edge-level feedback, which corresponds to [49, Theorem 3] and the key equation [47, Lemma 2, (28)]. For completeness, we also give the theorem statement here, which mainly follow the notations of [49].

$f(S, w, v)$ is the influence probability of seed set S to node v when the mean of the weights is vector w. $O(e)$ denotes the event that edge e is observed. Recall that an edge e is relevant with S, v means there exists a path ℓ from a seed node $s \in S$ to v such that (1) $e \in \ell$ and (2) ℓ does not contain another seed node other than s. In the following, we use boldface $\mathbf{w}$ to represent a random realization of the weight vector.

Theorem 4. *(restated) For any node $v \notin S$,*

$$f(S, U, v) - f(S, \bar{w}, v) \le \sum_{\substack{e \text{ is relevant with } S, v}} \mathbb{E}_{\bar{w}}[\mathbb{1}\{O(e)\} \cdot (U(e) - \bar{w}(e)) \mid S] \qquad (36)$$

Proof. Note that

$$f(S, U, v) = \mathbb{E}_{\mathbf{w}_1 \sim U} \mathbb{1}\{v \text{ is influenced under } \mathbf{w}_1\} ,$$
$$f(S, \bar{w}, v) = \mathbb{E}_{\mathbf{w}_2 \sim \bar{w}} \mathbb{1}\{v \text{ is influenced under } \mathbf{w}_2\} .$$

When we compute the difference of these two terms, we do not need to make these two $\mathbf{w}$ independent. Specifically, for each edge e, we can design $\mathbf{w}_1, \mathbf{w}_2$ in the following way. Suppose for each edge e we independently draw a uniform random variable $A(e)$ over $[0, 1]$, let

$$\begin{aligned} \mathbf{w}_1(e) = \mathbf{w}_2(e) = 1, & \qquad \text{if } A(e) \le \bar{w}(e) ; \\ \mathbf{w}_1(e) = 1, \mathbf{w}_2(e) = 0, & \qquad \text{if } A(e) \in (\bar{w}(e), U(e)] ; \\ \mathbf{w}_1(e) = \mathbf{w}_2(e) = 0, & \qquad \text{if } A(e) > U(e) . \end{aligned}$$

Such a design of $\mathbf{w}_2$ would introduce a subgraph of $\mathbf{w}_1$ and the marginal expected means of $\mathbf{w}_1, \mathbf{w}_2$ are $U, \bar{w}$ respectively. Then the difference would become much simpler

$$f(S, U, v) - f(S, \bar{w}, v) = \mathbb{E}_{\mathbf{w}_1, \mathbf{w}_2 \sim A}[f(S, \mathbf{w}_1, v) - f(S, \mathbf{w}_2, v)]$$

and $f(S, \mathbf{w}_1, v) - f(S, \mathbf{w}_2, v) = 0$ or 1.

$f(S, \mathbf{w}_1, v) - f(S, \mathbf{w}_2, v) = 1$ means $f(S, \mathbf{w}_1, v) = 1$ and $f(S, \mathbf{w}_2, v) = 0$. Thus for any path ℓ from S to v in $\mathbf{w}_1$, there is an edge $e \in \ell$ such that $e \notin \mathbf{w}_2$. We take first such $e = (u_1, u_2)$, thus the edges on ℓ before e are live in $\mathbf{w}_2$ and the starting node u_1 of e is activated under $\mathbf{w}_2$ without edge e. Therefore there is an edge $e = (u_1, u_2)$ on the path from S to v such that

1. u_1 is activated by $\mathbf{w}_2$ on the graph without edge e ;

2. $\mathbf{w}_1(e) = 1, \mathbf{w}_2(e) = 0$.

Such an edge e is relevant with S and v. Thus

$$\mathbb{E}_{\mathbf{w}_1, \mathbf{w}_2 \sim A}[f(S, \mathbf{w}_1, v) - f(S, \mathbf{w}_2, v)]$$
$$\leq \sum_{e \text{ is relevant with } S, v} \mathbb{E}_{\mathbf{w}_2}[\mathbb{1}\{e \text{ is observed under } \mathbf{w}_2\} \cdot (U(e) - \bar{w}(e))] .$$

$\square$

With the help of this theorem, we can get the same result of TPM conditions in the work [49, 47].