Journal of Machine Learning Research 21 (2020) 1-44 Submitted 10/16; Published 11/20

Spectral bandits

Tomáš Kocák[*] TOMAS.KOCAK@GMAIL.COM
ENS de Lyon, 15 Parvis René Descartes, 69342 Lyon, France

Rémi Munos[*] MUNOS@GOOGLE.COM
DeepMind Paris, 14 Rue de Londres, 75009 Paris, France

Branislav Kveton BKVETON@GOOGLE.COM
Google Research, 1600 Amphitheatre Parkway, Mountain View, CA 94043, United States

Shipra Agrawal SA3305@COLUMBIA.EDU
Columbia University, West 120th Street, New York, NY, 10027 United States

Michal Valko[*] VALKOM@DEEPMIND.COM
DeepMind Paris, 14 Rue de Londres, 75009 Paris, France

Editor: Peter Auer

Abstract

Smooth functions on graphs have wide applications in manifold and semi-supervised learning. In this work, we study a bandit problem where the payoffs of arms are smooth on a graph. This framework is suitable for solving online learning problems that involve graphs, such as content-based recommendation. In this problem, each item we can recommend is a node of an undirected graph and its expected rating is similar to the one of its neighbors. The goal is to recommend items that have high expected ratings. We aim for the algorithms where the cumulative regret with respect to the optimal policy would not scale poorly with the number of nodes. In particular, we introduce the notion of an *effective dimension,* which is small in real-world graphs, and propose three algorithms for solving our problem that scale linearly and sublinearly in this dimension. Our experiments on content recommendation problem show that a good estimator of user preferences for thousands of items can be learned from just tens of node evaluations.

1. Introduction

A smooth graph function is a function on a graph that returns similar values on neighboring nodes. This concept arises frequently in manifold and semi-supervised learning (Zhu, 2008; Valko et al., 2010), and reflects the fact that the outcomes on the neighboring nodes tend to be similar. It is well-known (Belkin et al., 2006, 2004) that a smooth graph function can be expressed as a linear combination of the eigenvectors of the graph Laplacian with smallest eigenvalues (see Figure 1 for an example). Therefore, the problem of learning such function can be cast as a regression problem on these eigenvectors. The present work brings this concept to bandits (Valko, 2016). In particular, we study a bandit problem where the arms are the nodes of a graph and the expected payoff of pulling an arm is a smooth function on this graph.

[*] also affiliated with Inria Lille – Nord Europe, SequeL team

We are motivated by a range of practical problems that involve graphs. One application is *targeted advertisement* in social networks. Here, the graph is a social network and our goal is to discover a part of the network that is interested in a given product. Interests of people in a social network tend to change smoothly (McPherson et al., 2001), because friends tend to have similar preferences. Therefore, we take advantage of this structure and formulate this problem as learning a smooth preference function on a graph.

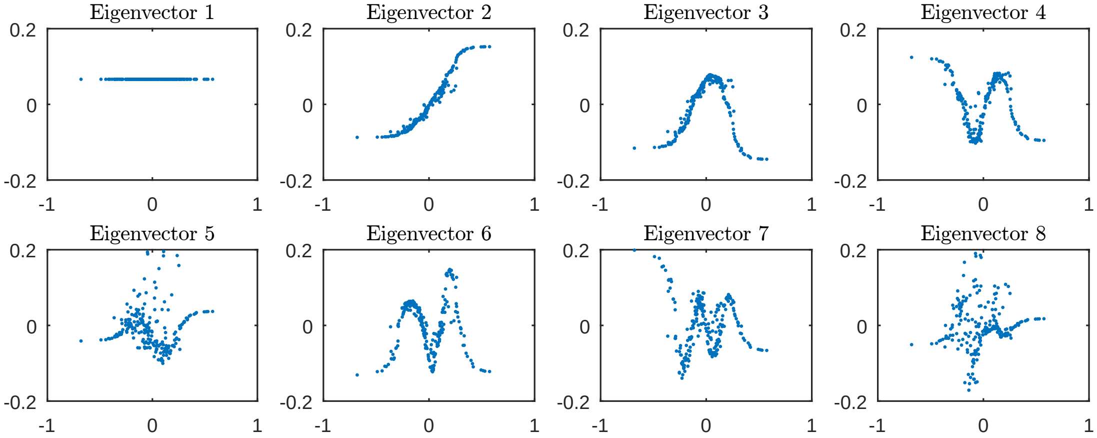


Figure 1: Eigenvectors from the Flixster dataset corresponding to the smallest few eigenvalues projected onto the first principal component of the data; x-axis represents components of the eigenvector sorted according to the projection onto the first principal component of the data while y-axis represent the value of the corresponding component of the eigenvector. To produce the figure above, we performed the following steps. (1) *Preprocessing:* We remove all the users that rated a small number of movies as well as the movies rated by only a few users. This leaves us with a $u \times m$ matrix $\mathbf{M}$ where u is the number of users and m is the number of movies and entry $\mathbf{M}_{i,j}$ of matrix $\mathbf{M}$ is the rating of movie j by user i, provided it exists. Note that matrix $\mathbf{M}$ might be missing some of the entries. (2) *Filling in the missing entries:* For this step, we use low-rank matrix factorization (Keshavan et al., 2009) to obtain $u \times r$ matrix $\mathbf{U}$ and $m \times r$ matrix $\mathbf{V}$, for some given rank r, such that $\mathbf{M} \approx \mathbf{U}\mathbf{V}^{\mathsf{T}}$. (3) *Constructing a similarity graph:* We construct the graph by creating an edge between movies i and j if the movie j is among 5 nearest neighbors of the movie i in the latent space of movies $\mathbf{V}$. (4) *Visualization:* Using the computed matrix $\mathbf{V}$, the matrix capturing the latent space of movies, we can find the direction of the highest variance of the data using PCA on $\mathbf{V}$. This gives us a way to visualize eigenvectors by projecting them on the first principal component. The above visualization shows that the eigenvectors corresponding to smaller eigenvalues tend to be smoother—the values corresponding to actions with their projections close to each other are similar. On the other hand, the eigenvectors corresponding to larger eigenvalues are more chaotic as values for nearby items can vary a lot. This gives a small insight into why a function created as a linear combination of the first few eigenvectors is a smooth reward function. We are more precise about the definition of smoothness and the connection of smooth functions and eigenvectors later in Section 4.

Another application of our work are *recommender systems* (Jannach et al., 2010). In content-based recommendation (Chau et al., 2011), the user is recommended items that are similar to the items that the user rated highly in the past. The assumption is that users prefer similar items similarly. The similarity of the items is measured for instance by a nearest-neighbor graph (Billsus et al., 2000), where each item is a node and its neighbors are the most similar items.

We consider the following learning setting. The graph is known in advance and its edges represent the similarity of the nodes. At round t, we choose a node and then observe its payoff. In targeted advertisement, this may correspond to showing an ad and then observing whether the person has clicked on it. In content-based recommendation, this may correspond to recommending an item and then observing the assigned rating. Based on the payoff, we update our model of the world and then the game proceeds into round $t+1$. In both applications described above, the learner (advertiser) has rarely the budget (time horizon T) to try all the options even once. Furthermore, imagine that the learner is a movie recommender system and would ask the user to rate all the movies before it starts producing relevant recommendations. Such a recommender system would be of little value. Yet, many bandit algorithms start with pulling each arm once. This is something that we cannot afford and therefore, contrary to standard bandits, we consider the case $T \ll N$, where the number of nodes N is huge. While we are mostly interested in the regime when $t < N$, our results are beneficial also for $t > N$. This regime is especially challenging since traditional multi-arm bandit algorithms need to try every arm.

If the smooth graph function is expressed as a linear combination of k eigenvectors of the graph Laplacian and k is small and known, our learning problem can be solved using ordinary linear bandits (Auer, 2002; Dani et al., 2008; Li et al., 2010; Agrawal and Goyal, 2013b; Abeille and Lazaric, 2017). In practice, k is problem specific and unknown. Moreover, the number of features k may approach the number of nodes N. Therefore, proper regularization is necessary, so that the regret of the learning algorithm does not scale with N. We are interested in the setting where the regret is independent of N and this makes the problem we study non-trivial.

Early short versions of our work appeared at *International Conference on Machine Learning* (Valko et al., 2014) and *AAAI Conference on Artificial Intelligence* (Kocák et al., 2014b). Compared to those, we give a new and improved definition of the effective dimension that is smaller than the old one, provide a matching lower bound, improved the regret bounds for two of our algorithm, and report a comprehensive empirical evaluation on artificial datasets as well as on the Movielens and Flixster datasets.

2. Setting

In this section, we formally define the spectral bandit setting. Let $\mathcal{G}$ be the given graph with the set of nodes $\mathcal{V}$ and denote $N \triangleq |\mathcal{V}|$ the number of nodes. Let $\mathcal{W}$ be the symmetric $N \times N$ matrix of similarities w_{ij} (edge weights) and $\mathcal{D}$ be the $N \times N$ diagonal matrix with entries $d_{ii} \triangleq \sum_j w_{ij}$ (node degrees). The graph Laplacian of $\mathcal{G}$ is defined as $\mathcal{L} \triangleq \mathcal{D} - \mathcal{W}$. Let $\{\lambda_k^{\mathcal{L}}, \mathbf{q}_k\}_{k=1}^N$ be the eigenvalues and eigenvectors of $\mathcal{L}$ ordered such that $0 = \lambda_1^{\mathcal{L}} \leq \lambda_2^{\mathcal{L}} \leq \cdots \leq \lambda_N^{\mathcal{L}}$. Equivalently, let $\mathcal{L} \triangleq \mathbf{Q}\mathbf{\Lambda}_{\mathcal{L}}\mathbf{Q}^\mathsf{T}$ be an eigendecomposition of $\mathcal{L}$, where $\mathbf{Q}$ is an $N \times N$ *orthogonal* matrix with eigenvectors in columns.

The eigenvectors of the graph Laplacian form a basis. Therefore, we can represent the reward function as a linear combination of the eigenvectors. For any set of weights $\boldsymbol{\alpha}$, let $f_{\boldsymbol{\alpha}} : \mathcal{V} \to \mathbb{R}$ be the function defined on nodes, linear in the basis of the eigenvectors of $\mathcal{L}$,

$$f_{\boldsymbol{\alpha}}(v) \triangleq \sum_{k=1}^{N} \alpha_k (\mathbf{q}_k)_v = \mathbf{x}_v^{\mathsf{T}} \boldsymbol{\alpha},$$

where $\mathbf{x}_v$ is the v-th row of $\mathbf{Q}$, i.e., $(\mathbf{x}_v)_i = (\mathbf{q}_i)_v$. If the weight coefficients of the true $\boldsymbol{\alpha}$ are such that the large coefficients correspond to the eigenvectors with the small eigenvalues and vice versa, then $f_{\boldsymbol{\alpha}}$ would be a smooth function on $\mathcal{G}$ (Belkin et al., 2006). For more details, see Section 4.1. Figure 1 displays the first few eigenvectors of the Laplacian constructed from the data that we use in our experiments. In the extreme case, the true $\boldsymbol{\alpha}$ may be of the form $[\alpha_1, \alpha_2, \ldots, \alpha_k, 0, 0, 0]_N^{\mathsf{T}}$ for some $k \ll N$. Had we known k in such case, the known linear bandit algorithms would work with the performance scaling with k instead of $D = N$. Unfortunately, first, we do not know k and second, we do not want to assume such an extreme case (i.e., $\alpha_i = 0$ for $i > k$). Therefore, we opt for the more plausible assumption that the coefficients with the high indexes are small. Consequently, we deliver algorithms with the performance that scale with the smoothness with respect to the graph.

We now define the learning setting. In each round $t \leq T$, the recommender chooses a node a_t and obtains a noisy reward such that

$$r_t \triangleq \mathbf{x}_{a_t}^{\mathsf{T}} \boldsymbol{\alpha} + \varepsilon_t,$$

where the noise ε_t is assumed to be zero mean and conditionally independent R-sub-Gaussian random variable for any t, that is, $\mathbb{E}\left[\exp(s\varepsilon_t)\right] \leq \exp(R^2 s^2/2)$, for all $s \in \mathbb{R}$ and $\mathbb{E}\left[\varepsilon_t\right] = 0$. In our setting, we have $\mathbf{x}_v \in \mathbb{R}^D$ and $\|\mathbf{x}_v\|_2 \leq 1$ for all $\mathbf{x}_v$. The goal of the recommender is to minimize the cumulative regret with respect to the strategy that always picks the best node w.r.t. $\boldsymbol{\alpha}$. Let a_t be the node picked (referred to as *pulling an arm*) by an algorithm at round t. The cumulative (pseudo-) regret of an algorithm is defined as

$$R_T \triangleq T \max_v f_{\boldsymbol{\alpha}}(v) - \sum_{t=1}^{T} f_{\boldsymbol{\alpha}}(a_t).$$

We call this bandit setting *spectral* since it is built on the spectral properties of a graph. Compared to the linear and multi-arm bandits, the number of arms K is equal to the number of nodes N and to the dimension of the basis D (the eigenvectors are of dimension N). However, a regret that scales with N or D that can be obtained using those approaches is not acceptable because the number of nodes can be large. While we are mostly interested in the setting with $K = N$, our algorithms and analyses are valid for any *finite* K.

3. Related work

The most related settings to our work are that of the linear and contextual linear bandits. For these settings, Auer (2002) proposed `SupLinRel` and showed that it obtains $\sqrt{DT}$ regret which matches the lower bound by Dani et al. (2008). However, the first practical and empirically successful algorithm was `LinUCB` (Li et al., 2010). Later, Chu et al. (2011)

analyzed SupLinUCB, which is a LinUCB equivalent of SupLinRel, to show that it also obtains $\sqrt{DT}$ regret. Abbasi-Yadkori et al. (2011) proposed OFUL for linear bandits which obtains $D\sqrt{T}$ regret. Using their analysis, it is possible to show that LinUCB obtains $D\sqrt{T}$ regret as well (Remark 25). Whether LinUCB matches the $\sqrt{DT}$ lower bound for this setting is still an open problem.

Apart from the above optimistic approaches, an older approach to the problem is Thompson sampling (TS, Thompson, 1933). It solves the exploration-exploitation dilemma by a simple and intuitive rule: when choosing the next action to play, choose it according to the probability that it is the best one; that is the one that maximizes the expected payoff. Chapelle and Li (2011) showed its practical relevance to the computational advertising. This motivated the researchers to explain the success of TS (Agrawal and Goyal, 2012; Kaufmann et al., 2012; May et al., 2012; Agrawal and Goyal, 2013a; Abeille and Lazaric, 2017). The most relevant results for our work are by Agrawal and Goyal (2013b), who bring a new martingale technique, enabling us to analyze cases where the payoffs of the actions are linear in some basis.

Abernethy et al. (2008) and Bubeck et al. (2012) studied a more difficult *adversarial* setting of linear bandits where the reward function is time-dependent. It is an open problem if this approaches would work in our setting and have an upper bound on the regret that scales better than with D.

Kleinberg et al. (2008), Slivkins (2009), and Bubeck et al. (2011) use similarity information between the context of arms, assuming a Lipschitz or more general properties. While such settings are indeed more general, the regret bounds scale worse with the relevant dimensions. Srinivas et al. (2010) and Valko et al. (2013) also perform maximization over the smooth functions that are either sampled from a Gaussian process prior or have a small RKHS norm. Their setting is also more general than ours since it already generalizes linear bandits. However, their regret bound in the linear case also scales with D. Moreover, the regret of these algorithms also depends on a quantity for which data-independent bounds exist only for some kernels, while our effective dimension is always computable given the graph.

Another bandit graph setting called the *gang of bandits* was studied by Cesa-Bianchi et al. (2013), where each node is a linear bandit with its own weight vector. These weight vectors are assumed to be smooth on the graph. Gentile et al. (2014) take a different approach to similarities in social networks by assuming that the actions are clustered into several unknown clusters and the actions within one cluster have the same expected reward. This approach can be applied also to the setting presented in our paper. The biggest advantage of the CLUB algorithm by Gentile et al. (2014) is that it constructs graph iteratively, starting with complete graph and removing edges which are not likely to be presented in the underlying clustering. Therefore, the algorithm does not need to know the similarity graph unlike in our setting. However, theoretical improvement of CLUB compared to the basic bandit algorithm comes from the small number of clusters. Therefore, if the number of clusters is close to the number of actions the algorithm does not bring any improvement while the algorithms in our setting still can leverage the similarity structure. Li et al. (2016) and Gentile et al. (2017) later extended the approach to *double-clustering* where both the users and the items are assumed to appear in clusters (with the underlying clustering unknown to the learner) and Korda et al. (2016) considers a distributed extension. Yet another assumption of a special graph reward structure is exploited by unimodal bandits (Yu and Mannor, 2011; Combes and Proutière, 2014). One of the settings considered by Yu and Mannor (2011) is a graph bandit setting where every path in the graph has unimodal rewards and therefore also imposes a specific kind of smoothness with respect to the graph topology. In networked bandits (Fang and Tao, 2014), the learner picks a node, but besides receiving the reward from that node, its reward is the sum of the rewards of the picked node and its neighborhood. The algorithm of Fang and Tao (2014), `NetBandits`, can also deal with changing topology, however, this has to be always revealed to the learner before it makes its decision.

Furthermore, bandits with side observations treat a different graph bandit setting where the learner obtains not only the reward from the selected action but also the rewards from the neighbors of the selected action. This setting was studied in both the stochastic case (Caron et al., 2012; Buccapatnam et al., 2014) and the adversarial one (Mannor and Shamir, 2011; Alon et al., 2013; Kocák et al., 2014a; Alon et al., 2017, 2015; Kocák et al., 2016a,b). For a comprehensive discussion, we refer to survey on graph bandits (Valko, 2016).

Spectral bandits with different objectives In the follow-up work on spectral bandits, there have been algorithms optimizing other objective function than the cumulative regret. First, in some sensor networks, sensing a node (pulling an arm) has an associated cost (Narang et al., 2013). In a particular, *cheap bandit* setting (Hanawal et al., 2015), it is cheaper to get an average of rewards of a set of nodes than a specific reward of a single one. More precisely, the learner pays the cost for the action which depends on the spectral properties of the graph while relying on the property that getting the average reward of many nodes is less costly than getting a reward of a single node. For this setting, Hanawal et al. (2015) proposed `CheapUCB` that reduces the cost of sampling by 1/4 as compared to `SpectralUCB`, while maintaining $\widetilde{\mathcal{O}}(d\sqrt{T})$ cumulative regret. Next, Gu and Han (2014) study the online classification setting on graphs with bandit feedback, very similar to spectral bandits; after predicting the class the oracle returns a single bit indicating whether the prediction is correct or not. The analysis of their algorithm delivers essentially the same bound on the regret, however, they need to know the number of relevant eigenvectors d. Moreover, Ma et al. (2015) consider several variants of Σ-*optimality* that favors specific exploration when selecting the nodes, for example, the learner is not allowed to play one arm multiple times. The authors were able to show a regret bound which scales with the effective dimension that we defined in our prior work (Valko et al., 2014).

4. Spectral bandits

In this section, we show how to leverage the smoothness of the rewards on a given graph. In our setting, the features of the arms (contexts) form a basis and therefore are *orthogonal* to each other. Thinking that the reward observed for an arm does not provide any information for other arms would not be correct because of the assumption that under another basis, the unknown parameter has a low norm. This provides an additional information across the arms through the estimation of the parameter $\boldsymbol{\alpha}$.

4.1 Smooth graph functions

There are several possible ways to define the *smoothness* of the function f with respect to the undirected graph G. We are using the one which is standard in the spectral clustering (von Luxburg, 2007) and semi-supervised learning (Belkin et al., 2006), defined as

$$S_G(f) \triangleq \frac{1}{2} \sum_{i,\, j \in [N]} w_{i,j} \left(f(i) - f(j)\right)^2.$$

Therefore, whenever the function values of the nodes connected by an edge with large weight are close, the smoothness of the function with respect to the graph is small and the function is smoother with respect to the graph. This definition has several useful properties. We are mainly interested in the following one,

$$S_G(f) = \mathbf{f}^\mathsf{T}\mathcal{L}\mathbf{f} = \mathbf{f}^\mathsf{T}\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\mathsf{T}\mathbf{f} = \boldsymbol{\alpha}^\mathsf{T}\boldsymbol{\Lambda}\boldsymbol{\alpha} = \|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}^2 = \sum_{i=1}^N \lambda_i \alpha_i^2,$$

where $\mathbf{f} = (f(1), \ldots, f(N))^\mathsf{T}$ is the vector of the function values, $\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\mathsf{T}$ is an eigendecomposition of the graph laplacian $\mathcal{L}$, and $\boldsymbol{\alpha} = \mathbf{Q}^\mathsf{T}\mathbf{f}$ is the representation of the vector $\mathbf{f}$ in the eigenbasis. The assumption on the smoothness of the reward function with respect to the underlying graph is reflected by the small value of $S_\mathcal{G}(f)$ and therefore, the components of $\boldsymbol{\alpha}$ corresponding to the large eigenvalues should be small as well.

As a result, we can think of our setting as an N-arm bandit problem where N is possibly larger than the time horizon T and the mean reward $f(k)$ for each arm k satisfies the property that under a change of coordinates, the vector $\mathbf{f}$ of mean rewards has small components, i.e., there exists a known orthogonal matrix $\mathbf{U}$ such that $\boldsymbol{\alpha} = \mathbf{U}\mathbf{f}$ has a low norm. As a consequence, we can estimate $\boldsymbol{\alpha}$ using penalization corresponding to the large eigenvalues and to recover $\mathbf{f}$. Given a vector of weights $\boldsymbol{\alpha}$, we define its $\boldsymbol{\Lambda}$-norm as

$$\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}} \triangleq \sqrt{\sum_{i=1}^N \lambda_i \alpha_i^2} = \sqrt{\boldsymbol{\alpha}^\mathsf{T}\boldsymbol{\Lambda}\boldsymbol{\alpha}}. \tag{1}$$

This norm is closely related to the smoothness of the function and we use it later in our algorithms by regularization which enforces small $\boldsymbol{\Lambda}$-norm of $\boldsymbol{\alpha}$.

4.2 Effective dimension

In order to present and analyze our algorithms, we use a notion of *effective dimension* denoted by (lower case) d. While we introduced a slightly different version of the effective dimension for spectral bandits previously (Valko et al., 2014), we now present an improved definition. This new definition of effective dimension enables us to prove tighter regret bounds for our algorithms. In the rest of the paper, we refer to the old definition of the effective dimension, introduced by Valko et al. (2014), as d_old. We keep using capital D to denote the ambient dimension (the number of features). Intuitively, the effective dimension is a proxy for the number of relevant dimensions. We first provide a formal definition and then discuss its properties, including $d < d_\text{old} \ll D$.

In general, we assume there exists a diagonal matrix $\mathbf{\Lambda}$ with the entries $0 < \lambda = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$ and a set of N vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N \in \mathbb{R}^N$ such that $\|\mathbf{x}_i\|_2 \leq 1$ for all i. Moreover, since $\mathbf{Q}$ is an orthonormal matrix, $\|\mathbf{x}_i\|_2 = 1$. Finally, since the first eigenvalue of a graph Laplacian is always zero, $\lambda_1^{\mathcal{L}} = 0$, we use $\mathbf{\Lambda} = \mathbf{\Lambda}_{\mathcal{L}} + \lambda\mathbf{I}$, in order to have $\lambda_1 = \lambda > 0$.

Definition 1 *The **effective dimension** d is defined as*

$$d \triangleq \left\lceil \frac{\max \log \prod_{i=1}^{N} \left(1 + \frac{t_i}{\lambda_i}\right)}{\log\left(1 + \frac{T}{K\lambda}\right)} \right\rceil,$$

where the maximum is taken over all possible non-negative real numbers $\{t_1, \ldots, t_N\}$, such that $\sum_{i=1}^N t_i = T$ and K is the number of zero eigenvalues of $\mathbf{\Lambda}_{\mathcal{L}}$. K is also the number of components of G.

Remark 2 *Note that if we first upper bound every $1/\lambda_i$ in the numerator by $1/\lambda$ then the maximum is acquired for t_i equal to T/N. Therefore, the right-hand side of the definition is bounded from above by $D = N$. This means that d is upper bounded by D. Later we show that in many practical situations, d is much smaller than D.*

For the comparison, we show the previous definition of the effective dimension (Valko et al., 2014) and from now we call it *old effective dimension* denoted by d_{old}.

Definition 3 (old effective dimension, Valko et al., 2014) *Let the **old effective dimension** d_{old} be the largest $d_{old} \in [N]$ such that*

$$(d_{old} - 1)\lambda_{d_{old}} \leq \frac{T}{\log(1 + T/\lambda)}.$$

Remark 4 *Note that from Lemma 5 and Lemma 6 by Valko et al. (2014), we see that the relation between the old and new definition of the effective dimension is: $d \leq 2d_{old}$. As we show later, the bounds using the effective dimension scale either with d or with $2d_{old}$. Moreover, we show that d is usually much smaller than $2d_{old}$ and therefore using the new definition of the effective dimension brings an improvement to the bound.*

The effective dimension d is small when the coefficients λ_i grow rapidly above T. This is the case when the dimension of the space D is much larger than T, such as in graphs from social networks with a very large number of nodes N. In contrast, when the coefficients λ_i are all small (if the graph is sparse, all eigenvalues of Laplacian are small) then d may be of the order of T. That would make the regret bounds useless.

The actual form of Definition 1 comes from Lemma 24 and becomes apparent in Section 6. The dependence of the effective dimension on T comes from the fact that d is related to the number of "non-negligible" dimensions characterizing the space where the solution to the penalized least-squares may lie, since this solution is basically constrained to an ellipsoid defined by the inverse of the eigenvalues. This ellipsoid is wide in the directions corresponding to the small eigenvalues and narrow in the directions corresponding to the large ones. After playing an action, the confidence ellipsoid shrinks in the directions of the action. Therefore, exploring in a direction where the ellipsoid is wide can reduce the volume of the ellipsoid much more than exploring in a direction where the ellipsoid is narrow. In fact, for a small T, the axes of the ellipsoid corresponding to the large eigenvalues of $\mathcal{L}$ are negligible. Consequently, d is related to the metric dimension of this ellipsoid. Therefore, when T tends to infinity, then all directions matter, thus the solution can be anywhere in a (bounded) space of dimension N. On the contrary, for a smaller T, the ellipsoid possesses a smaller number of "non-negligible" dimensions.

4.2.1 The computation of the effective dimension

All of the algorithms that we propose need to know the value of the effective dimension in order to leverage the structure of the problem. Therefore, it is necessary to compute it beforehand. when computing the effective dimension, we proceed in two steps:

1. Finding an N-tuple $(t_1, \ldots, t_N)$ which maximizes the expression from the definition of the effective dimension.
2. Plugging the N-tuple to the definition of the effective dimension.

We now focus on the first step. The following lemma gives us an efficient way to determine the N-tuple

Lemma 5 *Let $\omega \in [N]$ be the largest integer such that*

$$\frac{\sum_{i=1}^{\omega} \lambda_i}{\omega} + \frac{T}{\omega} - \lambda_\omega > 0,$$

then $t_1, \ldots, t_N$ that maximize the expression in the definition of the effective dimension are in the following form,

$$\begin{aligned} t_i &= \frac{\sum_{i=1}^{\omega} \lambda_i}{\omega} + \frac{T}{\omega} - \lambda_i \qquad && \text{for } i = 1, \ldots, \omega, \\ t_i &= 0 && \text{for } i = \omega + 1, \ldots, N. \end{aligned}$$

Proof First of all, we use the fact that logarithm is an increasing function and that the N-tuple which maximizes the expression is invariant to a multiplication of the expression by a constant,

$$\arg\max \log \prod_{i=1}^{N} \left(1 + \frac{t_i}{\lambda_i}\right) = \arg\max \prod_{i=1}^{N} \left(1 + \frac{t_i}{\lambda_i}\right) = \arg\max \prod_{i=1}^{N} (\lambda_i + t_i).$$

The last expression is easy to maximize since we know that for any $\Delta \geq \delta \geq 0$ and for any real number a we have

$$\begin{aligned} 0 &\leq \Delta^2 - \delta^2 \\ a^2 - \Delta^2 &\leq a^2 - \delta^2 \\ (a - \Delta)(a + \Delta) &\leq (a - \delta)(a + \delta). \end{aligned}$$

Therefore, if we take any two terms $(\lambda_i + t_i)$ and $(\lambda_j + t_j)$ from the expression which we are maximizing, we can potentially increase their product simply by balancing them,

$$t_i^{\text{new}} \triangleq \frac{\lambda_i + \lambda_j + t_i + t_j}{2} - \lambda_i$$
$$t_j^{\text{new}} \triangleq \frac{\lambda_i + \lambda_j + t_i + t_j}{2} - \lambda_j.$$

However, we still have to take into consideration that every t_i has to be positive. Therefore, if, for example, t_j^{new} is negative, we can simply set

$$t_i^{\text{new}} \triangleq t_i + t_j$$
$$t_j^{\text{new}} \triangleq 0.$$

We apply this argument to the expression we are trying to maximize to obtain the statement of the lemma. ■

The second part is straightforward. To avoid computational difficulties of multiplying N numbers, we use properties of logarithm to get

$$d = \left\lceil \frac{\max \log \prod_{i=1}^{N} \left(1 + \frac{t_i}{\lambda_i}\right)}{\log\left(1 + \frac{T}{K\lambda}\right)} \right\rceil = \left\lceil \frac{\max \sum_{i=1}^{N} \log \left(1 + \frac{t_i}{\lambda_i}\right)}{\log\left(1 + \frac{T}{K\lambda}\right)} \right\rceil.$$

Knowing an N-tuple which maximizes the expression, we simply plug it in and obtain the value of the effective dimension.

4.2.2 The old vs. new definition of the effective dimension

As we mentioned in Remark 4, our new effective dimension is always upperbounded by $2d_{\text{old}}$. In this section, we show that the gap between d and $2d_{\text{old}}$ can be significant. We demonstrate on the graphs constructed for several real-world datasets and also on several random graphs.

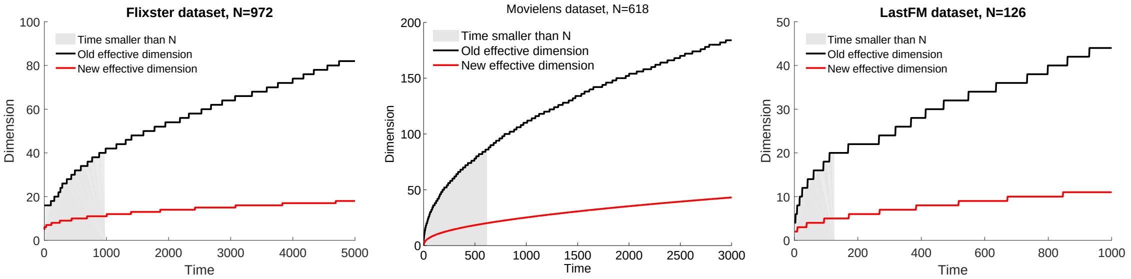


Figure 2: Difference between d and $2d_{\text{old}}$ for real world datasets. From left to right: Flixster dataset with $N = 972$, Movielens dataset with $N = 618$, and LastFM dataset with $N = 804$.

Figures 2 and 3 show how d behaves compared to $2d_{\text{old}}$ on the generated and the real Flixster, Movielens, and LastFM network graphs.[1] We use some of them for the experiments

[1] We set $\mathbf{\Lambda}$ to $\mathbf{\Lambda}_{\mathcal{L}} + \lambda\mathbf{I}$ with $\lambda = 0.1$, where $\mathbf{\Lambda}_{\mathcal{L}}$ is the graph Laplacian of the respective graph.

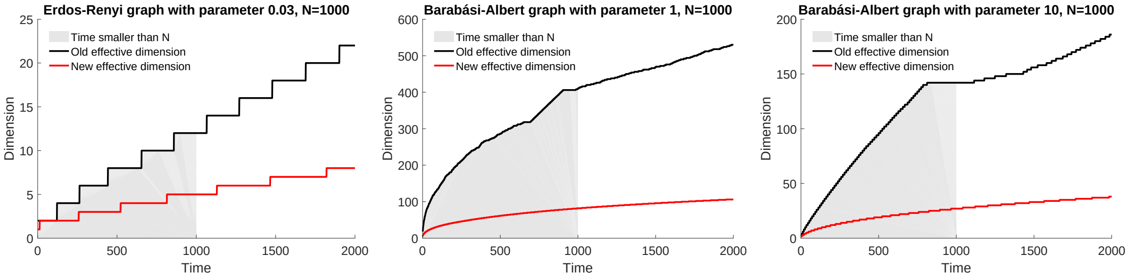


Figure 3: Difference between d and $2d_{\mathrm{old}}$ for random graphs on $N = 1000$ nodes. From left to right: Erdős-Renyi graph with the probability 0.03 of an edge, Barabási-Albert graph with one edge per added node, Barabási-Albert graph with ten edges per added node.

in Section 7. The figures clearly demonstrate the gap between d and $2d_{\mathrm{old}}$ while both of the quantities are much smaller then D. In fact, effective dimension d is much smaller than D even for $T > N$ (Figures 2 and 3). Therefore, spectral bandits can be used even for $T > N$ while maintaining the advantage of better regret bounds compared to the linear bandit algorithms.

4.3 Lower bound

In this section, we show a lower bound for the spectral setting. More precisely, for each possible value of effective dimension d and time horizon T, we show the existence of a “hard” problem with a lower bound of $\Omega(\sqrt{dT})$. We prove the theorem by reducing a carefully selected problem to a multi-arm bandit problem with d arms and using the following lower bound for it.

Theorem 6 (Auer et al., 2002) *For any number of actions $K \geq 2$ and for any time horizon T, there exists a distribution over the assignment of Bernoulli rewards such that the expected regret of any algorithm (where the expectation is taken with respect to both the randomization over rewards and the algorithms internal randomization) is at least*

$$R_T \geq \frac{1}{20} \min\left\{\sqrt{KT}, T\right\}.$$

Theorem 6 can also be proved without the randomization device. The constant 1/20 in the lower bound above can be improved into 1/8 (Cesa-Bianchi and Lugosi, 2006, Theorem 6.11). We now state a lower bound for spectral bandits, featuring the effective dimension d.

Theorem 7 *For any T and d, there exists a problem with effective dimension d and time horizon T such that the expected regret of any algorithm is of $\Omega(\sqrt{dT})$.*

Proof We define a problem with the set of actions consisting of $K = d$ blocks. Each block is a complete graph K_{M_T} on M_T vertices. Moreover, all weights of the edges inside a component are equal to one. We define M_T as a T-dependent constant such that the effective dimension of the problem d is exactly K. We specify the precise value of M_T later.

On top of the structure described above, we choose a reward function with smoothness 0, i.e., a constant on each of the components of the graph. In fact, even knowing that the reward function is constant on individual components, this problem is as difficult as the multi-arm bandit problem with K arms. Therefore, the lower bound of $\Omega(\sqrt{KT})$ of the K-arm bandit problem applies to our setting too. Consequently, we have the lower bound of $\Omega(\sqrt{dT})$, since $d = K$.

The last part of the proof is to show that $d = K$ and therefore, we have to specify value of M_T. The graph consists of K blocks consisting M_T vertices each. Therefore, the graph Laplacian is the following matrix

$$L = \begin{pmatrix} \boxed{\begin{matrix} (M_T-1) & -1 & \dots & -1 \\ -1 & \ddots & & \vdots \\ \vdots & & \ddots & -1 \\ -1 & \dots & -1 & (M_T-1) \end{matrix}} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \boxed{\begin{matrix} (M_T-1) & -1 & \dots & -1 \\ -1 & \ddots & & \vdots \\ \vdots & & \ddots & -1 \\ -1 & \dots & -1 & (M_T-1) \end{matrix}} \end{pmatrix}.$$

Now we compute eigenvalues of L to obtain

$$L \xrightarrow{\text{eigenvalues}} \overbrace{0, \dots, 0}^{K}, \overbrace{M_T, \dots, M_T}^{(M_T-1)K}.$$

We plug the above eigenvalues to the definition of the effective dimension and to set the value of M_T to obtain

$$d = \left\lceil \frac{\max \log \prod_{i=1}^{K} \left(1 + \frac{t_i}{\lambda}\right) \prod_{i=K+1}^{KM_T} \left(1 + \frac{t_i}{\lambda + M_T}\right)}{\log\left(1 + \frac{T}{K\lambda}\right)} \right\rceil.$$

By setting $M_T \geq T/K$, we have that the maximum in the definition is obtained for $t_1 = \dots = t_K = T/K$ and $t_{K+1} = \dots = t_{KM_T} = 0$. Therefore,

$$d = \left\lceil \frac{\log \prod_{i=1}^{K} \left(1 + \frac{T}{K\lambda}\right)}{\log\left(1 + \frac{T}{K\lambda}\right)} \right\rceil = \left\lceil \frac{\sum_{i=1}^{K} \log\left(1 + \frac{T}{K\lambda}\right)}{\log\left(1 + \frac{T}{K\lambda}\right)} \right\rceil = K.$$

This means that our problem is at least as difficult as the multi-arm bandit problem with $K = d$ arms and therefore, the lower bound for K-arm bandits (Theorem 6) applies. ■

5. Algorithms

In this section, we introduce the algorithms for spectral bandits: `SpectralUCB`, `SpectralTS`, and `SpectralEliminator`. For each algorithm, we state the regret bound and later we

Algorithm 1 SpectralUCB

```
Input:
    $N$: number of actions
    $T$: number of rounds
    $\{\boldsymbol{\Lambda}_{\mathcal{L}}, \mathbf{Q}\}$: spectral basis of a graph Laplacian $\mathcal{L}$
    $\lambda$, $\delta$: regularization and confidence parameters
    $R$, $C$: upper bounds on the noise and $\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}$
Initialization:
    $\mathbf{V}_1 \leftarrow \boldsymbol{\Lambda} \leftarrow \boldsymbol{\Lambda}_{\mathcal{L}} + \lambda \mathbf{I}$
    $\widehat{\boldsymbol{\alpha}}_1 \leftarrow 0_N$
    $d \leftarrow \lceil (\max \log \prod_{i=1}^{N} (1 + t_i/\lambda_i)) / \log(1 + T/(K\lambda)) \rceil$    (Definition 1)
    $c \leftarrow R\sqrt{2d \log(1 + T/(K\lambda)) + 8 \log(1/\delta)} + C$
Run:
for $t = 1$ to $T$ do
    Choose the node $a_t$ ($a_t$-th row of $\mathbf{Q}$): $a_t \leftarrow \arg\max_a \left(\mathbf{x}_a^{\mathsf{T}} \widehat{\boldsymbol{\alpha}}_t + c\|\mathbf{x}_a\|_{\mathbf{V}_t^{-1}}\right)$
    Observe a noisy reward $r_t \leftarrow \mathbf{x}_{a_t}^{\mathsf{T}} \boldsymbol{\alpha} + \varepsilon_t$
    Update the basis coefficients $\widehat{\boldsymbol{\alpha}}$:
        $\mathbf{V}_{t+1} \leftarrow \mathbf{V}_t + \mathbf{x}_{a_t} \mathbf{x}_{a_t}^{\mathsf{T}}$
        $\widehat{\boldsymbol{\alpha}}_{t+1} \leftarrow \mathbf{V}_{t+1}^{-1} \sum_{s=1}^{t} \mathbf{x}_{a_s} r_s$
end for
```

discuss the computational advantages and compare the theoretical regret bounds of the algorithms with the lower bound provided in the previous section. Full proofs are given in Section 6.

5.1 SpectralUCB

We first present SpectralUCB (Algorithm 1) which is based on LinUCB (Li et al., 2010) and uses the *spectral penalty* (1) in its least-square estimate. Here, we consider a *regularized* least-squares estimate $\widehat{\boldsymbol{\alpha}}_t$ of the form

$$\widehat{\boldsymbol{\alpha}}_t \triangleq \operatorname*{arg\,min}_{\mathbf{w} \in \mathbb{R}^N} \left(\sum_{s=1}^{t} \left[\mathbf{x}_{a_s}^{\mathsf{T}} \mathbf{w} - r_{a_s} \right]^2 + \|\mathbf{w}\|_{\boldsymbol{\Lambda}}^2 \right).$$

A key part of the algorithm is to define the $c_t\|\mathbf{x}\|_{\mathbf{V}_t^{-1}}$ confidence widths for the prediction of the rewards and consequently the upper confidence bounds (UCBs). We take advantage of our analysis (Section 6.3) to define c_t based on the effective dimension d which is tailored to our setting. This way we also avoid the computation of the determinant (see Section 6). The following theorem characterizes the performance of SpectralUCB and bounds the regret as a function of effective dimension d.

Theorem 8 *Let d be the effective dimension and λ be the minimum eigenvalue of $\boldsymbol{\Lambda}$. If $\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}} \leq C$ and for all $\mathbf{x}_a$, $\mathbf{x}_a^{\mathsf{T}} \boldsymbol{\alpha} \in [-1, 1]$, then the cumulative regret of* SpectralUCB is with probability at least $1-\delta$ bounded as

$$R_T \leq \left(2R\sqrt{2d\log\left(1+\frac{T}{K\lambda}\right)+8\log\left(\frac{1}{\delta}\right)}+2C+2\right)\sqrt{2dT\log\left(1+\frac{T}{K\lambda}\right)}$$
$$\leq \widetilde{\mathcal{O}}\left(d\sqrt{T}\right).$$

Remark 9 *The constant C needs to be such that $\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}} \leq C$. If we set C too small, the true $\boldsymbol{\alpha}$ will lie outside of the region and far from $\widehat{\boldsymbol{\alpha}}_t$, causing the algorithm to underperform. Alternatively, C can be time-dependent, e.g., $C_t \triangleq \log t$. In such case, we do not need to know an upper bound on $\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}$ in advance, but our regret bound would only hold after some t, in particular when $C_t \geq \|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}$.*

We provide the proof of Theorem 8 in Section 6 and examine the performance of our SpectralUCB experimentally in Section 7. The $d\sqrt{T}$ result of Theorem 8 is to be compared with the standard linear bandits, where LinUCB is the algorithm often used in practice (Li et al., 2010), achieving $D\sqrt{T}$ cumulative regret. As mentioned above and demonstrated in Figures 2 and 3, in the $T < N$ regime we can expect $d \ll D = N$ and obtain an improved performance.

5.2 SpectralTS

The second algorithm presented in this paper is SpectralTS which is based on LinearTS, analyzed by Agrawal and Goyal (2013b), and uses Thompson sampling to decide which arm to play. Specifically, we represent our current knowledge about $\boldsymbol{\alpha}$ as a normal distribution $\mathcal{N}(\widehat{\boldsymbol{\alpha}}_t, v^2\mathbf{V}_t^{-1})$, where $\widehat{\boldsymbol{\alpha}}_t$ is our actual approximation of the unknown vector $\boldsymbol{\alpha}$ and $v^2\mathbf{V}_t^{-1}$ reflects our uncertainty about it. As mentioned before, we assume that the reward function is a linear combination of eigenvectors of graph Laplacian $\mathcal{L}$ with large coefficients corresponding to the eigenvectors with small eigenvalues. We encode this assumption into our initial confidence ellipsoid by setting $\mathbf{V}_1 \triangleq \boldsymbol{\Lambda} \triangleq \boldsymbol{\Lambda}_{\mathcal{L}} + \lambda\mathbf{I}$, where λ is again a regularization parameter.

In every round t, we generate a sample $\widetilde{\boldsymbol{\alpha}}_t$ from the distribution $\mathcal{N}(\widehat{\boldsymbol{\alpha}}_t, v^2\mathbf{V}_t^{-1})$, choose an arm a_t which maximizes $\mathbf{x}_i^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t$, and receive a reward. Afterwards, we update our estimate of $\boldsymbol{\alpha}$ and the confidence of it, i.e., we compute $\widehat{\boldsymbol{\alpha}}_{t+1}$ and $\mathbf{V}_{t+1}$,

$$\mathbf{V}_{t+1} = \mathbf{V}_t + \mathbf{x}_{a_t}\mathbf{x}_{a_t}^\mathsf{T} \qquad \text{and} \qquad \widehat{\boldsymbol{\alpha}}_{t+1} = \mathbf{V}_{t+1}^{-1}\left(\sum_{s=1}^{t}\mathbf{x}_{a_s}r_s\right).$$

Remark 10 *Since* TS is a Bayesian approach, it requires a prior to run and we choose it here to be a Gaussian. However, this does not pose any assumption whatsoever about the actual data both for the algorithm and the analysis. The only assumptions we make about the data are: (a) that the mean payoff is linear in the features, (b) that the noise is sub-Gaussian, and (c) that we know a bound on the Laplacian norm of the mean reward function. We provide a **frequentist** bound on the regret (and not an average over the prior) which is a much stronger worst-case result.

Algorithm 2 SpectralTS

```
Input:
    $N$: number of actions
    $T$: number of rounds
    $\{\boldsymbol{\Lambda}_{\mathcal{L}}, \mathbf{Q}\}$: spectral basis of a graph Laplacian $\mathcal{L}$
    $\lambda$, $\delta$: regularization and confidence parameters
    $R$, $C$: upper bounds on the noise and $\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}$
Initialization:
    $\mathbf{V}_1 \leftarrow \boldsymbol{\Lambda} \leftarrow \boldsymbol{\Lambda}_{\mathcal{L}} + \lambda \mathbf{I}_N$
    $\widehat{\boldsymbol{\alpha}}_1 \leftarrow 0_N$
    $d \leftarrow \lceil (\max \log \prod_{i=1}^{N}(1 + t_i/\lambda_i)) / \log(1 + T/(K\lambda)) \rceil$    (Definition 1)
    $v \leftarrow R\sqrt{3d \log(1/\delta + T/(\delta\lambda K))} + C$
Run:
for $t = 1$ to $T$ do
    Sample $\widetilde{\boldsymbol{\alpha}}_t \sim \mathcal{N}(\widehat{\boldsymbol{\alpha}}_t, v^2 \mathbf{V}_t^{-1})$
    Choose the node $a_t$ ($a_t$-th row of $\mathbf{Q}$): $a_t \leftarrow \arg\max_a \mathbf{x}_a^{\mathsf{T}} \widetilde{\boldsymbol{\alpha}}$
    Observe a noisy reward $r_t \leftarrow \mathbf{x}_{a_t}^{\mathsf{T}} \boldsymbol{\alpha} + \varepsilon_t$
    Update the basis coefficients $\widehat{\boldsymbol{\alpha}}$:
        $\mathbf{V}_{t+1} \leftarrow \mathbf{V}_t + \mathbf{x}_{a_t} \mathbf{x}_{a_t}^{\mathsf{T}}$
        $\widehat{\boldsymbol{\alpha}}_{t+1} \leftarrow \mathbf{V}_{t+1}^{-1} \sum_{s=1}^{t} \mathbf{x}_{a_s} r_s$
end for
```

The following theorem upperbounds the cumulative regret of SpectralTS in terms of the effective dimension.

Theorem 11 *Let d be the effective dimension and λ be the minimum eigenvalue of $\boldsymbol{\Lambda}$. If $\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}} \leq C$ and for all $\mathbf{x}_a$, $\mathbf{x}_a^{\mathsf{T}}\boldsymbol{\alpha} \in [-1, 1]$, then the cumulative regret of SpectralTS is with probability at least $1 - \delta$ bounded as*

$$R_T \leq \frac{11g}{p}\sqrt{\frac{2+2\lambda}{\lambda} dT \log\left(1 + \frac{T}{K\lambda}\right)} + \frac{1}{T} + \frac{g}{p}\left(\frac{11}{\sqrt{\lambda}} + 2\right)\sqrt{2T \log\left(\frac{2}{\delta}\right)},$$

where $p = 1/(4e\sqrt{\pi})$ and

$$g = \sqrt{4 \log(TN)}\left(R\sqrt{3d \log\left(\frac{1}{\delta} + \frac{T}{\delta\lambda K}\right)} + C\right) + R\sqrt{d \log\left(\frac{T^2}{\delta} + \frac{T^3}{\delta\lambda K}\right)} + C.$$

Remark 12 *Substituting g and p, we see that the regret bound scales as $d\sqrt{T \log N}$. Note that $N = D$ could be exponential in d and therefore we need to consider factor $\sqrt{\log N}$ in our bound. On the other hand, if N is indeed exponential in d, then our algorithm scales with $\log D \sqrt{T \log D} = \log(D)^{3/2}\sqrt{T}$ which is even better.*

5.3 SpectralEliminator

Algorithm 3 SpectralEliminator

```
Input:
    $N$: number of nodes
    $T$: number of pulls
    $\{\boldsymbol{\Lambda}_\mathcal{L}, \mathbf{Q}\}$: spectral basis of a graph Laplacian $\mathcal{L}$
    $\lambda$: regularization parameter
    $\beta, \{t_j\}_j^J$: parameters of the elimination and phases where $J = \lfloor \log_2 T \rfloor + 1$
    $A_1 \leftarrow \{\mathbf{x}_1, \dots, \mathbf{x}_K\}$
for $j = 1$ to $J$ do
    $\mathbf{V}_{t_j} \leftarrow \boldsymbol{\Lambda}_\mathcal{L} + \lambda\mathbf{I}$
    for $t = t_j$ to $\min(t_{j+1} - 1, T)$ do
        Play available arm $a_t$ ($\mathbf{x}_{a_t} \in A_j$) with the largest width and observe $r_t$:
            $a_t \leftarrow \arg\max_{a|\mathbf{x}_a \in A_j} \|\mathbf{x}_a\|_{\mathbf{V}_t^{-1}}$
        $\mathbf{V}_{t+1} \leftarrow \mathbf{V}_t + \mathbf{x}_{a_t}\mathbf{x}_{a_t}^\mathsf{T}$
    end for
    Eliminate the arms that are not promising:
        $\widehat{\boldsymbol{\alpha}}_{j+1} \leftarrow \mathbf{V}_{t+1}^{-1}[\mathbf{x}_{t_j}, \dots, \mathbf{x}_t][r_{t_j}, \dots, r_t]^\mathsf{T}$
        $A_{j+1} \leftarrow \left\{\mathbf{x} \in A_j, \langle \widehat{\boldsymbol{\alpha}}_{j+1}, \mathbf{x}\rangle + \|\mathbf{x}\|_{\mathbf{V}_{t+1}^{-1}}\beta \geq \max_{\mathbf{x} \in A_j} \left[\langle \widehat{\boldsymbol{\alpha}}_{j+1}, \mathbf{x}\rangle - \|\mathbf{x}\|_{\mathbf{V}_{t+1}^{-1}}\beta\right]\right\}$
end for
```

It is known that the available upper bound for LinUCB, LinearTS, or OFUL is not tight for linear bandits with a *finite* number of arms in terms of dimension D. On the other hand, the algorithms SupLinRel or SupLinUCB achieve the optimal $\sqrt{DT}$ regret. In the following, we similarly provide an algorithm that also scales better with d and achieves a $\sqrt{dT}$ regret. The algorithm is called SpectralEliminator (Algorithm 3) and works in phases, eliminating the arms that are not promising. The phases are defined by the time indexes $t_1 = 1 \leq t_2 \leq \dots$ and depend on some parameter β. The algorithm is in a spirit similar to ImprovedUCB by Auer and Ortner (2010). As a special case and as a side result of independent interest, we give also the LinearEliminator algorithm. LinearEliminator achieves the optimal $\sqrt{DT}$ regret and uses *adaptive confidence intervals*, unlike SupLinRel or SupLinUCB, that use data-agnostic confidence intervals of the form 2^{-u} for $u \in \mathbb{N}_0$.

In the following theorem, we characterize the performance of SpectralEliminator and show that the upper bound on its regret has a $\sqrt{d}$ improvement over SpectralUCB and SpectralTS.

Theorem 13 *Choose the phases' starts as* $t_j \triangleq 2^{j-1}$. *Assume all rewards are in* $[0, 1]$ *and* $\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}} \leq C$. *For any* $\delta > 0$, *with probability at least* $1 - \delta$, *the cumulative regret of* SpectralEliminator run with parameter $\beta \triangleq R\sqrt{\log(2K(1 + \log_2 T)/\delta)} + C$ is bounded as

$$R_T \leq 2 + 8\left(R\sqrt{2\log\left(\frac{2K(1 + \log_2 T)}{\delta}\right)} + C + \frac{1}{2}\right)\sqrt{2dT\log_2(T)\log\left(1 + \frac{T}{\lambda K}\right)}.$$

5.4 Scalability and computational complexity

There are three main computational issues to address in order to make proposed algorithms scalable: the computation of N UCBs (applies to SpectralUCB), the matrix inversion, and obtaining the eigenbasis which serves as an input to any of the algorithms. First, to speed up the computation of N UCBs (that in general takes N^3 time) in each round, we use lazy updates (Desautels et al., 2012) which maintains a sorted queue of UCBs and using the fact that the UCB for every arm can only decrease after the update. Therefore, the algorithm does not need to update all UCBs in each round. In practice, lazy updates lead to light-speed gains. This issue does not apply to SpectralTS since we only need to sample $\widetilde{\boldsymbol{\alpha}}$ which can be done in N^2 time and find a maximum of $\mathbf{x}_i^\mathsf{T}\widetilde{\boldsymbol{\alpha}}$ which can be also done in N^2 time. In general, the computational complexity of sampling in SpectralTS is better than the complexity of computing the N UCBs in SpectralUCB. However, using lazy updates can significantly speed up SpectralUCB up to the point that SpectralUCB is comparable to SpectralTS.

Second, all of the proposed algorithms need to compute inverse if $N \times N$ matrix in each round which is costly. However, we can use Sherman-Morrison formula to invert the matrix iteratively and thus speed up the inversion since the matrix changes only by adding a rank-one matrix from one round to the next one.

Finally, while an eigendecomposition of a general matrix is computationally difficult, Laplacians are symmetric diagonally dominant (SDD). This enables us to use fast SDD solvers as CMG by Koutis et al. (2011). Furthermore, using CMG we can find good approximations to the first L eigenvectors in $\mathcal{O}(Lm\log m)$ time, where m is the number of edges in the graph (e.g., $m = 10N$ for Flixter data, Section 7.5). CMG can easily work with N in millions. In general, we have $L = N$ but from our experience, a smooth reward function can be often approximated by *dozens* of eigenvectors. In fact, L can be considered as an upper bound on the number of eigenvectors we actually need. Furthermore, by choosing small L we not only reduce the complexity of an eigendecomposition but also the complexity of the least-square problem that is solved in each round.

Choosing a small L can significantly reduce the computation but it is important to choose L large enough so that still less than L eigenvectors are enough. This way, the problem that we solve remains relevant and our analysis applies. In short, the problem cannot be solved trivially by choosing first k relevant eigenvectors because k is unknown. Therefore, in practice, we choose the largest L such that our method is able to run. In Section 7.3, we demonstrate that we can obtain good results with relatively small L.

6. Analysis

We are now ready to prove regret bounds for all algorithms. First, we show some general preliminary results (Section 6.1). Then, we present several auxiliary lemmas concerning confidence ellipsoid of the estimate (Section 6.2) and effective dimension (Section 6.3). Using these results we upperbound the regrets of SpectralUCB (Section 6.4), SpectralTS (Section 6.5), and SpectralEliminator (Section 6.6).

6.1 Preliminaries

The first lemma is a standard anti-concentration inequality for a Gaussian random variable.

Lemma 14 *For a Gaussian distributed random variable Z with mean m and variance σ^2, for any $z \geq 1$,*

$$\frac{1}{2\sqrt{\pi}z}e^{-\frac{z^2}{2}} \leq \mathbb{P}\left(|Z-m| > \sigma z\right) \leq \frac{1}{\sqrt{\pi}z}e^{-\frac{z^2}{2}}.$$

Multiple applications of Sylvester's determinant theorem gives our second preliminary lemma.

Lemma 15 *Let $\mathbf{V}_t = \mathbf{\Lambda} + \sum_{s=1}^{t-1} \mathbf{x}_s\mathbf{x}_s^\mathsf{T}$, then*

$$\log\frac{|\mathbf{V}_t|}{|\mathbf{\Lambda}|} = \sum_{s=1}^{t-1}\log\left(1 + \|\mathbf{x}_s\|^2_{\mathbf{V}_s^{-1}}\right).$$

Third lemma says that adding a rank-one matrix to a symmetric positive semi-definite matrix implies the following Löwner ordering for their inverses.

Lemma 16 *For any symmetric, positive semi-definite matrix $\mathbf{X}$, and any vectors $\mathbf{u}$ and $\mathbf{y}$,*

$$\mathbf{y}^\mathsf{T}(\mathbf{X}+\mathbf{u}\mathbf{u}^\mathsf{T})^{-1}\mathbf{y} \leq \mathbf{y}^\mathsf{T}\mathbf{X}^{-1}\mathbf{y}.$$

Proof Using Sherman-Morrison formula and the fact that inverse of a symmetric matrix is again symmetric, we have

$$\begin{aligned} -\frac{\left(\mathbf{u}^\mathsf{T}\mathbf{X}^{-1}\mathbf{y}\right)^\mathsf{T}\left(\mathbf{u}^\mathsf{T}\mathbf{X}^{-1}\mathbf{y}\right)}{1+\mathbf{u}^\mathsf{T}\mathbf{X}^{-1}\mathbf{u}} &\leq 0 \\ \mathbf{y}^\mathsf{T}\left(\mathbf{X}^{-1} - \frac{\mathbf{X}^{-1}\mathbf{u}\mathbf{u}^\mathsf{T}\mathbf{X}^{-1}}{1+\mathbf{u}^\mathsf{T}\mathbf{X}^{-1}\mathbf{u}}\right)\mathbf{y} &\leq \mathbf{y}^\mathsf{T}\mathbf{X}^{-1}\mathbf{y} \\ \mathbf{y}^\mathsf{T}\left(\mathbf{X}+\mathbf{u}\mathbf{u}^\mathsf{T}\right)^{-1}\mathbf{y} &\leq \mathbf{y}^\mathsf{T}\mathbf{X}^{-1}\mathbf{y}. \end{aligned}$$

■

Corollary 17 *Let $\mathbf{V}_t \triangleq \mathbf{\Lambda} + \sum_{s=1}^{t-1}\mathbf{x}_s\mathbf{x}_s^\mathsf{T}$. Then for any vector $\mathbf{x}$ and for any positive integers t_1 and t_2 satisfying $t_1 \leq t_2$,*

$$||\mathbf{x}||_{\mathbf{V}_{t_1}^{-1}} \geq ||\mathbf{x}||_{\mathbf{V}_{t_2}^{-1}}.$$

6.2 Confidence ellipsoid

We restate the two lemmas by Abbasi-Yadkori et al. (2011) for convenience.

Lemma 18 (Abbasi-Yadkori et al., 2011, Lemma 9) *Let $\mathbf{V}_t \triangleq \mathbf{\Lambda} + \sum_{s=1}^{t-1}\mathbf{x}_s\mathbf{x}_s^\mathsf{T}$ and define $\boldsymbol{\xi}_t \triangleq \sum_{s=1}^{t-1}\varepsilon_s\mathbf{x}_s$. With probability at least $1-\delta$, $\forall t \geq 1$,*

$$\|\boldsymbol{\xi}_t\|^2_{\mathbf{V}_t^{-1}} \leq 2R^2\log\left(\frac{|\mathbf{V}_t|^{1/2}}{\delta|\mathbf{\Lambda}|^{1/2}}\right).$$

Lemma 19 **(Abbasi-Yadkori et al., 2011, Lemma 11)** *For any round t, let us define* $\mathbf{V}_t \triangleq \mathbf{\Lambda} + \sum_{s=1}^{t-1} \mathbf{x}_s\mathbf{x}_s^\mathsf{T}$. *Then,*

$$\sum_{s=1}^{t} \min\left(1,\ \|\mathbf{x}_s\|^2_{\mathbf{V}_s^{-1}}\right) \le 2\log\frac{|\mathbf{V}_{t+1}|}{|\mathbf{\Lambda}|}.$$

The next lemma is a generalization of Theorem 2 by Abbasi-Yadkori et al. (2011) to the regularization with $\mathbf{\Lambda}$.

Lemma 20 *Let* $\mathbf{V}_t \triangleq \mathbf{\Lambda} + \sum_{s=1}^{t-1} \mathbf{x}_s\mathbf{x}_s^\mathsf{T}$ *and* $\|\boldsymbol{\alpha}\|_{\mathbf{\Lambda}} \le C$. *With probability at least* $1-\delta$, *for any* $\mathbf{x}$ *and* $t \ge 1$,

$$|\mathbf{x}^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t - \mathbf{x}^\mathsf{T}\boldsymbol{\alpha}| \le \|\mathbf{x}\|_{\mathbf{V}_t^{-1}}\left(R\sqrt{2\log\left(\frac{|\mathbf{V}_t|^{1/2}}{\delta|\mathbf{\Lambda}|^{1/2}}\right)} + C\right).$$

Proof We have that

$$\begin{aligned}|\mathbf{x}^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t - \mathbf{x}^\mathsf{T}\boldsymbol{\alpha}| &= |\mathbf{x}^\mathsf{T}(-\mathbf{V}_t^{-1}\mathbf{\Lambda}\boldsymbol{\alpha} + \mathbf{V}_t^{-1}\boldsymbol{\xi}_t)| \le |\mathbf{x}^\mathsf{T}\mathbf{V}_t^{-1}\mathbf{\Lambda}\boldsymbol{\alpha}| + |\mathbf{x}^\mathsf{T}\mathbf{V}_t^{-1}\boldsymbol{\xi}_t| \\ &\le |\mathbf{x}^\mathsf{T}\mathbf{V}_t^{-\frac{1}{2}}\mathbf{V}_t^{-\frac{1}{2}}\mathbf{\Lambda}\boldsymbol{\alpha}| + |\mathbf{x}^\mathsf{T}\mathbf{V}_t^{-\frac{1}{2}}\mathbf{V}_t^{-\frac{1}{2}}\boldsymbol{\xi}_t| \le \|\mathbf{x}\|_{\mathbf{V}_t^{-1}}\left(\|\boldsymbol{\xi}_t\|_{\mathbf{V}_t^{-1}} + \|\mathbf{\Lambda}\boldsymbol{\alpha}\|_{\mathbf{V}_t^{-1}}\right),\end{aligned}$$

where we use Cauchy-Schwarz inequality in the last step. Now, we bound $\|\boldsymbol{\xi}_t\|_{\mathbf{V}_t^{-1}}$ by Lemma 18 and using Corollary 17 we bound $||\mathbf{\Lambda}\boldsymbol{\alpha}||_{V_t^{-1}}$ as

$$\|\mathbf{\Lambda}\boldsymbol{\alpha}\|_{\mathbf{V}_t^{-1}} \le \|\mathbf{\Lambda}\boldsymbol{\alpha}\|_{\mathbf{V}_1^{-1}} = \|\mathbf{\Lambda}\boldsymbol{\alpha}\|_{\mathbf{\Lambda}^{-1}} = \|\boldsymbol{\alpha}\|_{\mathbf{\Lambda}} \le C.$$

■

6.3 Effective dimension

In Section 6.2, we show that several quantities scale with $\log(|\mathbf{V}_t|/|\mathbf{\Lambda}|)$, which can be of the order of D. Therefore, in this part, we present the key ingredient of our analysis, based on the geometrical properties of determinants (Lemmas 22 and 23), to upperbound $\log(|\mathbf{V}_t|/|\mathbf{\Lambda}|)$ by a term that scales with d (Lemma 24). Not only this allows us to show that the regret bound scales with d, but it also helps us to avoid the computation of the determinants in Algorithm 1.

Lemma 21 *For any real positive-definite matrix A with only simple eigenvalue multiplicities and any vector* $\mathbf{x}$ *such that* $\|\mathbf{x}\|_2 \le 1$, *we have that the determinant* $|\mathbf{A} + \mathbf{x}\mathbf{x}^\mathsf{T}|$ *is maximized by a vector* $\mathbf{x}$ *which is aligned with an eigenvector of* $\mathbf{A}$.

Proof Using Sylvester's determinant theorem, we have that

$$|\mathbf{A} + \mathbf{x}\mathbf{x}^\mathsf{T}| = |\mathbf{A}||\mathbf{I} + \mathbf{A}^{-1}\mathbf{x}\mathbf{x}^\mathsf{T}| = |\mathbf{A}|(1 + \mathbf{x}^\mathsf{T}\mathbf{A}^{-1}\mathbf{x}).$$

From the spectral theorem, there exists an orthonormal matrix $\mathbf{U}$, the columns of which are the eigenvectors of $\mathbf{A}$, such that $\mathbf{A} = \mathbf{U}\mathbf{D}\mathbf{U}^\mathsf{T}$ with $\mathbf{D}$ being a diagonal matrix with the positive eigenvalues of $\mathbf{A}$ on the diagonal. Thus,

$$\max_{\|\mathbf{x}\|_2 \leq 1} \mathbf{x}^\mathsf{T}\mathbf{A}^{-1}\mathbf{x} = \max_{\|\mathbf{x}\|_2 \leq 1} \mathbf{x}^\mathsf{T}\mathbf{U}\mathbf{D}^{-1}\mathbf{U}^\mathsf{T}\mathbf{x} = \max_{\|\mathbf{y}\|_2 \leq 1} \mathbf{y}^\mathsf{T}\mathbf{D}^{-1}\mathbf{y},$$

since $\mathbf{U}$ is a bijection from $\{\mathbf{x}, \|\mathbf{x}\|_2 \leq 1\}$ to itself.

As there are no multiplicities, it is easy to see that the quadratic mapping $\mathbf{y} \mapsto \mathbf{y}^\mathsf{T}\mathbf{D}^{-1}\mathbf{y}$ is maximized (under the constraint $\|\mathbf{y}\|_2 \leq 1$) by a canonical vector $\mathbf{e}_I$ corresponding to the lowest diagonal entry I of $\mathbf{D}$. Thus the maximum of $\mathbf{x} \mapsto \mathbf{x}^\mathsf{T}\mathbf{A}^{-1}\mathbf{x}$ is reached for $\mathbf{U}\mathbf{e}_I$, which is the eigenvector of $\mathbf{A}$ corresponding to its lowest eigenvalue. ■

Lemma 22 *Let $\mathbf{\Lambda} \triangleq \operatorname{diag}(\lambda_1, \ldots, \lambda_N)$ be any diagonal matrix with strictly positive entries. For any vectors $(\mathbf{x}_s)_{1 \leq s < t}$ such that $\|\mathbf{x}_s\|_2 \leq 1$ for all $1 \leq s < t$, we have that the determinant $|\mathbf{V}_t|$ of $\mathbf{V}_t \triangleq \mathbf{\Lambda} + \sum_{s=1}^{t-1} \mathbf{x}_s\mathbf{x}_s^\mathsf{T}$ is maximized when all $\mathbf{x}_s$ are aligned with the axes.*

Proof Let us write $d(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}) \triangleq |\mathbf{V}_t|$ the determinant of $\mathbf{V}_t$. We want to characterize

$$\max_{\mathbf{x}_1, \ldots, \mathbf{x}_{t-1} : \|\mathbf{x}_s\|_2 \leq 1, \forall 1 \leq s < t} d\left(\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}\right).$$

For any $1 \leq i < t$, let us define

$$\mathbf{V}_{-i} \triangleq \mathbf{\Lambda} + \sum_{\substack{s=1 \\ s \neq i}}^{t-1} \mathbf{x}_s\mathbf{x}_s^\mathsf{T}.$$

We have that $\mathbf{V}_t = \mathbf{V}_{-i} + \mathbf{x}_i\mathbf{x}_i^\mathsf{T}$. Consider the case with only simple eigenvalue multiplicities. In this case, Lemma 21 implies that $\mathbf{x}_i \mapsto d(\mathbf{x}_1, \ldots, \mathbf{x}_i, \ldots, \mathbf{x}_{t-1})$ is maximized when $\mathbf{x}_i$ is aligned with an eigenvector of $\mathbf{V}_{-i}$. Thus all $\mathbf{x}_s$, for $1 \leq s < t$, are aligned with an eigenvector of $\mathbf{V}_{-i}$ and therefore also with an eigenvector of $\mathbf{V}_t$. Consequently, the eigenvectors of $\sum_{s=1}^{t-1} \mathbf{x}_s\mathbf{x}_s^\mathsf{T}$ are also aligned with $\mathbf{V}_t$. Since $\mathbf{\Lambda} = \mathbf{V}_t - \sum_{s=1}^{t-1} \mathbf{x}_s\mathbf{x}_s^\mathsf{T}$ and $\mathbf{\Lambda}$ is diagonal, we conclude that $\mathbf{V}_t$ is diagonal and all $\mathbf{x}_s$ are aligned with the canonical axes.

Now in the case of eigenvalue multiplicities, the maximum of $|\mathbf{V}_t|$ may be reached by several sets of vectors $\{(\mathbf{x}_s^m)_{1 \leq s < t}\}_m$ but for some $m^\star$, the set $(\mathbf{x}_s^{m^\star})_{1 \leq s < t}$ will be aligned with the axes. In order to see that, consider a perturbed matrix $\mathbf{V}_{-i}^\varepsilon$ by a random perturbation of amplitude at most ε, i.e. such that $\mathbf{V}_{-i}^\varepsilon \to \mathbf{V}_{-i}$ when $\varepsilon \to 0$. Since the perturbation is random, then the probability that $\mathbf{\Lambda}^\varepsilon$, as well as all other $\mathbf{V}_{-i}^\varepsilon$ possess an eigenvalue of multiplicity bigger than 1 is zero. Since the mapping $\varepsilon \mapsto \mathbf{V}_{-i}^\varepsilon$ is continuous, we deduce that any adherent point $\overline{\mathbf{x}}_i$ of the sequence $(\mathbf{x}_i^\varepsilon)_\varepsilon$ (there exists at least one since the sequence is bounded in ℓ_2-norm) is aligned with the limit $\mathbf{V}_{-i}$ and we apply the previous reasoning. ■

Lemma 23 *For any t, let $\mathbf{V}_t \triangleq \sum_{s=1}^{t-1} \mathbf{x}_s\mathbf{x}_s^\mathsf{T} + \mathbf{\Lambda}$. Then,*

$$\log\frac{|\mathbf{V}_t|}{|\mathbf{\Lambda}|} \leq \max \sum_{i=1}^{N} \log\left(1+\frac{t_i}{\lambda_i}\right),$$

where the maximum is taken over all possible positive real numbers $\{t_1,\ldots,t_N\}$, such that $\sum_{i=1}^N t_i = t-1$.

Proof We want to bound the determinant $|\mathbf{V}_t|$ under the coordinate constraints $\|\mathbf{x}_s\|_2 \leq 1$. Let

$$M(\mathbf{x}_1,\ldots,\mathbf{x}_{t-1}) \triangleq \left|\mathbf{\Lambda} + \sum_{s=1}^{t-1}\mathbf{x}_s\mathbf{x}_s^\mathsf{T}\right|.$$

From Lemma 22, we deduce that the maximum of M is reached when all $\mathbf{x}_t$ are aligned with the axes,

$$\begin{aligned} M &= \max_{\mathbf{x}_1,\ldots,\mathbf{x}_{t-1};\mathbf{x}_s\in\{\mathbf{e}_1,\ldots,\mathbf{e}_N\}} \left|\mathbf{\Lambda} + \sum_{s=1}^{t-1}\mathbf{x}_s\mathbf{x}_s^\mathsf{T}\right| \\ &= \max_{t_1,\ldots,t_N \text{ positive integers},\sum_{i=1}^N t_i = t-1} |\operatorname{diag}(\lambda_i + t_i)| \\ &\leq \max_{t_1,\ldots,t_N \text{ positive reals},\sum_{i=1}^N t_i = t-1} \prod_{i=1}^N (\lambda_i + t_i), \end{aligned}$$

from which we obtain the result. ■

Lemma 24 *Let d be the effective dimension and $t \leq T+1$. Then,*

$$\log\frac{|\mathbf{V}_t|}{|\mathbf{\Lambda}|} \leq d\log\left(1+\frac{T}{K\lambda}\right)\cdot$$

Proof Using Lemma 23 and Definition 1 we have

$$\begin{aligned} \log\frac{|\mathbf{V}_t|}{|\mathbf{\Lambda}|} &\leq \max\sum_{i=1}^N \log\left(1+\frac{t_i}{\lambda_i}\right) \\ &= \frac{\max\sum_{i=1}^N \log\left(1+\frac{t_i}{\lambda_i}\right)}{\log(1+T/(K\lambda))}\log\left(1+\frac{T}{K\lambda}\right) \\ &\leq \left\lceil\frac{\max\sum_{i=1}^N \log\left(1+\frac{t_i}{\lambda_i}\right)}{\log(1+T/(K\lambda))}\right\rceil\log\left(1+\frac{T}{K\lambda}\right) \\ &= d\log\left(1+\frac{T}{K\lambda}\right)\cdot \end{aligned}$$

■

6.4 Regret bound of SpectralUCB

The analysis of SpectralUCB has two main ingredients. The first one is the derivation of the confidence ellipsoid for $\widehat{\boldsymbol{\alpha}}$, which is a straightforward update of the analysis of OFUL (Abbasi-Yadkori et al., 2011) using the self-normalized martingale inequality from Section 6.2. The second part is crucial for showing that the final regret bound scales only with the effective dimension d and not with the ambient dimension D. We achieve this by considering the geometrical properties of the determinant that hold in our setting (Section 6.3).

Proof [**Theorem** 8] Let $\mathbf{x}_\star \triangleq \arg\max_{\mathbf{x}_v} \mathbf{x}_v^\mathsf{T}\boldsymbol{\alpha}$ and let $R_T(t)$ denote the instantaneous regret at round t. With probability at least $1-\delta$, for all t:

$$\begin{align*}
R_T(t) &= \mathbf{x}_\star^\mathsf{T}\boldsymbol{\alpha} - \mathbf{x}_{a_t}^\mathsf{T}\boldsymbol{\alpha} \\
&\le \mathbf{x}_{a_t}^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t + c\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} - \mathbf{x}_{a_t}^\mathsf{T}\boldsymbol{\alpha} \tag{2}\\
&\le \mathbf{x}_{a_t}^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t + c\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} - \mathbf{x}_{a_t}^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t + c\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} \tag{3}\\
&= 2c\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}},
\end{align*}$$

where (2) is by algorithm design and reflects the optimistic principle of SpectralUCB. Specifically, $\mathbf{x}^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t + c\|\mathbf{x}_\star\|_{\mathbf{V}_t^{-1}} \le \mathbf{x}_{a_t}^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t + c\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}}$, from which

$$\mathbf{x}_\star^\mathsf{T}\boldsymbol{\alpha} \le \mathbf{x}_\star^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t + c\|\mathbf{x}_\star\|_{\mathbf{V}_t^{-1}} \le \mathbf{x}_{a_t}^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t + c\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}}.$$

In (3), we apply Lemma 20, $\mathbf{x}_{a_t}^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t \le \mathbf{x}_{a_t}^\mathsf{T}\boldsymbol{\alpha} + c\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}}$. Now, by Lemmas 19 and 24,

$$\begin{align*}
R_T = \sum_{t=1}^T R_T(t) &\le \sum_{t=1}^T \min\left(2,\ 2c\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}}\right) \le (2+2c)\sum_{t=1}^T \min\left(1,\ \|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}}\right) \\
&\le (2+2c)\sqrt{T\sum_{t=1}^T \min\left(1,\ \|\mathbf{x}_{a_t}\|^2_{\mathbf{V}_t^{-1}}\right)} \le (2+2c)\sqrt{2T\log\frac{|\mathbf{V}_{T+1}|}{|\boldsymbol{\Lambda}|}} \\
&\le (2+2c)\sqrt{2dT\log\left(1+\frac{T}{K\lambda}\right)}.
\end{align*}$$

By plugging c, we get that the theorem holds with probability at least $1-\delta$. ■

Remark 25 *Notice that if we set* $\boldsymbol{\Lambda} \triangleq \mathbf{I}$ *in Algorithm 1, we recover the* LinUCB algorithm. Since $\log(|\mathbf{V}_{T+1}|/|\boldsymbol{\Lambda}|)$is upperbounded by $D\log T$ (Abbasi-Yadkori et al., 2011), we obtain $\widetilde{\mathcal{O}}(D\sqrt{T})$ upper bound of regret of LinUCB as a corollary of Theorem 8. The known $\widetilde{\mathcal{O}}(\sqrt{DT})$ upper bound of Chu et al. (2011) applies to a related but *different* SupLinUCB, which is not efficient.

6.5 Regret bound of SpectralTS

The regret bound of SpectralTS is based on the proof technique of Agrawal and Goyal (2013b). Before applying the technique, we first give an intuitive explanation. Each round an arm is played, our algorithm improves the confidence about our actual estimate of $\boldsymbol{\alpha}$ via an update of $\mathbf{V}_t$ and thus the update of the confidence ellipsoid. However, when we play a suboptimal arm, the regret we obtain can be much higher than the improvement of our knowledge. To overcome this difficulty, the arms are divided into two groups of *saturated* and *unsaturated* arms, based on whether the standard deviation for an arm is smaller than the standard deviation of the optimal arm (Definition 27) or not. Consequently, the optimal arm is in the group of unsaturated arms. The idea is to bound the regret of playing an unsaturated arm in terms of standard deviation and to show that the probability that the saturated arm is played is small enough. This way, we overcome the difficulty of high regret and small knowledge obtained by playing an arm.

Definition 26 *We define $E^{\widehat{\boldsymbol{\alpha}}}(t)$ as the event when for all i,*

$$|\mathbf{x}_i^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t - \mathbf{x}_i^\mathsf{T}\boldsymbol{\alpha}| \leq \ell\|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}},$$

where

$$\ell \triangleq R\sqrt{d\log\left(\frac{T^2}{\delta} + \frac{T^3}{\delta\lambda K}\right)} + C,$$

and $E^{\widetilde{\boldsymbol{\alpha}}}(t)$ as the event when for all i,

$$|\mathbf{x}_i^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t - \mathbf{x}_i^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t| \leq v\|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}\sqrt{4\ln(TN)},$$

where

$$v \triangleq R\sqrt{3d\log\left(\frac{1}{\delta} + \frac{T}{\delta\lambda K}\right)} + C.$$

Definition 27 *Let $\Delta_i \triangleq \mathbf{x}_{a^\star}^\mathsf{T}\boldsymbol{\alpha} - \mathbf{x}_i^\mathsf{T}\boldsymbol{\alpha}$. We say that an arm i is **saturated** at round t if $\Delta_i > g\|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}$ and **unsaturated** otherwise, including the optimal arm $a_\star$. Let $C(t)$ denote the **set of saturated arms** at round t.*

Definition 28 *We define the filtration $\mathcal{F}_{t-1}$ as the union of the history until round $t-1$ and features,*

$$\mathcal{F}_{t-1} \triangleq \{\mathcal{H}_{t-1}\} \cup \{\mathbf{x}_i, i = 1, \ldots, N\}.$$

By definition, $\mathcal{F}_1 \subseteq \mathcal{F}_2 \subseteq \cdots \subseteq \mathcal{F}_{T-1}$.

Lemma 29 *For all t, $0 < \delta < 1$, $\mathbb{P}(E^{\widehat{\boldsymbol{\alpha}}}(t)) \geq 1 - \delta/T^2$, and for all possible filtrations $\mathcal{F}_{t-1}$,*

$$\mathbb{P}\left(E^{\widetilde{\boldsymbol{\alpha}}}(t)\,|\,\mathcal{F}_{t-1}\right) \geq 1 - \frac{1}{T^2}.$$

Proof **Bounding the probability of event $E^{\widehat{\boldsymbol{\alpha}}}(t)$:** Using Lemma 20, where C is such that $\|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}} \leq C$, for all i with probability at least $1-\delta'$ we have that

$$\begin{aligned}|\mathbf{x}_i^\mathsf{T}(\widehat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha})| &\leq \|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}\left(R\sqrt{2\log\left(\frac{|\mathbf{V}_t|^{1/2}}{\delta'|\boldsymbol{\Lambda}|^{1/2}}\right)} + C\right)\\ &= \|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}\left(R\sqrt{\log\frac{|\mathbf{V}_t|}{|\boldsymbol{\Lambda}|} + 2\log\left(\frac{1}{\delta'}\right)} + C\right).\end{aligned}$$

Therefore, using Lemma 24 and substituting $\delta' = \delta/T^2$, we get that with probability at least $1 - \delta/T^2$, for all i,

$$\begin{aligned} |\mathbf{x}_i^\mathsf{T}(\widehat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha})| &\leq \|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}} \left(R\sqrt{d\log\left(1 + \frac{T}{K\lambda}\right) + d\log\left(\frac{T^2}{\delta}\right)} + C \right) \\ &= \|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}} \left(R\sqrt{d\log\left(\frac{T^2}{\delta} + \frac{T^3}{\delta\lambda K}\right)} + C \right) = \ell\|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}. \end{aligned}$$

Bounding the probability of event $E^{\widetilde{\boldsymbol{\alpha}}}(t)$: The probability of each individual term $|\mathbf{x}_i^\mathsf{T}(\widetilde{\boldsymbol{\alpha}}_t - \widehat{\boldsymbol{\alpha}}_t)| < \sqrt{4\log(TN)}$ can be bounded using Lemma 14 to get

$$\mathbb{P}\left(|\mathbf{x}_i^\mathsf{T}(\widetilde{\boldsymbol{\alpha}}_t - \widehat{\boldsymbol{\alpha}}_t)| \geq v\|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}\sqrt{4\log(TN)} \right) \leq \frac{e^{-2\log TN}}{\sqrt{\pi 4\log(TN)}} \leq \frac{1}{T^2N}.$$

We complete the proof by taking a union bound over all N vectors $\mathbf{x}_i$. Notice that we took a *different* approach than Agrawal and Goyal (2013b) to avoid the dependence on the ambient dimension D. ■

Lemma 30 *For any filtration $\mathcal{F}_{t-1}$ such that $E^{\widehat{\boldsymbol{\alpha}}}(t)$ is true,*

$$\mathbb{P}\left(\mathbf{x}_{a_\star}^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t > \mathbf{x}_{a_\star}^\mathsf{T}\boldsymbol{\alpha} \,|\, \mathcal{F}_{t-1}\right) \geq \frac{1}{4e\sqrt{\pi}}.$$

Proof Since $\mathbf{x}_{a_\star}^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t$ is a Gaussian random variable with the mean $\mathbf{x}_{a_\star}^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t$ and the standard deviation $v\|\mathbf{x}_{a_\star}\|_{\mathbf{V}_t^{-1}}$, we can use the anti-concentration inequality from Lemma 14,

$$\mathbb{P}\left(\mathbf{x}_{a_\star}^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t \geq \mathbf{x}_{a_\star}^\mathsf{T}\boldsymbol{\alpha} \,|\, \mathcal{F}_{t-1}\right) = \mathbb{P}\left(\frac{\mathbf{x}_{a_\star}^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t - \mathbf{x}_{a_\star}^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t}{v\|\mathbf{x}_{a_\star}\|_{\mathbf{V}_t^{-1}}} \geq \frac{\mathbf{x}_{a_\star}^\mathsf{T}\boldsymbol{\alpha} - \mathbf{x}_{a_\star}^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t}{v\|\mathbf{x}_{a_\star}\|_{\mathbf{V}_t^{-1}}} \,|\, \mathcal{F}_{t-1} \right) \geq \frac{1}{4\sqrt{\pi}Z_t}e^{-Z_t^2},$$

where

$$|Z_t| \triangleq \left| \frac{\mathbf{x}_{a_\star}^\mathsf{T}\boldsymbol{\alpha} - \mathbf{x}_{a_\star}^\mathsf{T}\widehat{\boldsymbol{\alpha}}_t}{v\|\mathbf{x}_{a_\star}\|_{\mathbf{V}_t^{-1}}} \right|.$$

Since we consider filtration $\mathcal{F}_{t-1}$ such that $E^{\widehat{\boldsymbol{\alpha}}}(t)$ is true, we can upperbound the numerator to get

$$|Z_t| \leq \frac{\ell\|\mathbf{x}_{a_\star}\|_{\mathbf{V}_t^{-1}}}{v\|\mathbf{x}_{a_\star}\|_{\mathbf{V}_t^{-1}}} = \frac{\ell}{v} \leq 1.$$

Finally,

$$\mathbb{P}\left(\mathbf{x}_{a_\star}^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t > \mathbf{x}_{a_\star}^\mathsf{T}\boldsymbol{\alpha} \,|\, \mathcal{F}_{t-1}\right) \geq \frac{1}{4e\sqrt{\pi}}.$$

■

Lemma 31 *For any filtration $\mathcal{F}_{t-1}$ such that $E^{\widehat{\boldsymbol{\alpha}}}(t)$ is true,*

$$\mathbb{P}\left(a_t \notin C(t) \,|\, \mathcal{F}_{t-1}\right) \geq \frac{1}{4e\sqrt{\pi}} - \frac{1}{T^2}.$$

Proof The algorithm chooses the arm with the highest value of $\mathbf{x}_i^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t$ to be played at round t. Therefore, if $\mathbf{x}_{a_\star}^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t$ is greater than $\mathbf{x}_j^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t$ for all saturated arms, i.e., $\mathbf{x}_{a_\star}^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t > \mathbf{x}_j^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t$, $\forall j \in C(t)$, then one of the unsaturated arms (that include the optimal arm and other suboptimal unsaturated arms) must be played. Therefore,

$$\mathbb{P}\left(a_t \notin C(t) \,|\, \mathcal{F}_{t-1}\right) \geq \mathbb{P}\left(\mathbf{x}_{a_\star}^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t > \mathbf{x}_j^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t, \forall j \in C(t) \,|\, \mathcal{F}_{t-1}\right).$$

By definition, for all saturated arms $j \in C(t)$, $\Delta_j > g\|\mathbf{x}_j\|_{\mathbf{V}_t^{-1}}$. Now if both of the events $E^{\widehat{\boldsymbol{\alpha}}(t)}$ and $E^{\widetilde{\boldsymbol{\alpha}}(t)}$ are true, then, by definition of these events, for all $j \in C(t)$, $\mathbf{x}_j^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t \leq \mathbf{x}_j^\mathsf{T}\boldsymbol{\alpha}_t + g\|\mathbf{x}_j\|_{\mathbf{V}_t^{-1}}$. Therefore, given filtration $\mathcal{F}_{t-1}$, such that $E^{\widehat{\boldsymbol{\alpha}}}(t)$ is true, either $E^{\widetilde{\boldsymbol{\alpha}}}(t)$ is false, otherwise for all $j \in C(t)$,

$$\mathbf{x}_j^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t \leq \mathbf{x}_j^\mathsf{T}\boldsymbol{\alpha} + g\|\mathbf{x}_j\|_{\mathbf{V}_t^{-1}} \leq \mathbf{x}_{a_\star}^\mathsf{T}\boldsymbol{\alpha}.$$

Hence, for any $\mathcal{F}_{t-1}$ such that $E^{\widehat{\boldsymbol{\alpha}}}(t)$ is true,

$$\begin{aligned}\mathbb{P}\left(\mathbf{x}_{a_\star}^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t > \mathbf{x}_j^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t, \forall j \in C(t) \,|\, \mathcal{F}_{t-1}\right) &\geq \mathbb{P}\left(\mathbf{x}_{a_\star}^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t > \mathbf{x}_{a_\star}^\mathsf{T}\boldsymbol{\alpha} \,|\, \mathcal{F}_{t-1}\right) - \mathbb{P}\left(\overline{E^{\widehat{\boldsymbol{\alpha}}}(t)} \,|\, \mathcal{F}_{t-1}\right)\\ &\geq \frac{1}{4e\sqrt{\pi}} - \frac{1}{T^2},\end{aligned}$$

where in the last inequality we used Lemma 29 and Lemma 30. ■

Lemma 32 *For any filtration $\mathcal{F}_{t-1}$ such that $E^{\widehat{\boldsymbol{\alpha}}}(t)$ is true,*

$$\mathbb{E}\left[\Delta_{a_t} \,|\, \mathcal{F}_{t-1}\right] \leq \frac{11g}{p}\mathbb{E}\left[\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} \,|\, \mathcal{F}_{t-1}\right] + \frac{1}{T^2}.$$

Proof Let $\overline{a}_t$ denote the unsaturated arm with the smallest norm $\|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}$,

$$\overline{a}_t \triangleq \underset{i \notin C(t)}{\arg\min}\, \|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}.$$

Notice that since $C(t)$ and $\|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}$ for all i are fixed on fixing $\mathcal{F}_{t-1}$, so is $\overline{a}_t$. Now, using Lemma 31, for any $\mathcal{F}_{t-1}$ such that $E^{\widehat{\boldsymbol{\alpha}}}(t)$ is true,

$$\begin{aligned}\mathbb{E}\left[\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} \,|\, \mathcal{F}_{t-1}\right] &\geq \mathbb{E}\left[\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} \,|\, \mathcal{F}_{t-1}, a_t \notin C(t)\right] \mathbb{P}\left(a_t \notin C(t) \,|\, \mathcal{F}_{t-1}\right)\\ &\geq \|\mathbf{x}_{\overline{a}_t}\|_{\mathbf{V}_t^{-1}}\left(\frac{1}{4e\sqrt{\pi}} - \frac{1}{T^2}\right).\end{aligned}$$

Now, if events $E^{\widehat{\boldsymbol{\alpha}}}(t)$ and $E^{\widetilde{\boldsymbol{\alpha}}}(t)$ are true, then for all i, by definition, $\mathbf{x}_i^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t \leq \mathbf{x}_i^\mathsf{T}\boldsymbol{\alpha}+g\|\mathbf{x}_i\|_{\mathbf{V}_t^{-1}}$. Using this observation along with $\mathbf{x}_{a_t}^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t \geq \mathbf{x}_i^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t$ for all i,

$$\begin{aligned}
\Delta_{a_t} &= \Delta_{\overline{a}_t} + (\mathbf{x}_{\overline{a}_t}^\mathsf{T}\boldsymbol{\alpha} - \mathbf{x}_{a_t}^\mathsf{T}\boldsymbol{\alpha}) \\
&\leq \Delta_{\overline{a}_t} + (\mathbf{x}_{\overline{a}_t}^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t - \mathbf{x}_{a_t}^\mathsf{T}\widetilde{\boldsymbol{\alpha}}_t) + g\|\mathbf{x}_{\overline{a}_t}\|_{\mathbf{V}_t^{-1}} + g\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} \\
&\leq \Delta_{\overline{a}_t} + g\|\mathbf{x}_{\overline{a}_t}\|_{\mathbf{V}_t^{-1}} + g\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} \\
&\leq g\|\mathbf{x}_{\overline{a}_t}\|_{\mathbf{V}_t^{-1}} + g\|\mathbf{x}_{\overline{a}_t}\|_{\mathbf{V}_t^{-1}} + g\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}}.
\end{aligned}$$

Therefore, for any $\mathcal{F}_{t-1}$ such that $E^{\widehat{\boldsymbol{\alpha}}}(t)$ is true, either $\Delta_{a_t} \leq 2g\|\mathbf{x}_{\overline{a}_t}\|_{\mathbf{V}_t^{-1}} + g\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}}$, or $E^{\widetilde{\boldsymbol{\alpha}}}(t)$ is false. We can deduce that

$$\begin{aligned}
\mathbb{E}\left[\Delta_{a_t} \,|\, \mathcal{F}_{t-1}\right] &\leq \mathbb{E}\left[2g\|\mathbf{x}_{\overline{a}_t}\|_{\mathbf{V}_t^{-1}} + g\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} \,|\, \mathcal{F}_{t-1}\right] + \mathbb{P}\left(\overline{E^{\widetilde{\boldsymbol{\alpha}}}(t)}\right) \\
&\leq \frac{2g}{p-\frac{1}{T^2}}\mathbb{E}\left[\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} \,|\, \mathcal{F}_{t-1}\right] + g\mathbb{E}\left[\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} \,|\, \mathcal{F}_{t-1}\right] + \frac{1}{T^2} \\
&\leq \frac{11g}{p}\mathbb{E}\left[\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} \,|\, \mathcal{F}_{t-1}\right] + \frac{1}{T^2},
\end{aligned}$$

where in the last inequality we used that $1/(p-1/T^2) \leq 5/p$, which holds trivially for $T \leq 4$. For $T \geq 5$, we know that $p = 1/(4e\sqrt{\pi})$ and therefore $T^2 \geq 5e\sqrt{\pi}$, from which we get that $1/(p-1/T^2) \leq 5/p$ as well.

■

Definition 33 *We define* $R'_T(t) \triangleq R_T(t) \cdot I\left(E^{\widehat{\boldsymbol{\alpha}}}(t)\right)$.

Definition 34 *A sequence of random variables* $(Y_t;\, t \geq 0)$ *is called a* ***super-martingale*** *corresponding to a filtration* $\mathcal{F}_t$, *if for all* t, Y_t *is* $\mathcal{F}_t$*-measurable, and for* $t \geq 1$,

$$\mathbb{E}\left[Y_t - Y_{t-1} \,|\, \mathcal{F}_{t-1}\right] \leq 0.$$

Next, following Agrawal and Goyal (2013b), we establish a super-martingale process that forms the basis of our proof of the high-probability regret bound.

Definition 35 *Let*

$$\begin{aligned}
X_t &\triangleq R'_T(t) - \frac{11g}{p}\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} - \frac{1}{T^2} \quad \textit{and} \\
Y_t &\triangleq \sum_{w=1}^{t} X_w.
\end{aligned}$$

Lemma 36 $(Y_t;\, t = 0, \ldots, T)$ *is a super-martingale process with respect to filtration* $\mathcal{F}_t$.

Proof We need to prove that for all $t \in \{1, \ldots, T\}$ and any possible filtration $\mathcal{F}_{t-1}$, $\mathbb{E}[Y_t - Y_{t-1} \,|\, \mathcal{F}_{t-1}] \leq 0$, i.e.,

$$\mathbb{E}\left[R'_T(t) \,|\, \mathcal{F}_{t-1}\right] \leq \frac{11g}{p}\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} + \frac{1}{T^2}.$$

Note that whether $E^{\widehat{\boldsymbol{\alpha}}}(t)$ is true or not, is completely determined by $\mathcal{F}_{t-1}$. If $\mathcal{F}_{t-1}$ is such that $E^{\widehat{\boldsymbol{\alpha}}}(t)$ is not true, then $R'_T(t) = R_T(t) \cdot I\left(E^{\widehat{\boldsymbol{\alpha}}}(t)\right) = 0$, and the above inequality holds trivially. Moreover, for $\mathcal{F}_{t-1}$ such that $E^{\widehat{\boldsymbol{\alpha}}}(t)$ holds, the inequality follows from Lemma 32. ■

Note that unlike Agrawal and Goyal (2013b) and Abbasi-Yadkori et al. (2011), we do not want to require $\lambda \geq 1$. Therefore, we provide the following lemma that features the dependence of $\|\mathbf{x}_{a_t}\|^2_{\mathbf{V}_t^{-1}}$ on λ.

Lemma 37 *For all t,*

$$\|\mathbf{x}_{a_t}\|^2_{\mathbf{V}_t^{-1}} \leq \left(2 + \frac{2}{\lambda}\right) \log\left(1 + \|\mathbf{x}_{a_t}\|^2_{\mathbf{V}_t^{-1}}\right).$$

Proof Note, that $\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} \leq (1/\sqrt{\lambda})\|\mathbf{x}_{a_t}\| \leq (1/\sqrt{\lambda})$ and for all $0 \leq x \leq 1$, we have

$$x \leq 2\log(1 + x). \tag{4}$$

We now consider two cases depending on λ. If $\lambda \geq 1$, we know that $0 \leq \|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} \leq 1$ and therefore by (4),

$$\|\mathbf{x}_{a_t}\|^2_{\mathbf{V}_t^{-1}} \leq 2\log\left(1 + \|\mathbf{x}_{a_t}\|^2_{\mathbf{V}_t^{-1}}\right).$$

Similarly, if $\lambda < 1$, then $0 \leq \lambda\|\mathbf{x}_{a_t}\|^2_{\mathbf{V}_t^{-1}} \leq 1$ and we get

$$\|\mathbf{x}_{a_t}\|^2_{\mathbf{V}_t^{-1}} \leq \frac{2}{\lambda}\log\left(1 + \lambda\|\mathbf{x}_{a_t}\|^2_{\mathbf{V}_t^{-1}}\right) \leq \frac{2}{\lambda}\log\left(1 + \|\mathbf{x}_{a_t}\|^2_{\mathbf{V}_t^{-1}}\right).$$

Combining the two, we get that for all $\lambda \geq 0$,

$$\|\mathbf{x}_{a_t}\|^2_{\mathbf{V}_t^{-1}} \leq \max\left(2,\ \frac{2}{\lambda}\right)\log\left(1 + \|\mathbf{x}_{a_t}\|^2_{\mathbf{V}_t^{-1}}\right) \leq \left(2 + \frac{2}{\lambda}\right)\log\left(1 + \|\mathbf{x}_{a_t}\|^2_{\mathbf{V}_t^{-1}}\right).$$

■

Proof [**Theorem 11**] First, notice that X_t is bounded as

$$|X_t| \leq 1 + \frac{11g}{p\sqrt{\lambda}} + \frac{1}{T^2} \leq \frac{g}{p}\left(\frac{11}{\sqrt{\lambda}} + 2\right).$$

Thus, we can apply the Azuma-Hoeffding inequality to obtain that with probability at least $1 - \delta/2$,

$$\sum_{t=1}^{T} R'_T(t) \leq \sum_{t=1}^{T} \frac{11g}{p}\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} + \sum_{t=1}^{T}\frac{1}{T^2} + \sqrt{2\left(\sum_{t=1}^{T}\frac{g^2}{p^2}\left(\frac{11}{\sqrt{\lambda}} + 2\right)^2\right)\log\left(\frac{2}{\delta}\right)}.$$

Since p and g are constants, then with probability $1 - \delta/2$,

$$\sum_{t=1}^{T} R'_T(t) \leq \frac{11g}{p}\sum_{t=1}^{T}\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}} + \frac{1}{T} + \frac{g}{p}\left(\frac{11}{\sqrt{\lambda}} + 2\right)\sqrt{2T\log\left(\frac{2}{\delta}\right)}.$$

The last step is to upperbound $\sum_{t=1}^{T}\|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}}$. For this purpose, Agrawal and Goyal (2013b) rely on the analysis of Auer (2002) and the assumption that $\lambda \geq 1$. We provide an alternative approach using Cauchy-Schwartz inequality, Lemma 15, and Lemma 37 to get

$$\sum_{t=1}^{T}\|\mathbf{x}_{a_t)}\|_{\mathbf{V}_t^{-1}} \leq \sqrt{T\sum_{t=1}^{T}\|\mathbf{x}_{a_t}\|^2_{\mathbf{V}_t^{-1}}} \leq \sqrt{T\left(2+\frac{2}{\lambda}\right)\log\frac{|\mathbf{V}_T|}{|\mathbf{\Lambda}|}} \leq \sqrt{\frac{2+2\lambda}{\lambda}dT\log\left(1+\frac{T}{K\lambda}\right)}.$$

Finally, we know that $E^{\widehat{\boldsymbol{\alpha}}}(t)$ holds for all t with probability at least $1-\delta/2$ and $R'_T(t) = R_T(t)$ for all t with probability at least $1-\delta/2$. Hence, with probability $1-\delta$,

$$R_T \leq \frac{11g}{p}\sqrt{\frac{2+2\lambda}{\lambda}dT\log\left(1+\frac{T}{K\lambda}\right)} + \frac{1}{T} + \frac{g}{p}\left(\frac{11}{\sqrt{\lambda}}+2\right)\sqrt{2T\log\left(\frac{2}{\delta}\right)}.$$

■

6.6 Regret bound of SpectralEliminator

The probability space induced by the rewards $r_1, r_2, \ldots$ can be decomposed as a product of independent probability spaces induces by rewards in each phase $[t_j, t_{j+1}-1]$. Denote by $\mathcal{F}'_j$ the σ-algebra generated by the rewards $r_1, \ldots, r_{t_{j+1}-1}$, i.e., received before and during the phase j. We have the following two lemmas for any phase j. Let $\overline{\mathbf{V}}_j \triangleq \mathbf{\Lambda} + \sum_{s=t_{j-1}}^{t_j-1}\mathbf{x}_{a_s}\mathbf{x}_{a_s}^{\mathsf{T}}$ and let $\widehat{\boldsymbol{\alpha}}_j$ stand for $\widehat{\boldsymbol{\alpha}}_{t_j}$ for simplicity.

Lemma 38 *For any fixed* $\mathbf{x} \in \mathbb{R}^N$*, any* $\delta > 0$*, and* $\beta(\delta) \triangleq R\sqrt{2\log(2/\delta)} + \|\boldsymbol{\alpha}\|_{\mathbf{\Lambda}}$*, we have for all* j*,*

$$\mathbb{P}\left(|\mathbf{x}^{\mathsf{T}}(\widehat{\boldsymbol{\alpha}}_j - \boldsymbol{\alpha})| \leq \|\mathbf{x}\|_{\overline{\mathbf{V}}_j^{-1}}\beta(\delta)\right) \geq 1-\delta.$$

Proof Defining $\boldsymbol{\xi}_j \triangleq \sum_{s=t_{j-1}}^{t_j-1}\mathbf{x}_{a_s}\varepsilon_s$, we have

$$|\mathbf{x}^{\mathsf{T}}(\widehat{\boldsymbol{\alpha}}_j - \boldsymbol{\alpha})| = |\mathbf{x}^{\mathsf{T}}(-\overline{\mathbf{V}}_j^{-1}\mathbf{\Lambda}\boldsymbol{\alpha} + \overline{\mathbf{V}}_j^{-1}\boldsymbol{\xi}_j)| \leq |\mathbf{x}^{\mathsf{T}}\overline{\mathbf{V}}_j^{-1}\mathbf{\Lambda}\boldsymbol{\alpha}| + |\mathbf{x}^{\mathsf{T}}\overline{\mathbf{V}}_j^{-1}\boldsymbol{\xi}_j|. \tag{5}$$

The first term in the right hand side of (5) is bounded as

$$\begin{aligned}
|\mathbf{x}^{\mathsf{T}}\overline{\mathbf{V}}_j^{-1}\mathbf{\Lambda}\boldsymbol{\alpha}| &\leq \|\mathbf{x}^{\mathsf{T}}\overline{\mathbf{V}}_j^{-1}\mathbf{\Lambda}^{1/2}\|\|\mathbf{\Lambda}^{1/2}\boldsymbol{\alpha}\| \\
&= \|\boldsymbol{\alpha}\|_{\mathbf{\Lambda}}\sqrt{\mathbf{x}^{\mathsf{T}}\overline{\mathbf{V}}_j^{-1}\mathbf{\Lambda}\overline{\mathbf{V}}_j^{-1}\mathbf{x}} \\
&\leq \|\boldsymbol{\alpha}\|_{\mathbf{\Lambda}}\sqrt{\mathbf{x}^{\mathsf{T}}\overline{\mathbf{V}}_j^{-1}\mathbf{x}} = \|\boldsymbol{\alpha}\|_{\mathbf{\Lambda}}\|\mathbf{x}\|_{\overline{\mathbf{V}}_j^{-1}}.
\end{aligned}$$

Now, consider the second term in the right hand side of (5). We have

$$\left|\mathbf{x}^{\mathsf{T}}\overline{\mathbf{V}}_j^{-1}\boldsymbol{\xi}_j\right| = \left|\sum_{s=t_{j-1}}^{t_j-1}(\mathbf{x}^{\mathsf{T}}\overline{\mathbf{V}}_j^{-1}\mathbf{x}_{a_s})\varepsilon_s\right|.$$

Let us notice that the context vectors $(\mathbf{x}_{a_s})$ selected by the algorithm during phase $j-1$ only depend on their width $\|\mathbf{x}\|_{\mathbf{V}_s^{-1}}$, which does not depend on the rewards received during the phase $j-1$. Thus, given $\mathcal{F}'_{j-2}$, the values $\mathbf{x}^\mathsf{T}\overline{\mathbf{V}}_j^{-1}\mathbf{x}_{a_s}$ are deterministic for all rounds $t_{j-1} \leq s < t_j$. Consequently, we can use a variant of Hoeffding bound for *scaled* sub-Gaussians (Wainwright, 2015), in particular for $\mathbf{x}^\mathsf{T}\overline{\mathbf{V}}_j^{-1}\boldsymbol{\xi}_j = \sum_{s=t_{j-1}}^{t_j-1} \mathbf{x}^\mathsf{T}\overline{\mathbf{V}}_j^{-1}\mathbf{x}_{a_s}\varepsilon_s$, to get

$$\mathbb{P}\left(\left|\mathbf{x}^\mathsf{T}\overline{\mathbf{V}}_j^{-1}\boldsymbol{\xi}_j\right| \leq R\sqrt{2\log\left(\frac{2}{\delta}\right)\sum_{s=t_{j-1}}^{t_j-1}\left(\mathbf{x}^\mathsf{T}\overline{\mathbf{V}}_j^{-1}\mathbf{x}_{a_s}\right)^2}\right) \geq 1-\delta,$$

where ε_s is R-sub-Gaussian and $\mathbf{x}^\mathsf{T}\overline{\mathbf{V}}_j^{-1}\mathbf{x}_{a_s}$ is deterministic given $\mathcal{F}'_{j-2}$. We further deduce

$$\mathbb{P}\left(\left|\mathbf{x}^\mathsf{T}\overline{\mathbf{V}}_j^{-1}\boldsymbol{\xi}_j\right| \leq R\sqrt{2\log\left(\frac{2}{\delta}\right)\sum_{s=t_{j-1}}^{t_j-1}\left(\mathbf{x}^\mathsf{T}\overline{\mathbf{V}}_j^{-1}\mathbf{x}_{a_s}\mathbf{x}_{a_s}^\mathsf{T}\overline{\mathbf{V}}_j^{-1}\mathbf{x}\right)}\right) \geq 1-\delta$$
$$\mathbb{P}\left(\left|\mathbf{x}^\mathsf{T}\overline{\mathbf{V}}_j^{-1}\boldsymbol{\xi}_j\right| \leq R\sqrt{2\log\left(\frac{2}{\delta}\right)\mathbf{x}^\mathsf{T}\overline{\mathbf{V}}_j^{-1}\left(\sum_{s=t_{j-1}}^{t_j-1}\mathbf{x}_{a_s}\mathbf{x}_{a_s}^\mathsf{T}\right)\overline{\mathbf{V}}_j^{-1}\mathbf{x}}\right) \geq 1-\delta$$
$$\mathbb{P}\left(\left|\mathbf{x}^\mathsf{T}\overline{\mathbf{V}}_j^{-1}\boldsymbol{\xi}_j\right| \leq R\sqrt{2\log\left(\frac{2}{\delta}\right)\mathbf{x}^\mathsf{T}\overline{\mathbf{V}}_j^{-1}\mathbf{x}}\right) \geq 1-\delta,$$

since $\overline{\mathbf{V}}_j^{-1}$ is symmetric and $\sum_{s=t_{j-1}}^{t_j-1}\mathbf{x}_s\mathbf{x}_s^\mathsf{T} \prec \overline{\mathbf{V}}_j$ (Lemma 16). We conclude that

$$\mathbb{P}\left(\left|\mathbf{x}^\mathsf{T}\overline{\mathbf{V}}_j^{-1}\boldsymbol{\xi}_j\right| \leq R\|\mathbf{x}\|_{\overline{\mathbf{V}}_j^{-1}}\sqrt{2\log\left(\frac{2}{\delta}\right)}\right) \geq 1-\delta.$$

■

Lemma 39 *For all $\mathbf{x} \in A_j$, $j > 1$, we have that*

$$\min\left(1, \|\mathbf{x}\|_{\overline{\mathbf{V}}_j^{-1}}\right) \leq \frac{1}{t_j - t_{j-1}}\sum_{s=t_{j-1}}^{t_j-1}\min\left(1, \|\mathbf{x}_{a_s}\|_{\mathbf{V}_s^{-1}}\right).$$

Proof Using Lemma 16, we have that

$$
\begin{aligned}
(t_j - t_{j-1}) \min\left(1,\ \|\mathbf{x}\|_{\overline{\mathbf{V}}_j^{-1}}\right) &\leq \max_{\mathbf{x}\in A_j} \sum_{s=t_{j-1}}^{t_j-1} \min\left(1,\ \|\mathbf{x}\|_{\mathbf{V}_s^{-1}}\right) \\
&\leq \max_{\mathbf{x}\in A_{j-1}} \sum_{s=t_{j-1}}^{t_j-1} \min\left(1,\ \|\mathbf{x}\|_{\mathbf{V}_s^{-1}}\right) \\
&\leq \sum_{s=t_{j-1}}^{t_j-1} \min\left(1,\ \max_{\mathbf{x}\in A_{j-1}} \|\mathbf{x}\|_{\mathbf{V}_s^{-1}}\right) \\
&= \sum_{s=t_{j-1}}^{t_j-1} \min\left(1,\ \|\mathbf{x}_{a_s}\|_{\mathbf{V}_s^{-1}}\right),
\end{aligned}
$$

since the algorithm selects (during phase $j-1$) the arms with the largest width. ■

We now are ready to upperbound the cumulative regret of SpectralEliminator.

Proof [**Theorem 13**] Let $J \triangleq \lfloor \log_2 T \rfloor + 1$ and $t_j \triangleq 2^{j-1}$. We have that

$$
\begin{aligned}
R_T = \sum_{t=1}^{T} \mathbf{x}_{a^\star}^{\mathsf{T}} \boldsymbol{\alpha} - \mathbf{x}_{a_t}^{\mathsf{T}} \boldsymbol{\alpha} &\leq 2 + \sum_{j=2}^{J} \sum_{t=t_j}^{t_{j+1}-1} \min(2,\ \mathbf{x}_{a^\star}^{\mathsf{T}} \boldsymbol{\alpha} - \mathbf{x}_{a_t}^{\mathsf{T}} \boldsymbol{\alpha}) \\
&\leq 2 + \sum_{j=2}^{J} \sum_{t=t_j}^{t_{j+1}-1} \min\left(2,\ \mathbf{x}_{a^\star}^{\mathsf{T}} \widehat{\boldsymbol{\alpha}}_j - \mathbf{x}_{a_t}^{\mathsf{T}} \widehat{\boldsymbol{\alpha}}_j + \left(\|\mathbf{x}_\star\|_{\overline{\mathbf{V}}_j^{-1}} + \|\mathbf{x}_t\|_{\overline{\mathbf{V}}_j^{-1}}\right) \beta(\delta')\right),
\end{aligned}
$$

in an event ω of probability $1-\delta$, where we used Lemma 38 and the union bound in the last inequality for $\delta' \triangleq \delta/(KJ)$. By definition of the action subset A_j at phase $j > 1$, under ω, we have that

$$
\mathbf{x}_{a^\star}^{\mathsf{T}} \widehat{\boldsymbol{\alpha}}_j - \mathbf{x}_{a_t} \widehat{\boldsymbol{\alpha}}_j \leq \left(\|\mathbf{x}_{a^\star}\|_{\overline{\mathbf{V}}_j^{-1}} + \|\mathbf{x}_{a_t}\|_{\overline{\mathbf{V}}_j^{-1}}\right) \beta(\delta'),
$$

since $\mathbf{x}_{a^\star} \in A_j$ for all $j \leq J$. By previous two lemmas and the Cauchy-Schwarz inequality,

$$\begin{aligned}
R_T &\le 2 + \sum_{j=2}^{J} \sum_{t=t_j}^{t_{j+1}-1} \min\left(2,\ 4\beta(\delta')\|\mathbf{x}_{a_t}\|_{\overline{\mathbf{V}}_j^{-1}}\right) \\
&\le 2 + (4\beta(\delta')+2) \sum_{j=2}^{J} \sum_{t=t_j}^{t_{j+1}-1} \min\left(1,\ \|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}}\right) \\
&\le 2 + (4\beta(\delta')+2) \sum_{j=2}^{J} \frac{t_{j+1}-t_j}{t_j - t_{j-1}} \sum_{t=t_{j-1}}^{t_j-1} \min\left(1,\ \|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}}\right) \\
&\le 2 + (8\beta(\delta')+4) \sum_{j=2}^{J} \sum_{t=t_{j-1}}^{t_j-1} \min\left(1,\ \|\mathbf{x}_{a_t}\|_{\mathbf{V}_t^{-1}}\right) \\
&\le 2 + (8\beta(\delta')+4) \sqrt{T \sum_{j=2}^{J} \sum_{t=t_{j-1}}^{t_j-1} \min\left(1,\ \|\mathbf{x}_{a_t}\|^2_{\mathbf{V}_t^{-1}}\right)} \\
&\le 2 + (8\beta(\delta')+4) \sqrt{T \sum_{j=2}^{J} 2\log\frac{|\overline{\mathbf{V}}_j|}{|\mathbf{\Lambda}|}} \\
&\le 2 + (8\beta(\delta')+4) \sqrt{2dT\log_2(T)\log\left(1+\frac{T}{K\lambda}\right)}.
\end{aligned}$$

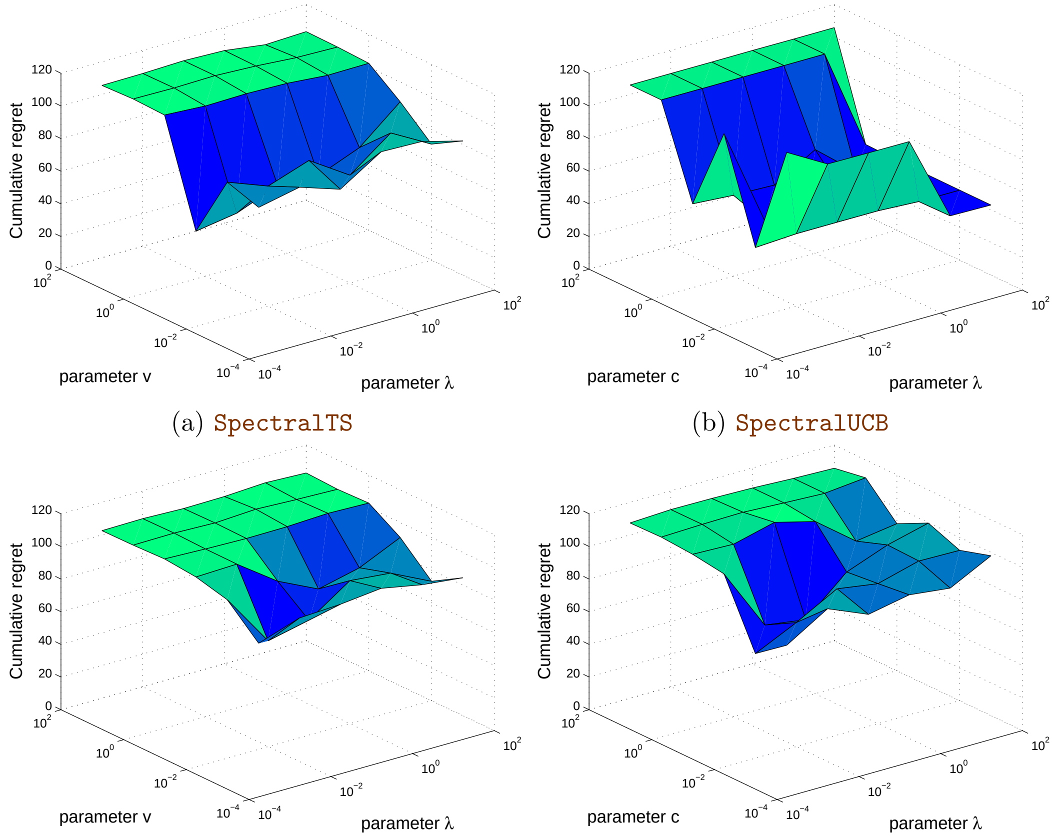
Finally, using $J = 1 + \lfloor \log_2 T \rfloor$, $\delta' = \delta/(KJ)$, and $\beta(\delta') \le \beta(\delta/(K(1+\log_2 T)))$, we obtain the result of Theorem 13. ■

Remark 40 *If we set $\mathbf{\Lambda} = \mathbf{I}$ in Algorithm 3 as in Remark 25, we get a new algorithm,* `LinearEliminator`, which is a competitor to `SupLinRel` (Auer, 2002) and `SupLinUCB` (Chu et al., 2011) and as a corollary to Theorem 13 also enjoys an $\widetilde{\mathcal{O}}(\sqrt{DT})$ upper bound on the cumulative regret. Compared to `SupLinRel` and `SupLinUCB`, `LinearEliminator` and its analysis are significantly simpler and more elegant. Furthermore, `LinearEliminator` is more data-adaptive since it uses self-normalized concentration bounds rather than data-agnostic confidence intervals of the form 2^{-u} for $u \in \mathbb{N}_0$, which are used in `SupLinRel` and `SupLinUCB`. Therefore, `LinearEliminator` narrows the gap between the practical algorithms and the algorithms with the optimal cumulative regret of $\widetilde{\mathcal{O}}(\sqrt{DT})$.

7. Experiments

In this section, we evaluate the empirical regret as well the empirical computational complexity of `SpectralTS`, `SpectralUCB`, `LinearTS`, and `LinUCB` on artificial datasets with different types of underlying graph structure as well as on MovieLens and Flixster datasets. We do not include `SpectralEliminator` in our experiments due to its impracticality for small time horizons.[2] We study the sensitivity of the algorithms to the important parame-

[2] since the algorithm updates confidence ellipsoid only at the end of the phase



(a) SpectralTS

(b) SpectralUCB

(c) LinearTS

(d) LinUCB

Figure 4: The dependence of cumulative regret on confidence and regularization parameters.

ters and comment on practical issues. Moreover, we study the effects of different speed-up techniques. In particular, we show the effect of the reduced basis on both the computational complexity and performance of the algorithms and the effect of Sherman-Morrison (the computation of matrix inversions) together with lazy updates (the computation of UCBs) on the running time. In all experiments, we set both the confidence parameter δ, use the uniformly distributed noise satisfying $R \leq 0.05$, and average over 5 runs. We performed but do not include the results for different values of δ and R since the results of the experiments are not sensitive to the values of these parameters and follow the same trend.

7.1 Artificial datasets

To demonstrate the benefit of spectral algorithms, we perform exhaustive experiments on artificial datasets with various underlying graphs. More precisely, we focus on problems where underlying graphs form a lattice or they are sampled either from the Barabási-Albert (BA) or Erdős-Rényi (ER) graph model. For all experiments on artificial datasets, we set the number of arms N to 500 and the time horizon T to 100. We sample a random vector $\boldsymbol{\alpha}$ such that reward function $\mathbf{f} \triangleq \mathbf{Q}\boldsymbol{\alpha}$ is smooth on the graph. We do it by settings only the first 20 elements of $\boldsymbol{\alpha}$ to be nonzero. For a more useful empirical comparison, we set the regularization parameter λ and confidence ellipsoid parameters v (`TS`) and c (`UCB`) respectively to the best empirical value over a grid search. We run the algorithms with several different values and select the values which minimized average cumulative regret after a few runs of algorithms. Figure 4 shows the dependence of cumulative regret on parameters with strong indication that `SpectralTS` and `SpectralUCB` can leverage smoothness of the reward function and outperform `LinearTS` and `LinUCB`.

7.1.1 Erdős-Rényi graphs

For this experiment, we construct the underlying graph as an Erdős-Rényi graph on 500 nodes with parameter 0.005 (the probability of edge appearance). The values of the parameters used for the experiment are listed in Table 1, which are the values where the algorithms perform the best.

Figure 5a shows the cumulative regrets of the algorithms with selected parameters. The regret of spectral algorithms tends to be sublinear while regret of linear algorithms appears to be linear for small T. Moreover, spectral algorithms reach much smaller empirical regrets than their linear counterparts.

`SpectralTS`		`SpectralUCB`		`LinearTS`		`LinUCB`	
$\lambda = 0.1$	$v = 0.1$	$\lambda = 1$	$c = 1$	$\lambda = 1$	$v = 0.1$	$\lambda = 0.1$	$c = 0.1$

Table 1: The best-performing empirical parameters for the Erdős-Rényi graph model.

7.1.2 Lattice graphs

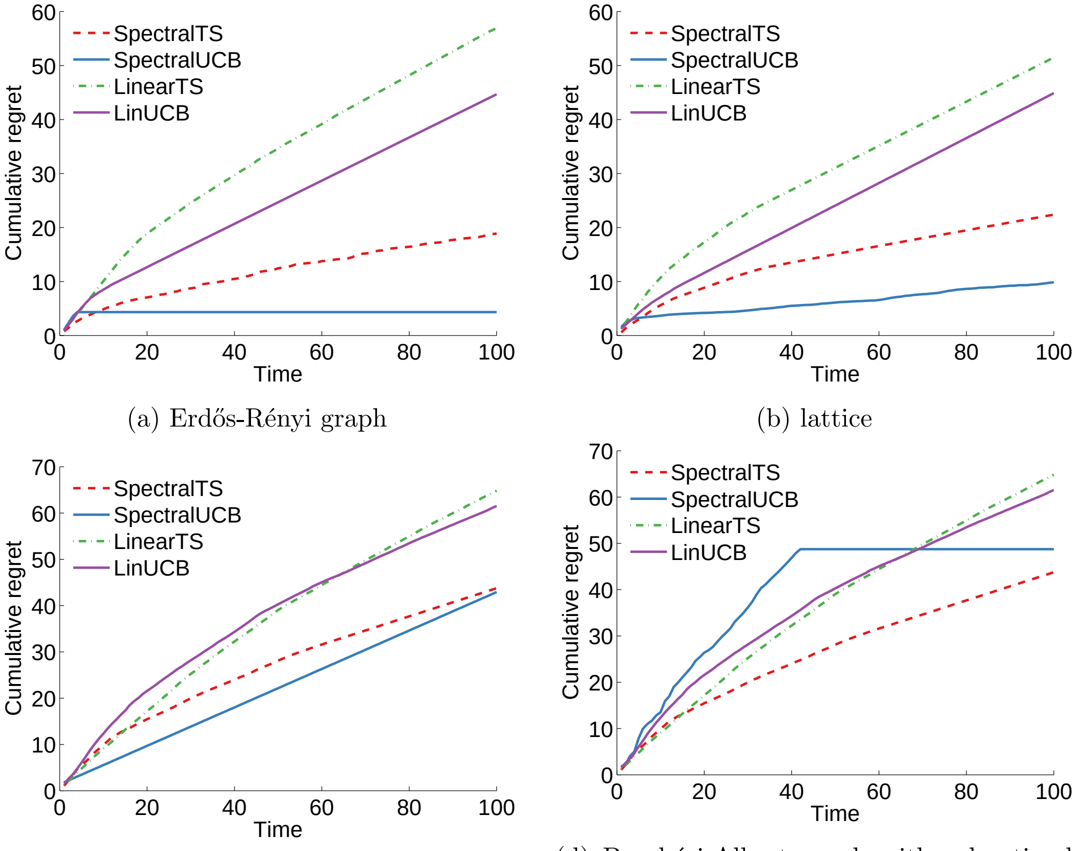
For lattices, we arrange 500 nodes to form a lattice and connect every pair of nodes by an edge if they are neighbors in the lattice. As for the other experiments, we empirically select the best set of parameters (Table 2) and use them to plot the cumulative regret of algorithms (Figure 5b). Even in this case, spectral algorithms perform well compared to the linear ones.

`SpectralTS`		`SpectralUCB`		`LinearTS`		`LinUCB`	
$\lambda = 0.01$	$v = 0.1$	$\lambda = 0.1$	$c = 1$	$\lambda = 1$	$v = 0.1$	$\lambda = 0.1$	$c = 0.1$

Table 2: The best-performing empirical parameters for lattices.

7.1.3 Barabási-Albert graphs

We construct the BA graph for our experiments in the following way. We start with k nodes ($k = 3$ in our case) without any connections between them. Then, we sequentially add one node at a time. Each new node is connected to $m \leq k$ previously added nodes and we sampled the connections according to the degrees of existing nodes: the higher the degree,



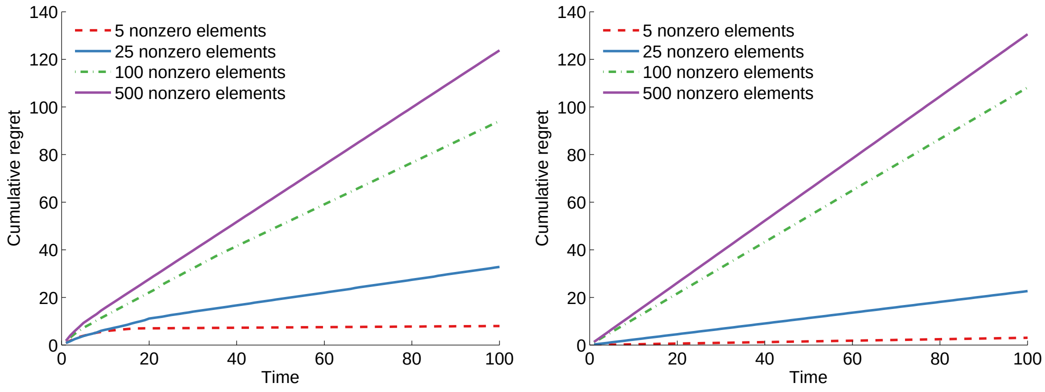
(a) Erdős-Rényi graph

(b) lattice

(c) Barabási-Albert graph

(d) Barabási-Albert graph with suboptimal parameters for SpectralUCB

Figure 5: Cumulative regret comparison of algorithms for different underlying graphs.



(a) SpectralTS

(b) SpectralUCB

Figure 6: Cumulative regret of SpectralTS and SpectralUCB for reward functions with different smoothness.

the bigger the chance of the connection. Table 3 summarizes the best empirical values of the parameters for the algorithms and

Figure 5c shows the performance of algorithms for the parameters in Table 3. Here we can clearly see that the spectral algorithms outperform the linear ones after just a few rounds. Note that the empirically optimal parameters can sometimes be too aggressive and force an algorithm to exploit more than it should. This is likely the case of `SpectralUCB` in Figure 5c since the curve of the cumulative regret of `SpectralUCB` appears to be linear for the time horizon used in our experiment. Therefore, we include Figure 5d, where we plot the cumulative regret of `SpectralUCB` for an empirically suboptimal value of $c = 1$ (close to the best theoretical value of c) to demonstrate the sublinear trend of the regret.

`SpectralTS`		`SpectralUCB`		`LinearTS`		`LinUCB`	
$\lambda = 0.001$	$v = 0.1$	$\lambda = 0.001$	$c = 0.01$	$\lambda = 0.01$	$v = 0.01$	$\lambda = 0.1$	$c = 0.1$

Table 3: The best-performing empirical parameters for the Barabási-Albert graph model.

7.2 The effect of smoothness on the regret

In this section, we study the effect of the smoothness of the reward function on the performance of spectral algorithms. We use a BA graph on 500 nodes for the experiment with time horizon 100 and the parameters of the algorithms are set according to table 3. The value of effective dimension is close to 8. We controll the smoothness by explicitly setting the number of eigenvectors used for constructing the reward function by letting 5, 25, 100, or 500 elements of $\boldsymbol{\alpha}$ to be nonzero. Note that the value of the effective dimension is the same for every reward function we used, since the definition of the effective dimension is independent of the reward function. Table 4 shows how the smoothness changes with the number of nonzero elements of $\boldsymbol{\alpha}$ and Figures 6a and 6b confirm that the spectral algorithms are able to leverage spectral properties of underlying graph better when the reward function is smoother. This is also supported by our analysis, since in our experiment, the smoothness of the reward function decreases with a smaller number of eigenvectors and the regret bounds of the spectral algorithms are decreasing with smoothness as well.

Number of nonzero components	**5**	**25**	**100**	**500**
Smoothness of the rewards ($\boldsymbol{\alpha}^\mathsf{T}\boldsymbol{\Lambda}\boldsymbol{\alpha}$)	1.56	11.16	58.12	216.89
Regret of `SpectralTS`	7.99	32.80	94.10	123.79
Regret of `SpectralUCB`	3.05	22.84	108.19	130.54

Table 4: The effect of smoothness on the cumulative regret for $T = 100$.

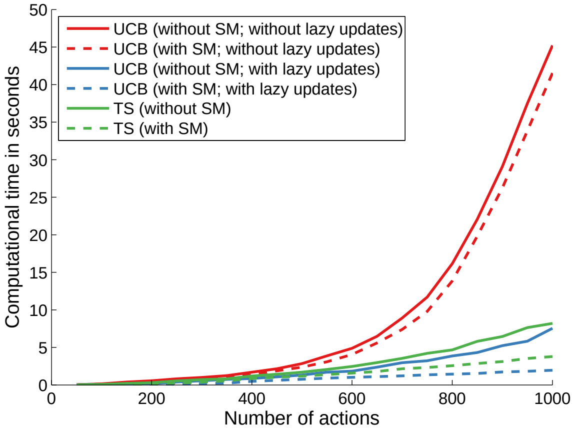


Figure 7: The impact of lazy updates and Sherman-Morrison formula on running time.

7.3 Computational complexity improvements

In general, the computation of N UCBs is computationally more expensive than sampling in TS. In Section 5.4, we discuss several possibilities to speed up algorithms. The impact of lazy updates for computing UCBs and effect of Sherman-Morrison formula on matrix inversion is demonstrated in Figure 7. The plot clearly shows that the lazy updates can improve the computation of UCBs to the point where the running time of SpectralUCB is comparable and in some cases even better than the running time of SpectralTS.

Another possible computational-time improvement, discussed in Section 5.4, can be made by extracting only the first $L \ll N$ eigenvectors of the graph Laplacian. First, the computational complexity of such operation is $\mathcal{O}(Lm \log m)$ where m is the number of edges. Second, the least-squares problem that we have to do in each round of the algorithm is only L-dimensional. In Figure 8 (right), we plot the cumulative regret and the total running time in seconds (log scale) of SpectralUCB for a single user from the MovieLens dataset. We vary L as 20, 200, and 2000 which corresponds to about 1%, 10%, and 100% of basis functions ($N = 2019$). The total running time also includes the computational savings from lazy updates and iterative matrix inversion. We see that with 10% of the eigenvectors, we achieve a similar performance as for the full set in the fraction of the running time.

7.4 MovieLens experiments

In this experiment, we take user preferences and the similarity graph over movies from the MovieLens dataset (Lam and Herlocker, 2012), a dataset of 6k users who rated one million movies. First, we extract a subset of 400 users and 618 movies with at least 500 ratings. Then we divide the dataset into three parts. The first is used to build our model of users, the rating that user i assigns to movie j. We factor the user-item matrix using low-rank matrix factorization (Keshavan et al., 2009) as $\mathbf{M} \approx \mathbf{U}\mathbf{V}'$, a standard approach to collaborative filtering. The rating that the user i assigns to movie j is estimated as $\widehat{r}_{i,j} = \langle \mathbf{u}_i, \mathbf{v}_j \rangle$, where $\mathbf{u}_i$ is the i-th row of $\mathbf{U}$ and $\mathbf{v}_j$ is the j-th row of $\mathbf{V}$. The rating $\widehat{r}_{i,j}$ is the payoff of pulling arm j when recommending to user i.

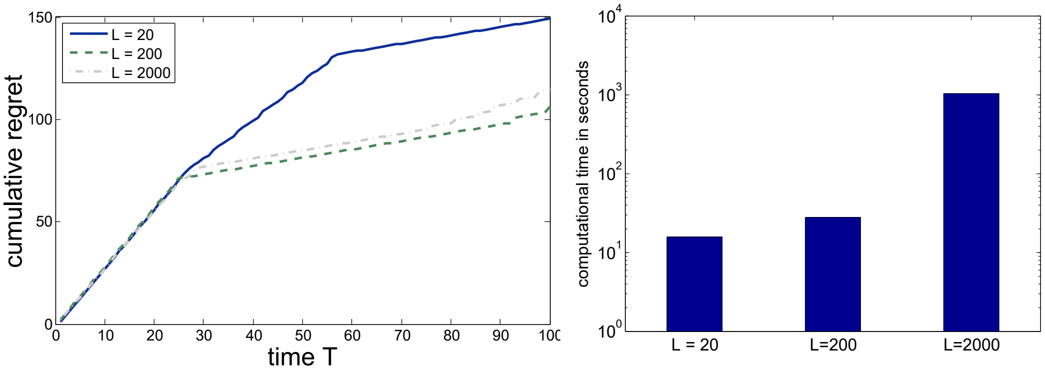


Figure 8: Cumulative regret and running time of SpectralUCB with reduced basis.

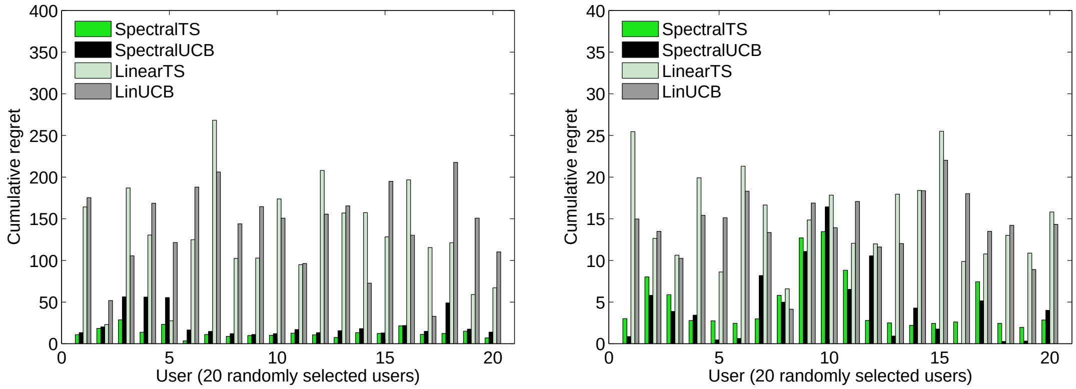
The second part of the dataset is used for parameter estimation. Similarly as for the first part, we complete ratings using low-rank factorization. The reason for using a different part of the dataset is to avoid dependencies.

The last part of the dataset is used to build our similarity graph over movies. We factor the dataset in the same way as the first two parts of the dataset. The graph contains an edge between movies i and i' if the movie i' is among 5 nearest neighbors of the movie i in the latent space of items $\mathbf{V}$. The weights on all edges are set to one. Notice that if two items are close in the item space, then their expected rating is expected to be similar. However, the opposite is not true. If two items have a similar expected rating, they do not have to be close in the item space. In other words, we take advantage of ratings but do not hardwire the two similarly-rated items to be similar.

Table 5 summarizes the best parameters learned on training part of the dataset. We use the parameters to run the algorithms on test part. Figure 9a shows 20 random users sampled from the testing part of the MovieLens dataset. We evaluate the regret of all four algorithms for $T = 500$ and compared the running time. We make few observations. First, spectral algorithms are consistently outperforming linear algorithms. Second, as we mention in Section 5.4, we use lazy updates for UCB algorithms which can improve the running time significantly. We see that in our experiment, the running time of UCB algorithms is better than the running time of TS algorithms even though in general, TS algorithms are computationally more efficient than UCB algorithms without lazy updates.

SpectralTS		SpectralUCB		LinearTS		LinUCB	
$\lambda = 0.001$	$v = 0.1$	$\lambda = 0.1$	$c = 1$	$\lambda = 100$	$v = 1$	$\lambda = 0.001$	$c = 0.001$

Table 5: The best-performing empirical parameters for Movielens.



(a) Movielens dataset, cumulative regret for 20 randomly selected users

(b) Flixster dataset, cumulative regret for 20 randomly selected users

Figure 9: Comparison of spectral and linear bandit algorithms for a subset of users.

7.5 Flixster experiments

We also perform experiments on users preferences from the movie recommendation website Flixster. The social network of the users was crawled by Jamali and Ester (2010) and then clustered by Graclus (2013) to obtain a strongly connected subgraph. Similarly as for Movielens, we extract a subset of users and movies, where each movie has at least 500 ratings. This results in a dataset of 972 movies and 1070 users. As with MovieLens, we complete the missing ratings by a low-rank matrix factorization and used it to construct a 5-NN similarity graph. For Figure 9b, we sample 20 random users and evaluate the regret of all four algorithms for $T = 50$. Similarly as for MovieLens, we set parameter λ to 0.01 while setting the parameter v of `SpectralTS` to be ten times smaller than the theoretical value.

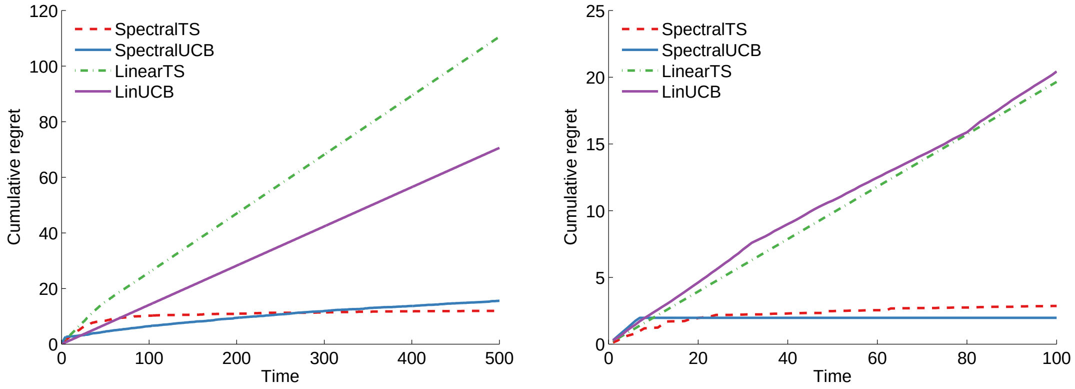
`SpectralTS`		`SpectralUCB`		`LinearTS`		`LinUCB`	
$\lambda = 0.01$	$v = 0.1$	$\lambda = 0.01$	$c = 0.11$	$\lambda = 1$	$v = 0.1$	$\lambda = 1$	$c = 1$

Table 6: The best-performing empirical parameters for Flixster.

7.6 Additional observations for improving the empirical performance

We give additional indications on how to improve the performance of the algorithms. This can be useful for the deployment of spectral bandits in practice.

- **Adjusting the number of edges in the graph.** Typically, the real-world datasets do not come with a graph structure. Therefore, we construct a nearest-neighbor graph which connects only the most similar actions. By reducing the number of neighbors, we are increasing the effective dimension (worsening the regret bound) and decreasing smoothness of the function (improving the regret bound). Finding a good trade-off



(a) Movielens dataset, cumulative regret for one random user

(b) Flixster dataset, cumulative regret for one random user

Figure 10: Comparison of spectral and linear bandit algorithms.

and adjusting the number of the edges can improve the performance of the algorithms significantly.

- **Scaling the confidence ellipsoid**, i.e., parameter c in SpectralUCB and parameter v in SpectralTS . Typically, the algorithms are too conservative and the bounds are too loose in order to include the worst-case case. Therefore, reducing the size of the confidence ellipsoid can sometimes improve the empirical performance of the algorithm at the price that some bounds might not hold anymore. In our experiments, we used the values for which the algorithms had good empirical performance.

- **The magnitude of regularization parameter λ.** By setting λ to a large value, all regularized eigenvalues become similar and therefore the algorithms take the graph structure into account less. On the other hand, if the regularization parameter λ is small, the algorithms depend on the graph structure more. Therefore, in order to leverage the graphs' structure, we have to find a good compromise while setting λ. In our experiments, we found that setting λ well was important and we tried several values of λ to pick the value with the best empirical performance.

- **Scaling the graph weights.** By scaling all the weights of the graph by some constant we scale the gap between the eigenvalues and therefore change the value of the effective dimension. Moreover, by scaling the weights we are also changing the smoothness of the reward function. Therefore, by simply scaling the weights we can make the graphs more useful for spectral bandits.

8. Conclusion

We presented spectral bandits inspired mostly by the applications in recommender systems and targeted advertisement in social networks. In this setting, we are asked to repeatedly maximize an unknown graph function, assumed to be smooth on a given similarity graph.

While standard linear bandits can be applied but their regret scales with the ambient dimension D, either linearly or as a square root, which can be very large.

Therefore, we introduced three algorithms, SpectralUCB, SpectralTS, and theoretically interesting SpectralEliminator. For all three algorithms, the regret bound only scales with the effective dimension d which is typically much smaller than D for real-world graphs. We also performed experiments and showed that spectral algorithms are able to leverage the structure of the problem when the reward function is smooth on the graph much better than their linear counterparts.

As two side results of independent interest, we provide the regret analysis of LinUCB with the upper bound of $\widetilde{\mathcal{O}}(D\sqrt{T})$ and define LinearEliminator for which we prove minimax-optimal regret bound of $\widetilde{\mathcal{O}}(\sqrt{DT})$. With adaptive confidence bounds and simpler analysis, LinearEliminator becomes a state-of-the-art algorithm among the ones with $\widetilde{\mathcal{O}}(\sqrt{DT})$ regret.

Acknowledgments

The authors wish to thank anynomous reviewers and Claudio Gentile for helpful suggestions. We are also grateful to András György for pointing out the connection between the effective dimension and the capacity formula of parallel Gaussian channels, and also for his suggestions on fixing the asymptotics of our lower bound.

The research presented in this paper was supported by French Ministry of Higher Education and Research, by European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n°270327 (project CompLACS), European CHIST-ERA project DELTA, French Ministry of Higher Education and Research, Inria and Otto-von-Guericke-Universität Magdeburg associated-team North-European project Allocate, Nord-Pas-de-Calais Regional Council, CPER Nord-Pas de Calais/FEDER DATA Advanced data science and technologies 2015-2020, French National Research Agency projects ExTra-Learn (n.ANR-14-CE24-0010-01) and BoB (n.ANR-16-CE23-0003), Intel Corporation, FMJH Program PGMO with the support to this program from CRITEO.

References

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

Marc Abeille and Alessandro Lazaric. Linear Thompson sampling revisited. In *International Conference on Artificial Intelligence and Statistics*, 2017.

Jacob D Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Conference on Learning Theory*, 2008.

Shipra Agrawal and Navin Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, 2012.

Shipra Agrawal and Navin Goyal. Further optimal regret bounds for Thompson sampling. In *International Conference on Artificial Intelligence and Statistics*, 2013a.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning*, 2013b.

Noga Alon, Nicolò Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. From bandits to experts: A tale of domination and independence. In *Neural Information Processing Systems*, 2013.

Noga Alon, Nicolò Cesa-Bianchi, Ofer Dekel, and Tomer Koren. Online learning with feedback graphs: Beyond bandits. In *Conference on Learning Theory*, 2015.

Noga Alon, Nicolò Cesa-Bianchi, Claudio Gentile, Shie Mannor, Yishay Mansour, and Ohad Shamir. Nonstochastic multi-armed bandits with graph-structured feedback. *SIAM Journal on Computing*, 46(6):1785–1826, 2017.

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

Peter Auer and Ronald Ortner. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 2010.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multi-armed bandit problem. *Journal on Computing*, 32(1):48–77, 2002.

Mikhail Belkin, Irina Matveeva, and Partha Niyogi. Regularization and semi-supervised learning on large graphs. In *Conference on Learning Theory*, 2004.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

Daniel Billsus, Michael J. Pazzani, and James Chen. A learning agent for wireless news access. In *International Conference on Intelligent User Interfaces*, 2000.

Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvári. $\mathcal{X}$-armed bandits. *Journal of Machine Learning Research*, 12:1587–1627, 2011.

Sébastien Bubeck, Nicolò Cesa-Bianchi, and Sham M. Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *Conference on Learning Theory*, 2012.

Swapna Buccapatnam, Atilla Eryilmaz, and Ness B. Shroff. Stochastic bandits with side observations on networks. In *International Conference on Measurement and Modeling of Computer Systems*, 2014.

Stéphane Caron, Branislav Kveton, Marc Lelarge, and Smriti Bhagat. Leveraging side observations in stochastic bandits. In *Uncertainty in Artificial Intelligence*, 2012.

Nicolò Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

Nicolò Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. A gang of bandits. In *Neural Information Processing Systems*, 2013.

Olivier Chapelle and Lihong Li. An empirical evaluation of Thompson sampling. In *Neural Information Processing Systems*. 2011.

Duen Horng Chau, Aniket Kittur, Jason I. Hong, and Christos Faloutsos. Apolo: Making sense of large network data by combining rich user interaction and machine learning. In *Conference on Human Factors in Computing Systems*, 2011.

Lei Chu, Lihong Li, Lev Reyzin, and Robert E Schapire. Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics*, 2011.

Richard Combes and Alexandre Proutière. Unimodal bandits: Regret lower bounds and optimal algorithms. In *International Conference on Machine Learning*, 2014.

Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. In *Conference on Learning Theory*, 2008.

Thomas Desautels, Andreas Krause, and Joel Burdick. Parallelizing exploration-exploitation tradeoffs in Gaussian process bandit optimization. In *International Conference on Machine Learning*, 2012.

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

Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. In *International Conference on Machine Learning*, 2014.

Claudio Gentile, Shuai Li, Purushottam Kar, Alexandros Karatzoglou, Giovanni Zappella, and Evans Etrue. On context-dependent clustering of bandits. In *International Conference on Machine Learning*, 2017.

Graclus. http://www.cs.utexas.edu/users/dml/software/graclus.html. *University of Texas*, 2013.

Quanquan Gu and Jiawei Han. Online spectral learning on a graph with bandit feedback. In *International Conference on Data Mining*, 2014.

Manjesh Hanawal, Venkatesh Saligrama, Michal Valko, and Rémi Munos. Cheap bandits. In *International Conference on Machine Learning*, 2015.

Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Conference on Recommender systems*, 2010.

Dietmar Jannach, Markus Zanker, Alexander Felfernig, and Gerhard Friedrich. *Recommender systems: An introduction*. Cambridge University Press, 2010.

Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. *Algorithmic Learning Theory*, 2012.

Raghunandan Keshavan, Sewoong Oh, and Andrea Montanari. Matrix completion from a few entries. In *International Symposium on Information Theory*, 2009.

Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Multi-armed bandit problems in metric spaces. In *Symposium on Theory Of Computing*, 2008.

Tomáš Kocák, Gergely Neu, Michal Valko, and Rémi Munos. Efficient learning by implicit exploration in bandit problems with side observations. In *Neural Information Processing Systems*, 2014a.

Tomáš Kocák, Michal Valko, Rémi Munos, and Shipra Agrawal. Spectral Thompson sampling. In *AAAI Conference on Artificial Intelligence*, 2014b.

Tomáš Kocák, Gergely Neu, and Michal Valko. Online learning with noisy side observations. In *International Conference on Artificial Intelligence and Statistics*, 2016a.

Tomáš Kocák, Gergely Neu, and Michal Valko. Online learning with Erdős-Rényi side-observation graphs. In *Uncertainty in Artificial Intelligence*, 2016b.

Nathan Korda, Balázs Szörényi, and Shuai Li. Distributed clustering of linear bandits in peer to peer networks. In *International Conference on Machine Learning*, 2016.

Ioannis Koutis, Gary L. Miller, and David Tolliver. Combinatorial preconditioners and multilevel solvers for problems in computer vision and image processing. *Computer Vision and Image Understanding*, 115(12):1638–1646, 2011.

Shyong Lam and Jon Herlocker. `http://www.grouplens.org/node/12`. *MovieLens 1M dataset*, 2012.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. *International World Wide Web Conference*, 2010.

Shuai Li, Alexandros Karatzoglou, and Claudio Gentile. Collaborative filtering bandits. In *Conference on Research and Development in Information Retrieval*, 2016.

Yifei Ma, Tzu-Kuo Huang, and Jeff Schneider. Active search and bandits on graphs using sigma-optimality. In *Uncertainty in Artificial Intelligence*, 2015.

Shie Mannor and Ohad Shamir. From bandits to experts: On the value of side-observations. In *Neural Information Processing Systems*, 2011.

Benedict C. May, Nathaniel Korda, Anthony Lee, and David S. Leslie. Optimistic Bayesian sampling in contextual-bandit problems. *Journal of Machine Learning Research*, 13(1): 2069–2106, 2012.

Miller McPherson, Lynn Smith-Lovin, and James Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–444, 2001.

Sunil K. Narang, Akshay Gadde, and Antonio Ortega. Signal processing techniques for interpolation in graph structured data. In *International Conference on Acoustics, Speech and Signal Processing*, 2013.

Aleksandrs Slivkins. Contextual bandits with similarity information. In *Conference on Learning Theory*, 2009.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *International Conference on Machine Learning*, 2010.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25:285–294, 1933.

Michal Valko. *Bandits on graphs and structures*. habilitation, École normale supérieure de Cachan, 2016.

Michal Valko, Branislav Kveton, Ling Huang, and Daniel Ting. Online semi-supervised learning on quantized graphs. In *Uncertainty in Artificial Intelligence*, 2010.

Michal Valko, Nathan Korda, Rémi Munos, Ilias Flaounas, and Nelo Cristianini. Finite-time analysis of kernelised contextual bandits. In *Uncertainty in Artificial Intelligence*, 2013.

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

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

Martin Wainwright. STAT 210B Advanced mathematical statistics. Lecture notes, University of California at Berkeley, 2015.

Jia Yuan Yu and Shie Mannor. Unimodal bandits. In *International Conference on Machine Learning*, 2011.

Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, University of Wisconsin-Madison, 2008.