Cheap Bandits

Manjesh Kumar Hanawal MHANAWAL@BU.EDU
Department of ECE, Boston University, Boston, Massachusetts, 02215 USA

Venkatesh Saligrama SRV@BU.EDU
Department of ECE, Boston University, Boston, Massachusetts, 02215 USA

Michal Valko MICHAL.VALKO@INRIA.FR
INRIA Lille - Nord Europe, SequeL team, 40 avenue Halley 59650, Villeneuve d'Ascq, France

Rémi Munos REMI.MUNOS@INRIA.FR
INRIA Lille - Nord Europe, SequeL team, France and Google DeepMind, United Kingdom

Abstract

We consider stochastic sequential learning problems where the learner can observe the *average reward of several actions*. Such a setting is interesting in many applications involving monitoring and surveillance, where the set of the actions to observe represent some (geographical) area. The importance of this setting is that in these applications, it is actually *cheaper* to observe average reward of a group of actions rather than the reward of a single action. We show that when the reward is *smooth* over a given graph representing the neighboring actions, we can maximize the cumulative reward of learning while *minimizing the sensing cost*. In this paper we propose CheapUCB, an algorithm that matches the regret guarantees of the known algorithms for this setting and at the same time guarantees a linear cost again over them. As a by-product of our analysis, we establish a $\Omega(\sqrt{dT})$ lower bound on the cumulative regret of spectral bandits for a class of graphs with effective dimension d.

1. Introduction

In many online learning and *bandit problems*, the learner is asked to select a *single action* for which it obtains a (possibly contextual) feedback. However, in many scenarios such as surveillance, monitoring and exploration of a large area or network, it is often cheaper to obtain an average reward for a *group of actions* rather than a reward for a single one. In this paper, we therefore study *group actions* and formalize this setting as *cheap bandits* on graph structured data. Nodes and edges in our graph model the geometric structure of the data and we associate signals (rewards) with each node. We are interested in problems where the actions are a *collection of nodes*. Our objective is to locate nodes with largest rewards.

The cost-aspect of our problem arises in *sensor networks* (SNETs) for target localization and identification. In SNETs sensors have limited sensing range (Ermis & Saligrama, 2010; 2005) and can reliably sense/identify targets only in their vicinity. To conserve battery power, sleep/awake scheduling is used (Fuemmeler & Veeravalli, 2008; Aeron et al., 2008), wherein a *group of sensors* is woken up sequentially based on probable locations of target. The *group of sensors* minimize transmit energy through coherent beamforming of sensed signal, which is then received as an average reward/signal at the receiver. While coherent beam forming is cheaper, it nevertheless increases target ambiguity since the sensed field degrades with distance from target. A similar scenario arises in aerial reconnaissance as well: Larger areas can be surveilled at higher altitudes more quickly (cheaper) but at the cost of more target ambiguity.

Moreover, sensing average rewards through group actions, in the initial phases, is also meaningful. Rewards in many applications are typically *smooth band-limited graph signals* (Narang et al., 2013) with the sensing field decaying smoothly with distance from the target. In addition to SNETs (Zhu & Rabbat, 2012), smooth graph signals also arise in *social networks* (Girvan & Newman, 2002), and *recommender systems*. Signals on graphs is an emerging area in *signal processing* (SP) but the emphasis is on reconstruction through sampling and interpolation from a

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

small subset of nodes (Shuman et al., 2013). In contrast, our goal is in locating the maxima of graph signals rather than reconstruction. Nevertheless, SP does provide us with the key insight that whenever the graph signal is smooth, we can obtain information about a location by sampling its neighborhood.

Our approach is to sequentially discover the nodes with optimal reward. We model this problem as an instance of *linear bandits* (Auer, 2002; Dani et al., 2008; Li et al., 2010) that links the reward of nodes through an unknown parameter. A bandit setting for smooth signals was recently studied by Valko et al. (2014), however *neglecting the signal cost*. While typically bandit algorithms aim to minimize the regret, we aim to minimize *both regret and the signal cost*. Nevertheless, we do not want to tradeoff the regret for cost. In particular, we are not compromising regret for cost, neither we seek a Pareto frontier of two objectives. We seek algorithms that minimize the cost of sensing and at the same time attain, the state-of-the-art regret guarantees.

Notice that our setting directly generalizes the traditional setting with single action per time step as the arms themselves are graph signals. We define cost of each arm in terms of their *graph Fourier transform*. The cost is quadratic in nature and assigns higher cost to arms that collect average information from a smaller set of neighbors. Our goal is to collect higher reward from the nodes while keeping the total cost small. However, there is a tradeoff in choosing low cost signals and higher reward collection: The arms collecting reward from individual nodes cost more, but give more specific information about node's reward and hence provide better estimates. On other hand, arms that collect average reward from subset of its neighbors cost less, but only give crude estimate of the reward function. In this paper, we develop an algorithm maximizing the reward collection while keeping the cost low.

2. Related Work

There are several other bandit and online learning settings that consider costs (Tran-Thanh et al., 2012; Badanidiyuru et al., 2013; Ding et al., 2013; Badanidiyuru et al., 2014; Zolghadr et al., 2013; Cesa-Bianchi et al., 2013a). The first set is referred to as *budgeted bandits* (Tran-Thanh et al., 2012) or *bandits with knapsacks* (Badanidiyuru et al., 2013), where each single arm is associated with a cost. This cost can be known or unknown (Ding et al., 2013) and can depend on a given context (Badanidiyuru et al., 2014). The goal there is in general to minimize the regret as a function of budget instead of time or to minimize regret under budget constraints, where there is no advantage in not spending all the budget. Our goal is different as we care both about minimizing the budget and minimizing the regret as a function of time. Another cost setting considers cost for observing features from which the learner can build its prediction (Zolghadr et al., 2013). This is different from our consideration of cost, which is inversely proportional to the sensing area. Finally, in the adversarial setting (Cesa-Bianchi et al., 2013a), considers cost for switching actions.

The most related graph bandits setting to ours is by Valko et al. (2014) on which we build this paper. Another graph bandit setting considers side information, when the learner obtains besides the reward of the node it chooses, also the rewards of the neighbors (Mannor & Shamir, 2011; Alon et al., 2013; Caron et al., 2012; Kocák et al., 2014). Finally a different graph bandit setup is gang of (multiple) bandits considered in (Cesa-Bianchi et al., 2013b) and online clustering of bandits in (Gentile et al., 2014).

Our main contribution is the incorporation of *sensing cost* into learning in linear bandit problems while simultaneously minimizing two performance metrics: cumulative regret and the cumulative sensing cost. We develop CheapUCB, the algorithm that guarantees regret bound of the order $d\sqrt{T}$, where d is the *effective dimension* and T is the number of rounds. This regret bound is of the same order as SpectralUCB (Valko et al., 2014) that does not take cost into consideration. However, we show that our algorithm provides a cost saving that is linear in T compared to the cost of SpectralUCB. The effective dimension d that appears in the bound is a dimension typically smaller in real-world graphs as compared to number of nodes N. This is in contrast with linear bandits that can achieve in this graph setting the regret of $N\sqrt{T}$ or $\sqrt{NT}$. However, our ideas of cheap sensing are directly applicable to the linear bandit setting as well. As a by-product of our analysis, we establish a $\Omega(\sqrt{dT})$ lower bound on the cumulative regret for a class of graphs with effective dimension d.

3. Problem Setup

Let $G = (\mathcal{V}, \mathcal{E})$ denote an undirected graph with number of nodes $|\mathcal{V}| = N$. We assume that degree of all the nodes is bounded by κ. Let $\mathbf{s} : \mathcal{V} \rightarrow \mathcal{R}$ denote a signal on G, and $\mathcal{S}$ the set of all possible signals on $\mathcal{G}$. Let $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ denote the unnormalized Laplacian of the graph G, where $\boldsymbol{A} = \{a_{ij}\}$ is the adjacency matrix and $\boldsymbol{D}$ is the diagonal matrix with $\boldsymbol{D}_{ii} = \sum_j a_{ij}$. We emphasize that our main results extend to weighted graphs if we replace the matrix $\boldsymbol{A}$ with the edge weight matrix $\boldsymbol{W}$. We work with matrix $\boldsymbol{A}$ for simplicity of exposition. We denote the eigenvalues of $\boldsymbol{L}$ as $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_N$, and the corresponding eigenvectors as $\mathbf{q_1}, \mathbf{q_2}, \cdots, \mathbf{q_N}$. Equivalently, we write $\boldsymbol{L} = \boldsymbol{Q}\Lambda_{\mathcal{L}}Q'$, where $\boldsymbol{\Lambda}_{\mathcal{L}} = diag(\lambda_1, \lambda_2, \cdots, \lambda_N)$ and $\mathbf{Q}$ is the $N \times N$ orthonormal matrix with eigenvectors in columns. We denote transpose of $\mathbf{a}$ as $\mathbf{a}'$, and all vectors are by default column vectors. For a given matrix $\boldsymbol{V}$, we denote $\boldsymbol{V}$-norm of a vector $\mathbf{a}$ as $\|\mathbf{a}\|_V = \sqrt{\mathbf{a}'\boldsymbol{V}\mathbf{a}}$.

3.1. Reward function

We define a reward function on a graph G as a linear combination of the eigenvectors. For a given parameter vector $\boldsymbol{\alpha} \in \mathcal{R}^N$, let $\boldsymbol{f_\alpha} : \mathcal{V} \to \mathcal{R}$ denote the reward function on the nodes defined as

$$\boldsymbol{f_\alpha} = \boldsymbol{Q\alpha}.$$

The parameter $\boldsymbol{\alpha}$ can be suitably penalized to control the smoothness of the reward function. For instance, if we choose $\boldsymbol{\alpha}$ such that large coefficients correspond to the eigenvectors associated with small eigenvalues then $f_{\boldsymbol{\alpha}}$ is a smooth function of G (Belkin et al., 2008). We denote the *unknown* parameter that defines the true reward function as $\boldsymbol{\alpha}^*$. We denote the reward of node i as $\boldsymbol{f}_{\boldsymbol{\alpha}^*}(i)$.

In our setting, the arms are nodes *and* the subsets of their neighbors. When an arm is selected, we observe only the average of the rewards of the nodes selected by that arm. To make this notion formal, we associate arms with *probe signals* on graphs.

3.2. Probes

Let $\mathcal{S} \subseteq \left\{\mathbf{s} \in [0,1]^N : \sum_{i=1}^N s_i = 1\right\}$ denote the set of probes. We use the word probe and action interchangeably. A probe is a signal with its width corresponding to the support of the signal s. For instance, it could correspond to the region-of-coverage or region-of-interest probed by a radar pulse. Thus each $\mathbf{s} \in \mathcal{S}$ is of the form $s_i = 1/\text{supp}(\mathbf{s})$, for all $i = 1, 2, \cdots, N$, where supp$(\mathbf{s})$ denotes the number of positive elements in $\mathbf{s}$. The inner product of $\boldsymbol{f}_{\boldsymbol{\alpha}^*}$ and a probe $\mathbf{s}$ is the average reward of supp$(\mathbf{s})$ number of nodes.

We parametrize a probe in terms of its width $w \in [N]$ and let the set of probes of width w to be $\tilde{\mathcal{S}}_w = \{\mathbf{s} \in \mathcal{S} : \text{supp}(\mathbf{s}) = w\}$. For a given $w > 0$, our focus in this paper is on probes with uniformly weighted components, which are limited to neighborhoods of each node on the graph. We denote the collection of these probes as $\mathcal{S}_w \subset \tilde{\mathcal{S}}_w$, which has N elements. We denote the element in $\mathcal{S}_w$ associated with node i as $\boldsymbol{s}_i^w$. Suppose node i has neighbors at $\{j_1, j_2, \cdots j_{w-1}\}$, then $\boldsymbol{s}_i^w$ is described as:

$$s_{ik}^w = \begin{cases} 1/w & \text{if } k = i \\ 1/w & \text{if } \ k = j_i, \ \ i = 1, 2, \cdots, w-1 \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

If node i has more than w neighbors, there can be multiple ways to define $\mathbf{s}_i^w$ depending on the choice of its neighbors. When w is less than degree of node i, in defining $\mathbf{s}_i^w$ we only consider neighbors with larger edge weights. If all the weights are the same, then we select w neighbors arbitrarily. Note that $|\mathcal{S}_w| = N$ for all w. In the following we write 'probing with $\mathbf{s}$' to mean that $\mathbf{s}$ is used to get information from nodes of graph G.

We define the arms as the set

$$\mathcal{S}_D := \{\mathcal{S}_w : w = 1, 2, \cdots, N\}.$$

Compared to multi-arm and linear bandits, the number of arms K is $\mathcal{O}(N^2)$ and the contexts have dimension N.

3.3. Cost of probes

The cost of the arms are defined using the spectral properties of their associated graph probes. Let $\tilde{\mathbf{s}}$ denote the *graph Fourier transform* (GFT) of probe $\mathbf{s} \in \mathcal{S}$. Analogous to Fourier transform of a continuous function, GFT gives amplitudes associated with graph frequencies. The GFT coefficient of a probe on frequency $\lambda_i, i = 1, 2 \cdots, N$ is obtained by projecting it on $\mathbf{q}_i$, i.e.,

$$\tilde{\mathbf{s}} = \mathbf{Q}'\mathbf{s},$$

where $\tilde{s}_i, i = 1, 2, \cdots, N$ is the GFT coefficient associated with frequency λ_i. Let $C : \mathcal{S} \to \mathcal{R}_+$ denote the cost function. Then the cost of the probe $\mathbf{s}$ is described by

$$C(\mathbf{s}) = \sum_{i \sim j} (s_i - s_j)^2,$$

where the summation is over all the unordered node pairs $\{i, j\}$ for which node i is adjacent to node j. We motivate this cost function from the SNET perspective where probes with large width are relatively cheap. We first observe that the cost of a constant probe is zero. For a probe, $\mathbf{s}_i^w \in \mathcal{S}_w$, of width w it follows that[1],

$$C(\mathbf{s}_i^w) = \frac{w-1}{w^2}\left(1 - \frac{1}{N}\right) + \frac{1}{w^2}. \tag{2}$$

Note that the cost of w- width probe associated with node i depends only on its width w. For $w = 1$, $C(\mathbf{s}_i^1) = 1$ for all $i = 1, 2, \cdots, N$. That is, the cost of probing individual nodes of the graph is the same. Also note that $C(\mathbf{s}_i^w)$ is decreasing in w, implying that probing a node is more costly than probing a subset of its neighbors.

Alternatively, we can associate probe costs with eigenvalues of the graph Laplacian. Constant probes corresponds to the zero eigenvalue of the graph Laplacian. More generally, we see that,

$$C(\mathbf{s}) = \sum_{i \sim j} (s_i - s_j)^2 = \mathbf{s}'\mathcal{L}\mathbf{s} = \sum_{i=1}^N \lambda_i \tilde{s}_i^2 = \tilde{\mathbf{s}}'\boldsymbol{\Lambda}_{\mathcal{L}}\tilde{\mathbf{s}}.$$

It follows that $C(\mathbf{s}) = \|\mathbf{s}\|_{\mathcal{L}}^2$. The operation of pulling an arm and observing a reward is equivalent to probing the

[1] We symmetrized the graph by adding self loops to all the nodes to make their degree (number of neighbors) N, and normalized the cost by N.

graph with a probe. This results in a value that is the inner product of the probe signal and graph reward function. We write the reward in the probe space $\mathcal{S}_D$ as follows. Let $F_G : \mathcal{S} \to \mathcal{R}$ defined as

$$F_G(\mathbf{s}) = \mathbf{s}'\boldsymbol{Q}\boldsymbol{\alpha}^* = \tilde{\mathbf{s}}'\boldsymbol{\alpha}^*$$

denote the reward obtained from probe $\mathbf{s}$. Thus, each arm gives a reward that is linear, and has quadratic cost, in its GFT coefficients. In terms of the linear bandit terminology, the GFT coefficients in $\mathcal{S}_D$ constitute the set of arms.

With the rewards defined in terms of the probes, the optimization of reward function is over the action space. Let $\mathbf{s}_* = \arg\max_{\mathbf{s}\in\mathcal{S}_D} F_G(\mathbf{s})$ denote the probe that gives the maximum reward. This is a straightforward linear optimization problem if the function parameter $\boldsymbol{\alpha}^*$ is known. When $\boldsymbol{\alpha}^*$ is unknown we can learn the function through a sequence of measurements.

3.4. Learning setting and performance metrics

Our learning setting is the following. The learner uses a policy $\pi : \{1, 2, \cdots, T\} \to \mathcal{S}_D$ that assigns at step $t \le T$, probe $\pi(t)$. In each step t, the recommender incurs a cost $C(\pi(t))$ and obtains a noisy reward such that

$$r_t = F_G(\pi(t)) + \varepsilon_t,$$

where ε_t is independent *R-sub Gaussian* for any t.

The cumulative regret of policy π is defined as

$$R_T = TF_G(\mathbf{s}_*) - \sum_{t=1}^{T} F_G(\pi(t)) \qquad (3)$$

and the total cost incurred up to time T is given by

$$C_T = \sum_{t=1}^{T} C(\pi(t)). \qquad (4)$$

The goal of the learner is to learn a policy π that minimizes total cost C_T while keeping the cumulative (pseudo) regret R_T as low as possible.

Node vs. Group actions: The set $\mathcal{S}_D$ allows actions that can probe a node (node-action) or a subset of nodes (group-action). Though the group actions have smaller cost, they only provide average reward information for the selected nodes. In contrast, node actions provide crisper information of the reward for the selected node, but at a cost premium. Thus, an algorithm that uses only node actions can provide a better regret performance compared to the one that takes group actions. But if the algorithms use only node actions, the cumulative cost can be high.

In the following, we first state the regret performance of the SpectralUCB algorithm (Valko et al., 2014) that uses only node actions. We then develop an algorithm that aims to achieve the same order of regret using group actions and reducing the total sensing cost.

4. Node Actions: Spectral Bandits

If we restrict the action set to $\mathcal{S}_D = \{\mathbf{e}_i : i = 1, 2, \cdots, n\}$, where $\mathbf{e}_i$ denotes a binary vector with i^{th} component set to 1 and all the other components set to 0, then only node actions are allowed in each step. In this setting, the cost is the same for all the actions, i.e., $C(\mathbf{e_i}) = 1$ for all i.

Using these node actions, Valko et al. (2014) developed SpectralUCB that aims to minimize the regret under the assumption that the reward function is smooth. The smoothness condition is characterized as follows:

$$\exists \;\; c > 0 \;\; \text{such that} \;\; \|\boldsymbol{\alpha^*}\|_{\boldsymbol{\Lambda}} \le c. \qquad (5)$$

Here $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}_{\mathcal{L}} + \lambda I$, and $\lambda > 0$ is used to make $\boldsymbol{\Lambda}_{\mathcal{L}}$ invertible. The bound c characterizes the smoothness of the reward. When c is small, the rewards on the neighboring nodes are more similar. In particular, when the reward function is a constant, then $c = 0$. To characterize the regret performance of SpectralUCB, Valko et al. (2014) introduced the notion of *effective dimension* defined as follows:

Definition 1 (Effective dimension) *For graph G, let us denote $\lambda = \lambda_1 \le \lambda_2 \cdots \le \lambda_N$ the diagonal elements of $\boldsymbol{\Lambda}$. Given T, effective dimension is the largest d such that:*

$$(d-1)\lambda_d \le \frac{T}{\log(T/\lambda + 1)} < d\lambda_{d+1}. \qquad (6)$$

Theorem 1 *(Valko et al., 2014) The cumulative regret of SpectralUCB is bounded with probability at least $1 - \delta$ as:*

$$R_T \le \left(8R\sqrt{d\log(1+T/\lambda) + 2\log(1/\delta)} + 4c\right) \times \sqrt{dT\log(1+T/\lambda)},$$

Lemma 1 *The total cost of the SpectralUCB is $C_T = T$.*

Note that effective dimension depends on T and also on how fast the eigenvalues grow. The regret performance of SpectralUCB is good when d is small, which occurs when the eigenspectrum exhibits large gaps. For these situations, SpectralUCB performance has a regret that scales as $O(d\sqrt{T})$ for a large range of values of T. To see this, notice that in relation (6) when λ_{d+1}/λ_d is large, the value of effective dimension remains unchanged over a large range of T implying that the regret bound of $O(d\sqrt{T})$ is valid for a large range of values of T with the same d.

There are many graphs for which the effective dimension is small. For example, random graphs are good expanders for which eigenvalues grow fast. Another setting are stochastic block models (Girvan & Newman, 2002), that exhibit large eigenvalue gap and are popular in the analysis of social, biological, citation, and information networks.

5. Group Actions: Cheap Bandits

Recall (Section 3.3) that group actions are cheaper than the node actions. Furthermore, that the cost of group actions is decreasing in group size. In this section, we develop a learning algorithm that aims to minimize the total cost without compromising on the regret using group actions. Specifically, given T and a graph with effective dimension d our objective is as follows:

$$\min_{\pi} C_T \quad \text{subject to} \quad R_T \lesssim d\sqrt{T}. \tag{7}$$

where optimization is over policies defined on the action set $\mathcal{S}_D$ given in subsection 3.2.

5.1. Lower bound

The action set used in the above optimization problem is larger than the set used in the SpectralUCB. This raises the question of whether or not the regret order of $d\sqrt{T}$ is too loose particularly when SpectralUCB can realize this bound using a much smaller set of probes.

In this section we derive a $\sqrt{dT}$ lower bound on the expected regret (worst-case) for any algorithm using action space $\mathcal{S}_D$ on graphs with effective dimension d. While this implies that our target in (7) should be $\sqrt{dT}$, we follow Valko et al. (2014) and develop a variation of SpectralUCB that obtains the target regret of $d\sqrt{T}$. We leave it as a future work to develop an algorithm that meets the target regret of $\sqrt{dT}$ while minimizing the cost.

Let $\mathcal{G}_d$ denote a set of graphs with effective dimension d. For a given policy $\pi, \boldsymbol{\alpha}^*, T$ and graph G. Define expected cumulative reward as

$$Regret(T, \pi, \boldsymbol{\alpha}^*, G) = \mathbb{E}\left[\sum_{t=1}^{T} \tilde{\boldsymbol{s}}_* \boldsymbol{\alpha}^* - \tilde{\boldsymbol{s}}_t \boldsymbol{\alpha}^* \middle| \boldsymbol{\alpha}^*\right]$$

where $\tilde{\boldsymbol{s}}_t = \pi'(t)\boldsymbol{Q}$.

Proposition 1 *For any policy π and time period T, there exists a graph $G \in \mathcal{G}_d$ and a $\boldsymbol{\alpha}^* \in \mathcal{R}^d$ representing a smooth reward such that*

$$Regret(T, \pi, \boldsymbol{\alpha}^*, G) = \Omega(\sqrt{dT})$$

The proof follows by construction of a graph with d disjoint cliques and restricting the rewards to be piecewise constant on the cliques. The problem then reduces to identifying the clique with the highest reward. We then reduce the problem to the multi-arm case, using Theorem 5.1 of Auer et al. (2003) and lower bound the minimax risk. See the supplementary material for a detailed proof.

5.2. Local smoothness

In this subsection we show that a smooth reward function on a graph with low effective dimension implies local smoothness of the reward function around each node. Specifically, we establish that the average reward around the neighborhood of a node provides good information about the reward of the node itself. Then, instead of probing a node, we can use group actions to probe its neighborhood and get good estimates of the reward at low cost.

From the discussion in Section 4, when d is small and there is a large gap between the λ_d and λ_{d+1}, SpectralUCB enjoys a small bound on the regret for a large range of values in the interval $[(d-1)\lambda_d,\, d\lambda_{d+1}]$. Intuitively, a large gap between the eigenvalues implies that there is a good partitioning of the graph into tight clusters. Furthermore, the smoothness assumption implies that the reward of a node and its neighbors within each cluster are similar.

Let $\mathcal{N}_i$ denote a set of neighbors of node i. The following result provides a relation between the reward of node i and the average reward from $\mathcal{N}_i$ of its neighbors.

Proposition 2 *Let d denote the effective dimension and $\lambda_{d+1}/\lambda_d \geq \mathcal{O}(d^2)$. Let $\boldsymbol{\alpha}^*$ satisfy (5). For any node i*

$$\left| \boldsymbol{f}_{\boldsymbol{\alpha}^*}(i) - 1/\mathcal{N}_i \sum_{j \in \mathcal{N}_i} \boldsymbol{f}_{\boldsymbol{\alpha}^*}(j) \right| \leq c' d/\lambda_{d+1} \tag{8}$$

for all $\mathcal{N}_i$, and $c' = 56\kappa\sqrt{2\kappa}c$.

The full proof is given in the supplementary material. It is based on k-way expansion constant together with bounds on higher order Cheeger inequality (Gharan & Trevisan, 2014). Note that (8) holds for all i. However, we only need this to hold for the node with the optimal reward to establish regret performance our algorithm. We rewrite (8) for the optimal i^* node using group actions as follows:

$$|F_G(\boldsymbol{s}_*) - F_G(\boldsymbol{s}_*^w)| \leq c' d/\lambda_{d+1} \quad \text{for all} \quad w \leq |\mathcal{N}_{i^*}|. \tag{9}$$

Though we give the proof of the above result under the technical assumption $\lambda_{d+1}/\lambda_d \geq \mathcal{O}(d^2)$, it holds in cases where eigenvalues grow fast. For example, for graphs with strong connectivity property this inequality is trivially satisfied. We can show that $|F_G(\mathbf{s}_*) - F_G(\mathbf{s}_*^w)| \leq c/\sqrt{\lambda_2}$ through a standard application of Cauchy-Schwartz inequality. For the model of Barabási-Albert we get $\lambda_2 = \Omega(N^\gamma)$ with $\gamma > 0$ and for the cliques we get $\lambda_2 = N$.

General graphs: When λ_{d+1} is much larger than λ_d, the above proposition gives a tight relationship between the optimal reward and the average reward from its neighborhood. However, for general graphs this eigenvalue gap assumption is not valid. Motivated by (9), we assume that the smooth reward function satisfies the following weaker version for the general graphs. For all $w \leq |\mathcal{N}_{i^*}|$

$$|F_G(\boldsymbol{s}_*) - F_G(\boldsymbol{s}_*^w)| \leq c'\sqrt{T}w/\lambda_{d+1}. \qquad (10)$$

These inequalities get progressively weaker in T and w and can be interpreted as follows. For small values of T, we have few rounds for exploration and require stronger assumptions on smoothness. On the other hand, as T increases we have the opportunity to explore and consequently the inequalities are more relaxed. This relaxation of the inequality as a function of the width w characterizes the fact that close neighborhoods around the optimal node provide better information about the optimal reward than a wider neighborhood.

5.3. Algorithm: CheapUCB

Below we present an algorithm similar to LinUCB (Li et al., 2010) and SpectralUCB (Valko et al., 2014) for regret minimization. The main difference between our algorithm and the SpectralUCB algorithm is the enlarged action space, which allows for selection of subsets of nodes and associated realization of average rewards. Note that when we probe a specific node instead of probing a subset of nodes, we get a more precise information (though noisy) about the node, but this results in higher cost.

As our goal is to minimize the cost while maintaining a low regret, we handle this requirement by moving sequentially from the least costly probes to expensive ones as we progress. In particular, we split the time horizon into J stages, and as we move from state j to $j+1$ we use more expensive probes. That means, we use probes with smaller widths as we progress through the different stages of learning. The algorithm uses the probes of different widths in each stage as follows. Stage $j = 1, \ldots, J$ consists of time steps from 2^{j-1} to $2^j - 1$ and uses of probes of weight j only.

At each time step $t = 1, 2, \ldots, T$, we estimate the value of $\boldsymbol{\alpha}^*$ by using l^2-regularized least square as follows. Let $\{\mathbf{s}_i := \pi(i), i = 1, 2, \ldots, t\}$ denote the probe selected till time t and $\{r_i, i = 1, 2, \ldots, t\}$ denote the corresponding rewards. The estimate of $\boldsymbol{\alpha}^*$ denoted $\hat{\boldsymbol{\alpha}}_t$ is computed as

$$\hat{\boldsymbol{\alpha}}_t = \arg\min_{\boldsymbol{\alpha}} \left(\sum_{i=1}^{t} \left[\mathbf{s}_i'\boldsymbol{Q}\boldsymbol{\alpha} - r_t\right]^2 + \|\boldsymbol{\alpha}\|_{\boldsymbol{\Lambda}}^2 \right).$$

Algorithm 1 CheapUCB

```
**Input:**
$G$: graph
$T$: number of steps
$\lambda, \delta$: regularization and confidence parameters
$R, c$: upper bound on noise and norm of $\boldsymbol{\alpha}$
**Initialization:**
$d \leftarrow \arg\max\{d : (d-1)\lambda_d \leq T/\log(1+T/\lambda)\}$
$\beta \leftarrow 2R\sqrt{d\log(1+T/\lambda) + 2\log(1/\delta)} + c$
$\boldsymbol{V}_0 \leftarrow \boldsymbol{\Lambda}_L + \lambda\boldsymbol{I}, \boldsymbol{S}_0 \leftarrow 0, r_0 \leftarrow 0$
for $j = 1 \to J$ do
  for $t = 2^{j-1} \to \min\{2^j - 1, T\}$ do
    $\boldsymbol{S}_t \leftarrow \boldsymbol{S}_{t-1} + r_{t-1}\tilde{\boldsymbol{s}}_{t-1}$
    $\boldsymbol{V}_t \leftarrow \boldsymbol{V}_{t-1} + \tilde{\mathbf{s}}_{t-1}\tilde{\mathbf{s}}_{t-1}'$
    $\hat{\boldsymbol{\alpha}}_t \leftarrow \boldsymbol{V}_t^{-1}\boldsymbol{S}_t$
    $\boldsymbol{s}_t \leftarrow \arg\max_{\boldsymbol{s}\in\,\mathcal{S}_{J-j+1}} \left(\tilde{\mathbf{s}}'\hat{\boldsymbol{\alpha}}_t + \beta\|\tilde{\mathbf{s}}\|_{\boldsymbol{V}_t^{-1}}\right)$
  end for
end for
```

Theorem 2 *Set $J = \lceil \log T \rceil$ in the algorithm. Let* d *be the effective dimension and λ be the smallest eigenvalue of Λ. Let $\tilde{\mathbf{s}}_t'\boldsymbol{\alpha}^* \in [-1, 1]$ for all $\mathbf{s} \in \mathcal{S}$, the cumulative regret of the algorithm is with probability at least $1-\delta$ bounded as:*

(i) If (5) holds and $\lambda_{d+1}/\lambda_d \geq \mathcal{O}(d^2)$, then

$$R_T \leq (8R\sqrt{d\log(1+T/\lambda) + 2\log(1/\delta)} + 4c) \times \sqrt{dT\log(1+T/\lambda)} + c'd^2\log_2(T/2)\log(T/\lambda+1),$$

(ii) If (5) and (10) hold, then

$$R_T \leq (8R\sqrt{d\log(1+T/\lambda) + 2\log(1/\delta)} + 4c) \times \sqrt{dT\log(1+T/\lambda)} + c'd\sqrt{T/4}\log_2(T/2)\log(T/\lambda+1),$$

Moreover, the cumulative cost of CheapUCB is bounded as

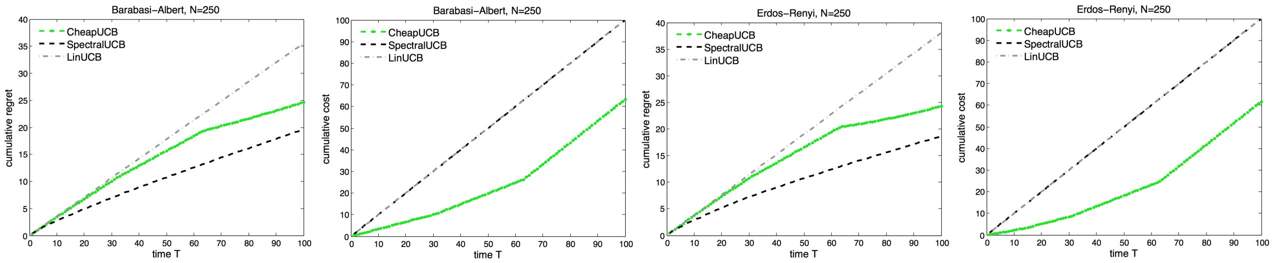
$$C_T \leq \sum_{j=1}^{J-1} \frac{2^{j-1}}{J-j+1} \leq \frac{3T}{4} - \frac{1}{2}$$

Remark 1 *Observe that when the eigenvalue gap is large, we get the regret to order $d\sqrt{T}$ within a constant factor satisfying the constraint (7). For the general case, compared to SpectralUCB, the regret bound of our algorithm increases by an amount of $cd\sqrt{T/2}\log_2(T/2)\log(T/\lambda+1)$, but still it is of the order $d\sqrt{T}$. However, the total cost in CheapUCB is smaller than in SpectralUCB by an amount of at least $T/4 + 1/2$, i.e., cost reduction of the order of T is achieved by our algorithm.*

Corollary 1 *CheapUCB matches the regret performance of SpectralUCB and provides a cost gain of $\mathcal{O}(T)$.*

5.4. Computational complexity and scalability

The computational and scalability issues of CheapUCB are essentially those associated with the SpectralUCB, i.e., obtaining eigenbasis of the graph Laplacian, matrix inversion



(a) Regret for BA graph (b) Cost for BA graph (c) Regret for ER graph (d) Cost for ER graph

Figure 1. Regret and Cost for Barabási-Albert (BA) and Erdős-Rényi (ER) graphs with N=250 nodes and $T = 100$

and computation of the UCBs. Though CheapUCB uses larger sets of arms or probes at each step, it needs to compute only N UCBs as $|\mathcal{S}_w| = N$ for all w. The i-th probe in the set $\mathcal{S}_w$ can be computed by sorting the elements of the edge weights $W(i, :)$ and assigning weight $1/w$ to the first w components can be done in order $N \log N$ computations. As Valko et al. (2014), we speed up matrix inversion using iterative update (Zhang, 2005), and compute the eigenbasis of symmetric Laplacian matrix using fast symmetric diagonally dominant solvers as CMG (Koutis et al., 2011).

6. Experiments

We evaluate and compare our algorithm with SpectralUCB which is shown to outperform its competitor LinUCB for learning on graphs with large number of nodes. To demonstrate the potential of our algorithm in a more realistic scenario we also provide experiments on Forest Cover Type dataset. We set $\delta = 0.001$, $R = 0.01$, and $\lambda = 0.01$.

6.1. Random graphs models

We generated graphs from two graph models that are widely used to analyze connectivity in social networks. First, we generated a *Erdős-Rényi* (ER) graph with each edge sampled with probability 0.05 independent of others. Second, we generated a *Barabási-Albert* (BA) graph with degree parameter 3. The weights of the edges of these graphs we assigned uniformly at random.

To obtain a reward function f, we randomly generate a sparse vector α^* with a small $k \ll N$ and use it to linearly combine the eigenvectors of the graph Laplacian as $f = \boldsymbol{Q\alpha}^*$, where $\boldsymbol{Q}$ is the orthonormal matrix derived from the eigendecomposition of the graph Laplacian. We ran our algorithm on each graph in the regime $T < N$. In the plots displayed we used $N = 250$, $T = 150$ and $k = 5$. We averaged the experiments over 100 runs.

From Figure 1, we see that the cumulative regret performance of CheapUCB is slightly worse than for SpectralUCB, but significantly better than for LinUCB. However, in terms of the cost CheapUCB provides a gain of at least 30 % as compared to both SpectralUCB and LinUCB.

6.2. Stochastic block models

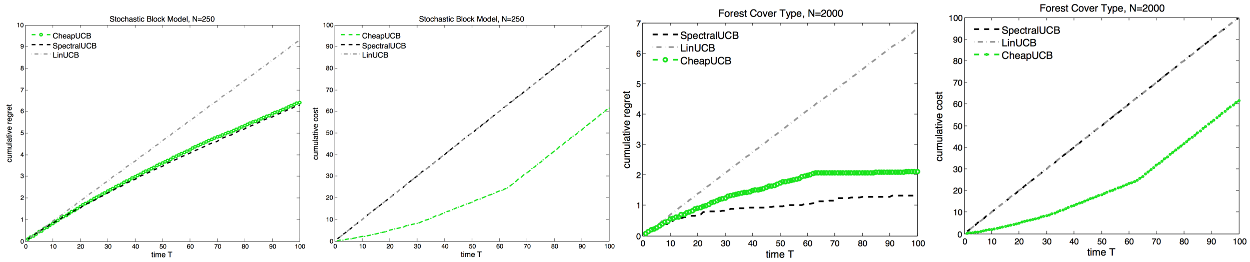
Community structure commonly arises in many networks. Many nodes can be naturally grouped together into a tightly knit collection of clusters with sparse connections among the different clusters. Graph representation of such networks often exhibit dense clusters with sparse connection between them. Stochastic block models are popular in modeling such community structure in many real-world networks (Girvan & Newman, 2002).

The adjacency matrix of SBMs exhibits a block triangular behavior. A generative model for SBM is based on connecting nodes within each block/cluster with high probability and nodes that are in two different blocks/clusters with low probability. For our simulations, we generated an SBM as follows. We grouped $N = 250$ nodes into 4 blocks of size 100, 60, 40 and 50, and connected nodes within each block with probability of 0.7. The nodes from the different blocks are connected with probability 0.02. We generated the reward function as in the previous subsection. The first 6 eigenvalues of the graph are $0, 3, 4, 5, 29, 29.6, \ldots$, i.e., there is a large gap between 4th and 5th eigenvalues, which confirms with our intuition that there should be 4 clusters (see Prop. 2). As seen from (a) and (b) in Figure 2, in this regime CheapUCB gives the same performance as SpectralUCB at a significantly lower cost, which confirms Theorem 2 (i) and Proposition 2.

6.3. Forest Cover Type data

As our motivation for cheap bandits comes from the scenario involving sensing costs, we performed experiments on the *Forest Cover Type* data, a collection of 581021 labeled samples each providing observations on $30m \times 30m$ region of a forest area. This dataset was chosen to match the radar motivation from the introduction, namely, we can view sensing the forest area from above, when vague sensing is cheap and specific sensing on low altitudes is costly. This dataset was already used to evaluate a bandit setting by Filippi et al. (2010).

The labels in Forest Cover Type data indicate the dominant species of trees (cover type) in a given region region.



(a) Regret for SBM (b) Cost for SBM (c) Regret for Forest data (d) Cost for Forest data

Figure 2. (a) Regret and (b) Cost for Stochastic block model with N=250 nodes and 4 blocks. (c) Regret and (d) Cost on the 'Cottonwood' cover type of the forest data.

The observations are 12 'cartographic' measures of the regions and are used as independent variables to derive the cover types. Ten of the cartographic measures are quantitative and indicate the distance of the regions with respect to some reference points. The other two are qualitative binary variables indicating presence of certain characteristics.

In a forest area, the cover type of a region depends on the geographical conditions which mostly remain similar in the neighboring regions. Thus, the cover types change smoothly over the neighboring regions and likely to be concentrated in some parts of forest. Our goal is to find the region where a particular cover type has the highest concentrated. For example, such requirement arises in aerial reconnaissance, where an air borne vehicle (like UAV) collects ground information through a series of measurements to identify the regions of interests. In such applications, larger areas can be sensed at higher altitudes more quickly (lower cost) but this sensing suffers a lower resolution. On the other hand, smaller areas can be sensed at lower altitudes but at much higher costs.

To find the regions of high concentration of a given cover type, we first clustered the samples using only the quantitative attributes ignoring all the qualitative measurements as done in (Filippi et al., 2010). We generated 2000 clusters (after normalizing the data to lie in the intervals $[0\ 1]$) using k-means with Euclidean distance as a distance metric. For each cover type, we defined reward on clusters as the fraction of samples in the cluster that have the given cover type. We then generated graphs taking cluster centers as nodes and connected them with edge weight 1 that have similar rewards using 10 nearest-neighbors method. Note that neighboring clusters are geographically closer and will have similar cover types making their rewards similar.

We first considered the 'Cottonwood/Willow' cover type for which nodes' rewards varies from 0 to 0.068. We plot the cumulative regret and cost in (c) and (d) in Figure 2 for $T = 100$. As we can see, the cumulative regret of the CheapUCB saturates faster than LinUCB and its performance is similar to that of SpectralUCB. And compared to both LinUCB and SpectralUCB total cost of CheapUCB is less by 35 %. We also considered reward functions for all the 7 cover types and the cumulative regret is shown in Figure 3. Again, the cumulative regret of CheapUCB is smaller than LinUCB and close to that of SpectralUCB with the cost gain same as in Figure 2(d) for all the cover types.

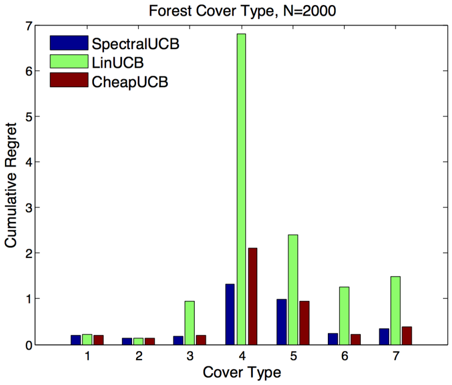


Figure 3. Cumulative regret for different cover types of the forest cover type data set with 2000 clusters: 1- Spruce/Fir, 2- Lodgepole Pine, 3- Ponderosa Pine, 4- Cottonwood/Willow, 5- Aspen, 6- Douglas-fir, 7- Krummholz.

7. Conclusion

We introduced *cheap bandits*, a new setting that aims to minimize sensing cost of the group actions while attaining the state-of-the-art regret guarantees in terms of effective dimension. The main advantage over typical bandit settings is that it models situations where getting the average reward from a set of neighboring actions is less costly than getting a reward from a single one. For the stochastic rewards, we proposed and evaluated CheapUCB, an algorithm that guarantees a cost gain linear in time. In future, we plan to extend this new sensing setting to other settings with limited feedback, such as contextual, combinatorial and non-stochastic bandits. As a by-product of our analysis, we establish a $\Omega(\sqrt{dT})$ lower bound on the cumulative regret for a class of graphs with effective dimension d.

Acknowledgment

This material is based upon work partially supported by NSF Grants CNS-1330008, CIF-1320566, CIF-1218992, and the U.S. Department of Homeland Security, Science and Technology Directorate, Office of University Programs, under Grant Award 2013-ST-061-ED0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the U.S. Department of Homeland Security or the National Science Foundation. This work was also supported by the French Ministry of Higher Education and Research and the French National Research Agency (ANR) under project ExTra-Learn n.ANR-14-CE24-0010-01.

References

Aeron, Shuchin, Saligrama, Venkatesh, and Castanon, David A. Efficient sensor management policies for distributed target tracking in multihop sensor networks. *IEEE Transactions on Signal Processing (TSP)*, 56(6): 2562–2574, 2008.

Alon, Noga, Cesa-Bianchi, Nicolò, Gentile, Claudio, and Mansour, Yishay. From Bandits to Experts: A Tale of Domination and Independence. In *Proceeding of Advance in Neural Information Processing Systems, NIPS*, Lake Tahoe, USA, 2013.

Auer, P., Cesa-Bianchi, N., Robert, Y. Freund, and Schapire, E. The non-stochastic multi-armed bandit problem. *SIAM Journal on Computing*, 32, 2003.

Auer, Peter. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422, March 2002. ISSN 1532-4435.

Badanidiyuru, A., Langford, J., and Slivkins, A. Resourceful contextual bandits. In *Proceeding of Conference on Learning Theory, COLT*, Barcelona, Spain, July 2014.

Badanidiyuru, Ashwinkumar, Kleinberg, Robert, and Slivkins, Aleksandrs. Bandits with knapsacks. In *Proceedings of Symposium on Foundations of Computer Science, FOCS*, California, USA, 2013.

Belkin, M., Niyogi, P., and Sindhwani, V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399—2434, 2008.

Caron, Stéphane, Kveton, Branislav, Lelarge, Marc, and Bhagat, Smriti. Leveraging Side Observations in Stochastic Bandits. In *Proceedings of Uncertainty in Artificial Intelligence, UAI*, pp. 142–151, Catalina Islands, USA, 2012.

Cesa-Bianchi, Nicolò, Dekel, Ofer, and Shamir, Ohad. Online Learning with Switching Costs and Other Adaptive Adversaries. In *Proceeding of Advances in Neural Information Processing Systems, NIPS*, pp. 1160–1168, Lake Tahoe, USA, 2013a.

Cesa-Bianchi, Nicolò, Gentile, Claudio, and Zappella, Giovanni. A Gang of Bandits. In *Procceding of Advances in Neural Information Processing Systems, NIPS*, Lake Tahoe, USA, 2013b.

Dani, V., Hayes, T. P., and Kakade, S. M. Stochastic linear optimization under bandit feedback. In *Proceeding of Conference on Learning Theory, COLT*, Helsinki, Finland, July 2008.

Ding, Wenkui, Qin, Tao, Zhang, Xu-dong, and Liu, Tieyan. Multi-Armed Bandit with Budget Constraint and Variable Costs. In *Proceedings of AAAI Conference on Artificial Intelligence*, Bellevue, Washington, USA, 2013.

Ermis, Erhan Baki and Saligrama, Venkatesh. Adaptive statistical sampling methods for decentralized estimation and detection of localized phenomena. In *Proceedings of Information Processing in Sensor Networks (IPSN)*, pp. 143–150, 2005.

Ermis, Erhan Baki and Saligrama, Venkatesh. Distributed detection in sensor networks with limited range multimodal sensors. *IEEE Transactions on Signal Processing*, 58(2):843–858, 2010.

Filippi, L., Cappe, O., Garivier, A., and Szepesvari, C. Parametric bandits: The generalized linear case. In *Proceeding of NIPS*, Vancouver, Canada, December 2010.

Fuemmeler, Jason A. and Veeravalli, Venugopal V. Smart sleeping policies for energy efficient tracking in sensor networks. *IEEE Transactions on Signal Processing*, 56 (5):2091–2101, 2008.

Gentile, Claudio, Li, Shuai, and Zappella, Giovanni. Online Clustering of Bandits. In *Proceeding of International Conference on Machine Learning, ICML*, Beijing, China, Jan 2014.

Gharan, S. O. and Trevisan, L. Partitioning into expanders. In *Proceeding of Symposium of Discrete Algorithms, SODA*, Portland, Oregon, USA, 2014.

Girvan, M. and Newman, M.E. Community structure in social and biological networks. In *Proceedings of Natl Acad Sci USA*, June 2002.

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

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

Li, L., Wei, C., Langford, J., and Schapire, R. E. A contextual-bandit approach to personalized news article recommendation. In *Proceeding of International Word Wide Web conference, WWW*, NC, USA, April 2010.

Mannor, Shie and Shamir, Ohad. From Bandits to Experts: On the Value of Side-Observations. In *Proceedings of Advances in Neural Information Processing Systems, NIPS*, Granada, Spain, 2011.

Narang, S. K., Gadde, A., and Ortega, A. Signal processing techniques for interpolation in graph structured data. In *Proceedings of International Conference of Acoustics, Speech and Signal Processing , ICASSP*, Vancouver, Canada, May 2013.

Shuman, D. I., Narang, S. K., Frossard, P., Ortega, A., and Vanderghenyst, P. The emerging filed of signal processing on graphs. In *IEEE Signal Processing Magazine*, May 2013.

Tran-Thanh, Long, Chapman, Archie C., Rogers, Alex, and Jennings, Nicholas R. Knapsack Based Optimal Policies for Budget-Limited Multi-Armed Bandits. In *Proceedings of AAAI conference on Artificial Intelligence*, Toronto, Canada, 2012.

Valko, Michal, Munos, Rémi, Kveton, Branislav, and Kocák, Tomáš. Spectral Bandits for Smooth Graph Functions. In *31th International Conference on Machine Learning*, 2014.

Zhang, F. The schur complement and its application. *Springer*, 4, 2005.

Zhu, X. and Rabbat, M. Graph spectral compressed sensing for sensor networks. In *Proceedings of International Conference of Acoustics, Speech and Signal Processing, ICASSP*, Kyoto, Japan, May 2012.

Zolghadr, Navid, Bartok, Gabor, Greiner, Russell, György, András, and Szepesvari, Csaba. Online Learning with Costly Features and Labels. In *Proceeding of Advances in Neural Information Processing Systems, NIPS*, Lake Tahoe, USA, 2013.

Supplementary Material: Cheap Bandits

1. Proof of Proposition 1

For a given policy $\pi, \boldsymbol{\alpha}^*, T$, and a graph G define expected cumulative reward as

$$Regret(T, \pi, \boldsymbol{\alpha}^*, G) = \mathbb{E}\left[\sum_{t=1}^{T} \tilde{\boldsymbol{s}}_* \boldsymbol{\alpha}_* - \tilde{\boldsymbol{s}}_t \boldsymbol{\alpha}^* \middle| \boldsymbol{\alpha}^*\right]$$

where $\tilde{\boldsymbol{s}}_t = \pi'(t)\boldsymbol{Q}$, and $\boldsymbol{Q}$ is the orthonormal basis matrix corresponding to Laplacian of G. Let $\mathcal{G}_d$ denote the family of graphs with effective dimension d. Define T-period risk of the policy π

$$Risk(T, \pi) = \max_{G \in \mathcal{G}_d} \max_{\substack{\boldsymbol{\alpha}^* \in \mathcal{R}^N \\ \|\boldsymbol{\alpha}^*\|_{\boldsymbol{\Lambda}} < c}} \left[Regret\left(T, \pi, \boldsymbol{\alpha}^*, G\right)\right]$$

We first establish that there exists a graph with effective dimension d and a class of smooth reward functions defined over it with parameters $\boldsymbol{\alpha}^*$'s in a d-dimensional vector space.

Lemma 1. *Given T, there exists a graph $\widehat{G} \in \mathcal{G}_d$ such that*

$$\max_{\substack{\boldsymbol{\alpha}^* \in \mathcal{R}^d \\ \|\boldsymbol{\alpha}^*\|_{\boldsymbol{\Lambda}} < c}} \left[Regret\left(T, \pi, \boldsymbol{\alpha}^*, \widehat{G}\right)\right] \leq Risk(T, \pi)$$

Proof: We prove the lemma by the explicit construction of a graph. Consider a graph G consisting of d disjoint connected subgraphs denoted as $G_j : j = 1, 2 \ldots, d$. Let the nodes in each subgraph have the same reward. The eigenvalues of the graph are $\{0, \widehat{\lambda}_1, \cdots, \widehat{\lambda}_{N-d}\}$, where eigenvalue 0 is repeated d times. Note that the set of eigenvalues of the graph is the union of the set of eigenvalues of the individual subgraphs. Without loss of generality, assume that $\widehat{\lambda_1} > T/d \log(T/\lambda{+}1)$. This is always possible, for example if subgraphs are cliques, which is what we assume. Then the effective dimension of the graph G is d. Since the graph separates into d disjoint subgraphs, we can split the reward function $\boldsymbol{f}_\alpha = \boldsymbol{Q}\boldsymbol{\alpha}$ into d parts, one corresponding to each subgraph. We write $\boldsymbol{f}_j = \boldsymbol{Q_j}\boldsymbol{\alpha}_j$ for $j = 1, 2, \ldots, d$, where $\boldsymbol{f}_j$ is the reward function associated with G_j, $\boldsymbol{Q}_j$ is the orthonormal matrix corresponding to Laplacian of G_j, and $\boldsymbol{\alpha}_i$ is a sub-vector of $\boldsymbol{\alpha}$ corresponding to node rewards on G_j.

Write $\boldsymbol{\alpha}_j = \boldsymbol{Q}_j'\boldsymbol{f}_j$. Since $\boldsymbol{f}_j$ is a constant vector, and except for one, all the columns in $\boldsymbol{Q}_j$ are orthogonal to $\boldsymbol{f}_j$, it is clear that $\boldsymbol{\alpha}_j$ has only one non-zero component. We conclude that for the reward functions that is constant on each subgraphs $\boldsymbol{\alpha}$ has only d non-zero components and is in a d-dimensional space. The proof of the lemma is completed by setting $\widehat{G} = G$. Note that a graph with effective dimension d cannot have more than d disjoint connected subgraphs. Next, we restrict our attention to graph $\widehat{G}$ and rewards that are piecewise constant on each clique. That means that the nodes in each clique have the same reward. Recall that action set $\mathcal{S}_D$ consists of actions that can probe a node or a group of neighboring nodes. Therefore, any group action will only allow us to observe average reward from a group of nodes within a clique but not across the cliques. Then, all node and group actions used to observe reward from within a clique are indistinguishable. Hence, the $\mathcal{S}_D$ collapses to set of d distinct actions one associated with each clique, and the problem reduces to that of selecting a clique with the highest reward. We henceforth treat each clique as an arm where all the nodes within the same clique share the same reward value.

We now provide a lower bound on the expected regret defined as follows

$$\widetilde{Risk}(T, \pi, \widehat{G}) = \mathbb{E}\left[Regret\left(T, \pi, \boldsymbol{\alpha}^*, \widehat{G}\right)\right], \tag{1}$$

where expectation is over the reward function on the arms.

To lower bound the regret we follow the argument of Auer et al. (2002) and their Theorem 5.1, where an adversarial setting is considered and the expectation in (1) is over the reward functions generated randomly according to Bernoulli distributions. We generalize this construction to our case with Gaussian noise. The reward generation process is as follows:

Without loss of generality choose cluster 1 to be the good cluster. At each time step t, sample reward of cluster 1 from the Gaussian distribution with mean $\frac{1}{2} + \xi$ and unit variance. For all other clusters, sample reward from the Gaussian distribution with mean $\frac{1}{2}$ and unit variance.

The rest of the proof of the arguments follows exactly as in the proof of Theorem 5.1 (Auer et al., 2002) except at their Equation 29. To obtain an equivalent version for Gaussian rewards, we use the relationship between the L_1 distance of Gaussian distributions and their KL divergence. We then apply the formula for the KL divergence between the Gaussian random variables to obtain equivalent version of their Equation 30. Now note that, $\log(1-\xi^2) \sim -\xi^2$ (within a constant). Then the proof follows silmilarly by setting $\xi = \sqrt{d/T}$ and noting that the L_2 norm of the mean rewards is bounded by c for an appropriate choice of λ.

2. Proof of Proposition 2

In the following, we first give some definitions and related results.

Definition 1 (k-way expansion constant by Lee et al., 2012)**.** *Consider a graph G and $\mathcal{X} \subset \mathcal{V}$. Let*

$$\phi_G(\mathcal{X}) := \phi(\mathcal{X}) = \frac{|\partial \mathcal{X}|}{V(\mathcal{X})},$$

where $V(\mathcal{X})$ denotes the sum of the degree of nodes in $\mathcal{X}$ and $|\partial \mathcal{X}|$ denotes the number of edges between the nodes in $\mathcal{X}$ and $\mathcal{V} \backslash \mathcal{X}$. For all $k > 0$, k-way expansion constant is defined as

$$\rho_{\mathcal{G}}(k) = \min\left\{\max \phi(\mathcal{V}^i) : \cap_{i=1}^k \mathcal{V}^i = \varnothing, |\mathcal{V}^i| \neq 0\right\}.$$

Let $\mu_1 \leq \mu_2, \ldots, \leq \mu_N$ denote the eigenvalues of the normalized Laplacian of G.

Theorem 1 (Gharan & Trevisan (2014), Lee et al. (2012))**.** *Let $\varepsilon > 0$ and $\rho(k+1) > (1+\varepsilon)\rho(k)$ holds for some $k > 0$. Then the following holds:*

$$\mu_k/2 \leq \rho(k) \leq \mathcal{O}(k^2)\sqrt{\mu_k} \tag{2}$$

There exist k partitions $\{\mathcal{V}^i : i = 1, 2, \ldots, k\}$ of $\mathcal{V}$ such that $\forall\ i = 1, 2, \ldots, k$

$$\begin{aligned} \phi(\mathcal{V}^i) &\leq k\rho(k) \quad \textit{and} & (3)\\ \phi(G[\mathcal{V}^i]) &\geq \varepsilon\rho(k+1)/14k & (4)\end{aligned}$$

where $\phi(G[\mathcal{X}])$ denotes the Cheeger's constant (conduntance) of the subgraph induced by $\mathcal{X}$.

Definition 2 (Isoperimetric number)**.**

$$\theta(G) = \left\{\min \frac{\partial \mathcal{X}}{|\mathcal{X}|} : |\mathcal{X}| \leq \mathcal{X}/2\right\}.$$

Let $\lambda_1 \leq \lambda_2, \ldots, \leq \lambda_N$ denote the eigenvalues of the unnormalized Laplacian of G. We remind the reader of the following standard result.

$$\lambda_2/2 \leq \theta(G) \leq \sqrt{2\kappa\lambda_2}. \tag{5}$$

Proof: The relation $\lambda_{k+1}/\lambda_k \geq \mathcal{O}(k^2)$ implies that $\mu_{k+1}/\mu_k \geq \mathcal{O}(k^2)$. Using the upper and lower bounds on the eigenvalues in (2), the relation $\rho_{k+1} \geq (1+\varepsilon)\rho_k$ holds for some $\varepsilon > 1/2$. Then, applying Theorem 1 we get k-partitions satisfying (3)-(4). Let $\boldsymbol{L}_i$ denote the Laplacian induced by the subgraph $G[\mathcal{V}^j] = (\mathcal{V}^j, \mathcal{E}^j)$ for $j = 1, 2, \cdots k$. By the quadratic property of the graph Laplacian we have

$$\begin{aligned}
\boldsymbol{f'Lf} &= \sum_{(u,v)\in\mathcal{E}} (f_u - f_v)^2 \\
&= \sum_{j=1}^{k} \sum_{(u,v)\in\mathcal{E}_j} (f_u - f_v)^2 \\
&= \sum_{j=1}^{k} \boldsymbol{f_j' L_j f_j}
\end{aligned}$$

where $\boldsymbol{f}_j$ denotes the reward vector on the induced subgraph $G_j := G[\mathcal{V}^j]$. In the following we just focus on the optimal node. The same arguments holds for any other node. Without loss of generality assume that the node with optimal reward lies in subgraph G_l for some $1 \leq l \leq d$. From the last relation above we have $\boldsymbol{f}_l'\boldsymbol{l}_l\boldsymbol{f}_l \leq c$. The reward functions on the subgraph G_l can be represented as $\boldsymbol{f}_l = \boldsymbol{Q}_l\boldsymbol{\alpha}_l$ for some $\boldsymbol{\alpha}_l$, where $\boldsymbol{Q}_l$ satisfies $\boldsymbol{L}_i = \boldsymbol{Q}_l'\boldsymbol{\Lambda}_L\boldsymbol{Q}_l$ and $\boldsymbol{\Lambda}_l$ denotes the diagonal matrix with eigenvalues of $\boldsymbol{\Lambda}_l$. We have

$$\begin{aligned}
|F_G(\boldsymbol{s}_*) - F_G((\boldsymbol{s}_*^w)| &= |F_{G_l}(\boldsymbol{s}_*) - F_{G_l}(\boldsymbol{s}_*^w)| \\
&\leq \|\boldsymbol{s}_* - \boldsymbol{s}_*^w\|\|\boldsymbol{Q}_l\boldsymbol{\alpha}_l\| \\
&\leq \left(1 - \frac{1}{w}\right)\|\boldsymbol{Q}_l\boldsymbol{\Lambda}_l^{-1/2}\|\|\boldsymbol{\Lambda}_l^{1/2}\boldsymbol{\alpha}_l\| \\
&\leq \frac{c}{\sqrt{\lambda_2(G_l)}} \quad \text{by Chauchy-Schwarz} \\
&\leq \frac{\sqrt{2\kappa}c}{\theta(G_l)} \quad \text{from (5)} \\
&\leq \frac{\sqrt{2\kappa}c}{\phi(G_l)} \quad \text{using } \theta(G_l) \geq \phi(G_l) \\
&\leq \frac{14k\sqrt{2\kappa}c}{\varepsilon\rho(k+1)} \quad \text{from Theorem 1, Equation 4} \\
&\leq \frac{56k\sqrt{2\kappa}c}{\mu_{k+1}} \quad \text{from Theorem 1, Equation 2} \\
&\leq \frac{56k\kappa\sqrt{2\kappa}c}{\lambda_{k+1}} \quad \text{using } \mu_{k+1} \geq \lambda_{k+1}/\kappa.
\end{aligned}$$

This completes the proof.

3. Analysis of CheapUCB

For a given confidence parameter δ define

$$\beta = 2R\sqrt{d\log\left(1 + \frac{T}{\lambda}\right) + 2\log\frac{1}{\delta}} + c,$$

and consider the ellipsoid around the estimate $\widehat{\boldsymbol{\alpha}}_t$

$$C_t = \{\boldsymbol{\alpha} : \|\widehat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha}\|_{V_t} \leq \beta\}.$$

We first state the following results by Abbasi-Yadkori et al. (2011), Dani et al. (2008), and Valko et al. (2014) that we use later in our analysis.

Lemma 2 (self-normalized bound)**.** *Let $\boldsymbol{\xi}_t = \sum_{i=1}^{t} \tilde{\mathbf{s}}_i\varepsilon_i$ and $\lambda > 0$. Then, for any $\delta > 0$, with probability at least $1 - \delta$ and for all $t > 0$,*

$$\|\boldsymbol{\xi_t}\|_{\boldsymbol{V_t^{-1}}} \leq \beta.$$

Lemma 3. *Let $V_0 = \lambda I$. We have:*

$$\log \frac{\det(V_t)}{\det(\lambda I)} \quad \leq \quad \sum_{i=1}^{t} \|\tilde{\boldsymbol{s}}_{\boldsymbol{i}}\|_{\boldsymbol{V_{i-1}^{-1}}} \|\tilde{\mathbf{s}}_i\|_{V_{i-1}^{-1}} \leq 2 \log \frac{\det(V_{t+1})}{\det(\lambda I)}$$

.

Lemma 4. *Let $\|\boldsymbol{\alpha}^*\|_2 \leq c$. Then, with probability at least $1-\delta$, for all $t \geq 0$ and for any $\mathbf{x} \in \mathcal{R}^n$ we have $\boldsymbol{\alpha}^* \in C_t$ and*

$$|\mathbf{x} \cdot (\widehat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha}^*)| \leq \|\mathbf{x}\|_{\mathbf{V_t^{-1}}} \beta.$$

Lemma 5. *Let d be the effective dimension and T be the time horizon of the algorithm. Then,*

$$\log \frac{\det(V_{T+1})}{\det(\Lambda)} \leq 2d \log \left(1 + \frac{T}{\lambda}\right).$$

3.1. Proof of Theorem 2

We first prove the case where degree of each node is at least $\log T$. Consider step $t \in [2^{j-1}, 2^j - 1]$ in stage $j = 1, 2, \cdots J-1$. Recall that in this step a probe of width $J-j+1$ is selected. Write $w_j := J-j+1$, and denote the probe of width $J-j+1$ associated with the optimal probe $\mathbf{s}_*$ as simply $\mathbf{s}_*^{w_j}$ and the corresponding GFT as $\tilde{\mathbf{s}}_*^{w_j}$. The probe selected at time t is denoted as $\mathbf{s}_t$. Note that both $\mathbf{s}_t$ and $\mathbf{s}_*^{w_j}$ lie in the set $\mathcal{S}_{J-j+1}$. For notational convenience let us denote

$$h(j) := \begin{cases} c'\sqrt{T}(J-j+1)/\lambda_{d+1} & \text{when (10) holds} \\ c'd/\lambda_{d+1} & \text{when (9) holds.} \end{cases}$$

The instantaneous regret in step t is

$$\begin{aligned} r_t = \tilde{\mathbf{s}}_* \cdot \boldsymbol{\alpha}^* - \tilde{\mathbf{s}}_t \cdot \boldsymbol{\alpha}^* \\ \leq \quad & \tilde{\mathbf{s}}_*^{w_j} \cdot \boldsymbol{\alpha}^* + h(j) - \tilde{\mathbf{s}}_t \cdot \boldsymbol{\alpha}^* \\ = \quad & \tilde{\mathbf{s}}_*^{w_j} \cdot (\boldsymbol{\alpha}^* - \widehat{\boldsymbol{\alpha}}_t) + \tilde{\mathbf{s}}_*^{j} \cdot \widehat{\boldsymbol{\alpha}}_t + \beta\|\tilde{\mathbf{s}}_*^{w_j}\|_{\mathbf{V_t^{-1}}} - \beta\|\tilde{\mathbf{s}}_*^{w_j}\|_{\mathbf{V_t^{-1}}} - \tilde{\mathbf{s}}_t \cdot \boldsymbol{\alpha}^* + h(j) \\ \leq \quad & \tilde{\mathbf{s}}_*^{w_j} \cdot (\boldsymbol{\alpha}^* - \widehat{\boldsymbol{\alpha}}_t) + \tilde{\mathbf{s}}_t \cdot \widehat{\boldsymbol{\alpha}}_t + \beta\|\tilde{\mathbf{s}}_t\|_{\mathbf{V_t^{-1}}} - \beta\|\tilde{\mathbf{s}}_*^{w_j}\|_{\mathbf{V_t^{-1}}} - \tilde{\mathbf{s}}_t \cdot \boldsymbol{\alpha}^* + h(j) \\ = \quad & \tilde{\mathbf{s}}_*^{w_j} \cdot (\boldsymbol{\alpha}^* - \widehat{\boldsymbol{\alpha}}_t) + \tilde{\mathbf{s}}_t \cdot (\widehat{\boldsymbol{\alpha}}_t - \boldsymbol{\alpha}^*) + \beta\|\tilde{\mathbf{s}}_t\|_{V_t^{-1}} - \beta\|\tilde{\mathbf{s}}_*^{w_j}\|_{\mathbf{V_t^{-1}}} + h(j) \\ \leq \quad & \beta\|\tilde{\mathbf{s}}_*^{w_j}\|_{\mathbf{V_t^{-1}}} + \beta\|\tilde{\mathbf{s}}_t\|_{\mathbf{V_t^{-1}}} + \beta\|\tilde{\mathbf{s}}_t\|_{\mathbf{V_t^{-1}}} - \beta\|\tilde{\mathbf{s}}_*^{w_j}\|_{\mathbf{V_t^{-1}}} + h(j) \\ = \quad & 2\beta\|\tilde{\mathbf{s}}_t\|_{\mathbf{V_t^{-1}}} + h(j). \end{aligned}$$

We used (9)/(10) in the first inequality. The second inequality follows from the algorithm design and the third inequality follows from Lemma 4. Now, the cumulative regret of the algorithm is given by

$$\begin{aligned} R_T \leq & \sum_{j=1}^{J} \sum_{t=2^{j-1}}^{2^j-1} \min\left\{2, 2\beta\|\tilde{\mathbf{s}}_t\|_{V_t^{-1}} + h(j)\right\} \\ \leq & \sum_{j=1}^{J} \sum_{t=2^{j-1}}^{2^j-1} \min\left\{2, 2\beta_t\|\tilde{\mathbf{s}}_t\|_{V_t^{-1}}\right\} + \sum_{j=1}^{J-1} \sum_{t=2^{j-1}}^{2^j-1} h(j) \\ \leq & \sum_{t=1}^{T} \min\left\{2, 2\beta_t\|\tilde{\mathbf{s}}_t\|_{V_t^{-1}}\right\} + \sum_{j=1}^{J-1} h(j) 2^{j-1}. \end{aligned}$$

Note that the summation in the second term includes only the first $J-1$ stages. In the last stage J, we use probes of width 1 and hence we do not need to use (9) or (10) in bounding the instantaneous regret. Next, we bound each term in the regret separately.

To bound the first term we use the same steps as in the proof of Theorem 1 of Valko et al. (2014). We repeat the steps below for convenience.

$$\begin{aligned}
\sum_{t=1}^{T} \min\{2, 2\beta\|\tilde{\mathbf{s}}_t\|_{V_t^{-1}}\} &\leq (2+2\beta)\sum_{t=1}^{T}\min\left\{1, \|\tilde{\mathbf{s}}_t\|_{\mathbf{V_t^{-1}}}\right\} \\
&\leq (2+2\beta)\sqrt{T\sum_{t=1}^{T}\min\left\{1, \beta_t\|\tilde{\mathbf{s}}_t\|_{\mathbf{V_t^{-1}}}\right\}^2} \\
&\leq 2(1+\beta)\sqrt{2T\log(|\boldsymbol{V}_{T+1}|/|\boldsymbol{\Lambda}|)} && (6)\\
&\leq 4(1+\beta)\sqrt{Td\log(1+T/\lambda)} && (7)\\
&\leq \left(8R\sqrt{2\log\frac{1}{\delta}+d\log\left(1+\frac{T}{\lambda}\right)}+4c+4\right)\sqrt{Td\log\left(1+\frac{T}{\lambda}\right)}.
\end{aligned}$$

We used Lemma 3 and 5 in inequalities (6) and (7) respectively. The final bound follows from plugging in the value of β.

3.2. For the case when (10) holds:

For this case we use $h(j) = c'\sqrt{T}(J-j+1)/\lambda_{d+1}$. First observe that $2^{j-1}h(j)$ is increasing in $1 \leq j \leq J-1$. We have

$$\begin{aligned}
\sum_{j=1}^{J-1}\frac{2^{j-1}c'\sqrt{T}(J-j+1)}{\lambda_{d+1}} &\leq (J-1)\frac{2^{J-1}\sqrt{T}c'}{\lambda_{d+1}} \\
&\leq (J-1)\frac{2^{\log_2 T-1}c'\sqrt{T}}{\lambda_{d+1}} \\
&\leq (J-1)\frac{c'\sqrt{T}(T/2)}{(T/d\log(T/\lambda+1))} \\
&\leq dc'\sqrt{T/4}\log_2(T/2)\log(T/\lambda+1).
\end{aligned}$$

In the second line we applied the definition of effective dimension.

3.3. For the case when $\lambda_{d+1}/\lambda_d \geq \mathcal{O}(d^2)$

For the case $\lambda_{d+1}/\lambda_d \geq \mathcal{O}(d^2)$ we use $h(j) = c'd/\lambda_{d+1}$.

$$\begin{aligned}
\sum_{j=1}^{J-1}\frac{2^{j-1}c'd}{\lambda_{d+1}} &\leq \frac{2^{J-1}c'd}{\lambda_{d+1}} \\
&\leq \quad c'd^2\log_2(T/2)\log(T/\lambda+1).
\end{aligned}$$

Now consider the case where minimum degree of the nodes is $1 < a \leq \log T$. In this case, we modify the algorithm to use only signals of width a in the first $\log T - a + 1$ stages and subsequently the signal width is reduced by one in each of the following stages. The previous analysis holds for this case and we get the same bounds on the cumulative regret and cost. When $a = 1$, CheapUCB is same as the SpectralUCB, hence total cost and regret is same as that of SpectralUCB.

To bound the total cost, note that in stage j we use signals of width $J-j+1$. Also, the cost of a signal given in (2) can be upper bounded as $C(\mathbf{s}_i^w) \leq \frac{1}{w}$. Then, we can upperbound total cost of signals used till step T as

$$\begin{aligned}
&\sum_{j=1}^{J} \frac{2^{j-1}}{J-j+1} \\
&\leq \quad \frac{1}{2}\sum_{j=1}^{J-1} 2^{j-1} + \frac{T}{2} \\
&\leq \quad \frac{1}{2}\left(\frac{T}{2}-1\right) + \frac{T}{2} \\
&= \quad \frac{3T}{4} - \frac{1}{2}.
\end{aligned}$$

References

Abbasi-Yadkori, Yasin, Pál, David, and Szepesvári, Csaba. Improved Algorithms for Linear Stochastic Bandits. In *Neural Information Processing Systems*. 2011.

Auer, Peter, Cesa-Bianchi, Nicolò, Freund, Yoav, and Schapire, Robert E. The Nonstochastic Multiarmed Bandit Problem. *SIAM Journal on Computing*, 32(1):48–77, January 2002.

Dani, Varsha, Hayes, Thomas P, and Kakade, Sham M. Stochastic Linear Optimization under Bandit Feedback. In *Conference on Learning Theory*, 2008.

Gharan, S. O. and Trevisan, L. Partitioning into expanders. In *Proceeding of Symposium of Discrete Algorithms, SODA*, Portland, Oregon, USA, 2014.

Lee, James R., Gharan, Shayan Oveis, and Trevisan, Luca. Multi-way spectral partitioning and higher-order cheeger inequalities. In *Proceeding of STOC*, 2012.

Valko, Michal, Munos, Rémi, Kveton, Branislav, and Kocák, Tomáš. Spectral Bandits for Smooth Graph Functions. In *31th International Conference on Machine Learning*, 2014.