Linear Bandits With Side Observations on Networks

Avik Kar, *Member, IEEE*, Rahul Singh, *Member, IEEE*, Fang Liu, Xin Liu, *Fellow, IEEE*, and Ness B. Shroff, *Fellow, IEEE*

***Abstract*— We investigate linear bandits in a network setting in the presence of side-observations across nodes in order to design recommendation algorithms for users connected via social networks. Users in social networks respond to their friends' activity and, hence, provide information about each other's preferences. In our model, when a learning algorithm recommends an article to a user, not only does it observe her response (e.g., an ad click) but also the side-observations, i.e., the response of her neighbors if they were presented with the same article. We model these observation dependencies by a graph $\mathcal{G}$ in which nodes correspond to users and edges to social links. We derive a problem/instance-dependent lower-bound on the regret of any consistent algorithm. We propose an optimization-based data-driven learning algorithm that utilizes the structure of $\mathcal{G}$ in order to make recommendations to users and show that it is asymptotically optimal, in the sense that its regret matches the lower-bound as the number of rounds $T \to \infty$. We show that this asymptotically optimal regret is upper-bounded as $O\left(|\chi(\mathcal{G})| \log T\right)$, where $|\chi(\mathcal{G})|$ is the domination number of $\mathcal{G}$. In contrast, a naive application of the existing learning algorithms results in $O\left(N \log T\right)$ regret, where N is the number of users.**

***Index Terms*— Multi-armed bandits, contextual bandits, networks.**

I. Introduction

THE linear multi-armed bandit model is popularly used in order to place ads and make personalized recommendations of news articles to users of web services [1], [2], [3]. In this model, both users and contents are represented by sets of features. For example, user features are obtained on the basis of their historical behavior and demographic information; while content feature depends upon its category and descriptive information. A learning algorithm sequentially recommends articles to users based on the information about the articles and preferences of users, while continually adapting its strategy to present articles on the basis of feedback, e.g., ad clicks, downloads, etc., received from users. Its goal is to maximize the cumulative reward, which is equal to the total number of user clicks in the long run.

We consider the problem of making recommendations to users of a social network such as Facebook, Goodreads, LinkedIn. If users' preferences were known, we could employ an optimal stationary strategy that maps the feature of each user to its optimal action, i.e., present her with an article that has the highest click-probability. Since users' preferences are typically unknown, one could employ an efficient linear-bandit learning algorithm as in [2], [3], and [4] on each user separately. This strategy achieves a regret of $O\left(N \log T\right)$, where N is the number of users. However, since the number of users can be very large (e.g., Facebook has 2.5 billion users [5]), this strategy is impractical.

Consider a social network modeled by an undirected graph $\mathcal{G}$ in which the nodes correspond to users, and undirected edges correspond to "social links," i.e., two users are connected by an edge if they are "friends." Since individual users are connected to a subset of the remaining users, each time the algorithm makes a recommendation to a user, it also obtains *side observations*, i.e., feedback from her "neighbors" regarding their potential interest in a similar offer. Side observations could be generated in several ways, we discuss two possibilities: (i) When a content is promoted to a user i, she may share it with her friend. Alternatively, this user may post her feedback about this content on her network so that her friends also get to share their feedback on it. These responses of user i's friends then constitute side observations [6], [7]. (ii) When user i is presented with a promotion x, her neighbors could be explicitly queried as follows: "Would you be interested in promotion x that was offered to your friend i?" [8], [9].

We design learning algorithms that incorporate these side-observations into the decision-making process for making recommendations. We show that the regret of the proposed algorithms scales at most as $O\left(|\chi(\mathcal{G})| \log T\right)$, where $|\chi(\mathcal{G})|$ is the domination number (see Definition 4) of graph $\mathcal{G}$. Since $|\chi(\mathcal{G})| \ll N$ for graphs describing social networks,[1] our algorithms drastically reduce the dependence of the regret on the number of users.

In our setup, choosing an action in the multi-armed bandit problem corresponds to making a recommendation to a *single* user in network. We work with *linear* bandit models, i.e., the

Manuscript received 21 February 2023; revised 27 September 2023 and 15 February 2024; accepted 5 June 2024; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor C. Joe-Wong. Date of publication 8 July 2024; date of current version 17 October 2024. The work of Avik Kar and Rahul Singh was supported in part by the Science and Engineering Research Board under Grant SRG/2021/00230. The work of Xin Liu was supported in part by Grant USDA-020-67021-32855 and Grant NSF OIA-2134901. The work of Fang Liu and Ness B. Shroff was supported in part by NSF under Grant CNS-2312836, Grant CNS-2223452, Grant CNS-2225561, Grant CNS-2112471, and Grant CNS-2106933. *(Corresponding author: Rahul Singh.)*

Avik Kar and Rahul Singh are with the Department of ECE, Indian Institute of Science Bengaluru, Bengaluru, Karnataka 560012, India (e-mail: avikkar@iisc.ac.in; rahulsingh0188@gmail.com; rahulsingh@iisc.ac.in).

Fang Liu is with Facebook, San Francisco, CA 94025 USA (e-mail: fangliu0302@gmail.com).

Xin Liu is with the Department of Computer Science, University of California at Davis, Davis, CA 95616 USA (e-mail: xinliu@ucdavis.edu).

Ness B. Shroff is with the Department of ECE, The Ohio State University, Columbus, OH 43210 USA (e-mail: shroff@ece.osu.edu).

Digital Object Identifier 10.1109/TNET.2024.3422323

[1] [10], [11] show that for social networks, $\chi(G)$ can be bounded by a sublinear function of the number of nodes; thus $\chi(G) \ll |\mathcal{V}|$ when the number of nodes is large.

one-step expected reward (e.g., user's ad-click probability) of an arm is a linear function of the arm. This (unknown) linear function depends upon the user on which this arm is played. For this bandit model with side observations, we derive a lower bound on the regret of any consistent algorithm and show that our algorithms are asymptotically (as $T \to \infty$) optimal since their asymptotic regret matches this lower bound.

Note that we assume that side observations are obtained from each neighbor. An alternative model would be to suggest an item to a neighbor only if it is liked by a user, i.e., to ask questions of the form "Would you be interested in a promotion that was liked by your friend" instead of our setup in which the query is "Would you be interested in a promotion that was offered to your friend." This model could be studied in a future work.

A. Related Work

We begin by describing existing works on linear bandits and then discuss works that derive learning algorithms for graphical bandits and cooperative bandits.

1) Bandits with Linear Pay-off Functions: Bandits with linear pay-off functions have been extensively studied. The efficiency of a learning algorithm is measured by its regret [12], [13], which is the expected value of the difference between the cumulative reward collected by an algorithm that knows the true parameters of the problem instance and hence makes the optimal choice in each round, and the reward collected by the learning algorithm. Upper Confidence Bound (UCB) [14], [15] based algorithms that use optimism in the face of uncertainty have been developed in works such as [1], [2], and [16]. Reference [16] analyzes LinUCB and shows that its minimax/worst-case regret scales as $\tilde{O}(\sqrt{dT})$, where d is the dimension of the feature space.[2] References [17] and [18] utilize Thompson sampling [19], [20] and prove that its regret scales as $\tilde{O}\left(\frac{d^2}{\epsilon}\sqrt{T^{1+\epsilon}}\right)$, where $0 < \epsilon < 1$. Reference [21] studies Reward Biased Maximum Likelihood Estimation (RBMLE) algorithm [22], [23], [24] for linear stochastic bandits, and shows that it enjoys a $\tilde{O}(d\sqrt{T})$ regret. However, we focus on developing algorithms that have provably optimal problem-dependent regret guarantees [12], [13]. Problem-dependent regret guarantees quantify the learning regret of an algorithm in terms of certain quantities of the problem instance; in contrast, problem-independent guarantees involve a worst-case (minmax) analysis and yield a bound that holds for a class of problem instances. Regret guarantee in Theorem 2 depends upon $\Delta_{\max,i}$ (4), which is the difference between rewards of the best arm and worst arm at node i, while that in Theorem 3 depends upon the value of optimization problem (8)-(10) that involves parameters of the underlying bandit problem. Both these quantities depend upon some properties of the problem instance. As has been shown in [4], the performance of learning algorithms based on UCB, or Thompson sampling can be arbitrarily far from optimal in this setting. For contextual MAB with similarity information other than linearity, see [25] and [26]. Finite-time problem-dependent guarantees for linear bandits have been derived in [27] and [28], however, these are far from optimal. Reference [4] studies problem-dependent regret in the asymptotic regime (when $T \to \infty$), and derives an algorithm that is asymptotically optimal. We focus exclusively on the problem-dependent setting and build upon the techniques of [4].

2) Graphical Bandits: A related setup is the graphical bandits model introduced in [29] and [30]. Reference [29], [30] considered the presence of *side-observations* in the adversarial multi-armed bandit setting [31], i.e., when the decision maker plays an arm, not only does it receive reward from the arm that was played, but it also gets to observe the rewards of "neighboring" arms of the arm that was played. The observation dependencies are encoded as an undirected graph $\mathcal{G}$ in which two nodes i, j are connected by an edge only if pulling an arm also reveals reward of the other arm. References [29] and [30] derive algorithms and analyze their learning regret under varied assumptions on the model. More specifically, [29], [30] study both the "uninformed setting" (where the feedback graph is not visible to the algorithm) as well as the "informed setting" and allow the feedback graph to be both directed (so that the feedbacks are unidirectional) as well as symmetric. They also allow the graph to be time-varying. The regret bounds specialized to the case of fixed, symmetric graph are $O(\sqrt{\alpha(\mathcal{G})T})$, where $\alpha(\mathcal{G})$ denotes the independence number of the graph $\mathcal{G}$, in all of these cases.

The works [8], [9], and [32] consider a setup similar to the graphical bandits problem of [29] and [30], but instead of assuming that the rewards are generated by an adversary, these works assume that the rewards are stochastic, i.e., the rewards from an arm are i.i.d. across time, and its distribution depends upon the arm. Reference [8] derives algorithms whose regret scales as $O(|\gamma(\mathcal{G})| \log T)$, where $|\gamma(\mathcal{G})|$ is the clique cover number of the graph[3] that describes observation dependencies. Reference [9] improves the regret to $O(|\chi(\mathcal{G})| \log T)$, where $|\chi(\mathcal{G})|$ is domination number of $\mathcal{G}$. The key insight gained from [8] and [9] is that in the presence of side-observations, not only does an efficient algorithm need to take into account the history of rewards obtained from an arm but also the location of the arm in the graph $\mathcal{G}$. Thus, for example, it might even be optimal to pull an arm with a low estimate of mean reward, because it is connected to relatively unexplored arms, and the "exploration gains" resulting from side-observations outweigh the (relatively larger) instantaneous regret of this arm. Reference [33] analyzes the feedback graph model in a setting in which the graph may vary from round to round, and is never revealed to the learner. Reference [8] develops UCB-based algorithms, while [9] additionally develops ϵ_t greedy-based policies in which the number of exploration steps that are to be spent on each arm is obtained by solving a linear program. Reference [7] considers a "networked bandit" setup in which dependencies amongst arms are described via a graph, and pulling an arm yields rewards of neighboring arms also in addition to the reward of the pulled arm. It proposes a

[2] $\tilde{O}(\cdot)$ hides factors that are logarithmic in number of rounds T.

[3] A clique cover of a given undirected graph is a partition of the vertices of the graph into cliques, i.e., subsets of vertices within which every two vertices are adjacent. Clique cover number is the smallest number of cliques using which nodes of $\mathcal{G}$ can be covered.

UCB-based algorithm and shows that its regret is bounded as $O(\sqrt{T})$, where the prefactor is linear in the number of arms. Our setup considers $|\mathcal{V}|$ parallel multi-armed bandit problems, one at each node of the graph. Reference [7] considers a single multi-armed bandit problem in which the total number of arms is $|\mathcal{V}|$, and upon pulling an arm, one gets to view the rewards of other neighboring arms. Moreover, our proposed algorithm incorporates side-observations while pulling arms so that the regret scales linearly with the domination number of the graph, rather than the number of nodes. Another difference with [7] is that we analyze instance-dependent regret, and show that this is $O(\log T)$, while the focus in [7] is on minmax regret, which scales as $O(\sqrt{T})$.

In departure from these works, our work considers a network setup in which side-observations occur across nodes and not arms, and moreover, the payoffs received are unknown linear functions of the arm. Note that in order to study this setup, we cannot use the approach/tools proposed [8] or [9]. This requires us to develop several novel tools and techniques. In effect, the resulting algorithms are vastly different from [8] and [9]. The resulting prefactor, interestingly, turns out to be the domination number of the connectivity graph. Our analysis builds upon the tools from [4]. For bandits with feedback graphs, [34] analyzes how the structure of the feedback graph affects the difficulty of the learning problem, while [30] studies online prediction problems in partial information regimes that interpolate between the classical bandit and expert settings and derives lower bounds for the multi-armed bandits with feedback graph model.

3) Cooperative Bandits: In this setup, multiple agents that can communicate via a network collectively solve a single instance of K-armed bandit problem. Reference [35] considers the case of a fixed network and runs consensus algorithm for generating each agent's estimate of mean rewards from its own rewards and the estimated rewards of its neighbors. It shows that the agents asymptotically recover the performance of a centralized agent. Reference [36] proposes an extension of the UCB1 algorithm [15] and analyzes its regret. Reference [37] develops an algorithm based on partitions of the communication graph, while [38] uses an accelerated consensus procedure to compute estimates of the average of rewards obtained by all the agents for each arm, and then uses an upper confidence bound algorithm that accounts for the delay and error of the estimates. References [39] and [40] consider a variation of this setup in which the agents have limited access to a local subset of arms. Reference [41] considers a variation in which some of the agents are malicious and can disrupt learning. Reference [42] shows that even a limited amount $(O(\log T))$ of communication about the identities of the arms played by the agents helps to "speed up" the learning process so that the prefactor in the per-agent regret scales down by the number of agents. Reference [43] extends this to the case when an agent is allowed to communicate with *only* a randomly chosen agent. In contrast with these works on collaborative bandits, in our case each node of the network has a different instance of a *linear* bandit problem. Hence, side-observation, in our case, corresponds to the reward generated when the same arm is played on a neighboring node's bandit problem instance (and not the common bandit problem instance).

4) Learning in Social Networks: Reference [44] derives algorithms that learn the state of the social network. Reference [45] considers the problem of choosing which articles should be published on a social network account so that the number of forwards is maximized. Reference [46] considers the case when there are multiple agents that operate in a decentralized way, each selling a different set of items to incoming customers. The agents recommend items to customers, obtain a reward even if they are able to sell the item of another agent. This work uses the contextual bandit framework to study how to maximize the cumulative profit of these agents. Reference [47] proposes targeted crawling algorithms using the theory of multi-armed bandits in order to find a profile matching some criteria in a social network.

B. Our Contributions

Our main contributions are as follows:

1) We consider linear multi-armed bandits in a network setting in which each node (user) of the network corresponds to a separate instance of linear bandit. Upon playing an arm on a node, in addition to collecting reward, one also receives "side-observations" from neighboring nodes. These side-observations are the rewards that would have been received from neighboring nodes if the same arm was played on them. We derive an instance dependent lower-bound on the regret of any consistent policy.[4] This bound is the optimal value of an optimization problem that is parametrized by the graph that describes the side-observations dependencies and the (unknown) sub-optimality gaps of arms.
2) We propose a UCB-type learning algorithm that explores the values of unknown coefficients of users using a barycentric spanner of the set of arms for each user in the network. It maintains confidence balls for the rewards of arms and uses a stopping rule in order to decide when to stop the exploration phase. At the end of exploration phase, it plays those arms that are optimal given the current estimates of coefficients. We analyze its finite-time regret, and show that it can be upper-bounded as $O(|\mathcal{V}|\log T)$, where $|\mathcal{V}|$ is the number of users, with a prefactor that depends upon the sub-optimality gaps of rewards of arms. This is a simple algorithm with a good regret bound, but does not match the lower bound.
3) To close the gap mentioned in 2) above, we develop a learning algorithm that is composed of three phases. During the *warm-up phase*, it samples each user's coefficient vectors for fixed $O(\log^{1/2} T)$ number of rounds. Thereafter, in the *success phase* it uses these samples to estimate the unknown sub-optimality gaps of arms, and inputs these estimates into the optimization problem (31)-(33). The solution of this problem then yields the number of times each arm is to be played. The algorithm uses a detector in order to constantly track

[4]A policy whose regret is smaller than $o(T^p)$, $\forall p > 0$ and all possible instances of problem.

the quality of estimates obtained at the end of warm-up phase. In the event it detects that these estimates are "bad," it enters into the *recovery phase* and switches to the UCB-type algorithm described in 2) above. We show that this "data-dependent" algorithm's regret asymptotically matches the lower-bound.

II. Problem Setting

The social network of interest is modeled by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, in which the nodes $\mathcal{V}$ represent users, while undirected edges $\mathcal{E}$ represent social connections. We let $N := |\mathcal{V}|$ denote the number of users. Associated with each node $i \in \mathcal{V}$ is a "coefficient vector" $\theta_i^\star \in \mathbb{R}^d$. In each round $t = 1, 2, \ldots, T$, the decision maker recommends articles to each $i \in \mathcal{V}$. Let $U_i(t) \in \mathcal{U} \subset \mathbb{R}^d$ denote the arm played on node i, i.e., feature of the article presented to user i at round t. The decision maker has to choose $U_i(t)$ for each user $i \in \mathcal{V}$ at times $t = 1, 2, \ldots, T$. Presenting article to a user i also reveals "side-observations" on its neighboring nodes $\mathcal{N}_i := \{j : (i,j) \in \mathcal{E}\}$. These are rewards that would have been obtained if the same article was presented to users in the set $\mathcal{N}_i$. We let $r_i(t)$ denote the reward received from recommendation to user i during round t. Also, let $y_{(i,j)}(t)$ denote the side-observation obtained from user j as a result of recomendation to i during round t. We let $\mathcal{F}_t$ be the sigma-algebra generated by $\left\{\{r_i(s)\}_{i\in\mathcal{V}}, \{y_{(i,j)}(s) : j \in \mathcal{N}_i\}_{i\in\mathcal{V}}, \{U_i(s)\}_{i\in\mathcal{V}}\right\}_{s=1}^{t}$. Thus, it is the sigma-algebra generated by the operational history until round t. The reward earned from i is given by

$$r_i(t) = U_i^\intercal(t)\ \theta_i^\star + \eta_i(t),\ i \in \mathcal{V}, \tag{1}$$

where $\eta_i(t) \sim \mathcal{N}(0,1)$ is Gaussian and independent of $\mathcal{F}_t$. Side-observations are given by,

$$y_{(i,j)}(t) = U_i^\intercal(t)\ \theta_j^\star + \eta_{(i,j)}(t),\ \forall (i,j) \in \mathcal{E}, \tag{2}$$

where $\eta_{(i,j)}(t) \sim \mathcal{N}(0,1)$ are independent of $\mathcal{F}_t$, and independent across social links. We assume that $r_i(t) \in [0,1],\ \forall i \in \mathcal{V}$ and $y_{(i,j)}(t) \in [0,1],\ \forall (i,j) \in \mathcal{E}$. Note that we assume that the decision-maker knows $\mathcal{G}$. However, this is not a very restrictive assumption since the owner of social network, for example Facebook, has access to this information. It may be possible to remove this assumption by learning $\mathcal{G}$ in an "online fashion." Reference [33] takes this approach in the setup of online learning problems with feedback graphs. Note that we use undirected edges instead of directed edges in order to represent social connections, the reason is that in the context of social networks, friendship is a symmetric relation. Thus, if i is a friend of j, then j is also a friend of i. We would like to remark that our results and proofs can be appropriately modified even if we use a directed graph to model the network. We assume that the neighboring nodes always provide feedback. An alternative is to assume that only a subset of the neighbors share their preferences, possibly with additional costs. Another possibility is to allow the algorithm to recommend limited items to only a subset of the users and not to all the users, as is the case currently. These could be analyzed in future works.

A. Notation

Denote $\theta^\star := (\theta_1^\star, \theta_2^\star, \ \ldots, \theta_N^\star) \in \mathbb{R}^{d\times N}$ the vector consisting of unknown coefficients of N users. An "action" a that corresponds to playing arm $u \in \mathcal{U}$ on node (user) i is denoted by the tuple $a = (i,u)$. For an action a, we let u_a denote its arm and i_a its node. We let $\mathcal{A}_i := \{(i,u) : u \in \mathcal{U}\}$ denote the set of actions that correspond to presenting an article to user i, and let $\mathcal{A} := \cup_{i\in\mathcal{V}}\mathcal{A}_i$ denote the set of all actions. We assume that $\mathcal{U}$, and hence $\mathcal{A}_i$ are finite. An optimal action b for node i satisfies $b \in \arg\max_{a\in\mathcal{A}_i}\{u_a^\intercal\theta_i^\star\}$. We denote the mean reward of action a by μ_a, i.e., $\mu_a := u_a^\intercal\theta_{i_a}^\star$. We assume that each node $i \in \mathcal{V}$ has exactly one optimal arm, which is denoted by $u_i^\star$. $\mathcal{A}_i^{(s)} := \mathcal{A}_i \setminus (i, u_i^\star)$ denotes the set of actions corresponding to presenting a sub-optimal article on node i, and $\mathcal{A}^{(s)} := \cup_{i\in\mathcal{V}}\mathcal{A}_i^{(s)}$ denotes the set of all sub-optimal actions. We also say that two actions $a_1 = (i_1, u_1)$, $a_2 = (i_2, u_2)$ are neighboring actions if $(i_1, i_2) \in \mathcal{E}$. By notational abuse, we let $\mathcal{N}_a$ denote the set of neighboring actions of action a. Define

$$\Delta_a := \max_{u\in\mathcal{U}} u^\intercal\theta_{i_a}^\star - u_a^\intercal\theta_{i_a}^\star,$$

to be the difference (gap) between the mean reward of action a and the optimal action for node i_a. Let $\Delta_{\min,i}$ ($\Delta_{\max,i}$) be the difference between the mean reward of the best arm and second-best arm (worst arm) at node i, i.e.,

$$\Delta_{\min,i} := \min_{a\in\mathcal{A}_i^{(s)}} (u_i^\star)^\intercal\,\theta_i^\star - u_a^\intercal\theta_i^\star, \tag{3}$$

$$\Delta_{\max,i} := \max_{a\in\mathcal{A}_i^{(s)}} (u_i^\star)^\intercal\,\theta_i^\star - u_a^\intercal\theta_i^\star. \tag{4}$$

Define

$$\Delta_{\min} := \min_{a\in\mathcal{A}^{(s)}} \Delta_a,\ \Delta_{\max} := \max_{a\in\mathcal{A}} \Delta_a$$

to be the minimum value of the gap over all suboptimal actions and the maximum gap over all actions, respectively. Since we observe the reward of an action that is played, we say that a node is a neighbor of itself, i.e., $i \in \mathcal{N}_i$, or equivalently $(i,i) \in \mathcal{E},\ \forall i \in \mathcal{V}$. Thus, we let $y_{(i,i)}(s) = r_i(s)$. This notation drastically simplifies the exposition. This property is called "strong observability" in the literature [34]. We note that the preferences of user i are reflected in his/her coefficient vector $\theta_i^\star$. In case the underlying graph is such that the preference of a neighbor j is similar to that of i, then the quantity $\|\theta_i^\star - \theta_j^\star\|$ would be small for such neighboring nodes. We allow the preferences of neighbors to be different, i.e., $\|\theta_i^\star - \theta_j^\star\|$ can be large for neighboring nodes i and j.

All vectors are assumed to be column vectors. $0_{m\times n}$ denotes an $m\times n$ matrix comprised of only zeros. For a matrix M, $M^\intercal$ denotes its transpose, while $trace(M)$ denotes its trace, and $col_k(M)$ denotes its k-th column. For two vectors $x, y \in \mathbb{R}^d$, $x^\intercal y$ denotes the dot product between x and y. We let $N_a(t)$ denote the number of times action a has been played until round t. For a vector $x \in \mathbb{R}^d$ we let $\|x\|$ denote its Euclidean norm, and for a positive-definite matrix H, we let $\|x\|_H^2 := x^\intercal Hx$. We use $[1,T]$ to denote the set $\{1, 2, \ldots, T\}$.

B. Learning Algorithm and Regret

A learning algorithm $\pi : \mathcal{F}_t \mapsto \otimes_{i\in\mathcal{V}}\mathcal{A}_i, t = 1, 2, \ldots, T$ maps the observational history until each round t, to a set of $|\mathcal{V}|$ actions, one for each node, where each action corresponds to playing an arm for a node. The performance of π is measured by its regret $R^{\pi}_{(\theta^\star,\mathcal{G},\mathcal{A})}(T)$,

$$R^{\pi}_{(\theta^\star,\mathcal{G},\mathcal{A})}(T) := \mathbb{E}\sum_{t=1}^{T}\left(\sum_{i\in\mathcal{V}}(u_i^\star)^\intercal\theta_i^\star - U_i(t)^\intercal\theta_i^\star\right), \quad (5)$$

where the expectation above is taken with respect to the probability measure induced by the algorithm π, and randomness of rewards. We will occasionally omit the dependence of regret upon $(\theta^\star, \mathcal{G}, \mathcal{A})$. Our objective is to design a learning algorithm that has a low regret. Hence, we will restrict ourselves to the following class of *consistent algorithms*.

Definition 1 (Consistent Algorithm): A learning algorithm π is called consistent if for all $\theta^\star, \mathcal{A}, \mathcal{G}$ and $p \in (0, 1)$, it satisfies $R^{\pi}_{(\theta^\star,\mathcal{G},\mathcal{A})}(T) = o\left(T^p\right)$.

C. Barycentric Spanner

Now, we introduce the concept of a barycentric spanner and generalize it to the network setting, which will be crucial in our algorithm design. The following can be found in [48].

Definition 2 (Barycentric Spanner of a subset $\mathcal{S}$ of $\mathbb{R}^d$): A set of vectors $\tilde{\mathcal{S}} \subseteq \mathcal{S}$ is called barycentric spanner of $\mathcal{S}$ if each $u \in \mathcal{S}$ can be written as follows,

$$u = \sum_{w\in\tilde{\mathcal{S}}} \alpha_w\, w, \qquad \textit{where } \alpha_w \in [-1, 1].$$

The following result is Proposition 2.2 and Proposition 2.4 of [48].

Lemma 1: Let $\mathcal{S} \subset \mathbb{R}^d$. There exists a barycentric spanner of $\mathcal{S}$ that has cardinality less than or equal to d.

It is an open question whether or not a barycentric spanner can be computed efficiently [48]. However, an approximation to barycentric spanner can be computed efficiently. A set is called a C-approximate barycentric spanner of $\mathcal{S}$ if every $u \in \mathcal{S}$ can be expressed as a linear combination of elements of this set using coefficients in $[-C, C]$. The following result is essentially Proposition 2.4 of [48]. It shows that if the set $\mathcal{S}$ satisfies certain properties, then it is possible to efficiently compute an approximate barycentric spanner for it with an arbitrary level of desired accuracy.

Proposition 1: Suppose $\mathcal{S}$ is not contained in any proper linear subspace. Given an oracle for optimizing linear functions over $\mathcal{S}$, for any $C > 1$, we can compute a C-approximate barycentric spanner for $\mathcal{S}$ in polynomial time, using $O\left(d^2 \log_C d\right)$ calls to the optimization oracle.

Definition 3 (Barycentric Spanner of $(\mathcal{A}, \mathcal{G})$): Consider the set $\mathcal{U}$ of arms, and let $\tilde{\mathcal{U}}$ be its barycentric spanner. Then, the set of actions $\mathcal{BS}$,

$$\mathcal{BS} := \left\{(i, u) : i \in \mathcal{V},\ u \in \tilde{\mathcal{U}}\right\},$$

is a barycentric spanner of $(\mathcal{A}, \mathcal{G})$. In what follows, we let $\mathcal{BS}$ be such a barycentric spanner of cardinality Nd.

Definition 4: (Dominating set of a graph) A dominating set $\mathcal{V}'$ of graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is a set of nodes such that each node $i \in \mathcal{V}$ either (i) belongs to this set, i.e. $i \in \mathcal{V}'$ or (ii) is a neighbor of some node belonging to $\mathcal{V}'$, i.e. there exists $i' \in \mathcal{V}'$ such that $(i, i') \in \mathcal{E}$. Let $\chi(\mathcal{G})$ be a dominating set with minimum cardinality. $|\chi(\mathcal{G})|$ is called the domination number of $\mathcal{G}$.

III. Lower Bounds on Regret

Define

$$G_i(t) := \sum_{s=1}^{t}\sum_{j\in\mathcal{N}_i} U_j(s)U_j(s)^\intercal, \quad (6)$$

$$\bar{G}_i(t) := \mathbb{E}\left(G_i(t)\right),\ \forall i \in \mathcal{V}. \quad (7)$$

We have the following lower bound on the regret of any consistent learning algorithm. Auxiliary results required while proving it are deferred to the Appendix.

Theorem 1: Consider the following optimization problem,

$$OPT: \min_{\left\{\zeta(a):a\in\mathcal{A}^{(s)}\right\}} \sum_{a\in\mathcal{A}^{(s)}} \zeta(a)\Delta_a \quad (8)$$

$$\textit{s.t. } \|u_a\|^2_{H^{-1}_{i_a}(\zeta)} \le \frac{\Delta_a^2}{2}, \quad \forall a \in \mathcal{A}^{(s)}, \quad (9)$$

$$\textit{where } H_i(\zeta) := \sum_{j\in\mathcal{N}_i}\sum_{\{a:i_a=j\}} \zeta(a)u_a u_a^\intercal, \quad (10)$$

where $\zeta(a) \in [0, \infty),\ \forall a$, and we let $\zeta = \{\zeta(a) : a \in \mathcal{A}\}$. Let $c(\theta^\star, \mathcal{G}, \mathcal{A})$ denote its optimal value. The regret $R^\pi(T)$ of any consistent learning algorithm π satisfies

$$\limsup_{T\to\infty} \frac{R^\pi(T)}{\log T} \ge c(\theta^\star, \mathcal{G}, \mathcal{A}). \quad (11)$$

Note that solving OPT requires us to know the values Δ_a.

Proof: We begin by showing that under any consistent learning algorithm, we have

$$\limsup_{T\to\infty}\ \log(T)\|u_a\|^2_{\bar{G}^{-1}_{i_a}(T)} \le \frac{\Delta_a^2}{2}, \quad \forall a \in \mathcal{A}^{(s)}. \quad (12)$$

Consider a sub-optimal action $a \in \mathcal{A}_i^{(s)}$. Recall that $u_i^\star$ is the optimal arm at node i. We also let $a_i^\star := (i, u_i^\star)$. We have,

$$\begin{aligned}\|u_a\|_{\bar{G}_i^{-1}(T)} &\le \|u_a - u_i^\star\|_{\bar{G}_i(T)^{-1}} + \|u_i^\star\|_{\bar{G}_i(T)^{-1}} \\ &\le \|u_a - u_i^\star\|_{\bar{G}_i(T)^{-1}} + \frac{\|u_i^\star\|}{\sqrt{N_{a_i^\star}(T)}}, \end{aligned} \quad (13)$$

where the first inequality follows from the triangle inequality, while the second follows since from (6) we have that $G_i(T) \ge N_{a_i^\star}(T)\ u_i^\star\left(u_i^\star\right)^\intercal$, which yields $\bar{G}_i(T)^{-1} \le \left(N_{a_i^\star}(T)\right)^{-1}\left[u_i^\star\left(u_i^\star\right)^\intercal\right]^\dagger$, where for a matrix A, we let $A^\dagger$ denote its pseudoinverse. After multiplying both sides of (13) by $\log^{1/2} T$, we obtain

$$\begin{aligned}&\limsup_{T\to\infty}\ \log^{1/2} T\|u_a\|_{\bar{G}_i(T)^{-1}} \\ &\quad\le \limsup_{T\to\infty} \log^{1/2} T\|u_a - u_i^\star\|_{\bar{G}_i(T)^{-1}} \\ &\quad+ \limsup_{T\to\infty} \frac{\log^{1/2} T}{\sqrt{N_{a_i^\star}(T)}}\|u_i^\star\|.\end{aligned}$$

Under a consistent policy, we have $\lim_{T\to\infty} N_{a_i^\star}(T)/T = 1$, so that the second term on the r.h.s. vanishes. It follows from Lemma 9 that the first term on the r.h.s. is upper-bounded by $\Delta_a/\sqrt{2}$. Substituting these into the above inequality yields the proof of (12).

We now show (11). Let π be a consistent policy, and let $N_a(T)$ denote the number of times it takes action a until round T. Define $\zeta^{(T)}(a) := \frac{\mathbb{E}N_a(T)}{\log T}$, and denote $\zeta^{(T)} := \left\{\zeta^{(T)}(a) : a \in \mathcal{A}\right\}$. Its regret $R^\pi(T)$ satisfies,

$$\frac{R^\pi(T)}{\log T} = \sum_{a\in\mathcal{A}^{(s)}} \zeta^{(T)}(a)\ \Delta_a. \tag{14}$$

Note that $\bar{G}_i(T) = (\log T) H_i(\zeta^{(T)})$ or $\bar{G}_i(T)^{-1} = (\log T)^{-1} H_i^{-1}(\zeta^{(T)})$, where the function $H_i(\cdot)$ is as defined in (10). Since π is consistent, it then follows from (12) that $\forall a \in \mathcal{A}^{(s)}$ we have,

$$\begin{aligned}\limsup_{T\to\infty} \|u_a\|^2_{H^{-1}_{i_a}(\zeta^{(T)})} &= \limsup_{T\to\infty} \log T \|u_a\|^2_{\bar{G}_{i_a}(T)^{-1}} \\ &\le \frac{\Delta_a^2}{2}.\end{aligned} \tag{15}$$

Let $\zeta^{(\infty)} = \left\{\zeta^{(\infty)}(a) : a \in \mathcal{A}\right\}$ be a limit point of $\zeta^{(T)}$. It follows from (15) that the vector $\zeta^{(\infty)}$ is feasible for (8)-(10), and hence we have that $\sum_{a\in\mathcal{A}^{(s)}} \zeta^{(\infty)}(a)\ \Delta_a \ge c(\theta^\star, \mathcal{G}, \mathcal{A})$. The proof is then completed by observing that from (14), the regret $R^\pi(T)$ satisfies $\limsup_{T\to\infty} \frac{R^\pi(T)}{\log T} \ge \sum_{a\in\mathcal{A}^{(s)}} \zeta^{(\infty)}(a)\ \Delta_a$. ■

Remark: Note that the optimization problem (8)-(10) is convex, and hence can be solved efficiently. To see this, we note that the objective function is linear in the decision variables $\{\zeta(a)\}$. The functions $\|u_a\|^2_{H^{-1}_{i_a}(\alpha)}$ associated with the constraints (9) are also convex, as is shown in Appendix G of [4]. The number of decision variables is equal to the total number of actions, and hence is equal to the product of the number of arms $|\mathcal{U}|$, and the number of nodes $|\mathcal{V}|$. Since the number of decision variables increases linearly with the number of users $|\mathcal{V}|$, the proposed approach is scalable and can be used for graphs of "large" size.

IV. Stopping Time Based Algorithm

We now propose an algorithm for linear bandits with side-observations. This algorithm is composed of two phases: (i) exploratory phase, followed by (ii) exploitation phase. The exploratory phase lasts until a stopping criterion is met. More details are as follows.

Exploratory Phase:

Only the actions in the barycentric spanner $\mathcal{BS}$ are played. Since $\mathcal{BS}$ is composed of d arms at each node i, we decompose this phase into "episodes" of duration d steps each, where the k-th episode consists of steps $\{kd+1, kd+2, \ldots, (k+1)d\}, k = 0, 1, \ldots$. Each action from $\mathcal{BS}$ is played exactly once during each episode. The algorithm maintains the empirical estimates $\{\hat{\theta}_i(t) : i \in \mathcal{V}\}$ of the unknown coefficients $\theta_i^\star$, which are obtained as follows,

$$\hat{\theta}_i(t) := G_i(t)^{-1} \left[\sum_{s=1}^{t} \sum_{j\in\mathcal{N}_i} y_{(j,i)}(s) U_j(s)\right], i \in \mathcal{V}, \tag{16}$$

where $G_i(t)$ is as in (6). Additionally, it also maintains confidence ball $\mathcal{B}_a(t)$ around the estimate of mean reward of each action a as follows,

$$\mathcal{B}_a(t) := \left\{\mu \in \mathbb{R} : |\mu - u_a^\intercal \hat{\theta}_{i_a}(t)| \le \alpha(t)\right\},\ a \in \mathcal{A}, \tag{17}$$

where

$$\alpha(t) := \sqrt{\frac{2\log\left(T\sum_{i\in\mathcal{V}} |\mathcal{A}_i|/\delta\right)}{t}}\ d. \tag{18}$$

It orders the balls $\{\mathcal{B}_a(t)\}_{a\in\mathcal{A}_i}$ at each node i in decreasing order of the corresponding values of the estimates of the mean rewards $\left\{u_a^\intercal \hat{\theta}_i(t) : a \in \mathcal{A}_i\right\}$. Let $\mathcal{B}^{(o)}_{i,m}(t)$ be the m-th such ball[5] at node i during round t. Define τ_i to be the following stopping time,

$$\begin{aligned}\tau_i := \inf \Big\{ & t : t = kd \text{ where } k \in \mathbb{N}, \\ & \mathcal{B}^{(o)}_{i,1}(t) \cap \mathcal{B}^{(o)}_{i,m}(t) = \emptyset, \quad \forall m = 2, 3, \ldots, |\mathcal{A}_i| \Big\},\end{aligned} \tag{19}$$

and,

$$\tau := \max_{i\in\mathcal{V}} \tau_i. \tag{20}$$

Exploratory phase ends at round τ.

Exploitation Phase:

Let $\hat{u}_i^\star(t)$ be the estimate of the optimal arm for node $i \in \mathcal{V}$ when the coefficient of i is equal to $\hat{\theta}_i(t)$, i.e., $\hat{u}_i^\star(t) \in \arg\max_{u\in\mathcal{U}} \left\{u^\intercal \hat{\theta}_i(t)\right\}$. Also let $\hat{a}_i^\star(t) := (i, \hat{u}_i^\star(t))$. During rounds $t > \tau$, algorithm plays only the actions $\{\hat{a}_i^\star(\tau), i \in \mathcal{V}\}$ at their corresponding nodes. Thus, it uses $\left\{\hat{\theta}_i(\tau) : i \in \mathcal{V}\right\}$ as a proxy for $\theta^\star$, and plays the resulting greedy decisions. Algorithm 1 summarizes this.

We will now derive an upper-bound on its regret. We begin by deriving bounds on the error associated with the estimates $\hat{\theta}_i(t)$. Upon substituting the expressions for rewards $r_i(s)$ and side-observations $y_{(j,i)}(s)$ from (1) and (2), we obtain the following,

$$\begin{aligned}e_i(t) :&= \hat{\theta}_i(t) - \theta_i^\star \\ &= G_i(t)^{-1} \sum_{s=1}^{t} \sum_{j\in\mathcal{N}_i} \eta_{(j,i)}(s) U_j(s).\end{aligned} \tag{21}$$

For $x \in \mathbb{R}^d$, consider:

$$x^\intercal e_i(t) = \sum_{s=1}^{t} \sum_{j\in\mathcal{N}_i} \eta_{(j,i)}(s) x^\intercal G_i(t)^{-1} U_j(s). \tag{22}$$

Define the following "error event,"

$$\begin{aligned}\mathcal{E}_i(x, \alpha, t) :&= \{\omega : |x^\intercal e_i(t)| > \alpha\}, \\ &\text{where } \alpha > 0, i \in \mathcal{V},\ t \in [1, T].\end{aligned} \tag{23}$$

[5]Superscript denotes ordered balls.

Algorithm 1 Stopping Time Based Algorithm

Input: Arms $\mathcal{A}$, Graph $\mathcal{G}$, Confidence parameter δ, Time horizon T

Initialize: Set $t := 1$, and estimates $\hat{\theta}(t) = (1, 1, \ldots, 1)$ for all $i \in \mathcal{V}$

// Exploratory Phase

while $\exists i \in \mathcal{V}$ such that $\mathcal{B}_{i,1}^{(o)}(t) \cap \mathcal{B}_{i,m}^{(o)}(t) \neq \emptyset$ for some m **do**

Play each arm $a \in \mathcal{BS}$ once

Update the estimates $\hat{\theta}_i(t)$ using (16)

Update the confidence balls $\mathcal{B}_a(t)$ using (17)

end while

Exploration phase ends at τ

Obtain estimates $\hat{\theta}_i(\tau)$ of the coefficients, and the optimal arms $\hat{a}_i^\star(\tau), i \in \mathcal{V}$

// Exploitation Phase

for $t = \tau + 1, \tau + 2, \ldots, T$ **do**

Play $\{\hat{a}_i^\star(\tau)\}_{i \in \mathcal{V}}$ on corresponding nodes

end for

Define

$$\mathcal{E} := \cup_{k \in \lfloor 1, T/d \rfloor, i \in \mathcal{V}, a_i \in \mathcal{A}_i} \mathcal{E}_i(a_i, \alpha(kd), kd), \tag{24}$$

where $\alpha(t)$ is as in (18). The following result derives an upper bound on the regret of Algorithm 1. Auxiliary results, used while proving it, are deferred to the Appendix.

Theorem 2: The regret $R(T)$ of Algorithm 1 is upper-bounded as

$$R(T) \leq \left(\sum_{i \in \mathcal{V}} \Delta_{\max,i} \right) \frac{2 \log \left(T \sum_{i \in \mathcal{V}} |\mathcal{A}_i| / \delta \right) d}{\left(\Delta_{\min}/2 \right)^2} + \delta T \left(\sum_{i \in \mathcal{V}} \Delta_{\max,i} \right).$$

With $\delta = 1/T$, we obtain the following upper-bound on regret,

$$R(T) \leq \left(\sum_{i \in \mathcal{V}} \Delta_{\max,i} \right) \frac{2 \log \left(T^2 \sum_{i \in \mathcal{V}} |\mathcal{A}_i| \right) d}{\left(\Delta_{\min}/2 \right)^2} + \left(\sum_{i \in \mathcal{V}} \Delta_{\max,i} \right).$$

Proof: We begin by showing that on $\mathcal{E}^c$, the regret of Algorithm 1 is 0 during rounds t greater than time τ. Note that $\hat{a}_i^\star(\tau_i)$ is the action that corresponds to $\mathcal{B}_{i,1}^{(o)}(\tau_i)$. On $\mathcal{E}^c$, we have $\mu_{a_i^\star} \in \mathcal{B}_{a_i^\star}(\tau_i)$, and also $\mu_a \in \mathcal{B}_a(\tau_i)$ for any sub-optimal $a \in \mathcal{A}_i^{(s)}$. This means that the ball $\mathcal{B}_{i,1}^{(o)}(\tau_i)$ is equal to the ball $\mathcal{B}_{a_i^\star}(\tau_i)$, since if this was not the case, then we would have a contradiction that $\mu_a > \mu_{a_i^\star}$ for some sub-optimal a. Hence we conclude that $\hat{a}_i^\star(\tau_i) = a_i^\star$ on $\mathcal{E}^c$. Thus, we have,

$$\mathbb{1}\left(\mathcal{E}^c\right) \sum_{t=1}^{T} \sum_{i \in \mathcal{V}} \Delta_{(i,U_i(t))} \leq \tau \left(\sum_{i \in \mathcal{V}} \Delta_{\max,i} \right). \tag{25}$$

We now derive an upper-bound on τ. To do so, we will bound τ_i. Note that on $\mathcal{E}^c$, the mean rewards of actions lie within their corresponding confidence balls. Hence in order for the ball $\mathcal{B}_{a_i^\star}$ and the ball $\mathcal{B}_a$, that corresponds to a sub-optimal $a \in \mathcal{A}_i^{(s)}$, to intersect during round t, we must necessarily have the following,

$$\mu_{a_i^\star} - \alpha(t) \leq \mu_a + \alpha(t), a \in \mathcal{A}_i^{(s)},$$

which gives $\alpha(t) \geq \frac{\Delta_a}{2},\ a \in \mathcal{A}_i^{(s)}$. Upon substituting for $\alpha(t)$ from (61) into the above inequality, we obtain,

$$\sqrt{\frac{2 \log \left(T \sum_{i \in \mathcal{V}} |\mathcal{A}_i| / \delta \right)}{t} d} \geq \frac{\Delta_a}{2},\ a \in \mathcal{A}_i^{(s)},$$

or

$$t \leq \frac{2 \log \left(T \sum_{i \in \mathcal{V}} |\mathcal{A}_i| / \delta \right) d}{\left(\Delta_a/2 \right)^2}, a \in \mathcal{A}_i^{(s)}.$$

This shows that on $\mathcal{E}^c$, $\mathcal{B}_{a_i^\star}(t)$ cannot intersect with $\mathcal{B}_a, a \in \mathcal{A}_i^{(s)}$ during rounds $t > \frac{2 \log \left(T \sum_{i \in \mathcal{V}} |\mathcal{A}_i| / \delta \right) d}{\left(\Delta_{\min,i}/2 \right)^2}, a \in \mathcal{A}_i^{(s)}$, and hence $\tau_i \leq \frac{2 \log \left(T \sum_{i \in \mathcal{V}} |\mathcal{A}_i| / \delta \right) d}{\left(\Delta_{\min,i}/2 \right)^2}$. Since $\tau = \max_i \tau_i$, we get $\tau \leq \frac{2 \log \left(T \sum_{i \in \mathcal{V}} |\mathcal{A}_i| / \delta \right) d}{\left(\Delta_{\min}/2 \right)^2}$. Upon substituting the upper-bound on τ into (25), we obtain the following,

$$\mathbb{E}\left(\mathbb{1}\left(\mathcal{E}^c\right) \sum_{t=1}^{T} \sum_{i \in \mathcal{V}} \Delta_{(i,U_i(t))} \right) \leq \frac{2 \log \left(T \sum_{i \in \mathcal{V}} |\mathcal{A}_i| / \delta \right) d}{\left(\Delta_{\min}/2 \right)^2} \left(\sum_{i \in \mathcal{V}} \Delta_{\max,i} \right). \tag{26}$$

Moreover, since the cumulative regret on any sample path is trivially upper-bounded by $T \left(\sum_{i \in \mathcal{V}} \Delta_{\max,i} \right)$, we have that,

$$\mathbb{E}\left(\mathbb{1}\left(\mathcal{E}\right) \sum_{t=1}^{T} \sum_{i \in \mathcal{V}} \Delta_{(i,U_i(t))} \right) \leq T \left(\sum_{i \in \mathcal{V}} \Delta_{\max,i} \right) \mathbb{P}(\mathcal{E}) \leq \delta T \left(\sum_{i \in \mathcal{V}} \Delta_{\max,i} \right), \tag{27}$$

where, the last inequality follows from Lemma 12. The proof then follows by combining the inequalities (26) and (27). ■

Algorithm 1 takes an "explore-then-commit approach" [13] using a naive exploration algorithm, and consequently it seems suboptimal. However, in the next section, we design a more sophisticated algorithm, in which Algorithm 1 is used as a "fallback algorithm" in the "bad event" when the estimates of sub-optimality gaps that have been obtained at the end of the "warm-up phase" turn out to be bad. Unless an algorithm such as Algorithm 1 is deployed in this bad event, the expected regret would be quite large since the regret on this bad event could be as large as the time horizon T. Note that the regret upper-bound in Theorem 1 scales linearly with the dimension d of the feature space, while the problem-dependent regret bound of other existing algorithms, such as that in [49] scales as d^2, d^3. However, the bound of [49] does not depend upon the number of arms, while our bound has a logarithmic growth with the number of arms.

V. Asymptotically Optimal Algorithm

The regret of Algorithm 1 scales as $O(\log T)$; if the parameters $\Delta_{\max}, \Delta_{\min}$ and the dimension d are kept constant, then the regret scales linearly with the number of nodes N. We now propose an algorithm that uses solution of optimization problem (8)-(10) in order to make decisions. We show that its regret matches the lower bound of Section III, and the prefactor can be upper-bounded by domination number $|\chi(\mathcal{G})|$ of the graph $\mathcal{G}$. Thus, for graphs $\mathcal{G}$ that satisfy $|\chi(\mathcal{G})| \ll |\mathcal{V}|$, this algorithm can be much more efficient than Algorithm 1.

Algorithm 2 Asymptotically Optimal Algorithm

Input: Arms $\mathcal{A}$, Graph $\mathcal{G}$, Confidence parameter δ, Time horizon T

// Warm-up Phase

Play each arm in spanning set $\mathcal{S}$ for $\log^{1/2} T$ times

// Success Phase

$\epsilon_T(t) \leftarrow \max_{a\in\mathcal{A}} \|a\|_{G_{i_a}^{-1}(d\log^{1/2}T)}\ \ g^{1/2}(T)$

$\hat{\Delta} \leftarrow \hat{\Delta}(d\log^{1/2}T), \hat{\mu} \leftarrow \hat{\mu}(d\log^{1/2}T)$

Solve $OPT(\hat{\Delta})$ (31)-(33) to obtain $\beta^\star(\hat{\Delta})$

while $t \le T$ and $|\hat{\mu}_a - \hat{\mu}_a(t-1)| \le 2\epsilon_T$ for all $a \in \mathcal{A}$ **do**

For each $i \in \mathcal{V}$ play actions in a round robin fashion with $N_a(t) \le \beta_a^\star(\hat{\Delta})$

end while

// Recovery Phase

Discard all data and play Algorithm 1 until $t = T$

We begin by introducing a few notations. Define,

$$f(t) := 2\log(t) + cd\log(d\log t) + 2, \tag{28}$$

$$g(t) := 2\log(\log t) + 2\frac{\log(\log t)}{\log t} + cd\log(d\log t), \tag{29}$$

where $c > 0$ is a constant. Let

$$\hat{\Delta}_a(t) := \max_{u\in\mathcal{U}} (u - u_a)^\intercal \hat{\theta}_{i_a}(t), \tag{30}$$

denote estimate of sub-optimality gap of action a during round t, and $\hat{\Delta}(t) := \left\{\hat{\Delta}_a(t) : a \in \mathcal{A}\right\}$. The proposed algorithm is composed of the following three phases.

Warm-up Phase:

Algorithm plays actions in the barycentric spanner $\mathcal{BS}$ for $d\log^{1/2} T$ rounds.

Success Phase:

Denote by $\hat{\Delta} = \{\hat{\Delta}_a : a \in \mathcal{A}\}, \hat{\mu} = \{\hat{\mu}_a : a \in \mathcal{A}\}$ the estimates of sub-optimality gaps and mean values of rewards that are obtained by using the information gained during the warm-up phase. Consider the following optimization problem obtained from OPT (8)-(10) by replacing the gaps Δ_a by their estimates $\hat{\Delta} = \left\{\hat{\Delta}_a : a \in \mathcal{A}\right\}$:

$$OPT(\hat{\Delta}) : \min_{\{\beta_a\}_{a\in\mathcal{A}}} \sum_{a\in\mathcal{A}} \beta_a \hat{\Delta}_a \tag{31}$$

$$\text{s.t. } f(T)\|u_a\|^2_{H_{i_a}^{-1}(\beta)} \le \frac{\hat{\Delta}_a^2}{2}, \quad \forall a \in \mathcal{A}, \tag{32}$$

$$\text{where } H_i(\beta) := \sum_{j\in\mathcal{N}_i} \sum_{\{a:i_a=j\}} \beta_a\ u_a\ u_a^\intercal,\ i \in \mathcal{V}. \tag{33}$$

Let $\beta^\star(\hat{\Delta}) = \left\{\beta_a^\star(\hat{\Delta}) : a \in \mathcal{A}\right\}$ be a solution of $OPT(\hat{\Delta})$. The algorithm uses estimates $\hat{\Delta}$ to solve (31)-(33), and obtains $\beta^\star(\hat{\Delta})$. It then plays each action a in a round-robin fashion until it has been played for $\beta_a^\star(\hat{\Delta})$ rounds. Meanwhile, it also continually keeps track of the quality of estimates $\hat{\mu}_a$ of rewards obtained at the end of warm-up phase as follows. Define

$$\epsilon_T(t) := \max_{a\in\mathcal{A}} \|a\|_{G_{i_a}^{-1}(t)}\ \sqrt{g(T)}, \tag{34}$$

where $g(\cdot)$ is as in (29). If during any round t, it observes that $|\hat{\mu}_a(t) - \hat{\mu}_a| > 2\epsilon_T(d\log^{1/2} T)$ for some action $a \in \mathcal{A}$, then it declares that the estimates $\hat{\mu}$ are bad, and in this event algorithm enters recovery phase.

Recovery Phase:

Algorithm discards all operational history and collected data, and starts playing Algorithm 1.

Algorithm 2 summarizes this. We next show that it is asymptotically optimal, i.e., as $T \to \infty$, its regret matches the lower bound derived in Theorem 1. Auxiliary results, required while proving it, are deferred to the Appendix. Define the following two events,

$$F := \bigcup_{a\in\mathcal{A}, t\in[1,T]} \left\{\omega : |\mu_a - \hat{\mu}_a(t)| \ge \|u_a\|_{G_{i_a}^{-1}(t)} g^{1/2}(T)\right\},$$

$$F' := \bigcup_{a\in\mathcal{A}, t\in[1,T]} \left\{\omega : |\mu_a - \hat{\mu}_a(t)| \ge \|u_a\|_{G_{i_a}^{-1}(t)} f^{1/2}(T)\right\},$$

where, the functions $f(\cdot)$ and $g(\cdot)$ are as in (28), (29). The following result is used while proving Theorem 3.

Lemma 2: Algorithm 2 never enters recovery phase on F^c.

Proof: On F^c we have the following,

$$|\mu_a - \hat{\mu}_a(t)| \le \|u_a\|_{G_{i_a}^{-1}(t)} g^{1/2}(T) \le \epsilon_T(t),\ \forall a \in \mathcal{A}.$$

Thus, for times $s, t \ge d\log^{1/2} T$, we have

$$|\hat{\mu}_a(s) - \hat{\mu}_a(t)| \le 2\epsilon_T(\min\{s,t\}) \le 2\epsilon_T(d\log^{1/2} T).$$

Since recovery phase occurs only when $|\hat{\mu}_a(t) - \hat{\mu}_a| > 2\epsilon_T(d\log^{1/2} T)$, where $\hat{\mu}_a$ is the estimate of μ_a at time $d\log^{1/2} T$, this shows that the algorithm does not enter the recovery phase on F^c. ∎

Theorem 3: The regret $R(T)$ of Algorithm 2 satisfies

$$\limsup_{T\to\infty} \frac{R(T)}{\log T} \le c(\mathcal{A}, \theta^\star, \mathcal{G}),$$

where $c(\mathcal{A}, \theta^\star, \mathcal{G})$ is the optimal value of optimization problem (8)-(10). It then follows from lower bound derived in Theorem 1 that Algorithm 2 is asymptotically optimal as $T \to \infty$.

Proof: Throughout this proof, we denote the regret of Algorithm 2 by $R(T)$. Let $\mathcal{T}_{warm}, \mathcal{T}_{succ}, \mathcal{T}_{rec}$ denote the rounds spent in the warm-up, success, and recovery phases,

respectively. The normalized cumulative regret $R(T)/\log T$ can be decomposed as follows,

$$\frac{R(T)}{\log T} = \frac{1}{\log T}\,\mathbb{E}\left(\sum_{t\in\mathcal{T}_{warm}}\sum_{i\in\mathcal{V}}\Delta_{(i,U_i(t))}\right) + \frac{1}{\log T}\,\mathbb{E}\left(\sum_{t\in\mathcal{T}_{rec}}\sum_{i\in\mathcal{V}}\Delta_{(i,U_i(t))} + \sum_{t\in\mathcal{T}_{succ}}\sum_{i\in\mathcal{V}}\Delta_{(i,U_i(t))}\right). \tag{35}$$

Since the warm-up phase lasts for $O(\log^{1/2} T)$ rounds, contribution of the first term is asymptotically 0 as $T\to\infty$. Hence, we focus on regret in the success and recovery phases.

Next, we analyze the regret during $\mathcal{T}_{rec}$. Using techniques similar to the proof of Theorem 8 of [4], we can show the following,

$$\mathbb{P}(F) \le \frac{1}{\log(T)}, \quad \mathbb{P}(F') \le \frac{1}{T}. \tag{36}$$

Upon combining this with Lemma 2, we conclude that $\mathbb{P}(\mathcal{T}_{rec}\neq\emptyset) \le 1/\log T$. From Theorem 2 we have that if the algorithm does enter the recovery phase, then its regret is upper-bounded as $O(\log T)$. Upon combining these two bounds, we have that the expected value of the second term in the summation in (35) is upper-bounded by a constant that does not depend upon T. Thus, the contribution of this summation term, when divided by $\log T$, is asymptotically 0.

The discussion so far shows that the first two summation terms in the r.h.s. of (35) asymptotically vanish. We finally analyze the regret in the success phase. We analyze this regret separately on the following sets: (i) F', (ii) $F\cap(F')^c$, (iii) F^c. Since from (36) we have that $\mathbb{P}(F')\le 1/T$, and moreover the regret on any sample-path can be trivially upper-bounded as $O(T)$, we conclude that the regret on F' is upper-bounded by a constant that does not depend upon T. This term, when divided by $\log T$, asymptotically vanishes. Using techniques similar to the proof of Lemma 13 of [4], it follows that this regret on the set $F\cap(F')^c$ asymptotically vanishes. Thus, it only remains to analyze the regret during the success phase on the set F^c, which we now show is asymptotically upper-bounded by $c(\theta^\star,\mathcal{G},\mathcal{A})$. In what follows, we use $\hat{\Delta}$ and ϵ_T in lieu of $\hat{\Delta}(d\log^{1/2}T)$ and $\epsilon_T(d\log^{1/2}(T))$ respectively. Recall that $\beta^\star(\Delta)$ is the number of plays calculated by solving the optimization problem (31)-(33). $\beta^\star(\Delta)$ satisfies the following,

$$\limsup_{T\to\infty}\frac{\sum_{a\in\mathcal{A}^{(s)}}\beta_a^\star(\Delta)\Delta_a}{\log T} = c(\theta^\star,\mathcal{G},\mathcal{A}). \tag{37}$$

The regret that occurred during the success phase satisfies,

$$\begin{aligned}\mathbb{1}(F^c)\sum_{t\in\mathcal{T}_{succ}}\sum_{i\in\mathcal{V}}\Delta_{(i,U_i(t))} &\le \sum_{a\in\mathcal{A}^{(s)}}\beta_a^\star(\hat{\Delta})\Delta_a\\ &= \sum_{a\in\mathcal{A}^{(s)}}\beta_a^\star(\hat{\Delta})\hat{\Delta}_a + \sum_{a\in\mathcal{A}^{(s)}}\beta_a^\star(\hat{\Delta})\left[\Delta_a-\hat{\Delta}_a\right]\\ &\le (1+\delta_T)\sum_{a\in\mathcal{A}^{(s)}}\beta_a^\star(\Delta)\hat{\Delta}_a + 2\epsilon_T\sum_{a\in\mathcal{A}^{(s)}}\beta_a^\star(\hat{\Delta})\\ &\le (1+\delta_T)\sum_{a\in\mathcal{A}^{(s)}}\beta_a^\star(\Delta)\Delta_a + 2\epsilon_T(d\log^{1/2}(T))\\ &\quad\times\left[(1+\delta_T)\sum_{a\in\mathcal{A}^{(s)}}\beta_a^\star(\Delta)+\sum_{a\in\mathcal{A}^{(s)}}\beta_a^\star(\hat{\Delta})\right],\end{aligned} \tag{38}$$

where the first inequality follows since under Algorithm 2, the number of plays of an arm a is atmost equal to $\beta_a^\star(\hat{\Delta})$, the second inequality follows from (62) and the fact that on F^c we have $|\mu_a-\hat{\mu}_a(t)|\le\epsilon_T$. We now use the results of Lemma 13 and Lemma 15 in the inequality (38), and also choose the operating horizon T to be sufficiently large so as to satisfy $2\epsilon_T\le\Delta_{\min}/2$, and obtain the following,

$$\mathbb{1}(F^c)\sum_{t\in\mathcal{T}_{succ}}\sum_{i\in\mathcal{V}}\Delta_{(i,U_i(t))} \le \left(1+\frac{16\epsilon_T}{\Delta_{\min}}\right)\sum_{a\in\mathcal{A}^{(s)}}\beta_a^\star(\Delta)\Delta_a + 2\epsilon_T\left[2+2d^3f(T)\frac{\Delta_{\max}}{\Delta_{\min}^3}\right]. \tag{39}$$

We have

$$\epsilon_T = O\left(\frac{\log^{1/2}(\log T)}{\log^{1/4}T}\right). \tag{40}$$

We now divide both sides of (39) by $\log T$, and substitute (40) in (39) in order to obtain the following,

$$\begin{aligned}&\limsup_{T\to\infty}\frac{1}{\log T}\mathbb{1}(F^c)\sum_{t\in\mathcal{T}_{succ}}\sum_{i\in\mathcal{V}}\Delta_{(i,U_i(t))}\\ &\le\limsup_{T\to\infty}\frac{1}{\log T}\sum_{a\in\mathcal{A}^{(s)}}\beta_a^\star(\Delta)\Delta_a \le c(\theta^\star,\mathcal{G},\mathcal{A}),\end{aligned} \tag{41}$$

where the last inequality follows from (37). It thus follows from Fatou's lemma [50] that the regret during the success phase on F^c is upper-bounded by $c(\theta^\star,\mathcal{G},\mathcal{A})$. The proof is then completed by substituting the bounds on different terms into the relation (35). ∎

Corollary 1: Optimal value of problem (8)-(10), $c(\mathcal{A},\theta^\star,\mathcal{G})$, is less than or equal to $\frac{\Delta_{\max}}{\Delta_{\min}}|\chi(\mathcal{G})|$. Thus, the regret $R(T)$ of Algorithm 2 satisfies

$$\limsup_{T\to\infty}\frac{R(T)}{\log T}\le\frac{\Delta_{\max}}{\Delta_{\min}}|\chi(\mathcal{G})|.$$

Proof: Consider the following optimization problem

$$OPT_1: \min_{\{w_a\}_{a\in\mathcal{A}}}\Delta_{\max}\sum_{a\in\mathcal{A}}w_a \tag{42}$$

$$\text{s.t. } n_i(w)\ge\frac{\sqrt{2}}{\Delta_{\min}},\quad \forall i\in\mathcal{V}, \tag{43}$$

$$\text{where } n_i(w) := \sum_{j\in\mathcal{N}_i}\sum_{\{a:i_a=j\}}w_a,\ i\in\mathcal{V}., \tag{44}$$

$$w_a=0 \text{ if } a\notin\mathcal{BS}, \text{ and } w_a=w_b,\ \forall a,b\in\mathcal{BS}\cap\mathcal{A}_{i_a}. \tag{45}$$

It is easily verified that any vector feasible for OPT_1 is also feasible for OPT. Moreover, its objective function is also greater than the objective of OPT. Thus, its optimal value, denoted $c(\mathcal{A},\theta^\star,\mathcal{G})_1$ is greater than $c(\mathcal{A},\theta^\star,\mathcal{G})$. Consider now a scaled version of OPT_1,

$$OPT_{1,s}: \min_{\{w_a\}_{a\in\mathcal{A}}}\sum_{a\in\mathcal{A}}w_a \tag{46}$$

$$\text{s.t. } n_i(w) \geq 1, \quad \forall i \in \mathcal{V}, \tag{47}$$

$$\text{where } n_i(w) := \sum_{j \in \mathcal{N}_i} \sum_{\{a: i_a = j\}} w_a, \ i \in \mathcal{V}, \tag{48}$$

$$w_a = 0 \text{ if } a \notin \mathcal{S}, \text{ and } w_a = w_b, \ \forall a, b \in \mathcal{S} \cap \mathcal{A}_{i_a}. \tag{49}$$

It is evident that if x is feasible for OPT_1, then $x\Delta_{\min}/\sqrt{2}$ is feasible for $OPT_{1,s}$, and if y is feasible for $OPT_{1,s}$, then $y\sqrt{2}/\Delta_{\min}$ is feasible for OPT_1. Thus, if $c(\mathcal{A}, \theta^\star, \mathcal{G})_{1,s}$ denotes optimal value of $OPT_{1,s}$, then we have $c(\mathcal{A}, \theta^\star, \mathcal{G})_1 = c(\mathcal{A}, \theta^\star, \mathcal{G})_{1,s} \left(\sqrt{2}/\Delta_{\min}\right) \Delta_{\max}$. The proof is completed by noting that the optimal value of $OPT_{1,s}$ is a lower bound on $|\chi(\mathcal{G})|$. ∎

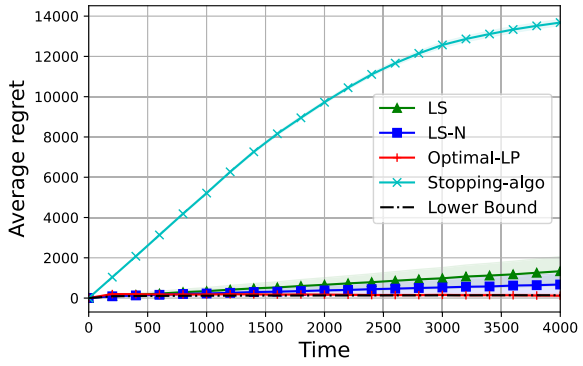
Remark: We note that the preferences of user i are reflected in his/her coefficient vector $\theta_i^\star$, while the side observations revealed from recommendations depend upon the graph $\mathcal{G}$. Since the proposed algorithm uses a solution of (31)-(33) in order to make decisions, it takes into account both these quantities while making recommendations. This means that even if the coefficient vectors $\theta_i^\star$ and $\theta_j^\star$ of neighbors i, j are close, the recommendations made to i and j could be different if their sets of neighbors are different. Also, note that in our model, the preferences of neighbors are allowed to be different, i.e., $\|\theta_i^\star - \theta_j^\star\|$ can be large for neighboring nodes i and j.

VI. Experiments

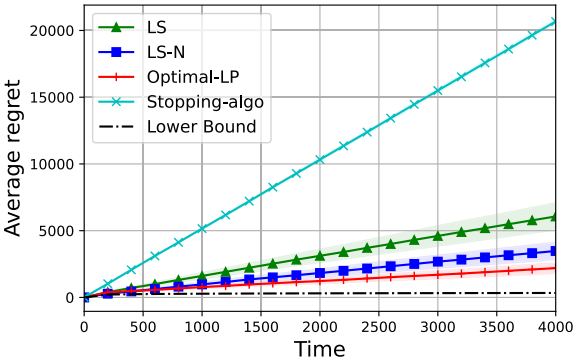
A. Synthetic Data Experiment

The vector $\theta^\star = (\theta_1^\star, \theta_2^\star, \ \ldots, \theta_N^\star)$ that contains the coefficients of the users, and the arms constituting the set $\mathcal{U}$, are generated randomly. More specifically, $\{\theta_i^\star\}_{i=1}^N$ and the arms are drawn from a uniform distribution with support in the set $[0, 1]^d$. In order to generate the graph $\mathcal{G}$, the edges connecting the nodes are drawn randomly before the experiment begins; thus, any two nodes $i, j \in \mathcal{V}$ are connected with a probability p. Note that the graph $\mathcal{G}$ is kept fixed throughout the experiment. The noise $\eta_i(t), \eta_{(i,j)}(t)$ associated with the rewards and the side-observations (1), (2) are assumed to be Gaussian with standard deviation σ. We compare the performance of Algorithm 2 with the algorithm of [4], which is denoted LS,[6] In order to make decisions regarding which actions should be played, LS algorithm solves N optimization problems, one for each node. The optimization problem for node i is similar to (31)-(33), but involves only those actions which correspond to playing an arm on node i. Similarly, the estimates $\hat{\theta}_i(t)$ are also calculated without taking the side observations into account. We also consider a naive adaptation of LS to the graphical setting and denote it by LS-N. LS-N differs from LS in that it uses side observations to enhance the estimation after warm-up phase, but in the success phase, it does not utilize the structure of $\mathcal{G}$ in order to cleverly choose the number of times that an arm should be played. Instead of solving (31)-(33), it solves a separate optimization problem for each $i \in \mathcal{V}$; see [4] for more details. The computational complexity of all the three algorithms is similar since they involve solving an optimization problem in which number of decision variables is equal to the number of actions. Our experimental results show the potential gains from using the side observations alone. However, by leveraging the graph structure, our optimal algorithm shows significant regret reduction and verifies our theoretical claims. We summarize the results of this evaluation in Fig. 1, where we plot the regret of the algorithms as a function of rounds. Algorithm 1 and Algorithm 2 are denoted Stopping-algo and Optimal-LP, respectively, in all the plots. In Fig. 1, Algorithm 2 is seen to outperform other algorithms. Next, in order to study the dependence of regret on the link-probability, in Fig. 2, we compare the cumulative regrets of different algorithms as this probability is varied.

[6]Abbreviation for Lattimore, Szepesvári.



(a) $d = 4, N = 10, K = 4, p = 0.4, \sigma = 1.0$



(b) $d = 6, N = 10, K = 10, p = 0.4, \sigma = 1.0$

Fig. 1. Comparison of regret of Algorithm 1, denoted Stopping-algo, and Algorithm 2, denoted Optimal-LP, with the algorithms LS and LS-N of [4], on two randomly generated bandit instances. The plots are obtained after averaging the results of 20 runs. N and K denote the number of nodes and set of arms at each node, respectively.

Since a higher value of link generation probability leads to more connections in the graph, and on average, this would mean that the graph would have a lower domination number, we expect the cumulative regret of the optimal algorithm to decrease with p. Note that the plots in Fig. 2 are obtained after averaging over 40 randomly generated graphs. Fig. 2 shows that the terminal cumulative regret of the optimal algorithm reduces as the link generation probability increases, this is in coherence with the $|\chi(\mathcal{G})|$-dependence of regret upper-bound that was derived in Corollary 1. Algorithm 2 is seen to consistently outperform others.

B. MovieLens Data Experiment

The MovieLens 20M dataset consists of reviews on more than $100,000$ movies by more than $10,000$ users. Each movie

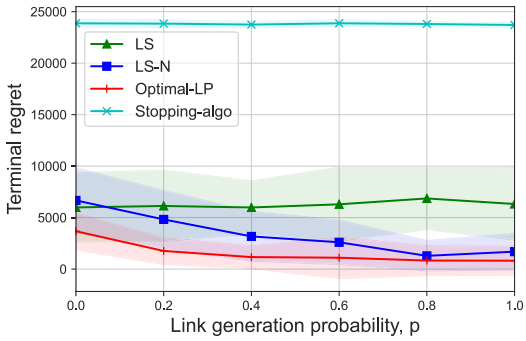
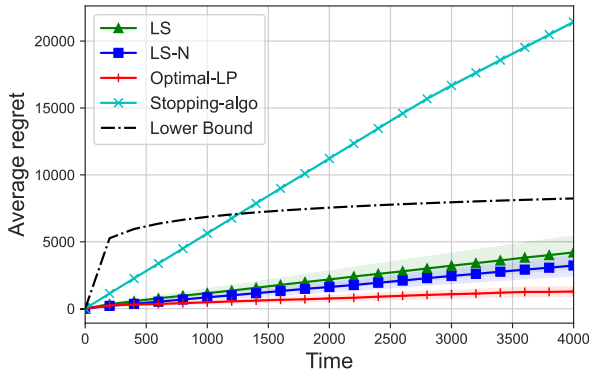
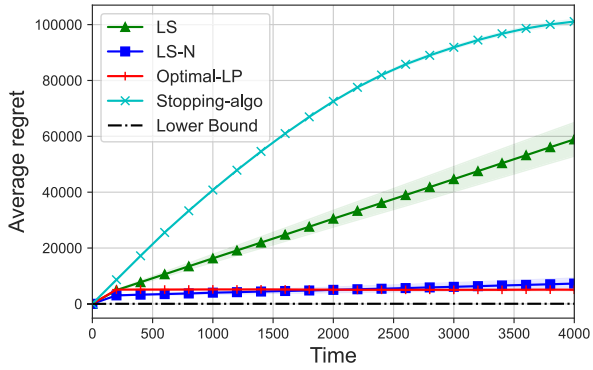


Fig. 2. Plot of cumulative regret at $T = 4000$ as p is varied. $\{\theta_i^\star\}_{i=1}^N$ and the arms are drawn from a uniform distribution with support in the set $[0,1]^d$. We use $d = 4, N = 10, K = 8,\ \sigma = 1.5$. At each value of p, the terminal regret for 40 different graphs are averaged.



(a) $N = 10, K = 4, p = 0.3, \sigma = 1.0$



(b) $N = 25, K = 30, p = 0.3, \sigma = 1.5$

Fig. 3. Comparison of regret of Algorithm 1 and Algorithm 2 with LS and LS-N [4] on two bandit instances generated using the MovieLens dataset. The plots are obtained after averaging the results of 20 runs. N and K denote the number of nodes (users) and set of arms (movies) at each node, respectively. A link between every pair of users is generated independently at random with probability p.

belongs to one or more genres and has user-given tags, which can be thought of as features. In order to reduce the dimension of the feature space, we performed *principle component analysis* [51] upon the feature matrix of the movies and chose the first 10 dominant eigenvectors. Thereafter, we obtained the coefficient vector $\theta_i^\star$ for each user i by performing a least squares fit. Fig. 3 compares the performance of algorithms for two problem instants, and Algorithm 2 is seen to yield the best performance.

Next, we empirically verify that the regret of Algorithm 2 scales as the domination number of the graph and not as the

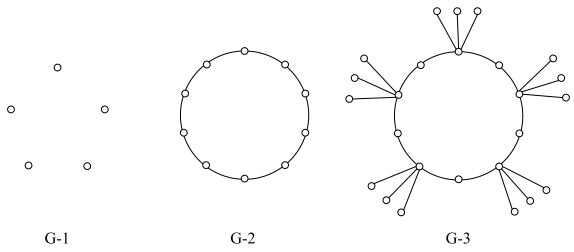


Fig. 4. Three graphs having different numbers of nodes but the same domination number of 5.

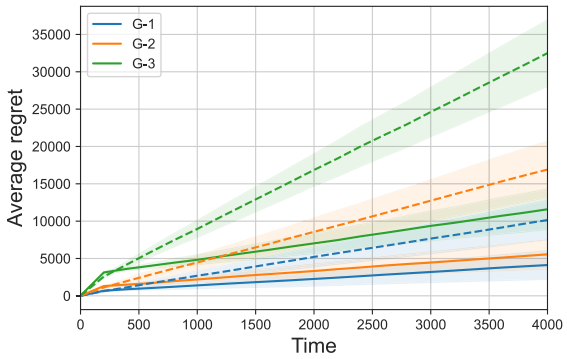


Fig. 5. Plot of regret for Algorithm 2 and LS-N for the three graphs G-1, G-2 and G-3 depicted in Fig. 4. All of them have the same domination number. Unbroken lines and dashed lines indicate Optimal-LP and LS-N, respectively.

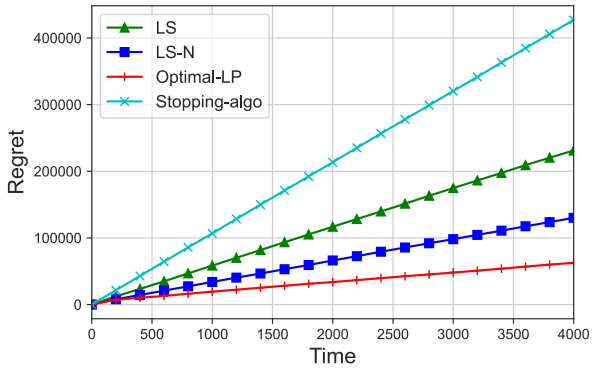
number of users. For this, consider the three graphs shown in Fig. 4 that have different numbers of users but the same domination number of 5. Fig. 5 shows the regret as a function of time when different algorithms are applied to these three graphs. Number of articles is kept fixed at 20, and the standard deviation of noise is 1.5. These plots suggest that the regret of Algorithm 2 does not scale with the number of users, since the domination number of the graphs is fixed.

Next, we compare the performance of algorithms on random graphs of size 200, 400, and 800 nodes. We reduce the dimension of the feature space to 4 and set the number of articles equal to 10. The standard deviation of the noise is set equal to 1. Figure 6 plots the regret for various algorithms.

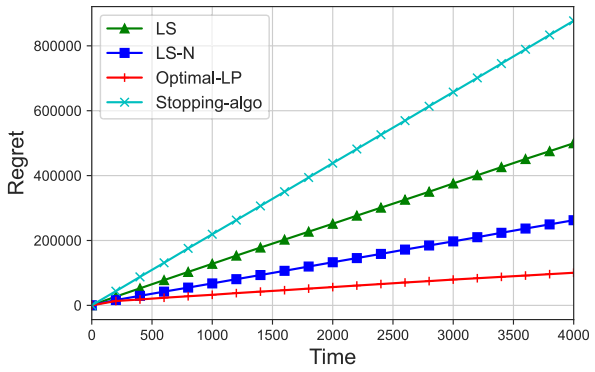
Remark: The following heuristic could be used in order to reduce computation time when running the algorithm on large graphs. Partition the graph $\mathcal{G}$ into M smaller sub-graphs and solve the resulting M subproblems separately using different processors. This procedure yields a linear speedup so that the computation time reduces by a factor of M. The optimization problem for a subgraph is obtained by restricting the nodes and edges in problem (31)-(33) to only those belonging to this sub-graph. For each node i, choose its actions by using the solution of the optimization problem corresponding to its subgraph. Since this approach essentially ignores the possibility of obtaining any side-information using those edges that connect different subgraphs, it is expected that it does not affect the regret much when the number of such edges is small as compared with the total number of edges.

VII. Discussion

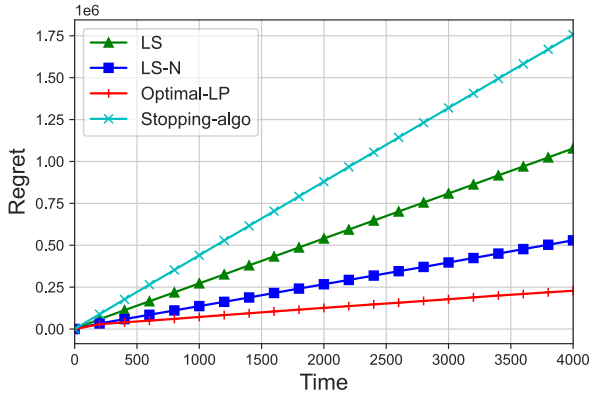
In this paper, we introduce a framework to make optimal decisions for linear bandits in a network setting by incorporat-



(a) $N = 200, K = 10$, Average degree= 3, $\sigma = 1.0$



(b) $N = 400, K = 10$, Average degree= 3, $\sigma = 1.0$



(b) $N = 800, K = 10$, Average degree= 3, $\sigma = 1.0$

Fig. 6. Plot comparing the regret of Algorithm 1 and Algorithm 2 with LS and LS-N [4] on two problem instances generated using the MovieLens dataset. N and K denote the number of nodes (users) and set of arms (movies) at each node, respectively. In each graph, on average each user is connected with 3 other users.

ing side-observations. We derive an instance-dependent lower bound on the regret of any learning algorithm and also an optimal algorithm whose regret matches these lower bounds asymptotically as $T \to \infty$. This work can be extended in several interesting directions. We plan to extend the results to the case when the set of arms to choose from is non-stationary. The current model assumes the neighboring nodes would always provide feedback for "free." In real-world applications, it is more reasonable to consider a subset of the neighbors that would share their preferences, probably with certain additional costs. Another interesting extension is to allow the algorithm to recommend limited items to only a subset of the users and not choose arms for all the users, as is the case currently. Note that currently, we assume that users provide feedback irrespective of whether they like the recommendation or not. Yet another possibility is to collect feedback from a neighbor only when the user reacts positively to the item presented to him/her.

APPENDIX

A. Auxiliary Results Used in Proof of Theorem 1

The following result is Lemma 5 of [4].

Lemma 3: Let $\mathbb{P}$ and $\mathbb{P}'$ be measures on the same measurable space $(\Omega, \mathcal{F})$. Then, for any event $A \in \mathcal{F}$, we have,

$$\mathbb{P}(A) + \mathbb{P}'(A^c) \geq \frac{1}{2}\exp(-KL(\mathbb{P}, \mathbb{P}')),$$

where $KL(\mathbb{P}, \mathbb{P}')$ denotes the relative entropy between $\mathbb{P}$ and $\mathbb{P}'$, which is defined as $+\infty$ if $\mathbb{P}$ is not absolutely continuous with respect to $\mathbb{P}'$, and is equal to $\int_\Omega d\mathbb{P}(\omega) \log \frac{d\mathbb{P}}{d\mathbb{P}'}(\omega)$ otherwise.

The following result shows that the relative entropy between the probability measures induced by a learning algorithm on sequences of outcomes for two different multi-armed bandit problem instances can be decomposed in terms of the expected number of times each arm is chosen and the relative entropies of the distributions of the arms. We omit its proof since it is a minor modification of the proof of Lemma 6 of [4].

Lemma 4: (Information Processing Lemma) Consider a learning algorithm π applied to two different problem instances, in which the users' coefficients are equal to $\{\theta_i^\star\}_{i\in\mathcal{V}}$, and $\{\theta_i'\}_{i\in\mathcal{V}}$, while the graph and the set of arms are the same in both the instances and given by $\mathcal{G}$ and $\mathcal{A}$. Let $\mathbb{P}, \mathbb{P}'$ denote the probability measures induced by π on the sequence of rewards $\{r_i(s) : i \in \mathcal{V}\}_{s=1}^t$, side-observations $\{y_{(i,j)}(s) : (i,j) \in \mathcal{E}\}_{s=1}^t$ and arms $\{U_i(s) : i \in \mathcal{V}\}_{s=1}^t$. Furthermore, assume that $\theta^\star$ and θ' differ only on the value at a single node i, i.e., $\theta_j^\star = \theta_j'$, $\forall j \neq i$, and $\theta_i^\star \neq \theta_i'$. Then we have the following,

$$KL(\mathbb{P}, \mathbb{P}') = \frac{1}{2}(\theta_i^\star - \theta_i')^\intercal \bar{G}_i(T)\ (\theta_i^\star - \theta_i'),$$

where $\bar{G}_i(T)$ is as in (6), and the expectation is taken when $\theta^\star$ is the true parameter.

Constructing Modified Coefficient Vector θ'

Recall that when the coefficient vector is equal to $\theta^\star$, $a_i^\star$ is the unique optimal action for node i. We will now construct a coefficient vector θ' so that the resulting optimal action for node i will be $b^\star$, where $b^\star \neq a_i^\star$. Since we do not modify the coefficients at other nodes, the optimal arms for other nodes $v \in \mathcal{V}\backslash\{i\}$ remain unchanged. Let $H > 0$ be a positive-definite matrix that will be specified soon. We let

$$\theta_v' = \begin{cases} \theta_v^\star, & \text{if } v \in \mathcal{V} \setminus \{i\}, \\ \theta_i^\star + \dfrac{1}{\|u_{b^\star} - u_i^\star\|_H^2} H(u_{b^\star} - u_i^\star)(\Delta_{b^\star} + \epsilon) & \text{if } v = i, \end{cases} \tag{50}$$

where $u_i^\star$ is the optimal arm for node i under $\theta^\star$. Note that under θ_i', the mean reward of $b^\star$ is more than that of $a_i^\star$ since,

$$\begin{aligned}&(\theta_i')^\intercal\left(u_{b^\star}-u_i^\star\right)=\\&\left(\theta_i^\star+\frac{1}{\|u_{b^\star}-u_i^\star\|_H^2}H(u_{b^\star}-u_i^\star)(\Delta_{b^\star}+\epsilon)\right)^\intercal(u_{b^\star}-u_i^\star)\\&=-\Delta_{b^\star}+\Delta_{b^\star}+\epsilon\\&=\epsilon.\end{aligned}\tag{51}$$

Let $R^\pi_{(\theta^\star,\mathcal{G},\mathcal{A})}(T), R^\pi_{(\theta',\mathcal{G},\mathcal{A})}(T)$ denote the regret incurred by the learning algorithm π, when the coefficients are equal to $\theta^\star$ and θ' respectively. We have the following lower-bound on $R^\pi_{(\theta^\star,\mathcal{G},\mathcal{A})}(T)+R^\pi_{(\theta',\mathcal{G},\mathcal{A})}(T)$. Recall that $N_a(T)$ is the number of plays of action a until round T.

Lemma 5: Let θ' be the coefficient vector constructed as in (50), and $\mathbb{P},\mathbb{P}'$ denote the probability measures induced by a learning algorithm π on the sequence of rewards, side-observations and actions when users' coefficients in are equal to $\{\theta_i^\star\}_{i\in\mathcal{V}}$, and $\{\theta_i'\}_{i\in\mathcal{V}}$ respectively. Furthermore, let $\epsilon<\Delta_{\min,i}$, where $\Delta_{\min,i}$ is as in (3). We then have that,

$$\begin{aligned}&R^\pi_{(\theta^\star,\mathcal{G},\mathcal{A})}(T)+R^\pi_{(\theta',\mathcal{G},\mathcal{A})}(T)\geq\\&\frac{\epsilon T}{2}\left[\mathbb{P}\left(N_{a_i^\star}(T)\leq T/2\right)+\mathbb{P}'\left(N_{a_i^\star}(T)>T/2\right)\right].\end{aligned}$$

Proof: Clearly, $R^\pi_{(\theta^\star,\mathcal{G},\mathcal{A})}(T)\geq\frac{T}{2}\Delta_{\min,i}\mathbb{P}\left(N_{a_i^\star}(T)\leq T/2\right)$. Similarly, it follows from (51) that $R^\pi_{(\theta',\mathcal{G},\mathcal{A})}(T)\geq\frac{T}{2}\epsilon\,\mathbb{P}'\left(N_{a_i^\star}(T)\geq T/2\right)$. The proof then follows by adding the above two inequalities and utilizing $\epsilon<\Delta_{\min,i}$. ∎

Lemma 6: Let θ' be the coefficient vector constructed as in (50), and $b^\star$ be the optimal action for node i under θ'. Define

$$\delta a^\star:=u_{b^\star}-u_i^\star,\tag{52}$$

where $u_i^\star$ is the optimal arm for node i when its coefficient is equal to $\theta_i^\star$. For $H>0$ define

$$\begin{aligned}\rho_i(T;H):=&\|\delta a^\star\|^2_{H^\intercal\bar{G}_i(T)H}\times\\&\|\delta a^\star\|^2_{\bar{G}_i^{-1}(T)}\left(\|\delta a^\star\|_H^4\right)^{-1}.\end{aligned}\tag{53}$$

We then have that,

$$\begin{aligned}\frac{(\Delta_{b^\star}+\epsilon)^2}{2}\frac{\rho_i(T;H)}{\log T\;\|\delta a^\star\|^2_{\bar{G}_i^{-1}(T)}}\geq 1+\frac{\log\epsilon-\log 2}{\log T}\\-\frac{\log\left(R^\pi_{(\theta^\star,\mathcal{G},\mathcal{A})}(T)+R^\pi_{(\theta',\mathcal{G},\mathcal{A})}(T)\right)}{\log T}.\end{aligned}\tag{54}$$

Proof: It follows from Lemma 3 and Lemma 5 that

$$R^\pi_{(\theta^\star,\mathcal{G},\mathcal{A})}(T)+R^\pi_{(\theta',\mathcal{G},\mathcal{A})}(T)\geq\frac{\epsilon T}{2}\exp(-KL(\mathbb{P},\mathbb{P}')).\tag{55}$$

Substituting the expression for $KL(\mathbb{P},\mathbb{P}')$ from Lemma 4 into the above inequality, and taking logarithms, we obtain the following,

$$\begin{aligned}&\frac{1}{2}(\theta_i^\star-\theta_i')^\intercal\bar{G}_i(T)\;(\theta_i^\star-\theta_i')\geq\\&\log\left(\frac{\epsilon T}{2}\right)-\log\left(R^\pi_{(\theta^\star,\mathcal{G},\mathcal{A})}(T)+R^\pi_{(\theta',\mathcal{G},\mathcal{A})}(T)\right).\end{aligned}$$

We then substitute the value of θ_i' from (50) in the above inequality and perform some algebraic manipulations in order to obtain (54). ∎

Lemma 7: Let $\delta a^\star$ be as in (52), and π be a consistent learning algorithm. We then have that,

$$\liminf_{T\to\infty}\frac{\rho_i(T;H)}{\log T\;\|\delta a^\star\|^2_{\bar{G}_i^{-1}(T)}}\geq\frac{2}{\Delta_b^2},$$

where $\rho_i(T;H)$ is as defined in (53).

Proof: Since π is a consistent learning algorithm, we have

$$\limsup_{T\to\infty}\frac{\log\left(R^\pi_{(\theta^\star,\mathcal{G},\mathcal{A})}(T)+R^\pi_{(\theta',\mathcal{G},\mathcal{A})}(T)\right)}{\log T}\leq 0.$$

Substituting this into the inequality (54) yields

$$\frac{(\Delta_{b^\star}+\epsilon)^2}{2}\liminf_{T\to\infty}\frac{\rho_i(T;H)}{\log T\;\|\delta a^\star\|^2_{\bar{G}_i^{-1}(T)}}\geq 1.$$

The result then follows since the bound holds true for an arbitrary choice of $b^\star$, and for all $\epsilon>0$. ∎

Next, define $c:=\limsup_{T\to\infty}\log T\;\|\delta a^\star\|^2_{\bar{G}_i^{-1}(T)}$, and let $d\in\mathbb{R}$ be such that $d\leq\liminf_{T\to\infty}\frac{\rho_i(T;H)}{\log T\;\|\delta a^\star\|^2_{\bar{G}_i^{-1}(T)}}$. We then have that

$$c\leq\frac{\liminf_{T\to\infty}\rho_i(T;H)}{d},\tag{56}$$

where $H>0$. It follows from Lemma 7 that d can be taken to be $2/\Delta_{b^\star}^2$. We now obtain an upper-bound on $\liminf_{T\to\infty}\rho_i(T;H)$ which will give us an upper-bound on c.

Lemma 8: Define, $\tilde{H}_i(T):=\frac{\bar{G}_i^{-1}(T)}{\|\bar{G}_i^{-1}(T)\|}$, and let $\tilde{H}_i(\infty)$ be a limit point of $\tilde{H}_i(T)$. We then have that

$$\liminf_{T\to\infty}\rho_i(T;\tilde{H}_i(\infty))\leq 1.\tag{57}$$

Proof: We have

$$\begin{aligned}\rho_i(T;H)&=\frac{\|\delta a^\star\|^2_{H^\intercal\bar{G}_i(T)H}\|\delta a^\star\|^2_{\bar{G}_i^{-1}(T)}}{\|\delta a^\star\|_H^4}\\&=\|\delta a^\star\|^2_{\tilde{H}_i(T)}\;\;\|\delta a^\star\|^2_{H\tilde{H}_i(T)H}\;\;\|\delta a^\star\|_H^{-4}.\end{aligned}$$

The last expression computes to 1 with H set equal to $\tilde{H}_i(T)$. It then follows that $\liminf_{T\to\infty}\rho_i(T;\tilde{H}_i(\infty))\leq 1$. ∎

Lemma 9: Under any consistent learning algorithm π, we have that

$$\limsup_{T\to\infty}\log T\;\|u_b-u_i^\star\|^2_{\bar{G}_i^{-1}(T)}\leq\frac{\Delta_b^2}{2},\quad\forall b\in\mathcal{A}_i^{(s)}.$$

Proof: Follows by substituting (57) into the inequality (56), and choosing d to be equal to $2/\Delta_b^2$. ∎

B. Auxiliary Results Used in Proof of Theorem 2

We will derive an upper-bound on the probability of the event $\mathcal{E}$.

Lemma 10: Let the decisions $\{U_i(t):i\in\mathcal{V}\}_{t\in[1,T]}$ be deterministic. We then have that,

$$\mathbb{P}\left(\mathcal{E}_i(x,\alpha,t)\right)\leq 2\exp\left(-\frac{\alpha^2}{2\|x\|^2_{\bar{G}_i^{-1}(t)}}\right).\tag{58}$$

Proof: For $\lambda > 0$, it follows from Chebyshev's inequality that,

$$\mathbb{P}\left(x^\intercal e_i(t) > \alpha\right) \leq \exp(-\lambda\alpha)\mathbb{E}\exp(\lambda x^\intercal e_i(t)). \tag{59}$$

Substituting the expression for $x^\intercal e_i(t)$ from (22), we obtain, $\mathbb{E}\exp(\lambda x^\intercal e_i(t)) = \exp\left(\frac{\lambda^2}{2}\|x\|^2_{G_i^{-1}(t)}\right)$. Substituting the above into the inequality (59), we obtain $\mathbb{P}\left(x^\intercal e_i(t) > \alpha\right) \leq \exp(-\lambda\alpha)\exp\left(\frac{\lambda^2}{2}\|x\|^2_{G_i^{-1}(t)}\right)$. For $\lambda = \alpha/\|x\|^2_{G_i^{-1}(t)}$, this bound reduces to $\mathbb{P}\left(x^\intercal e_i(t) > \alpha\right) \leq \exp\left(-\frac{\alpha^2}{2\|x\|^2_{G_i^{-1}(t)}}\right)$.

A similar bound can be derived for the probability of the event $\{x^\intercal e_i(t) < -\alpha\}$. Combining these bounds completes the proof. ∎

Note that the exploration phase is composed of "episodes," such that each action in the barycentric spanner $\mathcal{BS}$ is played exactly once during an episode. Thus, an episode lasts for d rounds. After t episodes, the matrices $G_i(td)$ are given as follows, $G_i(td) = t\left(\sum_{j\in\mathcal{N}_i}\sum_{a\in\mathcal{BS}\cap\mathcal{A}_j} u_a u_a^\intercal\right) = t|\mathcal{N}_i|\left(\sum_{u\in\tilde{\mathcal{U}}} u\ u^\intercal\right)$, so that $G_i^{-1}(td) = \frac{1}{t|\mathcal{N}_i|}\left(\sum_{u\in\tilde{\mathcal{U}}} u\ u^\intercal\right)^{-1}$.

Lemma 11: If the decisions $\{U_i(t) : i \in \mathcal{V}\}_{t\in[1,T]}$ are such that only the actions in $\mathcal{BS}$ are played in a round-robin manner, then,

$$\|u_a\|^2_{G_{i_a}^{-1}(kd)} \leq \frac{d}{k|\mathcal{N}_i|},\ \forall a \in \mathcal{A},\ k = 1, 2, \ldots, \lfloor T/d\rfloor, \tag{60}$$

where for $x \in \mathbb{R}$, $\lfloor x\rfloor$ denotes the greatest integer less than or equal to x.

Proof: Within this proof, we let i denote the node i_a at which action a is played. Since $\tilde{\mathcal{U}}$ is a barycentric spanner for $\mathcal{U}$, we have $u_a = \sum_{u\in\tilde{\mathcal{U}}}\alpha_u u$, where $\alpha_u \in [-1, 1]$, $\forall u \in \tilde{\mathcal{U}}$. Thus,

$$\begin{aligned}
\|u_a\|^2_{G_i^{-1}(kd)} &= \sum_{u\in\tilde{\mathcal{U}}} \alpha_u^2\ u^\intercal G_i^{-1}(kd)\ u \\
&\leq \frac{1}{k|\mathcal{N}_i|}\sum_{u\in\tilde{\mathcal{U}}} \alpha_u^2\ u^\intercal\left(u\ u^\intercal\right)^\dagger u \\
&\leq \frac{1}{k|\mathcal{N}_i|}\sum_{u\in\tilde{\mathcal{U}}} u^\intercal\left(u\ u^\intercal\right)^\dagger u \\
&\leq \frac{d}{k|\mathcal{N}_i|},
\end{aligned}$$

where for a matrix A, we let $A^\dagger$ denote its pseudoinverse, the first inequality follows since $G_i^{-1}(td) = \frac{1}{t|\mathcal{N}_i|}\left(\sum_{u\in\tilde{\mathcal{U}}} u\ u^\intercal\right)^{-1}$, and the second inequality follows since $|\alpha_u| \leq 1$. ∎

Recall that the size of confidence intervals, $\alpha(t)$, is as follows,

$$\alpha(t) = \sqrt{\frac{2\log\left(T\sum_{i\in\mathcal{V}}|\mathcal{A}_i|/\delta\right)}{t}}\ d. \tag{61}$$

Lemma 12: We have the following upper-bound on the probability of event $\mathcal{E}$ while playing the arms in the set $\mathcal{BS}$ in a round-robin manner, $\mathbb{P}(\mathcal{E}) \leq \delta$.

Proof: Substituting the bound (60) for $\|u_a\|^2_{G_{i_a}^{-1}(td)}$ into the inequality (58), we obtain,

$$\begin{aligned}
\mathbb{P}\left(\mathcal{E}_i(a_i, \alpha(kd), kd)\right) &\leq \exp\left(-\frac{\alpha^2(kd)|\mathcal{N}_i|kd}{2d}\right) \\
&\leq \frac{\delta}{T\sum_{i\in\mathcal{V}}|\mathcal{A}_i|}.
\end{aligned}$$

The proof then follows by using the union bound, i.e., $\mathbb{P}(\mathcal{E}) \leq \sum_{k,i}\sum_{a\in\mathcal{A}_i}\mathbb{P}\left(\mathcal{E}_i(u_a, \alpha(kd), kd)\right)$. ∎

C. Auxiliary Results Used in Proof of Theorem 3

Recall that $\hat{\Delta} = \left\{\hat{\Delta}_a : a \in \mathcal{A}\right\}$ denotes the estimates (30) of sub-optimality gaps of arms, obtained at the end of the warm-up phase.

Lemma 13: Consider the optimization problem OPT($\hat{\Delta}$) (31)-(33), solving which requires the estimates $\hat{\Delta}$ as an input. We then have that,

$$\sum_{a\in\mathcal{A}^{(s)}} \beta_a^\star(\hat{\Delta}) \leq 2d^3 f(T)\frac{\hat{\Delta}_{\max}}{\hat{\Delta}^3_{\min}},$$

where $\beta^\star(\hat{\Delta}) = \left\{\beta_a^\star(\hat{\Delta})\right\}_{a\in\mathcal{A}}$ is a solution of (31)-(33).

Proof: We omit the proof since it closely follows the proof of Lemma 12 of [4]. ∎

Lemma 14: Define $\delta_T := \max_{a\in\mathcal{A}:\hat{\Delta}_a>0}\ \frac{\Delta_a^2}{\hat{\Delta}_a^2} - 1$. We then have that,

$$\sum_{a\in\mathcal{A}} \beta_a^\star(\hat{\Delta})\hat{\Delta}_a \leq (1+\delta_T)\sum_{a\in\mathcal{A}}\beta_a^\star(\Delta)\hat{\Delta}_a. \tag{62}$$

Proof: For an action a, we have $\|u_a\|^2_{H_{i_a}^{-1}((1+\delta_T)\beta^\star(\Delta))} = \|u_a\|^2_{H_{i_a}^{-1}(\beta^\star(\Delta))}/(1+\delta_T) \leq \Delta_a^2/(1+\delta_T)f(T) \leq \hat{\Delta}_a^2/f(T)$, where $H_i(\cdot)$ is as defined in (33), the first inequality follows since $\beta^\star(\Delta)$ is feasible for $OPT(\Delta)$, and the last inequality follows from the definition of δ_T. It follows from the above inequality that the vector $\{(1+\delta_T)\beta_a^\star(\Delta) : a \in \mathcal{A}\}$ is feasible for $OPT(\hat{\Delta})$. Hence, the optimal value of $OPT(\hat{\Delta})$ is upper-bounded by $(1+\delta_T)\sum_{a\in\mathcal{A}}\beta_a^\star(\Delta)\hat{\Delta}_a$. This completes the proof. ∎

Lemma 15: If $2\epsilon_T(d\log^{1/2}T) \leq \Delta_{\min}/2$, then we have the following upper-bound on the quantity δ_T that was defined in Lemma 14, $\delta_T \leq \frac{16\epsilon_T(d\log^{1/2}T)}{\Delta_{\min}}$.

Proof: Within this proof we use ϵ_T to denote $\epsilon_T(d\log^{1/2}T)$. We have

$$\begin{aligned}
1+\delta_T &= \max_{a\in\mathcal{A}:\hat{\Delta}_a>0}\frac{\Delta_a^2}{\hat{\Delta}_a^2} \\
&\leq \max_{a\in\mathcal{A}:\hat{\Delta}_a>0}\frac{\Delta_a^2}{\left(\Delta_a - 2\epsilon_T\right)^2} \\
&\leq \max_{a\in\mathcal{A}:\hat{\Delta}_a>0}\left(1 + \frac{4\left(\Delta_a - \epsilon_T\right)\epsilon_T}{\left(\Delta_a - 2\epsilon_T\right)^2}\right) \\
&\leq 1 + \frac{16\epsilon_T}{\Delta_{\min}}.
\end{aligned}$$

∎

REFERENCES

[1] P. Auer, "Using confidence bounds for exploitation-exploration tradeoffs," *J. Mach. Learn. Res.*, vol. 3, pp. 397–422, Nov. 2002.

[2] L. Li, W. Chu, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proc. 19th Int. Conf. World Wide Web*, 2010, pp. 661–670.

[3] J. Langford and T. Zhang, "The epoch-greedy algorithm for multi-armed bandits with side information," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 817–824.

[4] T. Lattimore and C. Szepesvári, "The end of optimism? An asymptotic analysis of finite-armed linear bandits," in *Proc. Artif. Intell. Statist.*, 2017, pp. 728–737.

[5] Wikipedia. (2020). *Facebook*. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Facebook&oldid=957360060

[6] S. A. Myers, C. Zhu, and J. Leskovec, "Information diffusion and external influence in networks," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2012, pp. 33–41.

[7] M. Fang and D. Tao, "Networked bandits with disjoint linear payoffs," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2014, pp. 1106–1115.

[8] S. Caron, B. Kveton, M. Lelarge, and S. Bhagat, "Leveraging side observations in stochastic bandits," in *Proc. 28th Conf. Uncertainty Artif. Intell.*, N. de Freitas and K. P. Murphy, Eds., Catalina Island, CA, USA: AUAI Press, Aug. 2012, pp. 142–151. [Online]. Available: https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=2277&proceeding_id=28

[9] S. Buccapatnam, A. Eryilmaz, and N. B. Shroff, "Stochastic bandits with side observations on networks," *SIGMETRICS Perform. Eval. Rev.*, vol. 42, no. 1, pp. 289–300, Jun. 2014, doi: 10.1145/2637364.2591989.

[10] F. Molnár, N. Derzsy, É. Czabarka, L. Székely, B. K. Szymanski, and G. Korniss, "Dominating scale-free networks using generalized probabilistic methods," *Sci. Rep.*, vol. 4, no. 1, pp. 1–9, Sep. 2014.

[11] A. Bonato, M. Lozier, D. Mitsche, X. Pérez-Giménez, and P. Prałat, "The domination number of on-line social networks and random geometric graphs," in *Theory and Applications of Models of Computation: 12th Annual Conference, TAMC 2015, Singapore, May 18–20, 2015, Proceedings 12*. Singapore: Springer, 2015, pp. 150–163.

[12] S. Bubeck and N. Cesa-Bianchi, "Regret analysis of stochastic and nonstochastic multi-armed bandit problems," *Found. Trends Mach. Learn.*, vol. 5, no. 1, pp. 1–122, 2012.

[13] T. Lattimore and C. Szepesvári, *Bandit Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2020.

[14] T. L. Lai and H. Robbins, "Asymptotically efficient adaptive allocation rules," *Adv. Appl. Math.*, vol. 6, no. 1, pp. 4–22, 1985.

[15] P. Auer, N. Cesa-Bianchi, and P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Mach. Learn.*, vol. 47, nos. 2–3, pp. 235–256, May 2002.

[16] W. Chu, L. Li, L. Reyzin, and R. Schapire, "Contextual bandits with linear payoff functions," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 208–214.

[17] O. Chapelle and L. Li, "An empirical evaluation of Thompson sampling," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2249–2257.

[18] S. Agrawal and N. Goyal, "Thompson sampling for contextual bandits with linear payoffs," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 127–135.

[19] S. Agrawal and N. Goyal, "Analysis of Thompson sampling for the multi-armed bandit problem," in *Proc. 25th Conf. Learn. Theory*, 2012, pp. 1–39.

[20] D. J. Russo, B. V. Roy, A. Kazerouni, I. Osband, and Z. Wen, "A tutorial on Thompson sampling," *Found. Trends Mach. Learn.*, vol. 11, no. 1, pp. 1–96, 2018.

[21] Y.-H. Hung, P.-C. Hsieh, X. Liu, and P. R. Kumar, "Reward-biased maximum likelihood estimation for linear stochastic bandits," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 9, pp. 7874–7882.

[22] P. Kumar and A. Becker, "A new family of optimal adaptive controllers for Markov chains," *IEEE Trans. Autom. Control*, vol. AC-27, no. 1, pp. 137–146, Feb. 1982.

[23] X. Liu, P.-C. Hsieh, Y. H. Hung, A. Bhattacharya, and P. R. Kumar, "Exploration through reward biasing: Reward-biased maximum likelihood estimation for stochastic multi-armed bandits," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 6248–6258.

[24] A. Mete, R. Singh, X. Liu, and P. R. Kumar, "Reward biased maximum likelihood estimation for reinforcement learning," in *Proc. Learn. Dyn. Control*, 2021, pp. 815–827.

[25] A. Slivkins, "Contextual bandits with similarity information," in *Proc. 24th Annu. Conf. Learn. Theory*, 2011, pp. 679–702.

[26] A. Krishnamurthy, Z. S. Wu, and V. Syrgkanis, "Semiparametric contextual bandits," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2776–2785.

[27] P. Rusmevichientong and J. N. Tsitsiklis, "Linearly parameterized bandits," *Math. Oper. Res.*, vol. 35, no. 2, pp. 395–411, May 2010.

[28] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári, "Improved algorithms for linear stochastic bandits," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 24, 2011, pp. 2312–2320.

[29] S. Mannor and O. Shamir, "From bandits to experts: On the value of side-observations," in *Proc. NIPS*, 2011, pp. 684–692.

[30] N. Alon, N. Cesa-Bianchi, C. Gentile, S. Mannor, Y. Mansour, and O. Shamir, "Nonstochastic multi-armed bandits with graph-structured feedback," *SIAM J. Comput.*, vol. 46, no. 6, pp. 1785–1826, Jan. 2017.

[31] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire, "The nonstochastic multiarmed bandit problem," *SIAM J. Comput.*, vol. 32, no. 1, pp. 48–77, 2002.

[32] S. Buccapatnam, F. Liu, A. Eryilmaz, and N. B. Shroff, "Reward maximization under uncertainty: Leveraging side-observations on networks," *J. Mach. Learn. Res.*, vol. 18, pp. 216:1–216:34, 2017. [Online]. Available: http://jmlr.org/papers/v18/16-340.html

[33] A. Cohen, T. Hazan, and T. Koren, "Online learning with feedback graphs without the graphs," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 811–819.

[34] N. Alon, N. Cesa-Bianchi, O. Dekel, and T. Koren, "Online learning with feedback graphs: Beyond bandits," in *Proc. Conf. Learn. Theory*, 2015, pp. 23–35.

[35] P. Landgren, V. Srivastava, and N. E. Leonard, "Distributed cooperative decision-making in multiarmed bandits: Frequentist and Bayesian algorithms," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Dec. 2016, pp. 167–172.

[36] R. K. Kolla, K. Jagannathan, and A. Gopalan, "Collaborative learning of stochastic bandits over a social network," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1782–1795, Aug. 2018.

[37] P. Landgren, V. Srivastava, and N. E. Leonard, "Social imitation in cooperative multiarmed bandits: Partition-based algorithms with strictly local information," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 5239–5244.

[38] D. Martínez-Rubio, V. Kanade, and P. Rebeschini, "Decentralized cooperative stochastic bandits," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 4529–4540.

[39] L. Yang, Y.-Z. J. Chen, S. Pasteris, M. Hajiesmaili, J. Lui, and D. Towsley, "Cooperative stochastic bandits with asynchronous agents and constrained feedback," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 8885–8897.

[40] L. Yang, Y. J. Chen, M. H. Hajiemaili, J. C. S. Lui, and D. Towsley, "Distributed bandits with heterogeneous agents," in *Proc. IEEE Conf. Comput. Commun.*, May 2022, pp. 200–209.

[41] D. Vial, S. Shakkottai, and R. Srikant, "Robust multi-agent bandits over undirected graphs," *ACM Meas. Anal. Comput. Syst.*, vol. 6, no. 3, pp. 1–57, 2022.

[42] A. Sankararaman, A. Ganesh, and S. Shakkottai, "Social learning in multi agent multi armed bandits," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 3, no. 3, pp. 1–35, 2019.

[43] R. Chawla, A. Sankararaman, A. Ganesh, and S. Shakkottai, "The gossiping insert-eliminate algorithm for multi-agent bandits," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 3471–3481.

[44] O. T. Odeyomi, "Learning the truth in social networks using multi-armed bandit," *IEEE Access*, vol. 8, pp. 137692–137701, 2020.

[45] R. Lage, L. Denoyer, P. Gallinari, and P. Dolog, "Choosing which message to publish on social networks: A contextual bandit approach," in *Proc. IEEE/ACM Int. Conf. Adv. Social Netw. Anal. Mining (ASONAM)*, Aug. 2013, pp. 620–627.

[46] C. Tekin, S. Zhang, and M. van der Schaar, "Distributed online learning in social recommender systems," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 4, pp. 638–652, Aug. 2014.

[47] Z. Bnaya, R. Puzis, R. Stern, and A. Felner, "Bandit algorithms for social network queries," in *Proc. Int. Conf. Social Comput.*, Sep. 2013, pp. 148–153.

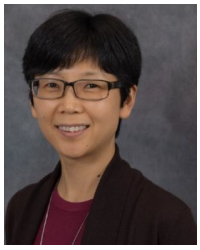
[48] B. Awerbuch and R. D. Kleinberg, "Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches," in *Proc. 36th Annu. ACM Symp. Theory Computing*, Chicago, IL, USA, Jun. 2004, pp. 45–53, doi: 10.1145/1007352.1007367.

[49] V. Dani, T. P. Hayes, and S. M. Kakade, "Stochastic linear optimization under bandit feedback," in *Proc. 21st Annu. Conf. Learn. Theory*, Helsinki, Finland, 2008, pp. 355–366.
[50] G. B. Folland, *Real Analysis: Modern Techniques and Their Applications*. Hoboken, NJ, USA: Wiley, 2013.
[51] I. T. Jolliffe and J. Cadima, "Principal component analysis: A review and recent developments," *Phil. Trans. Roy. Soc. A: Math., Phys. Eng. Sci.*, vol. 374, no. 2065, Apr. 2016, Art. no. 20150202.

Avik Kar (Member, IEEE) received the M.Tech. degree from Indian Institute of Technology Kharagpur, India, in 2021. He is currently pursuing the Ph.D. degree with Indian Institute of Science Bengaluru, India. His research interests include reinforcement learning and online machine learning. He was a recipient of the Keshab Kanti Parhi Endowment Award for his M.Tech. Thesis and a recipient of the Prime Minister's Research Fellowship.

Rahul Singh (Member, IEEE) received the B.Tech. degree in electrical engineering from Indian Institute of Technology Kanpur, India, in 2009, the M.S. degree in electrical engineering from the University of Notre Dame, South Bend, USA, in 2011, and the Ph.D. degree from Texas A & M University, College Station, in 2015. Currently, he is an Assistant Professor with the Department of Electrical Communication Engineering, Indian Institute of Science Bengaluru, India. He was a Post-Doctoral Scholar with the Laboratory for Information and Decision Systems (LIDS), Massachusetts Institute of Technology, and The Ohio State University. His research interests include machine learning, networks, and stochastic control. His article was runner-up for the Best Paper Award of ACM MobiHoc 2020.

Fang Liu received the B.S. degree in information engineering from the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, in 2014, and the Ph.D. degree in electrical and computer engineering from The Ohio State University in 2019. He was a Student Intern with the AT&T Laboratories Research in Summer 2018 and was a Software Engineer Intern with Facebook NYC in Summer 2019. Currently, he is a Research Scientist with Facebook. His research interests are statistics and machine learning. He was a recipient of the Litton Fellowship, UAI-18 Scholarship, and AAAI-18 Scholarship during the Ph.D. degree.



Xin Liu (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Purdue University in 2002. She is currently a Professor with the Department of Computer Science, University of California at Davis. Her current research interests fall in the general areas of machine learning algorithm development and machine learning applications in human and animal healthcare, food systems, and communication networks. Her research on networking includes cellular networks, cognitive radio networks, wireless sensor networks, network information theory, network security, and the IoT systems.

Ness B. Shroff (Fellow, IEEE) received the Ph.D. degree from Columbia University, NY, USA, in 1994. He joined Purdue university immediately thereafter as an Assistant Professor. At Purdue, he became a Professor with the School of Electrical and Computer Engineering and the Director of CWSA in 2004, a university-wide center on wireless systems and applications. In July 2007, he joined the ECE Department and the CSE Department, The Ohio State University, where he holds Ohio Eminent Scholar Chaired Professorship of Networking and Communications. He was a Guest Chaired Professor in wireless communications with Tsinghua University and an Honorary Guest Professor with Shanghai Jiaotong University, China. He currently holds a visiting professor position with Indian Institute of Technology Bombay.

His research interests span the areas of communication, networking, computing, storage, cloud, recommender, social, cyber-physical systems, fundamental problems in machine learning, design, control, performance, pricing, and security of these complex systems. He is a National Science Foundation CAREER Awardee. His papers have received numerous awards at various top-tier venues. He is on the list of highly cited researchers from Thomson Reuters ISI in 2014 and 2015 and in the Thomson Reuters Book on The World's Most Influential Scientific Minds in 2014. He received the IEEE INFOCOM Achievement Award for seminal contributions to scheduling and resource allocation in wireless networks in 2014. He is currently leading an NSF AI Institute for Future Edge Networks and Distributed Intelligence.