Model-Independent Online Learning for Influence Maximization

Sharan Vaswani [1] **Branislav Kveton** [2] **Zheng Wen** [2] **Mohammad Ghavamzadeh** [3] **Laks V.S. Lakshmanan** [1] **Mark Schmidt** [1]

Abstract

We consider *influence maximization* (IM) in social networks, which is the problem of maximizing the number of users that become aware of a product by selecting a set of "seed" users to expose the product to. While prior work assumes a known model of information diffusion, we propose a novel parametrization that not only makes our framework agnostic to the underlying diffusion model, but also statistically efficient to learn from data. We give a corresponding monotone, submodular surrogate function, and show that it is a good approximation to the original IM objective. We also consider the case of a new marketer looking to exploit an existing social network, while simultaneously learning the factors governing information propagation. For this, we propose a pairwise-influence semi-bandit feedback model and develop a LinUCB-based bandit algorithm. Our model-independent analysis shows that our regret bound has a better (as compared to previous work) dependence on the size of the network. Experimental evaluation suggests that our framework is robust to the underlying diffusion model and can efficiently learn a near-optimal solution.

1. Introduction

The aim of viral marketing is to spread awareness about a specific product via word-of-mouth information propagation over a social network. More precisely, marketers (agents) aim to select a fixed number of influential users (called *seeds*) and provide them with free products or discounts. They assume that these users will influence their neighbours and, transitively, other users in the social network to adopt the product. This will thus result in information propagating across the network as more users adopt or become aware of the product. The marketer has a budget on the number of free products and must choose seeds in order to maximize the *influence spread* which is the expected number of users that become aware of the product. This problem is referred to as *influence maximization* (IM).

Existing solutions to the IM problem require as input, the underlying diffusion model which describes how information propagates through the network. The IM problem has been studied under various probabilistic diffusion models such as independent cascade (IC) and linear threshold (LT) models (Kempe et al., 2003). Under these common models, there has been substantial work on developing efficient heuristics and approximation algorithms (Chen et al., 2009; Leskovec et al., 2007; Goyal et al., 2011b;a; Tang et al., 2014; 2015).

Unfortunately, knowledge of the underlying diffusion model and its parameters is essential for the existing IM algorithms to perform well. For example, Du et al. (2014) empirically showed that misspecification of the diffusion model can lead to choosing bad seeds and consequently to a low spread. In practice, it is not clear how to choose from amongst the increasing number of plausible diffusion models (Kempe et al., 2003; Gomez Rodriguez et al., 2012; Li et al., 2013). Even if we are able to choose a diffusion model according to some prior information, the number of parameters for these models scales with the size of the network (for example, it is equal to the number of edges for both the IC and LT models) and it is not clear how to set these. Goyal et al. (2011a) showed that even when assuming the IC or LT model, correct knowledge of the model parameters is critical to choosing good seeds that lead to a large spread. Some papers try to learn these parameters from past propagation data (Saito et al., 2008; Goyal et al., 2010; Netrapalli & Sanghavi, 2012). However in practice, such data is hard to obtain and the large number of parameters makes this learning challenging.

To overcome these difficulties, we propose a novel parametrization for the IM problem in terms of *pairwise reachability probabilities* (Section 2). This parametrization depends only on the state of the network after the information diffusion has taken place. Since it does not depend on *how* information diffuses, it is agnostic to the underlying diffusion model. To select seeds based on these reachability probabilities, we propose a monotone and submodular surrogate objective function based on the notion of *maximum reachability* (Section 3). Our surrogate function can be optimized efficiently and is a a good approximation to

[1]University of British Columbia [2]Adobe Research [3]DeepMind (The work was done when the author was with Adobe Research). Correspondence to: Sharan Vaswani <sharanv@cs.ubc.ca>.

Proceedings of the 34^{th} International Conference on Machine Learning, Sydney, Australia, PMLR 70, 2017. Copyright 2017 by the author(s).

the IM objective. We theoretically bound the quality of this approximation. Our parametrization may be of independent interest to the IM community.

Next, we consider learning how to choose good seeds in an online setting. Specifically, we focus on the case of a new marketer looking to exploit an existing network to market their product. They need to choose a good seed set, while simultaneously learning the factors affecting information propagation. This motivates the learning framework of IM semi-bandits (Vaswani et al., 2015; Chen et al., 2016; Wen et al., 2017). In these works, the marketer performs IM over multiple "rounds" and learns about the factors governing the diffusion on the fly. Each round corresponds to an IM attempt for the same or similar products. Each attempt incurs a loss in the influence spread (measured in terms of *cumulative regret*) because of the lack of knowledge about the diffusion process. The aim is to minimize the cumulative regret incurred across multiple such rounds. This leads to the classic exploration-exploitation trade-off where the marketer must either choose seeds that either improve their knowledge about the diffusion process ("exploration") or find a seed set that leads to a large expected spread ("exploitation"). Note that all previous works on IM semi-bandits assume the IC model.

We propose a novel semi-bandit feedback model based on pairwise influence (Section 4). Our feedback model is weaker than the edge-level feedback proposed in (Chen et al., 2016; Wen et al., 2017). Under this feedback, we formulate IM semi-bandit as a linear bandit problem and propose a scalable LinUCB-based algorithm (Section 5). We bound the cumulative regret of this algorithm (Section 6) and show that our regret bound has the optimal dependence on the time horizon, is linear in the cardinality of the seed set, and as compared to the previous literature, has a better dependence on the size of the network. In Section 7, we describe how to construct features based on the graph Laplacian eigenbasis and describe a practical implementation of our algorithm. Finally, in Section 8, we empirically evaluate our proposed algorithm on a real-world network and show that it is statistically efficient and robust to the underlying diffusion model.

2. Influence Maximization

The IM problem is characterized by the triple $(\mathcal{G}, \mathcal{C}, \mathcal{D})$, where $\mathcal{G}$ is a directed graph encoding the topology of the social network, $\mathcal{C}$ is the collection of feasible seed sets, and $\mathcal{D}$ is the underlying diffusion model. Specifically, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, 2, \ldots, n\}$ and $\mathcal{E}$ are the node and edge sets of $\mathcal{G}$, with cardinalities $n = |\mathcal{V}|$ and $m = |\mathcal{E}|$, respectively. The collection of feasible seed sets $\mathcal{C}$ is determined by a *cardinality constraint* on the sets and possibly some *combinatorial constraints* (e.g. matroid constraints) that rule out some subsets of $\mathcal{V}$. This implies that $\mathcal{C} \subseteq \{\mathcal{S} \subseteq \mathcal{V} : |\mathcal{S}| \leq K\}$, for some $K \leq n$. The diffusion model $\mathcal{D}$ specifies the stochastic process under which influence is propagated across the social network once a seed set $\mathcal{S} \in \mathcal{C}$ is selected. Without loss of generality, we assume that all stochasticity in $\mathcal{D}$ is encoded in a random vector $\mathbf{w}$, referred to as the *diffusion random vector*. Note that throughout this paper, we denote vectors in bold case. We assume that each diffusion has a corresponding $\mathbf{w}$ sampled independently from an underlying probability distribution $\mathbb{P}$ specific to the diffusion model. For the widely-used models IC and LT, $\mathbf{w}$ is an m-dimensional binary vector encoding edge activations for all the edges in $\mathcal{E}$, and $\mathbb{P}$ is parametrized by m *influence probabilities*, one for each edge. Once $\mathbf{w}$ is sampled, we use $\mathcal{D}(\mathbf{w})$ to refer to the particular realization of the diffusion model $\mathcal{D}$. Note that by definition, $\mathcal{D}(\mathbf{w})$ is deterministic, conditioned on $\mathbf{w}$.

Given the above definitions, an IM attempt can be described as: the marketer first chooses a seed set $\mathcal{S} \in \mathcal{C}$ and then nature independently samples a diffusion random vector $\mathbf{w} \sim \mathbb{P}$. Note that the influenced nodes in the diffusion are completely determined by $\mathcal{S}$ and $\mathcal{D}(\mathbf{w})$. We use the indicator $\mathbb{1}\big(\mathcal{S}, v, \mathcal{D}(\mathbf{w})\big) \in \{0, 1\}$ to denote if the node v is influenced under the seed set $\mathcal{S}$ and the particular realization $\mathcal{D}(\mathbf{w})$. For a given $(\mathcal{G}, \mathcal{D})$, once a seed set $\mathcal{S} \subseteq \mathcal{C}$ is chosen, for each node $v \in \mathcal{V}$, we use $F(\mathcal{S}, v)$ to denote the probability that v is influenced under the seed set $\mathcal{S}$, i.e.,

$$F(\mathcal{S}, v) = \mathbb{E}\left[\mathbb{1}\big(\mathcal{S}, v, \mathcal{D}(\mathbf{w})\big) \big| \mathcal{S}\right] \tag{1}$$

where the expectation is over all possible realizations $\mathcal{D}(\mathbf{w})$. We denote by $F(\mathcal{S}) = \sum_{v \in \mathcal{V}} F(\mathcal{S}, v)$, the expected number of nodes that are influenced when the seed set $\mathcal{S}$ is chosen. The aim of the IM problem is to maximize $F(\mathcal{S})$ subject to the constraint $\mathcal{S} \in \mathcal{C}$, i.e., to find $\mathcal{S}^* \in \arg\max_{\mathcal{S} \in \mathcal{C}} F(\mathcal{S})$. Although IM is an NP-hard problem in general, under common diffusion models such as IC and LT, the objective function $F(\mathcal{S})$ is monotone and submodular, and thus, a near-optimal solution can be computed in polynomial time using a greedy algorithm (Nemhauser et al., 1978). In this work, we assume that $\mathcal{D}$ is any diffusion model satisfying the following monotonicity assumption:

Assumption 1. *For any $v \in \mathcal{V}$ and any subsets $\mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq \mathcal{V}$, if $F(\mathcal{S}_1, v) \leq F(\mathcal{S}_2, v)$, then $F(\mathcal{S}, v)$ is monotone in $\mathcal{S}$.*

Note that all progressive diffusion models (models where once the user is influenced, they can not change their state), including those in (Kempe et al., 2003; Gomez Rodriguez et al., 2012; Li et al., 2013) satisfy Assumption 1.

3. Surrogate Objective

We now motivate and propose a surrogate objective for the IM problem based on the notion of *maximal pairwise reachability*. We start by defining some useful notation. For any set $\mathcal{S} \subseteq \mathcal{V}$ and any set of "pairwise probabilities" $p : \mathcal{V} \times \mathcal{V} \to [0, 1]$, for all nodes $v \in \mathcal{V}$, we define

$$f(\mathcal{S}, v, p) = \max_{u \in \mathcal{S}} \; p_{u,v} \tag{2}$$

where $p_{u,v}$ is the pairwise probability associated with the ordered node pair (u, v). We further define $f(\mathcal{S}, p) = \sum_{v\in\mathcal{V}} f(\mathcal{S}, v, p)$. Note that for all p, $f(\mathcal{S}, p)$ is always monotone and submodular in $\mathcal{S}$ (Krause & Golovin, 2012).

For any pair of nodes $u, v \in \mathcal{V}$, we define the *pairwise reachability* from u to v as $p^*_{u,v} = F(\{u\}, v)$, i.e., the probability that v will be influenced, if u is the only seed node under graph $\mathcal{G}$ and diffusion model $\mathcal{D}$. Throughout this paper, we use "source node" and "seed" interchangeably and refer to the nodes *not* in the seed set $\mathcal{S}$ as "target" nodes. We define $f(\mathcal{S}, v, p^*) = \max_{u\in\mathcal{S}} p^*_{u,v}$ as the *maximal pairwise reachability* from the seed set $\mathcal{S}$ to the target node v.

Our proposed surrogate objective for the IM problem is $f(\mathcal{S}, p^*) = \sum_{v\in\mathcal{V}} f(\mathcal{S}, v, p^*)$. Based on this objective, an approximate solution $\widetilde{\mathcal{S}}$ to the IM problem can be obtained by maximizing $f(\mathcal{S}, p^*)$ under the constraint $\mathcal{S} \in \mathcal{C}$,

$$\widetilde{\mathcal{S}} \in \arg\max_{\mathcal{S}\in\mathcal{C}} f(\mathcal{S}, p^*) \tag{3}$$

Recall that $\mathcal{S}^*$ is the optimal solution to the IM problem. To quantify the quality of the surrogate, we define the *surrogate approximation factor* as $\rho = f(\widetilde{\mathcal{S}}, p^*)/F(\mathcal{S}^*)$. The following theorem, (proved in Appendix A) states that we can obtain the following upper and lower bounds on ρ:

Theorem 1. *For any graph $\mathcal{G}$, seed set $\mathcal{S} \in \mathcal{C}$, and diffusion model $\mathcal{D}$ satisfying Assumption 1,*

1 $f(\mathcal{S}, p^*) \le F(\mathcal{S})$,

2 *If $F(\mathcal{S})$ is submodular in $\mathcal{S}$, then* $1/K \le \rho \le 1$.

The above theorem implies that for any progressive model satisfying Assumption 1, maximizing $f(\mathcal{S}, p^*)$ is equivalent to maximizing a lower-bound on the true spread $F(\mathcal{S})$. For both IC and LT models, $F(\mathcal{S})$ is both monotone and submodular, and the approximation factor can be bounded from below by $1/K$. In Section 8, we empirically show that in cases of practical interest, $f(\mathcal{S}, p^*)$ is a good approximation to $F(\mathcal{S})$ and that ρ is much larger than $1/K$.

Finally, note that solving $\widetilde{\mathcal{S}} \in \arg\max_{\mathcal{S}\in\mathcal{C}} f(\mathcal{S}, p^*)$ exactly might be computationally intractable and thus we need to compute a near-optimal solution based on an approximation algorithm. In this paper, we refer to such approximation algorithms as *oracles* to distinguish them from learning algorithms. Let `ORACLE` be a specific oracle and let $\widehat{\mathcal{S}} \triangleq \texttt{ORACLE}(\mathcal{G}, \mathcal{C}, p)$ be the seed set output by it. For any $\alpha \in [0, 1]$, we say that `ORACLE` is an α-approximation algorithm if for all $p : \mathcal{V} \times \mathcal{V} \to [0, 1]$, $f(\widehat{\mathcal{S}}, p) \ge \alpha \max_{\mathcal{S}\in\mathcal{C}} f(\mathcal{S}, p)$. For our particular case, since $f(\mathcal{S}, p^*)$ is submodular, a valid oracle is the greedy algorithm which gives an $\alpha = 1 - 1/e$ approximation (Nemhauser et al., 1978). Hence, given the knowledge of p^*, we can obtain an approcimate solution to the IM problem without knowing the exact underlying diffusion model.

4. Influence Maximization Semi-Bandits

We now focus on the case of a new marketer trying to learn the pairwise reachabilities by repeatedly interacting with the network. We describe the observable feedback (Section 4.2) and the learning framework (Section 4.3).

4.1. Influence Maximization Semi-Bandits

In an influence maximization semi-bandit problem, the agent (marketer) knows both $\mathcal{G}$ and $\mathcal{C}$, but does not know the diffusion model $\mathcal{D}$. Specifically, the agent knows neither the model of $\mathcal{D}$, for instance whether $\mathcal{D}$ is the IC or LT model; nor its parameters, for instance the influence probabilities in the IC or LT model. Consider a scenario in which the agent interacts with the social network for T rounds. At each round $t \in \{1, \dots, T\}$, the agent first chooses a seed set $\mathcal{S}_t \in \mathcal{C}$ based on its prior knowledge and past observations and then nature independently samples a diffusion random vector $\mathbf{w}_t \sim \mathbb{P}$. Influence thus diffuses in the social network from $\mathcal{S}_t$ according to $\mathcal{D}(\mathbf{w}_t)$. The agent's reward at round t is the number of the influenced nodes

$$r_t = \textstyle\sum_{v\in\mathcal{V}} \mathbb{1}\big(\mathcal{S}_t, v, \mathcal{D}(\mathbf{w}_t)\big).$$

Recall that by definition, $\mathbb{E}\left[r_t | \mathcal{S}_t, \mathcal{D}(\mathbf{w}_t)\right] = F\left(\mathcal{S}_t\right)$. After each such IM attempt, the agent observes the *pairwise influence feedback* (described next) and uses it to improve the subsequent IM attempts. The agent's objective is to maximize the expected cumulative reward across the T rounds, i.e., to maximize $\mathbb{E}\left[\sum_{t=1}^{T} r_t\right]$. This is equivalent to minimizing the cumulative regret defined subsequently.

4.2. Pairwise Influence Feedback Model

We propose a novel IM semi-bandit feedback model referred to as *pairwise influence feedback*. Under this feedback model, at the end of each round t, the agent observes $\mathbb{1}\big(\{u\}, v, \mathcal{D}(\mathbf{w}_t)\big)$ for all $u \in \mathcal{S}_t$ and all $v \in \mathcal{V}$. In other words, it observes whether or not v would be influenced, if the agent selects $\mathcal{S} = \{u\}$ as the seed set under the diffusion model $\mathcal{D}(\mathbf{w}_t)$. This form of semi-bandit feedback is plausible in most IM scenarios. For example, on sites like Facebook, we can identify the user who influenced another user to "share" or "like" an article, and thus, can transitively trace the propagation to the seed which started the diffusion. Note that our assumption is strictly weaker than (and implied by) edge level semi-bandit feedback (Chen et al., 2016; Wen et al., 2017): from edge level feedback, we can identify the edges along which the diffusion travelled, and thus, determine whether a particular source node is responsible for activating a target node. However, from pairwise feedback, it is impossible to infer a unique edge level feedback.

4.3. Linear Generalization

Parametrizing the problem in terms of reachability probabilities results in $O(n^2)$ parameters that need to be

learned. Without any structural assumptions, this becomes intractable for large networks. To develop statistically efficient algorithms for large-scale IM semi-bandits, we make a linear generalization assumption similar to (Wen et al., 2015; 2017). Assume that each node $v \in \mathcal{V}$ is associated with two vectors of dimension d, the seed (source) weight $\boldsymbol{\theta}_v^* \in \Re^d$ and the target feature $\boldsymbol{x}_v \in \Re^d$. We assume that the target feature $\boldsymbol{x}_v$ is known, whereas $\boldsymbol{\theta}_v^*$ is unknown and needs to be learned. The linear generalization assumption is stated as:

Assumption 2. *For all $u, v \in \mathcal{V}$, $p_{u,v}^*$ can be "well approximated" by the inner product of $\boldsymbol{\theta}_u^*$ and $\boldsymbol{x}_v$, i.e.,*

$$p_{u,v}^* \approx \langle \boldsymbol{\theta}_u^*, \boldsymbol{x}_v \rangle \triangleq \boldsymbol{x}_v^\top \boldsymbol{\theta}_u^*$$

Note that for the *tabular case* (the case without generalization across $p_{u,v}^*$), we can always choose $\boldsymbol{x}_v = e_v \in \Re^n$ and $\boldsymbol{\theta}_u^* = \left[p_{u,1}^*, \ldots, p_{u,n}^*\right]^T$, where e_v is an n-dimensional indicator vector with the v-th element equal to 1 and all other elements equal to 0. However, in this case $d = n$, which is not desirable. Constructing target features when $d \ll n$ is non-trivial. We discuss a feature construction approach based on the unweighted graph Laplacian in Section 7. We use matrix $X \in \Re^{d \times n}$ to encode the target features. Specifically, for $v = 1, \ldots, n$, the v-th column of X is set as $\boldsymbol{x}_v$. Note that $X = I \in \Re^{n \times n}$ in the tabular case.

Finally, note that under Assumption 2, estimating the reachability probabilities becomes equivalent to estimating n (one for each source) d-dimensional weight vectors. This implies that Assumption 2 reduces the number of parameters to learn from $O(n^2)$ to $O(dn)$, and thus, is important for developing statistically efficient algorithms for large-scale IM semi-bandits.

4.4. Performance Metric

We benchmark the performance of an IM semi-bandit algorithm by comparing its spread against the attainable influence assuming perfect knowledge of $\mathcal{D}$. Since it is NP-hard to compute the optimal seed set even when with perfect knowledge, similar to (Wen et al., 2017; Chen et al., 2016), we measure the performance of an IM semi-bandit algorithm by *scaled cumulative regret*. Specifically, if $\mathcal{S}_t$ is the seed set selected by the IM semi-bandit algorithm at round t, for any $\kappa \in (0, 1)$, the κ-scaled cumulative regret $R^\kappa(T)$ in the first T rounds is defined as

$$R^\kappa(T) = T \cdot F(\mathcal{S}^*) - \frac{1}{\kappa}\mathbb{E}\left[\sum_{t=1}^T F(\mathcal{S}_t)\right]. \quad (4)$$

5. Algorithm

In this section, we propose a LinUCB-based IM semi-bandit algorithm, called *diffusion-independent LinUCB* (`DILinUCB`), whose pseudocode is in Algorithm 1. As its name suggests, `DILinUCB` is applicable to IM semi-bandits with any diffusion model $\mathcal{D}$ satisfying Assumption]refassum:monotone. The only requirement to apply `DILinUCB` is that the IM semi-bandit provides the pairwise influence feedback described in Section 4.2.

Algorithm 1 Diffusion-Independent LinUCB (`DILinUCB`)

1: **Input:** $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{C}$, oracle `ORACLE`, target feature matrix $X \in \mathbb{R}^{d \times n}$, algorithm parameters $c, \lambda, \sigma > 0$
2: Initialize $\Sigma_{u,0} \leftarrow \lambda I_d$, $\boldsymbol{b}_{u,0} \leftarrow \mathbf{0}$, $\widehat{\boldsymbol{\theta}}_{u,0} \leftarrow \mathbf{0}$ for all $u \in \mathcal{V}$, and UCB $\overline{p}_{u,v} \leftarrow 1$ for all $u, v \in \mathcal{V}$
3: **for** $t = 1$ **to** T **do**
4: Choose $\mathcal{S}_t \leftarrow$ `ORACLE` $(\mathcal{G}, \mathcal{C}, \overline{p})$
5: **for** $u \in \mathcal{S}_t$ **do**
6: Get pairwise influence feedback $\boldsymbol{y}_{u,t}$
7: $\boldsymbol{b}_{u,t} \leftarrow \boldsymbol{b}_{u,t-1} + X\boldsymbol{y}_{u,t}$
8: $\Sigma_{u,t} \leftarrow \Sigma_{u,t-1} + \sigma^{-2} X X^T$
9: $\widehat{\boldsymbol{\theta}}_{u,t} \leftarrow \sigma^{-2}\Sigma_{u,t}^{-1}\boldsymbol{b}_{u,t}$
10: $\overline{p}_{u,v} \leftarrow \mathrm{Proj}_{[0,1]}\left[\langle \widehat{\boldsymbol{\theta}}_{u,t}\boldsymbol{x}_v \rangle + c\|\boldsymbol{x}_v\|_{\Sigma_{u,t}^{-1}}\right], \forall v \in \mathcal{V}$
11: **end for**
12: **for** $u \notin \mathcal{S}_t$ **do**
13: $\boldsymbol{b}_{u,t} = \boldsymbol{b}_{u,t-1}$
14: $\Sigma_{u,t} = \Sigma_{u,t-1}$
15: **end for**
16: **end for**

The inputs to `DILinUCB` include the network topology $\mathcal{G}$, the collection of the feasible sets $\mathcal{C}$, the optimization algorithm `ORACLE`, the target feature matrix X, and three algorithm parameters $c, \lambda, \sigma > 0$. The parameter λ is a regularization parameter whereas σ is proportional to the noise in the observations and hence controls the learning rate. For each source node $u \in \mathcal{V}$ and time t, we define the Gram matrix $\Sigma_{u,t} \in \Re^{d \times d}$, and $\boldsymbol{b}_{u,t} \in \Re^d$ as the vector summarizing the past propagations from u. The vector $\boldsymbol{\theta}_{u,t}$ is the source weight estimate for node u at round t. The mean reachability probability from u to v is given by $\langle \widehat{\boldsymbol{\theta}}_{u,t}, \boldsymbol{x}_v \rangle$, whereas its variance is given as $\|\boldsymbol{x}_v\|_{\Sigma_{u,t}^{-1}} = \sqrt{\boldsymbol{x}_v^T \Sigma_{u,t}^{-1} \boldsymbol{x}_v}$. Note that Σ_u and $\boldsymbol{b}_u$ are sufficient statistics for computing UCB estimates $\overline{p}_{u,v}$ for all $v \in \mathcal{V}$. The parameter c trades off the mean and variance in the UCB estimates and thus controls the "degree of optimism" of the algorithm.

All the Gram matrices are initialized to λI_d, where I_d denotes the d-dimensional identity matrix whereas the vectors $\boldsymbol{b}_{u,0}$ and $\boldsymbol{\theta}_{u,0}$ are set to d-dimensional all-zeros vectors. At each round t, `DILinUCB` first uses the existing UCB estimates to compute the seed set $\mathcal{S}_t$ based on the given oracle `ORACLE` (line 4 of Algorithm 1). Then, it observes the pairwise reachability vector $\boldsymbol{y}_{u,t}$ for all the selected seeds in $\mathcal{S}_t$. The vector $\boldsymbol{y}_{u,t}$ is an n-dimensional column vector such that $\boldsymbol{y}_{u,t}(v) = \mathbb{1}\left(\{u\}, v, \mathcal{D}(\mathbf{w}_t)\right)$ indicating whether node v is reachable from the source u at round t. Finally, for each of the K selected seeds $u \in \mathcal{S}_t$, `DILinUCB` up-

dates the sufficient statistics (lines 7 and 8 of Algorithm 1) and the UCB estimates (line 10 of Algorithm 1). Here, $\text{Proj}_{[0,1]}[\cdot]$ projects a real number onto the $[0, 1]$ interval.

6. Regret Bound

In this section, we derive a regret bound for DILinUCB, under (1) Assumption 1, (2) perfect linear generalization i.e. $p^*_{u,v} = \langle \boldsymbol{\theta}^*_u, \boldsymbol{x}_v \rangle$ for all $u, v \in \mathcal{V}$, and (3) the assumption that $||\boldsymbol{x}_v||_2 \leq 1$ for all $v \in \mathcal{V}$. Notice that (2) is the standard assumption for linear bandit analysis (Dani et al., 2008), and (3) can always be satisfied by rescaling the target features. Our regret bound is stated below:

Theorem 2. *For any $\lambda, \sigma > 0$, any feature matrix X, any α-approximation oracle* ORACLE*, and any c satisfying*

$$c \geq \frac{1}{\sigma}\sqrt{dn\log\left(1+\frac{nT}{\sigma^2\lambda d}\right)+2\log\left(n^2T\right)}+\sqrt{\lambda}\max_{u\in\mathcal{V}}\|\boldsymbol{\theta}^*_u\|_2, \tag{5}$$

if we apply DILinUCB *with input* $(\text{ORACLE}, X, c, \lambda, \sigma)$*, then its $\rho\alpha$-scaled cumulative regret is upper-bounded as*

$$R^{\rho\alpha}(T) \leq \frac{2c}{\rho\alpha}n^{\frac{3}{2}}\sqrt{\frac{dKT\log\left(1+\frac{nT}{d\lambda\sigma^2}\right)}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}}+\frac{1}{\rho}. \tag{6}$$

For the tabular case $X = I$, we obtain a tighter bound

$$R^{\rho\alpha}(T) \leq \frac{2c}{\rho\alpha}n^{\frac{3}{2}}\sqrt{\frac{KT\log\left(1+\frac{T}{\lambda\sigma^2}\right)}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}}+\frac{1}{\rho}. \tag{7}$$

Recall that ρ specifies the quality of the surrogate approximation. Notice that if we choose $\lambda = \sigma = 1$, and choose c s.t. Inequality 5 is tight, then our regret bound is $\widetilde{O}(n^2 d\sqrt{KT}/(\alpha\rho))$ for general feature matrix X, and $\widetilde{O}(n^{2.5}\sqrt{KT}/(\alpha\rho))$ in the tabular case. Here the $\widetilde{O}$ hides log factors. We now briefly discuss the tightness of our regret bounds. First, note that the $O(1/\rho)$ factor is due to the surrogate objective approximation discussed in Section 3, and the $O(1/\alpha)$ factor is due to the fact that ORACLE is an α-approximation algorithm. Second, note that the $\widetilde{O}(\sqrt{T})$-dependence on time is near-optimal, and the $\widetilde{O}(\sqrt{K})$-dependence on the cardinality of the seed sets is standard in the combinatorial semi-bandit literature (Kveton et al., 2015). Third, for general X, notice that the $\widetilde{O}(d)$-dependence on feature dimension is standard in linear bandit literature (Dani et al., 2008; Wen et al., 2015). To explain the $\widetilde{O}(n^2)$ factor in this case, notice that one $O(n)$ factor is due to the magnitude of the reward (the reward is from 0 to n, rather than 0 to 1), whereas one $\widetilde{O}(\sqrt{n})$ factor is due to the statistical dependence of the pairwise reachabilities. Assuming statistical independence between these reachabilities (similar to Chen et al. (2016)), we can shave off this $\widetilde{O}(\sqrt{n})$ factor. However this assumption is unrealistic in practice. Another $\widetilde{O}(\sqrt{n})$ is due to the fact that we learn one $\boldsymbol{\theta}^*_u$ for each source node u (i.e. there is no generalization across the source nodes). Finally, for the tabular case $X = I$, the dependence on d no longer exists, but there is another $\widetilde{O}(\sqrt{n})$ factor due to the fact that there is no generalization across target nodes.

We conclude this section by sketching the proof for Theorem 2 (the detailed proof is available in Appendix B and Appendix C). We define the "good event" as

$$\mathcal{F} = \{|\boldsymbol{x}_v^T(\widehat{\boldsymbol{\theta}}_{u,t-1} - \boldsymbol{\theta}^*_u)| \leq c\|\boldsymbol{x}_v\|_{\Sigma^{-1}_{u,t-1}} \ \forall u, v \in \mathcal{V}, t \leq T\},$$

and the "bad event" $\overline{\mathcal{F}}$ as the complement of $\mathcal{F}$. We then decompose the $\rho\alpha$-scaled regret $R^{\rho\alpha}(T)$ over $\mathcal{F}$ and $\overline{\mathcal{F}}$, and obtain the following inequality:

$$R^{\rho\alpha}(T) \leq \frac{2c}{\rho\alpha}\mathbb{E}\left\{\sum_{t=1}^{T}\sum_{u\in\mathcal{S}_t}\sum_{v\in\mathcal{V}}\|\boldsymbol{x}_v\|_{\Sigma^{-1}_{u,t-1}}\middle|\mathcal{F}\right\}+\frac{P(\overline{\mathcal{F}})}{\rho}nT,$$

where $P(\overline{\mathcal{F}})$ is the probability of $\overline{\mathcal{F}}$. The regret bounds in Theorem 2 are derived based on worst-case bounds on $\sum_{t=1}^{T}\sum_{u\in\mathcal{S}_t}\sum_{v\in\mathcal{V}}\|x_v\|_{\Sigma^{-1}_{u,t-1}}$ (Appendix B.2), and a bound on $P(\overline{\mathcal{F}})$ based on the "self-normalized bound for matrix-valued martingales" developed in Theorem 3 (Appendix C).

7. Practical Implementation

In this section, we briefly discuss how to implement our proposed algorithm, DILinUCB, in practical semi-bandit IM problems. Specifically, we will discuss how to construct features in Section 7.1, how to enhance the practical performance of DILinUCB based on Laplacian regularization in Section 7.2, and how to implement DILinUCB computationally efficiently in real-world problems in Section 7.3.

7.1. Target Feature Construction

Although DILinUCB is applicable with any target feature matrix X, in practice, its performance is highly dependent on the "quality" of X. In this subsection, we motivate and propose a systematic feature construction approach based on the unweighted Laplacian matrix of the network topology $\mathcal{G}$. For all $u \in \mathcal{V}$, let $p^*_u \in \Re^n$ be the vector encoding the reachabilities from the seed u to all the target nodes $v \in \mathcal{V}$. Intuitively, p^*_u tends to be a smooth graph function in the sense that target nodes close to each other (e.g., in the same community) tend to have similar reachabilities from u. From (Belkin et al., 2006; Valko et al., 2014), we know that a smooth graph function (in this case, the reachability from a source) can be expressed as a linear combination of eigenvectors of the weighted Laplacian of the network. In our case, the edge weights correspond to influence probabilities and are unknown in the IM semi-bandit setting. However, we use the above intuition to construct target features based on the unweighted Laplacian of $\mathcal{G}$. Specifically, for a given $d = 1, 2, \ldots, n$, we set the feature matrix X to be the bottom d eigenvectors (associated with d smallest eigenvalues) of the unweighted Laplacian of $\mathcal{G}$. Other approaches to construct target features include the neighbour-

hood preserving node-level features as described in (Grover & Leskovec, 2016; Perozzi et al., 2014). We leave the investigation of other feature construction approaches to future work.

7.2. Laplacian Regularization

One limitation of our proposed DILinUCB algorithm is that it does not generalize across the seed nodes u. Specifically, it needs to learn the source node feature $\boldsymbol{\theta}^*_u$ for each source node u separately, which is inefficient for large-scale semi-bandit IM problems. Similar to target features, the source features also tend to be smooth in the sense that $\|\boldsymbol{\theta}^*_{u_1} - \boldsymbol{\theta}^*_{u_2}\|_2$ is "small" if nodes u_1 and u_2 are adjacent. We use this idea to design a prior which ties together the source features for different nodes, and hence transfers information between them. This idea of Laplacian regularization has been used in multi-task learning (Evgeniou et al., 2005) and for contextual-bandits in (Cesa-Bianchi et al., 2013; Vaswani et al., 2017). Specifically, at each round t, we compute $\widehat{\boldsymbol{\theta}}_{u,t}$ by minimizing the following objective w.r.t $\boldsymbol{\theta}_u$:

$$\sum_{j=1}^{t} \sum_{u \in \mathcal{S}_t} (\mathbf{y}_{u,j} - X^T \boldsymbol{\theta}_u)^2 + \lambda_2 \sum_{(u_1,u_2) \in \mathcal{E}} ||\boldsymbol{\theta}_{u_1} - \boldsymbol{\theta}_{u_2}||_2^2$$

where $\lambda_2 \geq 0$ is the regularization parameter. The implementation details are provided in Appendix D.

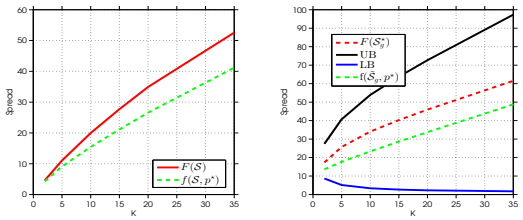
7.3. Computational Complexity

We now characterize the computational complexity of DILinUCB, and discuss how to implement it efficiently. Note that at each time t, DILinUCB needs to first compute a solution $\mathcal{S}_t$ based on ORACLE, and then update the UCBs. Since $\Sigma_{u,t}$ is positive semi-definite, the linear system in line 9 of Algorithm 1 can be solved using conjugate gradient in $O(d^2)$ time. It is straightforward to see the computational complexity to update the UCBs is $O(Knd^2)$. The computational complexity to compute $\mathcal{S}_t$ is dependent on ORACLE. For the classical setting in which $\mathcal{C} = \{\mathcal{S} \subseteq \mathcal{V} : |\mathcal{S}| \leq K\}$ and ORACLE is the greedy algorithm, the computational complexity is $O(Kn)$. To speed this up, we use the idea of lazy evaluations for submodular maximization proposed in (Minoux, 1978; Leskovec et al., 2007). It is known that this results in improved running time in practice.

8. Experiments

8.1. Empirical Verification of Surrogate Objective

In this subsection, we empirically verify that the surrogate $f(\mathcal{S}, p^*)$ proposed in Section 3 is a good approximation of the true IM objective $F(\mathcal{S})$. We conduct our tests on random Kronecker graphs, which are known to capture many properties of real-world social networks (Leskovec et al., 2010). Specifically, we generate a *social network instance* $(\mathcal{G}, \mathcal{D})$ as follows: we randomly sample $\mathcal{G}$ as a Kronecker graph with $n = 256$ and *sparsity* equal to 0.03 [1] (Leskovec et al., 2005). We choose $\mathcal{D}$ as the IC model and sample each of its influence probabilities independently from the uniform distribution $U(0, 0.1)$. Note that this range of influence probabilities is guided by the empirical evidence in (Goyal et al., 2010; Barbieri et al., 2013). To weaken the dependence on a particular instance, all the results in this subsection are averaged over 10 randomly generated instances.



(a) (b)

Figure 1. Experimental verification of surrogate objective.

We first numerically estimate the pairwise reachabilities p^* for all 10 instances based on social network simulation. In a simulation, we randomly sample a seed set $\mathcal{S}$ with cardinality K between 1 and 35, and record the pairwise influence indicator $\boldsymbol{y}_u(v)$ from each source $u \in \mathcal{S}$ to each target node v in this simulation. The reachability $p^*_{u,v}$ is estimated by averaging the $\boldsymbol{y}_u(v)$ values across 50k such simulations.

Based on the p^* so estimated, we compare $f(\mathcal{S}, p^*)$ and $F(\mathcal{S})$ as K, the seed set cardinality, varies from 2 to 35. For each K and each social network instance, we randomly sample 100 seed sets $\mathcal{S}$ with cardinality K. Then, we evaluate $f(\mathcal{S}, p^*)$ based on the estimated p^*; and numerically evaluate $F(\mathcal{S})$ by averaging results of 500 influence simulations (diffusions). For each K, we average both $F(\mathcal{S})$ and $f(\mathcal{S}, p^*)$ across the random seed sets in each instance as well as across the 10 instances. We plot the average $F(\mathcal{S})$ and $f(\mathcal{S}, p^*)$ as a function of K in Figure 1(a). The plot shows that $f(\mathcal{S})$ is a good lower bound on the true expected spread $F(\mathcal{S})$, especially for low K.

Finally, we empirically quantify the surrogate approximation factor ρ. As before, we vary K from 2 to 35 and average across 10 instances. Let $\alpha^* = 1 - e^{-1}$. For each instance and each K, we first use the estimated p^* and the greedy algorithm to find an α^*-approximation solution $\widetilde{\mathcal{S}}_g$ to the surrogate problem $\max_{\mathcal{S}} f(\mathcal{S}, p^*)$. We then use the state-of-the-art IM algorithm (Tang et al., 2014) to compute an α^*-approximation solution $\mathcal{S}^*_g$ to the IM problem $\max_{\mathcal{S}} F(\mathcal{S})$. Since $F(\mathcal{S}^*_g) \geq \alpha F(S^*)$ (Nemhauser et al., 1978), UB $\triangleq F(\mathcal{S}^*_g)/\alpha^*$ is an upper bound on $F(S^*)$. From Theorem 1, LB $\triangleq F(\mathcal{S}^*_g)/K \leq F(S^*)/K$ is a lower

[1] Based on the sparsity of typical social networks.

bound on $f(\widetilde{\mathcal{S}}, p^*)$. We plot the average values (over 10 instances) of $F(\mathcal{S}_g^*)$, $f(\widetilde{\mathcal{S}}_g, p^*)$, UB and LB against K in Figure 1(b). We observe that the difference in spreads does not increase rapidly with K. Although ρ is lower-bounded with $\frac{1}{K}$, in practice for all $K \in [2, 35]$, $\rho \geq \frac{\alpha^* f(\widetilde{\mathcal{S}}_g, p^*)}{F(\mathcal{S}_g^*)} \geq 0.55$. This shows that in practice, our surrogate approximation is reasonable even for large K.

8.2. Performance of DILinUCB

We now demonstrate the performances of variants of DILinUCB and compare them with the start of the art. We choose the social network topology $\mathcal{G}$ as a subgraph of the Facebook network available at (Leskovec & Krevl, 2014), which consists of $n = 4$k nodes and $m = 88$k edges. Since true diffusion model is unavailable, we assume the diffusion model $\mathcal{D}$ is either an IC model or an LT model, and sample the edge influence probabilities independently from the uniform distribution $U(0, 0.1)$. We also choose $T = 5$k rounds.

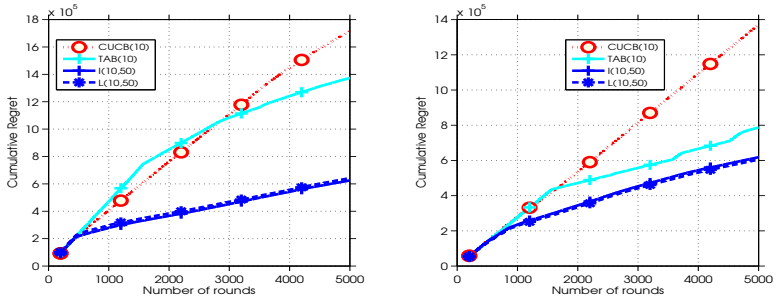
We compare DILinUCB against the CUCB algorithm (Chen et al., 2016) in both the IC model and the LT model, with $K = 10$. CUCB (referred to as CUCB(K) in plots) assumes the IC model, edge-level feedback and learns the influence probability for each edge independently. We demonstrate the performance of three variants of DILinUCB - the tabular case with $X = I$, independent estimation for each source node using target features (Algorithm 1) and Laplacian regularized estimation with target features (Appendix D). In the subsequent plots, to emphasize the dependence on K and d, these are referred to as TAB(K), I(K,d) and L(K,d) respectively. We construct features as described in Section 7.1. Similar to spectral clustering (Von Luxburg, 2007), the gap in the eigenvalues of the unweighted Laplacian can be used to choose the number of eigenvectors d. In our case, we choose the bottom $d = 50$ eigenvectors for constructing target features and show the effect of varying d in the next experiment. Similar to (Gentile et al., 2014), all hyper-parameters for our algorithm are set using an initial validation set of 500 rounds. The best validation performance was observed for $\lambda = 10^{-4}$ and $\sigma = 1$.

We now briefly discuss the performance metrics used in this section. For all $\mathcal{S} \subseteq \mathcal{V}$ and all $t = 1, 2 \ldots$, we define $r_t(\mathcal{S}) = \sum_{v \in \mathcal{V}} I(\mathcal{S}, v, \mathcal{D}(\mathbf{w}_t))$, which is the realized reward at time t if $\mathcal{S}$ is chosen at that time. One performance metric is the *per-step reward*. Specifically, in one simulation, the per-step reward at time t is defined as $\frac{\sum_{s=1}^{t} r_s}{t}$. Another performance metric is the *cumulative regret*. Since it is computationally intractable to derive S^*, our regret is measured with respect to $\mathcal{S}_g^*$, the α^*-approximation solution discussed in Section 8.1. In one simulation, the cumulative regret at time t is defined as $R(t) = \sum_{s=1}^{t} \left[r_s(\mathcal{S}_g^*) - r_s(\mathcal{S}_s) \right]$. All the subsequent results are averaged across 5 independent simulations.

Figures 2(a) and 2(b) show the cumulative regret when the underlying diffusion model is IC and LT, respectively. We have the following observations: (i) As compared to CUCB, the cumulative regret increases at a slower rate for all variants of DILinUCB, under both the IC and LT models, and for both the tabular case and case with features. (ii) Exploiting target features (linear generalization) in DILinUCB leads to a much smaller cumulative regret. (iii) CUCB is not robust to model misspecification: it has a near linear cumulative regret under LT model. (iv) Laplacian regularization has little effect on the cumulative regret in these two cases. These observations clearly demonstrate the two main advantages of DILinUCB: it is both statistically efficient and robust to diffusion model misspecification. To explain (iv), we argue that the current combination of T, K, d and n results in sufficient feedback for independent estimation to perform well and hence it is difficult to observe any additional benefit of Laplacian regularization. We provide additional evidence for this argument in the next experiment.

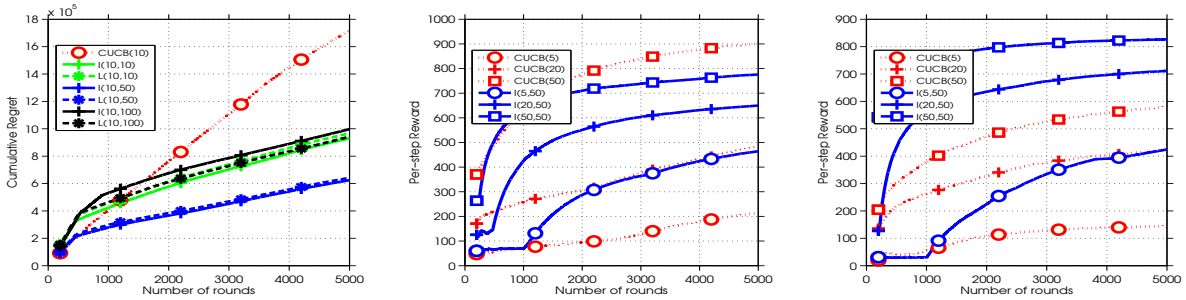
In Figure 3(a), we quantify the effect of varying d when the underlying diffusion model is IC and make the following observations: (i) The cumulative regret for both $d = 10$ and $d = 100$ is higher than that for $d = 50$. (ii) Laplacian regularization leads to observably lower cumulative regret when $d = 100$. Observation (iii) implies that $d = 10$ does not provide enough expressive power for linear generalization across the nodes of the network, whereas it is relatively difficult to estimate 100-dimensional $\boldsymbol{\theta}_u^*$ vectors within 5k rounds. Observation (iv) implies that tying source node estimates together imposes an additional bias which becomes important while learning higher dimensional coefficients. This shows the potential benefit of using Laplacian regularization for larger networks, where we will need higher d for linear generalization across nodes. We obtain similar results under the LT model.

In Figures 3(b) and 3(c), we show the effect of varying K on the per-step reward. We compare CUCB and the independent version of our algorithm when the underlying model is IC and LT. We make the following observations: (i) For both IC and LT, the per-step reward for all methods increases with K. (ii) For the IC model, the per-step reward for our algorithm is higher than CUCB when $K = \{5, 10, 20\}$, but the difference in the two spreads decreases with K. For $K = 50$, CUCB outperforms our algorithm. (iii) For the LT model, the per-step reward of our algorithm is substantially higher than CUCB for all K. Observation (i) is readily explained since both IC and LT are progressive models, and satisfy Assumption 1. To explain (ii), note that CUCB is correctly specified for the IC model. As K becomes higher, more edges become active and CUCB observes more feedback. It is thus able to learn more efficiently, leading to a higher per-step reward compared to our algorithm when $K = 50$. Observation (iii) again demonstrates that CUCB is not robust to diffusion model misspecification, while DILinUCB is.



(a) IC Model (b) LT Model

Figure 2. Comparing `DILinUCB` and `CUCB` on the Facebook subgraph with $K = 10$.



(a) Effect of d in IC (b) Effect of K in IC (c) Effect of K in LT

Figure 3. Effects of varying d or K.

9. Related Work

IM semi-bandits have been studied in several recent papers (Wen et al., 2017; Chen et al., 2016; Vaswani et al., 2015; Carpentier & Valko, 2016). Chen et al. (2016) studied IM semi-bandit under edge-level feedback and the IC diffusion model. They formulated it as a combinatorial multi-armed bandit problem and proposed a UCB algorithm (`CUCB`). They only consider the tabular case, and derive an $O(n^3)$ regret bound that also depends on the reciprocal of the minimum observation probability p of an edge. This can be problematic in for example, a line graph with L edges where all edge weights are 0.5. Then $1/p$ is 2^{L-1}, implying an exponentially large regret. Moreover, they assume that source nodes influence the target nodes independently, which is not true in most practical social networks. In contrast, both our algorithm and analysis are diffusion independent, and our analysis does not require the "independent influence" assumption made in (Chen et al., 2016). Our regret bound is $O(n^{2.5})$ in the tabular case and $O(n^2 d)$ in the general linear bandit case. Vaswani et al. (2015) use ε-greedy and Thompson sampling algorithms for a different and more challenging feedback model, where the learning agent observes influenced nodes but not the edges. They do not give any theoretical guarantees. Concurrent to our work, Wen et al. (2017) consider a linear generalization model across edges and prove regret bounds under edge-level feedback. Note that all of the above papers assume the IC diffusion model.

Carpentier & Valko (2016); Fang & Tao (2014) consider a simpler local model of influence, in which information does not transitively diffuse across the network. Lei et al. (2015) consider the related, but different, problem of maximizing the number of unique activated nodes across multiple rounds. They do not provide any theoretical analysis.

10. Conclusion

In this paper, we described a novel model-independent parametrization and a corresponding surrogate objective function for the IM problem. We used this parametrization to propose `DILinUCB`, a diffusion-independent learning algorithm for IM semi-bandits. We conjecture that with an appropriate generalization across source nodes, it may be possible to get a more statistically efficient algorithm and get rid of an additional $O(\sqrt{n})$ factor in the regret bound. In the future, we hope to address alternate bandit algorithms such Thompson sampling, and feedback models such as node-level in Vaswani et al. (2015).

Acknowledgements: This research was supported by the Natural Sciences and Engineering Research Council of Canada.

References

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

Barbieri, Nicola, Bonchi, Francesco, and Manco, Giuseppe. Topic-aware social influence propagation models. *Knowledge and information systems*, 37(3): 555–584, 2013.

Belkin, Mikhail, Niyogi, Partha, and Sindhwani, Vikas. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7(Nov):2399–2434, 2006.

Carpentier, Alexandra and Valko, Michal. Revealing graph bandits for maximizing local influence. In *International Conference on Artificial Intelligence and Statistics*, 2016.

Cesa-Bianchi, Nicolo, Gentile, Claudio, and Zappella, Giovanni. A gang of bandits. In *Advances in Neural Information Processing Systems*, pp. 737–745, 2013.

Chen, Wei, Wang, Yajun, and Yang, Siyu. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 199–208. ACM, 2009.

Chen, Wei, Wang, Yajun, Yuan, Yang, and Wang, Qinshi. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *Journal of Machine Learning Research*, 17(50):1–33, 2016.

Dani, Varsha, Hayes, Thomas P, and Kakade, Sham M. Stochastic linear optimization under bandit feedback. In *COLT*, pp. 355–366, 2008.

Du, Nan, Liang, Yingyu, Balcan, Maria-Florina, and Song, Le. Influence Function Learning in Information Diffusion Networks. *Journal of Machine Learning Research*, 32:2016–2024, 2014. URL http://machinelearning.wustl.edu/mlpapers/papers/icml2014c2{_}du14.

Evgeniou, Theodoros, Micchelli, Charles A, and Pontil, Massimiliano. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6(Apr): 615–637, 2005.

Fang, Meng and Tao, Dacheng. Networked bandits with disjoint linear payoffs. In *Internattional Conference on Knowledge Discovery and Data Mining*, 2014.

Gentile, Claudio, Li, Shuai, and Zappella, Giovanni. Online clustering of bandits. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 757–765, 2014.

Gomez Rodriguez, M, Schölkopf, B, Pineau, Langford J, et al. Influence maximization in continuous time diffusion networks. In *29th International Conference on Machine Learning (ICML 2012)*, pp. 1–8. International Machine Learning Society, 2012.

Goyal, Amit, Bonchi, Francesco, and Lakshmanan, Laks VS. Learning influence probabilities in social networks. In *Proceedings of the third ACM international conference on Web search and data mining*, pp. 241–250. ACM, 2010.

Goyal, Amit, Bonchi, Francesco, and Lakshmanan, Laks VS. A data-based approach to social influence maximization. *Proceedings of the VLDB Endowment*, 5(1):73–84, 2011a.

Goyal, Amit, Lu, Wei, and Lakshmanan, Laks VS. Simpath: An efficient algorithm for influence maximization under the linear threshold model. In *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pp. 211–220. IEEE, 2011b.

Grover, Aditya and Leskovec, Jure. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864. ACM, 2016.

Hestenes, Magnus Rudolph and Stiefel, Eduard. *Methods of conjugate gradients for solving linear systems*, volume 49. 1952.

Kempe, David, Kleinberg, Jon, and Tardos, Éva. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137–146. ACM, 2003.

Krause, Andreas and Golovin, Daniel. Submodular function maximization. *Tractability: Practical Approaches to Hard Problems*, 3(19):8, 2012.

Kveton, Branislav, Wen, Zheng, Ashkan, Azin, and Szepesvari, Csaba. Tight regret bounds for stochastic combinatorial semi-bandits. In *AISTATS*, 2015.

Lei, Siyu, Maniu, Silviu, Mo, Luyi, Cheng, Reynold, and Senellart, Pierre. Online influence maximization. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*, pp. 645–654, 2015.

Leskovec, Jure and Krevl, Andrej. SNAP Datasets: Stanford large network dataset collection. http://snap.stanford.edu/data, June 2014.

Leskovec, Jure, Krause, Andreas, Guestrin, Carlos, Faloutsos, Christos, VanBriesen, Jeanne, and Glance, Natalie.

Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 420–429. ACM, 2007.

Leskovec, Jure, Chakrabarti, Deepayan, Kleinberg, Jon, Faloutsos, Christos, and Ghahramani, Zoubin. Kronecker graphs: An approach to modeling networks. *The Journal of Machine Learning Research*, 11:985–1042, 2010.

Leskovec, Jurij, Chakrabarti, Deepayan, Kleinberg, Jon, and Faloutsos, Christos. Realistic, mathematically tractable graph generation and evolution, using kronecker multiplication. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 133–145. Springer, 2005.

Li, Yanhua, Chen, Wei, Wang, Yajun, and Zhang, Zhi-Li. Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 657–666. ACM, 2013.

Minoux, Michel. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques*, pp. 234–243. Springer, 1978.

Nemhauser, George L, Wolsey, Laurence A, and Fisher, Marshall L. An analysis of approximations for maximizing submodular set functions. *Mathematical Programming*, 14(1):265–294, 1978.

Netrapalli, Praneeth and Sanghavi, Sujay. Learning the graph of epidemic cascades. In *ACM SIGMETRICS Performance Evaluation Review*, volume 40, pp. 211–222. ACM, 2012.

Perozzi, Bryan, Al-Rfou, Rami, and Skiena, Steven. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 701–710. ACM, 2014.

Saito, Kazumi, Nakano, Ryohei, and Kimura, Masahiro. Prediction of information diffusion probabilities for independent cascade model. In *Knowledge-Based Intelligent Information and Engineering Systems*, pp. 67–75. Springer, 2008.

Tang, Youze, Xiao, Xiaokui, and Yanchen, Shi. Influence maximization: Near-optimal time complexity meets practical efficiency. 2014.

Tang, Youze, Shi, Yanchen, and Xiao, Xiaokui. Influence maximization in near-linear time: A martingale approach. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, pp. 1539–1554, 2015. ISBN 978-1-4503-2758-9.

Valko, Michal, Munos, Rémi, Kveton, Branislav, and Kocák, Tomáš. Spectral bandits for smooth graph functions. In *31th International Conference on Machine Learning*, 2014.

Vaswani, Sharan, Lakshmanan, Laks. V. S., and Mark Schmidt. Influence maximization with bandits. Technical report, http://arxiv.org/abs/1503.00024, 2015. URL http://arxiv.org/abs/1503.00024.

Vaswani, Sharan, Schmidt, Mark, and Lakshmanan, Laks. Horde of bandits using gaussian markov random fields. In *Artificial Intelligence and Statistics*, pp. 690–699, 2017.

Von Luxburg, Ulrike. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

Wen, Zheng, Kveton, Branislav, and Ashkan, Azin. Efficient learning in large-scale combinatorial semi-bandits. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, 2015.

Wen, Zheng, Kveton, Branislav, Valko, Michal, and Vaswani, Sharan. Online influence maximization under independent cascade model with semi-bandit feedback. *arXiv preprint arXiv:1605.06593v2*, 2017.

Appendices

A. Proof of Theorem 1

Proof. Theorem 1 can be proved based on the definitions of monotonicity and submodularity. Note that from Assumption 1, for any seed set $\mathcal{S} \in \mathcal{C}$, any seed node $u \in \mathcal{S}$, and any target node $v \in \mathcal{V}$, we have $F(\{u\}, v) \leq F(\mathcal{S}, v)$, which implies that

$$f(\mathcal{S}, v, p^*) = \max_{u \in \mathcal{S}} F(\{u\}, v) \leq F(\mathcal{S}, v),$$

hence

$$f(\mathcal{S}, p^*) = \sum_{v \in \mathcal{V}} f(\mathcal{S}, v, p^*) \leq \sum_{v \in \mathcal{V}} F(\mathcal{S}, v) = F(\mathcal{S}).$$

This proves the first part of Theorem 1.

We now prove the second part of the theorem. First, note that from the first part, we have

$$f(\widetilde{\mathcal{S}}, p^*) \leq F(\widetilde{\mathcal{S}}) \leq F(\mathcal{S}^*),$$

where the first inequality follows from the first part of this theorem, and the second inequality follows from the definition of $\mathcal{S}^*$. Thus, we have $\rho \leq 1$. To prove that $\rho \geq 1/K$, we assume that $\mathcal{S} = \{u_1, u_2, \ldots, u_K\}$, and define $\mathcal{S}_k = \{u_1, u_2, \ldots, u_k\}$ for $k = 1, 2, \ldots, K$. Thus, for any $\mathcal{S} \subseteq \mathcal{V}$ with $|\mathcal{S}| = K$, we have

$$\begin{aligned} F(\mathcal{S}) &= F(\mathcal{S}_1) + \sum_{k=1}^{K-1} [F(\mathcal{S}_{k+1}) - F(\mathcal{S}_k)] \\ &\leq \sum_{k=1}^{K} F(\{u_k\}) = \sum_{k=1}^{K} \sum_{v \in \mathcal{V}} F(\{u_k\}, v) \\ &\leq \sum_{v \in \mathcal{V}} K \max_{u \in \mathcal{S}} F(\{u\}, v) = K \sum_{v \in \mathcal{V}} f(\mathcal{S}, v, p^*) = K f(\mathcal{S}, p^*), \end{aligned}$$

where the first inequality follows from the submodularity of $F(\cdot)$. Thus we have

$$F(\mathcal{S}^*) \leq K f(\mathcal{S}^*, p^*) \leq K f(\widetilde{\mathcal{S}}, p^*),$$

where the second inequality follows from the definition of $\widetilde{\mathcal{S}}$. This implies that $\rho \geq 1/K$. □

B. Proof of Theorem 2

We start by defining some useful notations. We use $\mathcal{H}_t$ to denote the "history" by the end of time t. For any node pair $(u, v) \in \mathcal{V} \times \mathcal{V}$ and any time t, we define the upper confidence bound (UCB) $U_t(u, v)$ and the lower confidence bound (LCB) $L_t(u, v)$ respectively as

$$\begin{aligned} U_t(u, v) &= \mathrm{Proj}_{[0,1]} \left(\langle \widehat{\boldsymbol{\theta}}_{u,t-1}, \boldsymbol{x}_v \rangle + c \sqrt{\boldsymbol{x}_v^T \Sigma_{u,t-1}^{-1} \boldsymbol{x}_v} \right) \\ L_t(u, v) &= \mathrm{Proj}_{[0,1]} \left(\langle \widehat{\boldsymbol{\theta}}_{u,t-1}, \boldsymbol{x}_v \rangle - c \sqrt{\boldsymbol{x}_v^T \Sigma_{u,t-1}^{-1} \boldsymbol{x}_v} \right) \end{aligned} \tag{8}$$

Notice that U_t is the same as the UCB estimate $\overline{p}$ defined in Algorithm 1. Moreover, we define the "good event" $\mathcal{F}$ as

$$\mathcal{F} = \left\{ |x_v^T (\widehat{\boldsymbol{\theta}}_{u,t-1} - \boldsymbol{\theta}_u^*)| \leq c \sqrt{x_v^T \Sigma_{u,t-1}^{-1} x_v}, \, \forall u, v \in \mathcal{V}, \, \forall t \leq T \right\}, \tag{9}$$

and the "bad event" $\overline{\mathcal{F}}$ as the complement of $\mathcal{F}$.

B.1. Regret Decomposition

Recall that the realized scaled regret at time t is $R_t^{\rho\alpha} = F(S^*) - \frac{1}{\rho\alpha}F(\mathcal{S}_t)$, thus we have

$$R_t^{\rho\alpha} = F(S^*) - \frac{1}{\rho\alpha}F(\mathcal{S}_t) \overset{(a)}{=} \frac{1}{\rho}f(\widetilde{\mathcal{S}}, p^*) - \frac{1}{\rho\alpha}F(\mathcal{S}_t) \overset{(b)}{\leq} \frac{1}{\rho}f(\widetilde{\mathcal{S}}, p^*) - \frac{1}{\rho\alpha}f(\mathcal{S}_t, p^*), \tag{10}$$

where equality (a) follows from the definition of ρ (i.e. ρ is defined as $\rho = f(\widetilde{\mathcal{S}}, p^*)/F(S^*)$), and inequality (b) follows from $f(\mathcal{S}_t, p^*) \leq F(\mathcal{S}_t)$ (see Theorem 1). Thus, we have

$$\begin{aligned}
R^{\rho\alpha}(T) &= \mathbb{E}\left[\sum_{t=1}^{T} R_t^{\rho\alpha}\right] \\
&\leq \frac{1}{\rho}\mathbb{E}\left\{\sum_{t=1}^{T}\left[f(\widetilde{\mathcal{S}}, p^*) - f(\mathcal{S}_t, p^*)/\alpha\right]\right\} \\
&= \frac{P(\mathcal{F})}{\rho}\mathbb{E}\left\{\sum_{t=1}^{T}\left[f(\widetilde{\mathcal{S}}, p^*) - f(\mathcal{S}_t, p^*)/\alpha\right]\middle|\mathcal{F}\right\} + \frac{P(\overline{\mathcal{F}})}{\rho}\mathbb{E}\left\{\sum_{t=1}^{T}\left[f(\widetilde{\mathcal{S}}, p^*) - f(\mathcal{S}_t, p^*)/\alpha\right]\middle|\overline{\mathcal{F}}\right\} \\
&\leq \frac{1}{\rho}\mathbb{E}\left\{\sum_{t=1}^{T}\left[f(\widetilde{\mathcal{S}}, p^*) - f(\mathcal{S}_t, p^*)/\alpha\right]\middle|\mathcal{F}\right\} + \frac{P(\overline{\mathcal{F}})}{\rho}nT,
\end{aligned} \tag{11}$$

where the last inequality follows from the naive bounds $P(\mathcal{F}) \leq 1$ and $f(\widetilde{\mathcal{S}}, p^*) - f(\mathcal{S}_t, p^*)/\alpha \leq n$. Notice that under "good" event $\mathcal{F}$, we have

$$L_t(u, v) \leq p^*_{uv} = x_v^T\theta_u^* \leq U_t(u, v) \tag{12}$$

for all node pair (u, v) and for all time $t \leq T$. Thus, we have $f(\mathcal{S}, L_t) \leq f(\mathcal{S}, p^*) \leq f(\mathcal{S}, U_t)$ for all $\mathcal{S}$ and $t \leq T$ under event $\mathcal{F}$. So under event $\mathcal{F}$, we have

$$f(\mathcal{S}_t, L_t) \overset{(a)}{\leq} f(\mathcal{S}_t, p^*) \overset{(b)}{\leq} f(\widetilde{\mathcal{S}}, p^*) \overset{(c)}{\leq} f(\widetilde{\mathcal{S}}, U_t) \leq \max_{\mathcal{S}\in\mathcal{C}} f(\mathcal{S}, U_t) \overset{(d)}{\leq} \frac{1}{\alpha}f(\mathcal{S}_t, U_t)$$

for all $t \leq T$, where inequalities (a) and (c) follow from (12), inequality (b) follows from $\widetilde{\mathcal{S}} \in \arg\max_{\mathcal{S}\in\mathcal{C}} f(\mathcal{S}, p^*)$, and inequality (d) follows from the fact that `ORACLE` is an α-approximation algorithm. Specifically, the fact that `ORACLE` is an α-approximation algorithm implies that $f(\mathcal{S}_t, U_t) \geq \alpha \max_{\mathcal{S}\in\mathcal{C}} f(\mathcal{S}, U_t)$.

Consequently, under event $\mathcal{F}$, we have

$$\begin{aligned}
f(\widetilde{\mathcal{S}}, p^*) - \frac{1}{\alpha}f(\mathcal{S}_t, p^*) &\leq \frac{1}{\alpha}f(\mathcal{S}_t, U_t) - \frac{1}{\alpha}f(\mathcal{S}_t, L_t) \\
&= \frac{1}{\alpha}\sum_{v\in\mathcal{V}}\left[\max_{u\in\mathcal{S}_t} U_t(u, v) - \max_{u\in\mathcal{S}_t} L_t(u, v)\right] \\
&\leq \frac{1}{\alpha}\sum_{v\in\mathcal{V}}\sum_{u\in\mathcal{S}_t}\left[U_t(u, v) - L_t(u, v)\right] \\
&\leq \sum_{v\in\mathcal{V}}\sum_{u\in\mathcal{S}_t}\frac{2c}{\alpha}\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v}.
\end{aligned} \tag{13}$$

So we have

$$R^{\rho\alpha}(T) \leq \frac{2c}{\rho\alpha}\mathbb{E}\left\{\sum_{t=1}^{T}\sum_{u\in\mathcal{S}_t}\sum_{v\in\mathcal{V}}\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v}\middle|\mathcal{F}\right\} + \frac{P(\overline{\mathcal{F}})}{\rho}nT. \tag{14}$$

In the remainder of this section, we will provide a worst-case bound on $\sum_{t=1}^{T}\sum_{u\in\mathcal{S}_t}\sum_{v\in\mathcal{V}}\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v}$ (see Appendix B.2) and a bound on the probability of "bad event" $P(\overline{\mathcal{F}})$ (see Appendix B.3).

B.2. Worst-Case Bound on $\sum_{t=1}^{T}\sum_{u\in\mathcal{S}_t}\sum_{v\in\mathcal{V}}\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v}$

Notice that

$$\sum_{t=1}^{T}\sum_{u\in\mathcal{S}_t}\sum_{v\in\mathcal{V}}\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v}=\sum_{u\in\mathcal{V}}\sum_{t=1}^{T}\mathbf{1}\left[u\in\mathcal{S}_t\right]\sum_{v\in\mathcal{V}}\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v}$$

For each $u\in\mathcal{V}$, we define $K_u=\sum_{t=1}^{T}\mathbf{1}\left[u\in\mathcal{S}_t\right]$ as the number of times at which u is chosen as a source node, then we have the following lemma:

Lemma 1. *For all $u\in\mathcal{V}$, we have*

$$\sum_{t=1}^{T}\mathbf{1}\left[u\in\mathcal{S}_t\right]\sum_{v\in\mathcal{V}}\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v}\le\sqrt{nK_u}\sqrt{\frac{dn\log\left(1+\frac{nK_u}{d\lambda\sigma^2}\right)}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}}.$$

Moreover, when $X=I$, we have

$$\sum_{t=1}^{T}\mathbf{1}\left[u\in\mathcal{S}_t\right]\sum_{v\in\mathcal{V}}\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v}\le\sqrt{nK_u}\sqrt{\frac{n\log\left(1+\frac{K_u}{\lambda\sigma^2}\right)}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}}.$$

Proof. To simplify the exposition, we use Σ_t to denote $\Sigma_{u,t}$, and define $z_{t,v}=\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v}$ for all $t\le T$ and all $v\in\mathcal{V}$. Recall that

$$\Sigma_t=\Sigma_{t-1}+\frac{\mathbf{1}\left[u\in\mathcal{S}_t\right]}{\sigma^2}XX^T=\Sigma_{t-1}+\frac{\mathbf{1}\left[u\in\mathcal{S}_t\right]}{\sigma^2}\sum_{v\in\mathcal{V}}x_vx_v^T.$$

Note that if $u\notin\mathcal{S}_t$, $\Sigma_t=\Sigma_{t-1}$. If $u\in\mathcal{S}_t$, then for any $v\in\mathcal{V}$, we have

$$\begin{aligned}
\det\left[\Sigma_t\right]&\ge\det\left[\Sigma_{t-1}+\frac{1}{\sigma^2}x_vx_v^T\right]\\
&=\det\left[\Sigma_{t-1}^{\frac{1}{2}}\left(I+\frac{1}{\sigma^2}\Sigma_{t-1}^{-\frac{1}{2}}x_vx_v^T\Sigma_{t-1}^{-\frac{1}{2}}\right)\Sigma_{t-1}^{\frac{1}{2}}\right]\\
&=\det\left[\Sigma_{t-1}\right]\det\left[I+\frac{1}{\sigma^2}\Sigma_{t-1}^{-\frac{1}{2}}x_vx_v^T\Sigma_{t-1}^{-\frac{1}{2}}\right]\\
&=\det\left[\Sigma_{t-1}\right]\left(1+\frac{1}{\sigma^2}x_v^T\Sigma_{t-1}^{-1}x_v\right)=\det\left[\Sigma_{t-1}\right]\left(1+\frac{z_{t-1,v}^2}{\sigma^2}\right).
\end{aligned}$$

Hence, we have

$$\det\left[\Sigma_t\right]^n\ge\det\left[\Sigma_{t-1}\right]^n\prod_{v\in\mathcal{V}}\left(1+\frac{z_{t-1,v}^2}{\sigma^2}\right). \tag{15}$$

Note that the above inequality holds for any X. However, if $X=I$, then all Σ_t's are diagonal and we have

$$\det\left[\Sigma_t\right]=\det\left[\Sigma_{t-1}\right]\prod_{v\in\mathcal{V}}\left(1+\frac{z_{t-1,v}^2}{\sigma^2}\right). \tag{16}$$

As we will show later, this leads to a tighter regret bound in the tabular ($X=I$) case.

Let's continue our analysis for general X. The above results imply that

$$n\log\left(\det\left[\Sigma_t\right]\right)\ge n\log\left(\det\left[\Sigma_{t-1}\right]\right)+\mathbf{1}\left(u\in\mathcal{S}_t\right)\sum_{v\in\mathcal{V}}\log\left(1+\frac{z_{t-1,v}^2}{\sigma^2}\right)$$

and hence

$$n\log\left(\det\left[\Sigma_T\right]\right) \geq n\log\left(\det\left[\Sigma_0\right]\right) + \sum_{t=1}^{T}\mathbf{1}\left(u\in\mathcal{S}_t\right)\sum_{v\in\mathcal{V}}\log\left(1+\frac{z_{t-1,v}^2}{\sigma^2}\right)$$
$$= nd\log(\lambda) + \sum_{t=1}^{T}\mathbf{1}\left(u\in\mathcal{S}_t\right)\sum_{v\in\mathcal{V}}\log\left(1+\frac{z_{t-1,v}^2}{\sigma^2}\right). \tag{17}$$

On the other hand, we have that

$$\operatorname{Tr}\left[\Sigma_T\right] = \operatorname{Tr}\left[\Sigma_0 + \sum_{t=1}^{T}\frac{\mathbf{1}\left[u\in\mathcal{S}_t\right]}{\sigma^2}\sum_{v\in\mathcal{V}}x_v x_v^T\right]$$
$$= \operatorname{Tr}\left[\Sigma_0\right] + \sum_{t=1}^{T}\frac{\mathbf{1}\left[u\in\mathcal{S}_t\right]}{\sigma^2}\sum_{v\in\mathcal{V}}\operatorname{Tr}\left[x_v x_v^T\right]$$
$$= \lambda d + \sum_{t=1}^{T}\frac{\mathbf{1}\left[u\in\mathcal{S}_t\right]}{\sigma^2}\sum_{v\in\mathcal{V}}\|x_v\|^2 \leq \lambda d + \frac{nK_u}{\sigma^2}, \tag{18}$$

where the last inequality follows from the assumption that $\|x_v\| \leq 1$ and the definition of K_u. From the trace-determinant inequality, we have $\frac{1}{d}\operatorname{Tr}\left[\Sigma_T\right] \geq \det\left[\Sigma_T\right]^{\frac{1}{d}}$. Thus, we have

$$dn\log\left(\lambda+\frac{nK_u}{d\sigma^2}\right) \geq dn\log\left(\frac{1}{d}\operatorname{Tr}\left[\Sigma_T\right]\right) \geq n\log\left(\det\left[\Sigma_T\right]\right) \geq dn\log(\lambda) + \sum_{t=1}^{T}\mathbf{1}\left(u\in\mathcal{S}_t\right)\sum_{v\in\mathcal{V}}\log\left(1+\frac{z_{t-1,v}^2}{\sigma^2}\right).$$

That is

$$\sum_{t=1}^{T}\mathbf{1}\left(u\in\mathcal{S}_t\right)\sum_{v\in\mathcal{V}}\log\left(1+\frac{z_{t-1,v}^2}{\sigma^2}\right) \leq dn\log\left(1+\frac{nK_u}{d\lambda\sigma^2}\right)$$

Notice that $z_{t-1,v}^2 = x_v^T\Sigma_{t-1}^{-1}x_v \leq x_v^T\Sigma_0^{-1}x_v = \frac{\|x_v\|^2}{\lambda} \leq \frac{1}{\lambda}$. Moreover, for all $y\in[0,1/\lambda]$, we have $\log\left(1+\frac{y}{\sigma^2}\right) \geq \lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)y$ based on the concavity of $\log(\cdot)$. Thus, we have

$$\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)\sum_{t=1}^{T}\mathbf{1}\left(u\in\mathcal{S}_t\right)\sum_{v\in\mathcal{V}}z_{t-1,v}^2 \leq dn\log\left(1+\frac{nK_u}{d\lambda\sigma^2}\right).$$

Finally, from Cauchy-Schwarz inequality, we have that

$$\sum_{t=1}^{T}\mathbf{1}\left(u\in\mathcal{S}_t\right)\sum_{v\in\mathcal{V}}z_{t-1,v} \leq \sqrt{nK_u}\sqrt{\sum_{t=1}^{T}\mathbf{1}\left(u\in\mathcal{S}_t\right)\sum_{v\in\mathcal{V}}z_{t-1,v}^2}.$$

Combining the above results, we have

$$\sum_{t=1}^{T}\mathbf{1}\left(u\in\mathcal{S}_t\right)\sum_{v\in\mathcal{V}}z_{t-1,v} \leq \sqrt{nK_u}\sqrt{\frac{dn\log\left(1+\frac{nK_u}{d\lambda\sigma^2}\right)}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}}. \tag{19}$$

This concludes the proof for general X. Based on (16), the analysis for the tabular ($X=I$) case is similar, and we omit the detailed analysis. In the tabular case, we have

$$\sum_{t=1}^{T}\mathbf{1}\left(u\in\mathcal{S}_t\right)\sum_{v\in\mathcal{V}}z_{t-1,v} \leq \sqrt{nK_u}\sqrt{\frac{n\log\left(1+\frac{K_u}{\lambda\sigma^2}\right)}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}}. \tag{20}$$

□

We now develop a worst-case bound. Notice that for general X, we have

$$\begin{aligned}\sum_{u\in\mathcal{V}}\sum_{t=1}^{T}\mathbf{1}\left[u\in\mathcal{S}_t\right]\sum_{v\in\mathcal{V}}\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v} &\leq \sum_{u\in\mathcal{V}}\sqrt{nK_u}\sqrt{\frac{dn\log\left(1+\frac{nK_u}{d\lambda\sigma^2}\right)}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}}\\ &\overset{(a)}{\leq} n\sqrt{\frac{d\log\left(1+\frac{nT}{d\lambda\sigma^2}\right)}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}}\sum_{u\in\mathcal{V}}\sqrt{K_u}\\ &\overset{(b)}{\leq} n\sqrt{\frac{d\log\left(1+\frac{nT}{d\lambda\sigma^2}\right)}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}}\sqrt{n}\sqrt{\sum_{u\in\mathcal{V}}K_u}\\ &\overset{(c)}{=} n^{\frac{3}{2}}\sqrt{\frac{dKT\log\left(1+\frac{nT}{d\lambda\sigma^2}\right)}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}},\end{aligned} \tag{21}$$

where inequality (a) follows from the naive bound $K_u \leq T$, inequality (b) follows from Cauchy-Schwarz inequality, and equality (c) follows from $\sum_{u\in\mathcal{V}} K_u = KT$. Similarly, for the special case with $X = I$, we have

$$\sum_{u\in\mathcal{V}}\sum_{t=1}^{T}\mathbf{1}\left[u\in\mathcal{S}_t\right]\sum_{v\in\mathcal{V}}\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v} \leq \sum_{u\in\mathcal{V}}\sqrt{nK_u}\sqrt{\frac{n\log\left(1+\frac{K_u}{\lambda\sigma^2}\right)}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}} \leq n^{\frac{3}{2}}\sqrt{\frac{KT\log\left(1+\frac{T}{\lambda\sigma^2}\right)}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}}. \tag{22}$$

This concludes the derivation of a worst-case bound.

B.3. Bound on $P\left(\overline{\mathcal{F}}\right)$

We now derive a bound on $P\left(\overline{\mathcal{F}}\right)$ based on the "Self-Normalized Bound for Matrix-Valued Martingales" developed in Theorem 3 (see Theorem 3). Before proceeding, we define $\mathcal{F}_u$ for all $u \in \mathcal{V}$ as

$$\mathcal{F}_u = \left\{|x_v^T(\widehat{\boldsymbol{\theta}}_{u,t-1} - \boldsymbol{\theta}_u^*)| \leq c\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v},\ \forall v\in\mathcal{V},\ \forall t\leq T\right\}, \tag{23}$$

and the $\overline{\mathcal{F}}_u$ as the complement of $\mathcal{F}_u$. Note that by definition, $\overline{\mathcal{F}} = \bigcup_{u\in\mathcal{V}}\overline{\mathcal{F}}_u$. Hence, we first develop a bound on $P\left(\overline{\mathcal{F}}_u\right)$, then we develop a bound on $P\left(\overline{\mathcal{F}}\right)$ based on union bound.

Lemma 2. *For all $u \in \mathcal{V}$, all $\sigma, \lambda > 0$, all $\delta \in (0,1)$, and all*

$$c \geq \frac{1}{\sigma}\sqrt{dn\log\left(1+\frac{nT}{\sigma^2\lambda d}\right)+2\log\left(\frac{1}{\delta}\right)} + \sqrt{\lambda}\|\boldsymbol{\theta}_u^*\|_2$$

we have $P\left(\overline{\mathcal{F}}_u\right) \leq \delta$.

Proof. To simplify the expositions, we omit the subscript u in this proof. For instance, we use θ^*, Σ_t, $\mathbf{y}_t$ and $\mathbf{b}_t$ to respectively denote θ_u^*, $\Sigma_{u,t}$, $\mathbf{y}_{u,t}$ and $\mathbf{b}_{u,t}$. We also use $\mathcal{H}_t$ to denote the "history" by the end of time t, and hence $\{\mathcal{H}_t\}_{t=0}^{\infty}$ is a filtration. Notice that U_t is $\mathcal{H}_{t-1}$-adaptive, and hence $\mathcal{S}_t$ and $\mathbf{1}\left[u\in\mathcal{S}_t\right]$ are also $\mathcal{H}_{t-1}$-adaptive. We define

$$\eta_t = \begin{cases}\mathbf{y}_t - X^T\theta^* & \text{if } u\in\mathcal{S}_t\\ 0 & \text{otherwise}\end{cases} \in \Re^n \quad \text{and} \quad X_t = \begin{cases}X & \text{if } u\in\mathcal{S}_t\\ 0 & \text{otherwise}\end{cases} \in \Re^{d\times n} \tag{24}$$

Note that X_t is $\mathcal{H}_{t-1}$-adaptive, and η_t is $\mathcal{H}_t$-adaptive. Moreover, $\|\eta_t\|_\infty \leq 1$ always holds, and $\mathbb{E}\left[\eta_t|\mathcal{H}_{t-1}\right] = 0$. To simplify the expositions, we further define $\mathbf{y}_t = 0$ for all t s.t. $u \notin \mathcal{S}_t$. Note that with this definition, we have

$\eta_t = \mathbf{y}_t - X_t^T\theta^*$ for all t. We further define

$$\begin{aligned}\overline{V}_t &= n\sigma^2\Sigma_t = n\sigma^2\lambda I + n\sum_{s=1}^{t} X_s X_s^T \\ \overline{S}_t &= \sum_{s=1}^{t} X_s\eta_s = \sum_{s=1}^{t} X_s\left[\mathbf{y}_s - X_s^T\theta^*\right] = \mathbf{b}_t - \sigma^2\left[\Sigma_t - \lambda I\right]\theta^*\end{aligned} \tag{25}$$

Thus, we have $\Sigma_t\widehat{\boldsymbol{\theta}}_t = \sigma^{-2}\mathbf{b}_t = \sigma^{-2}\overline{S}_t + [\Sigma_t - \lambda I]\,\boldsymbol{\theta}^*$, which implies

$$\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^* = \Sigma_t^{-1}\left[\sigma^{-2}\overline{S}_t - \lambda\boldsymbol{\theta}^*\right]. \tag{26}$$

Consequently, for any $v \in \mathcal{V}$, we have

$$\begin{aligned}\left|x_v^T\left(\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*\right)\right| &= \left|x_v^T\Sigma_t^{-1}\left[\sigma^{-2}\overline{S}_t - \lambda\boldsymbol{\theta}^*\right]\right| \le \sqrt{x_v^T\Sigma_t^{-1}x_v}\|\sigma^{-2}\overline{S}_t - \lambda\boldsymbol{\theta}^*\|_{\Sigma_t^{-1}} \\ &\le \sqrt{x_v^T\Sigma_t^{-1}x_v}\left[\|\sigma^{-2}\overline{S}_t\|_{\Sigma_t^{-1}} + \|\lambda\boldsymbol{\theta}^*\|_{\Sigma_t^{-1}}\right],\end{aligned} \tag{27}$$

where the first inequality follows from Cauchy-Schwarz inequality and the second inequality follows from triangular inequality. Note that $\|\lambda\boldsymbol{\theta}^*\|_{\Sigma_t^{-1}} = \lambda\|\boldsymbol{\theta}^*\|_{\Sigma_t^{-1}} \le \lambda\|\boldsymbol{\theta}^*\|_{\Sigma_0^{-1}} = \sqrt{\lambda}\|\boldsymbol{\theta}^*\|_2$. On the other hand, since $\Sigma_t^{-1} = n\sigma^2\overline{V}_t^{-1}$, we have $\|\sigma^{-2}\overline{S}_t\|_{\Sigma_t^{-1}} = \frac{\sqrt{n}}{\sigma}\|\overline{S}_t\|_{\overline{V}_t^{-1}}$. Thus, we have

$$\left|x_v^T\left(\widehat{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*\right)\right| \le \sqrt{x_v^T\Sigma_t^{-1}x_v}\left[\frac{\sqrt{n}}{\sigma}\|\overline{S}_t\|_{\overline{V}_t^{-1}} + \sqrt{\lambda}\|\boldsymbol{\theta}^*\|_2\right]. \tag{28}$$

From Theorem 3, we know with probability at least $1-\delta$, for all $t \le T$, we have

$$\|S_t\|^2_{\overline{V}_t^{-1}} \le 2\log\left(\frac{\det\left(\overline{V}_t\right)^{1/2}\det\left(V\right)^{-1/2}}{\delta}\right) \le 2\log\left(\frac{\det\left(\overline{V}_T\right)^{1/2}\det\left(V\right)^{-1/2}}{\delta}\right),$$

where $V = n\sigma^2\lambda I$. Note that from the trace-determinant inequality, we have

$$\det\left[\overline{V}_T\right]^{\frac{1}{d}} \le \frac{\mathrm{Tr}\left[\overline{V}_T\right]}{d} \le \frac{n\sigma^2\lambda d + n^2T}{d},$$

where the last inequality follows from $\mathrm{Tr}\left[X_tX_t^T\right] \le n$ for all t. Note that $\det\left[V\right] = \left[n\sigma^2\lambda\right]^d$, with a little bit algebra, we have

$$\|S_t\|_{\overline{V}_t^{-1}} \le \sqrt{d\log\left(1 + \frac{nT}{\sigma^2\lambda d}\right) + 2\log\left(\frac{1}{\delta}\right)} \quad \forall t \le T$$

with probability at least $1-\delta$. Thus, if

$$c \ge \frac{1}{\sigma}\sqrt{dn\log\left(1 + \frac{nT}{\sigma^2\lambda d}\right) + 2\log\left(\frac{1}{\delta}\right)} + \sqrt{\lambda}\|\boldsymbol{\theta}^*\|_2,$$

then $\mathcal{F}_u$ holds with probability at least $1-\delta$. This concludes the proof of this lemma. □

Hence, from the union bound, we have the following lemma:

Lemma 3. *For all $\sigma, \lambda > 0$, all $\delta \in (0,1)$, and all*

$$c \ge \frac{1}{\sigma}\sqrt{dn\log\left(1 + \frac{nT}{\sigma^2\lambda d}\right) + 2\log\left(\frac{n}{\delta}\right)} + \sqrt{\lambda}\max_{u\in\mathcal{V}}\|\boldsymbol{\theta}_u^*\|_2 \tag{29}$$

we have $P\left(\overline{\mathcal{F}}\right) \leq \delta$.

Proof. This lemma follows directly from the union bound. Note that for all c satisfying Equation 29, we have $P\left(\overline{\mathcal{F}}_u\right) \leq \frac{\delta}{n}$ for all $u \in \mathcal{V}$, which implies $P\left(\overline{\mathcal{F}}\right) = P\left(\bigcup_{u\in\mathcal{V}} \overline{\mathcal{F}}_u\right) \leq \sum_{u\in\mathcal{V}} P\left(\overline{\mathcal{F}}_u\right) \leq \delta$. □

B.4. Conclude the Proof

Note that if we choose

$$c \geq \frac{1}{\sigma}\sqrt{dn\log\left(1+\frac{nT}{\sigma^2\lambda d}\right)+2\log\left(n^2T\right)}+\sqrt{\lambda}\max_{u\in\mathcal{V}}\|\boldsymbol{\theta}_u^*\|_2, \tag{30}$$

we have $P\left(\overline{\mathcal{F}}\right) \leq \frac{1}{nT}$. Hence for general X, we have

$$\begin{aligned} R^{\rho\alpha}(T) &\leq \frac{2c}{\rho\alpha}\mathbb{E}\left\{\sum_{t=1}^{T}\sum_{u\in\mathcal{S}_t}\sum_{v\in\mathcal{V}}\sqrt{x_v^T\Sigma_{u,t-1}^{-1}x_v}\middle|\mathcal{F}\right\}+\frac{1}{\rho} \\ &\leq \frac{2c}{\rho\alpha}n^{\frac{3}{2}}\sqrt{\frac{dKT\log\left(1+\frac{nT}{d\lambda\sigma^2}\right)}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}}+\frac{1}{\rho}. \end{aligned} \tag{31}$$

Note that with $c = \frac{1}{\sigma}\sqrt{dn\log\left(1+\frac{nT}{\sigma^2\lambda d}\right)+2\log\left(n^2T\right)}+\sqrt{\lambda}\max_{u\in\mathcal{V}}\|\boldsymbol{\theta}_u^*\|_2$, this regret bound is $\widetilde{O}\left(\frac{n^2d\sqrt{KT}}{\rho\alpha}\right)$. Similarly, for the special case $X = I$, we have

$$R^{\rho\alpha}(T) \leq \frac{2c}{\rho\alpha}n^{\frac{3}{2}}\sqrt{\frac{KT\log\left(1+\frac{T}{\lambda\sigma^2}\right)}{\lambda\log\left(1+\frac{1}{\lambda\sigma^2}\right)}}+\frac{1}{\rho}. \tag{32}$$

Note that with $c = \frac{n}{\sigma}\sqrt{\log\left(1+\frac{T}{\sigma^2\lambda}\right)+2\log\left(n^2T\right)}+\sqrt{\lambda}\max_{u\in\mathcal{V}}\|\boldsymbol{\theta}_u^*\|_2 \leq \frac{n}{\sigma}\sqrt{\log\left(1+\frac{T}{\sigma^2\lambda}\right)+2\log\left(n^2T\right)}+\sqrt{\lambda n}$, this regret bound is $\widetilde{O}\left(\frac{n^{\frac{5}{2}}\sqrt{KT}}{\rho\alpha}\right)$.

C. Self-Normalized Bound for Matrix-Valued Martingales

In this section, we derive a "self-normalized bound" for matrix-valued Martingales. This result is a natural generalization of Theorem 1 in Abbasi-Yadkori et al. (2011).

Theorem 3. *(Self-Normalized Bound for Matrix-Valued Martingales) Let* $\{\mathcal{H}_t\}_{t=0}^{\infty}$ *be a filtration, and* $\{\eta_t\}_{t=1}^{\infty}$ *be a* $\Re^K$*-valued Martingale difference sequence with respect to* $\{\mathcal{H}_t\}_{t=0}^{\infty}$*. Specifically, for all* t*,* η_t *is* $\mathcal{H}_t$*-measurable and satisfies (1)* $\mathbb{E}\left[\eta_t|\mathcal{H}_{t-1}\right] = 0$ *and (2)* $\|\eta_t\|_\infty \leq 1$ *with probability 1 conditioning on* $\mathcal{H}_{t-1}$*. Let* $\{X_t\}_{t=1}^{\infty}$ *be a* $\Re^{d\times K}$*-valued stochastic process such that* X_t *is* $\mathcal{H}_{t-1}$ *measurable. Assume that* $V \in \Re^{d\times d}$ *is a positive-definite matrix. For any* $t \geq 0$*, define*

$$\overline{V}_t = V + K\sum_{s=1}^{t}X_sX_s^T \qquad S_t = \sum_{s=1}^{t}X_s\eta_s. \tag{33}$$

Then, for any $\delta > 0$*, with probability at least* $1-\delta$*, we have*

$$\|S_t\|^2_{\overline{V}_t^{-1}} \leq 2\log\left(\frac{\det\left(\overline{V}_t\right)^{1/2}\det\left(V\right)^{-1/2}}{\delta}\right) \quad \forall t \geq 0. \tag{34}$$

We first define some useful notations. Similarly as Abbasi-Yadkori et al. (2011), for any $\lambda \in \Re^d$ and any t, we define D_t^λ as

$$D_t^\lambda = \exp\left(\lambda^T X_t\eta_t - \frac{K}{2}\|X_t^T\lambda\|_2^2\right), \tag{35}$$

and $M_t^\lambda = \prod_{s=1}^{t} D_s^\lambda$ with convention $M_0^\lambda = 1$. Note that both D_t^λ and M_t^λ are $\mathcal{H}_t$-measurable, and $\left\{M_t^\lambda\right\}_{t=0}^{\infty}$ is a

supermartingale with respect to the filtration $\{\mathcal{H}_t\}_{t=0}^{\infty}$. To see it, notice that conditioning on $\mathcal{H}_{t-1}$, we have

$$\lambda^T X_t \eta_t = (X_t^T \lambda)^T \eta_t \leq \|X_t^T \lambda\|_1 \|\eta_t\|_\infty \leq \|X_t^T \lambda\|_1 \leq \sqrt{K}\|X_t^T \lambda\|_2$$

with probability 1. This implies that $\lambda^T X_t \eta_t$ is conditionally $\sqrt{K}\|X_t^T \lambda\|_2$-subGaussian. Thus, we have

$$\mathbb{E}\left[D_t^\lambda \middle| \mathcal{H}_{t-1}\right] = \mathbb{E}\left[\exp\left(\lambda^T X_t \eta_t\right) \middle| \mathcal{H}_{t-1}\right] \exp\left(-\frac{K}{2}\|X_t^T \lambda\|_2^2\right) \leq \exp\left(\frac{K}{2}\|X_t^T \lambda\|_2^2 - \frac{K}{2}\|X_t^T \lambda\|_2^2\right) = 1.$$

Thus,

$$\mathbb{E}\left[M_t^\lambda \middle| \mathcal{H}_{t-1}\right] = M_{t-1}^\lambda \mathbb{E}\left[D_t^\lambda \middle| \mathcal{H}_{t-1}\right] \leq M_{t-1}^\lambda.$$

So $\left\{M_t^\lambda\right\}_{t=0}^{\infty}$ is a supermartingale with respect to the filtration $\{\mathcal{H}_t\}_{t=0}^{\infty}$. Then, following Lemma 8 of Abbasi-Yadkori et al. (2011), we have the following lemma:

Lemma 4. *Let τ be a stopping time with respect to the filtration $\{\mathcal{H}_t\}_{t=0}^{\infty}$. Then for any $\lambda \in \Re^d$, M_τ^λ is almost surely well-defined and $\mathbb{E}\left[M_\tau^\lambda\right] \leq 1$.*

Proof. First, we argue that M_τ^λ is almost surely well-defined. By Doob's convergence theorem for nonnegative supermartingales, $M_\infty^\lambda = \lim_{t\to\infty} M_t^\lambda$ is almost surely well-defined. Hence M_τ^λ is indeed well-defined independent of $\tau < \infty$ or not. Next, we show that $\mathbb{E}\left[M_\tau^\lambda\right] \leq 1$. Let $Q_t^\lambda = M_{\min\{\tau,t\}}^\lambda$ be a stopped version of $\left\{M_t^\lambda\right\}_{t=1}^{\infty}$. By Fatou's Lemma, we have $\mathbb{E}\left[M_\tau^\lambda\right] = \mathbb{E}\left[\liminf_{t\to\infty} Q_t^\lambda\right] \leq \liminf_{t\to\infty} \mathbb{E}\left[Q_t^\lambda\right] \leq 1$. □

The following results follow from Lemma 9 of Abbasi-Yadkori et al. (2011), which uses the "method of mixtures" technique. Let Λ be a Gaussian random vector in $\Re^d$ with mean 0 and covariance matrix V^{-1}, and independent of all the other random variables. Let $\mathcal{H}_\infty$ be the tail σ-algebra of the filtration, i.e. the σ-algebra generated by the union of all events in the filtration. We further define $M_t = \mathbb{E}\left[M_t^\Lambda \middle| \mathcal{H}_\infty\right]$ for all $t = 0, 1, \ldots$ and $t = \infty$. Note that M_∞ is almost surely well-defined since M_∞^λ is almost surely well-defined.

Let τ be a stopping time with respect to the filtration $\{\mathcal{H}_t\}_{t=0}^{\infty}$. Note that M_τ is almost surely well-defined since M_∞ is almost surely well-defined. Since $\mathbb{E}\left[M_\tau^\lambda\right] \leq 1$ from Lemma 4, we have

$$\mathbb{E}\left[M_\tau\right] = \mathbb{E}\left[M_\tau^\Lambda\right] = \mathbb{E}\left[\mathbb{E}\left[M_\tau^\Lambda \middle| \Lambda\right]\right] \leq 1.$$

The following lemma follows directly from the proof for Lemma 9 of Abbasi-Yadkori et al. (2011), which can be derived by algebra. The proof is omitted here.

Lemma 5. *For all finite $t = 0, 1, \ldots$, we have*

$$M_t = \left(\frac{\det(V)}{\det(\overline{V}_t)}\right)^{1/2} \exp\left(\frac{1}{2}\|S_t\|_{\overline{V}_t^{-1}}\right). \tag{36}$$

Note that Lemma 5 implies that for finite t, $\|S_t\|_{\overline{V}_t^{-1}}^2 > 2\log\left(\frac{\det(\overline{V}_t)^{1/2}\det(V)^{-1/2}}{\delta}\right)$ and $M_t > \frac{1}{\delta}$ are equivalent. Consequently, for any stopping time τ, the event

$$\left\{\tau < \infty,\ \|S_\tau\|_{\overline{V}_\tau^{-1}}^2 > 2\log\left(\frac{\det\left(\overline{V}_\tau\right)^{1/2}\det\left(V\right)^{-1/2}}{\delta}\right)\right\}$$

is equivalent to $\left\{\tau < \infty,\ M_\tau > \frac{1}{\delta}\right\}$. Finally, we prove Theorem 3:

Proof. We define the "bad event" at time $t = 0, 1, \ldots$ as:

$$B_t(\delta) = \left\{\|S_t\|_{\overline{V}_t^{-1}}^2 > 2\log\left(\frac{\det\left(\overline{V}_t\right)^{1/2}\det\left(V\right)^{-1/2}}{\delta}\right)\right\}.$$

We are interested in bounding the probability of the "bad event" $\bigcup_{t=1}^{\infty} B_t(\delta)$. Let Ω denote the sample space, for any outcome $\omega \in \Omega$, we define $\tau(\omega) = \min\{t \geq 0 : \omega \in B_t(\delta)\}$, with the convention that $\min \emptyset = +\infty$. Thus, τ is a stopping time. Notice that $\bigcup_{t=1}^{\infty} B_t(\delta) = \{\tau < \infty\}$. Moreover, if $\tau < \infty$, then by definition of τ, we have $\|S_\tau\|^2_{\overline{V}_\tau^{-1}} > 2\log\left(\frac{\det\left(\overline{V}_\tau\right)^{1/2}\det(V)^{-1/2}}{\delta}\right)$, which is equivalent to $M_\tau > \frac{1}{\delta}$ as discussed above. Thus we have

$$
\begin{aligned}
P\left(\bigcup_{t=1}^{\infty} B_t(\delta)\right) &\overset{(a)}{=} P\left(\tau < \infty\right) \\
&\overset{(b)}{=} P\left(\|S_\tau\|^2_{\overline{V}_\tau^{-1}} > 2\log\left(\frac{\det\left(\overline{V}_\tau\right)^{1/2}\det\left(V\right)^{-1/2}}{\delta}\right),\, \tau < \infty\right) \\
&\overset{(c)}{=} P\left(M_\tau > 1/\delta,\, \tau < \infty\right) \\
&\leq P\left(M_\tau > 1/\delta\right) \\
&\overset{(d)}{\leq} \delta,
\end{aligned}
$$

where equalities (a) and (b) follow from the definition of τ, equality (c) follows from Lemma 5, and inequality (d) follows from Markov's inequality. This concludes the proof for Theorem 3. □

We conclude this section by briefly discussing a special case. If for any t, the elements of η_t are statistically independent conditioning on $\mathcal{H}_{t-1}$, then we can prove a variant of Theorem 3: with $\overline{V}_t = V + \sum_{s=1}^{t} X_s X_s^T$ and $S_t = \sum_{s=1}^{t} X_s \eta_s$, Equation 34 holds with probability at least $1-\delta$. To see it, notice that in this case

$$
\begin{aligned}
\mathbb{E}\left[\exp\left(\lambda^T X_t \eta_t\right)\middle|\mathcal{H}_{t-1}\right] &= \mathbb{E}\left[\prod_{k=1}^{K}\exp\left((X_t^T\lambda)(k)\eta_t(k)\right)\middle|\mathcal{H}_{t-1}\right] \\
&\overset{(a)}{=} \prod_{k=1}^{K}\mathbb{E}\left[\exp\left((X_t^T\lambda)(k)\eta_t(k)\right)\middle|\mathcal{H}_{t-1}\right] \\
&\overset{(b)}{\leq} \prod_{k=1}^{K}\exp\left(\frac{(X_t^T\lambda)(k)^2}{2}\right) = \exp\left(\frac{\left\|X_t^T\lambda\right\|^2}{2}\right),
\end{aligned} \tag{37}
$$

where (k) denote the k-th element of the vector. Note that the equality (a) follows from the conditional independence of the elements in η_t, and inequality (b) follows from $|\eta_t(k)| \leq 1$ for all t and k. Thus, if we redefine $D_t^\lambda = \exp\left(\lambda^T X_t \eta_t - \frac{1}{2}\|X_t^T\lambda\|_2^2\right)$, and $M_t^\lambda = \prod_{s=1}^{t} D_s^\lambda$, we can prove that $\{M_t^\lambda\}_t$ is a supermartingale. Consequently, using similar analysis techniques, we can prove the variant of Theorem 3 discussed in this paragraph.

D. Laplacian Regularization

As explained in section 7, enforcing Laplacian regularization leads to the following optimization problem:

$$
\widehat{\boldsymbol{\theta}}_t = \arg\min_{\boldsymbol{\theta}}\Big[\sum_{j=1}^{t}\sum_{u\in\mathcal{S}_t}(y_{u,j} - \boldsymbol{\theta}_u X)^2 + \lambda_2 \sum_{(u_1,u_2)\in\mathcal{E}} ||\boldsymbol{\theta}_{u_1} - \boldsymbol{\theta}_{u_2}||_2^2\Big]
$$

Here, the first term is the data fitting term, whereas the second term is the Laplacian regularization terms which enforces smoothness in the source node estimates. This can optimization problem can be re-written as follows:

$$
\widehat{\boldsymbol{\theta}}_t = \arg\min_{\boldsymbol{\theta}}\left[\sum_{j=1}^{t}\sum_{u\in\mathcal{S}_t}(y_{u,j} - \boldsymbol{\theta}_u X)^2 + \lambda_2 \boldsymbol{\theta}^T (L \otimes I_d)\boldsymbol{\theta}\right]
$$

Here, $\boldsymbol{\theta} \in \Re^{dn}$ is the concatenation of the n d-dimensional $\boldsymbol{\theta}_u$ vectors and $A \otimes B$ refers to the Kronecker product of matrices A and B. Setting the gradient of equation 38 to zero results in solving the following linear system:

$$[XX^T \otimes I_n + \lambda_2 L \otimes I_d]\widehat{\boldsymbol{\theta}}_t = b_t \tag{38}$$

Here b_t corresponds to the concatenation of the n d-dimensional vectors $b_{u,t}$. This is the Sylvester equation and there exist sophisticated methods of solving it. For simplicity, we focus on the special case when the features are derived from the Laplacian eigenvectors (Section 7).

Let β_t be a diagonal matrix such that $\beta_t u, u$ refers to the number of times node u has been selected as the source. Since the Laplacian eigenvectors are orthogonal, when using Laplacian features, $XX^T \otimes I_n = \beta \otimes I_d$. We thus obtain the following system:

$$[(\beta + \lambda_2 L) \otimes I_d]\widehat{\boldsymbol{\theta}}_t = b_t \tag{39}$$

Note that the matrix $(\beta + \lambda_2 L)$ and thus $(\beta + \lambda_2 L) \otimes I_d$ is positive semi-definite and can be solved using conjugate gradient (Hestenes & Stiefel, 1952).

For conjugate gradient, the most expensive operation is the matrix-vector multiplication $(\beta + \lambda_2 L) \otimes I_d]\mathbf{v}$ for an arbitrary vector $\mathbf{v}$. Let vec be an operation that takes a $d \times n$ matrix and stacks it column-wise converting it into a dn-dimensional vector. Let V refer to the $d \times n$ matrix obtained by partitioning the vector $\mathbf{v}$ into columns of V. Given this notation, we use the property that $(B^T \otimes A)\mathbf{v} = vec(AVB)$. This implies that the matrix-vector multiplication can then be rewritten as follows:

$$(\beta + \lambda_2 L) \otimes I_d \mathbf{v} = \text{vec}(V\left(\beta + \lambda_2 L^T\right)) \tag{40}$$

Since β is a diagonal matrix, $V\beta$ is an $O(dn)$ operation, whereas VL^T is an $O(dm)$ operation since there are only m non-zeros (corresponding to edges) in the Laplacian matrix. Hence the complexity of computing the mean $\widehat{\boldsymbol{\theta}}_t$ is an order $O((d(m+n))\kappa)$ where κ is the number of conjugate gradient iterations. In our experiments, similar to (Vaswani et al., 2017), we warm-start with the solution at the previous round and find that $\kappa = 5$ is enough for convergence.

Unlike independent estimation where we update the UCB estimates for only the selected nodes, when using Laplacian regularization, the upper confidence values for each reachability probability need to be recomputed in each round. Once we have an estimate of $\boldsymbol{\theta}$, calculating the mean estimates for the reachabilities for all u, v requires $O(dn^2)$ computation. This is the most expensive step when using Laplacian regularization.

We now describe how to compute the confidence intervals. For this, let $\boldsymbol{D}$ denote the diagonal of $(\beta + \lambda_2 L)^{-1}$. The UCB value $z_{u,v,t}$ can then be computed as:

$$z_{u,v,t} = \sqrt{\boldsymbol{D}_u}||x_v||_2 \tag{41}$$

The ℓ_2 norms for all the target nodes v can be pre-computed. If we maintain the $\boldsymbol{D}$ vector, the confidence intervals for all pairs can be computed in $O(n^2)$ time.

Note that $\boldsymbol{D}_t$ requires $O(n)$ storage and can be updated across rounds in $O(K)$ time using the Sherman Morrison formula. Specifically, if $\boldsymbol{D}_{u,t}$ refers to the u^{th} element in the vector $\boldsymbol{D}_t$, then

$$\boldsymbol{D}_{u,t+1} = \begin{cases} \dfrac{\boldsymbol{D}_{u,t}}{(1 + \boldsymbol{D}_{u,t})}, & \text{if} u \in \mathcal{S}_t \\ \boldsymbol{D}_{u,t}, & \text{otherwise} \end{cases}$$

Hence, the total complexity of implementing Laplacian regularization is $O(dn^2)$. We need to store the $\boldsymbol{\theta}$ vector, the Laplacian and the diagonal vectors β and $\boldsymbol{D}$. Hence, the total memory requirement is $O(dn + m)$.