Be Greedy: How Chromatic Number meets Regret Minimization in Graph Bandits

Shreyas S *
Flipkart,
Bangalore, India
s.shreyas@flipkart.com

Aadirupa Saha *
Indian Institute of Science,
Bangalore, India
aadirupa@iisc.ac.in

Chiranjib Bhattacharyya
Indian Institute of Science,
Bangalore, India
chiru@iisc.ac.in

Abstract

We study the classical linear bandit problem on *graphs* modelling arm rewards through an underlying graph structure $G(V,E)$ such that rewards of neighboring nodes are similar. Previous attempts along this line have primarily considered the arm rewards to be a smooth function over graph Laplacian, which however failed to characterize the inherent problem complexity in terms of the graph structure.We bridge this gap by showing a regret guarantee of $\tilde{O}(\chi(\overline{G})\sqrt{T})$ [1] that scales only with the chromatic number of the complement graph $\chi(\overline{G})$, assuming the rewards to be a smooth function over a general class of graph embeddings—*Orthonormal Representations*. Our proposed algorithms yield a regret guarantee of $\tilde{O}(r\sqrt{T})$ for any general embedding of rank r. Moreover, if the rewards correspond to a minimum rank embedding, the regret boils down to $\tilde{O}(\chi(\overline{G})\sqrt{T})$– none of the existing works were able to bring out such influences of graph structures over arm rewards. Finally, noting that computing the above minimum rank embedding is NP-Hard, we also propose an alternative $O(|V| + |E|)$ time computable embedding scheme—*Greedy Embeddings*—based on greedy graph coloring, with which our algorithms perform optimally on a large family of graphs, e.g. union of cliques, complement of k-colorable graphs, regular graphs, trees etc, and are also shown to outperform state-of-the-art methods on real datasets. Our findings open up new roads for exploiting graph structures on regret performance.

1 Introduction

The problem of multiarmed bandit (MAB) is extremely well studied in machine learning literature which is widely used to model online decision making problems under uncertainty [7, 37]. Due to their implicit exploration-vs-exploitation tradeoff, they are quite prevalent in clinical treatment, movie recommendations, job scheduling etc. Over the years several variants of MAB has been studied in the literature, introducing arm features [33, 24, 39], side information [28, 23, 8, 22], contextual scenarios [26, 27] etc. However, perhaps surprisingly, very few attempts have been made towards exploring the problem under graph based assumptions—precisely when bandit arms are known to be connected by a known relational graph.

Undoubtedly graph structural representations of data are extremely relevant in various real world scenarios where the edge connections model the item similarities e.g. connection among friends in a social network, or similar movies in a recommender systems etc. One might argue to model arm similarities in terms of feature representations, however in principle a relational graph has much more realistic interpretation. Moreover, the information of exact features may not even be available in reality—can we actually learn faster (achieve smaller regret) in such scenarios with just the knowledge of the graph?

Problem Statement. We consider the setting of MAB with an additional graph structure $G(V, E)$ over the N arms (nodes), i.e. $|V| = N$ and we denote $V = [N]$ henceforth, such that neighboring arms are similar in terms of their underlying rewards. In particular, if f_i is the expected reward associated to each arm i, we assume $f_i = \sum_{j\in\mathcal{N}_G(i)} S(i,j)\alpha_j$, where $\mathbf{S} \in \mathbb{R}^{N\times N}$ represents similarity matrix, unknown to the learner: For any item pair $i, j \in [N]$, $S(i, j)$ denotes their degree of similarity as a function of their edge information $E(i, j)$; $S(i, j) = 0$, if $(i, j) \notin E$. $\mathcal{N}_G(i)$ denotes the neighboring nodes of i in G. $\alpha_j \in \mathbb{R}$ can be seen as the 'contribution' or 'weight' factor of arm j on reward of the i^{th} arm, f_i.

Clearly, the above structure models the rewards of two similar nodes (i.e. with similar neighborhood w.r.t. the graph) similarly—question is does this additional struc-

[*] Both authors contributed equally to the paper.

[1] $\tilde{O}(\cdot)$ notation hides dependencies on $\log T$.

ture helps us to achieve a smaller regret guarantee? Intuitively, *it must*, as for a fixed N, one can expect to estimate the arm rewards faster in a denser graph compared to a sparser one, as in the former case, the knowledge of reward of a particular node reveals a lot more information about its neighboring nodes, possibly leading to a faster learning rate. We hence aim to characterize the complexity of the MAB problem in terms the underlying graph structure. *But how to achieve that? What is the right dependency of the underlying graph complexity parameter?*

Related Works. One can certainly apply the classical MAB algorithms [6, 7] for the purpose, but that leads to a regret of $O(\sqrt{NT})$, which could be arbitrarily bad for large N, precisely due to their inability to exploit the additional reward structure on $\boldsymbol{f}$ modeled w.r.t. G. The *linear bandit* algorithms [33, 39, 11] also fail due to the absence of any graph information leading to a regret bound of $\tilde{O}(\sqrt{dT})$ in terms of the feature dimension d. [35] also addresses a similar setting of linear contextual bandits, and derive an $\tilde{O}(\sqrt{\tilde{d}T})$ regret guarantee, with $\tilde{d}$ being a notion of dimensionality that decides the underlying problem complexity. However above regret boils down to $\tilde{O}(\sqrt{NT})$ for linear finite arm case [29], and moreover their setting does not consider any graph structure. [10] addresses a similar problem as that of [35], but their setting is specifically catered to Gaussian Process kernels.

[28, 23, 22, 2, 3] studied the MAB problem assuming relation graph over the nodes, however their setting also requires to reveal reward of a neighboring set of the pulled arm which boils to a semi-bandit (side information) setting, unlike our setting which is a pure bandit feedback model that reveals only a noisy reward of the selected arm. Few of them also requires [16] also consider a stochastic sequential learning problems on graphs but here the learner gets to observe the average reward of a group of graph nodes rather than a single one. The *online clustering of bandits* line of works [9, 14, 36] also attempts to exploit the item similarities through graphs, however their setting assumes the graph to be initially unknown while the goal is to learn the edge connections using additional contextual information per round. [29, 34] assume the arm rewards to be a smooth function over a graph, however their setting is restricted only to the time regime $T < N$ as their regret is $\tilde{O}(\sqrt{\tilde{d}(G,T)T})$ which depends on a term $\tilde{d}(G,T)$ called 'effective dimension'—an increasing function of T, and shoots up to N for large T where their guarantee stands vacuous (see Sec. 4.1 for few examples). Moreover the quantity $\tilde{d}(G,T)$ is poorly understood in graph theoretic literature, as it does not relate to any graph property, e.g. chromatic number, node/edge connectivity; nor lends itself to any structural information like graph sparsity etc., which fails to capture the regret guarantee in terms of the underlying graph structure.

Our results. We thus seek for a more interpretable regret bounds in terms of known graph theoretic quantities that directly relates to structural properties of G. Towards this we formulate the problem of graph bandits over N arms, where the reward vector $\boldsymbol{f} = \mathbf{S}\boldsymbol{\alpha} \in \mathbb{R}^N$ arms is defined in terms of a similarity matrix $\mathbf{S} \in \mathbb{R}^{N \times N}$ modeled through an underlying graph structure $G([N], E)$ over N bandit arms. Our key idea approaches the problem from the viewpoint of finding the *"right embedding"* that *best fits* the graph, by using a rich class of graph embeddings—*Orthonormal Graph Representations*—going beyond the usual choice of Laplacian embeddings. Using this, the regret guarantees of our proposed algorithms are shown to be of $\tilde{O}\big(\chi(\overline{G})\sqrt{T}\big)$, $\chi(\overline{G})$ being the chromatic number of the complement graph $\overline{G}$, which bridges our quest for relating the problem complexity with the graph structure. Note that the added advantage of above regret bounds is that for graphs with $\chi(\overline{G}) = O(1)$, we drive a regret of just $\tilde{O}(\sqrt{T})$ (independent of N), whereas the state-of-the art methods [29, 34] still lead to a $\tilde{O}(\sqrt{NT})$ regret with large enough T. See Sec. 4.1 for specific examples.

- We study the problem of *Bandits on Graphs* using a rich family of graph embedding–*Orthonormal Representations*–moving beyond the usual choice of Laplacian embedding, the only embedding used in the existing literature for any graph based learning problems [4, 17, 29, 34]. (Sec. 4.2).
- Under above embedding, our proposed algorithms, *OUCB* and *SupOUCB* are shown to achieve a regret guarantee of $\tilde{O}(r\sqrt{T})$ (Thm. 6, 8), r being the embedding rank, potentially much smaller than N of data dimension d, and thus improves upon the $O(\sqrt{NT})$ or $\tilde{O}(\sqrt{dT})$ regret of classical MAB or linear bandit algorithms respectively, or state-of-the-art regret bound $\tilde{O}(\sqrt{\tilde{d}(G,T)T})$ [29, 34]. (Sec. 3).
- Our main contribution lies in showing how embedding rank r relates to $\chi(\overline{G})$ under an optimal choice of embedding leading to a regret guarantee of $\tilde{O}(\chi(\overline{G})\sqrt{T})$ (Cor. 11), $\chi(\overline{G})$ being *chromatic number* of the complement graph $\overline{G}$. Thus we relate the learning rate to the underlying graph structure as $\chi(\overline{G})$ is a function of the graph connectivity. This, for the first time, brings out the inherent complexity of the underlying problem in terms of well studied graph theoretic measures—a much desired result, yet unattained so far. Clearly, a denser graph implies a small regret rate—an intuitive result. (Sec. 4).
- However, finding the optimal embeddings being NP-Complete (Cor. 11), we propose to work with a *'near optimal embedding'* based on *greedy graph colorings* – called *Greedy Embeddings* (Algo. 3), which

works in $O(N)$ time, given a valid coloring of $\overline{G}$ that can be easily obtained using any polytime approximate graph coloring algorithm, e.g. greedy coloring (runs in $O(N + |E|)$ time complexity) etc. The resulting algorithm is shown to perform with $\tilde{O}(c\sqrt{T})$ regret guarantee (Cor. 13), if c is the total number of colors used, which is in fact optimal as long as $c = O(\chi(\overline{G}))$ and holds good for a large family of graphs: e.g. regular graphs, union of cliques, planar graphs, trees, $G(n,p)$ random graphs etc. (Sec. 4.3).

- Finally, we also prove a matching regret lower bound of $\Omega\Big(\sqrt{\chi(\overline{G})T}\Big)$ on specific graph instances, proving optimality our regret bounds for such cases (up to a factor of $\sqrt{\chi(\overline{G})\log T}$, see Thm. 16),Sec. 5).

Efficacy of proposed algorithms and greedy embedding schemes are evaluated on different synthetic and real world graphs, where our proposed method is shown to outperform all the state of art algorithms (Sec. 6).

Organization. Sec. 2 introduces the preliminary notations and the problem setting along with the related works. Sec. 3 describes our proposed algorithms, *OUCB* and *SupOUCB* , with their regret guarantees. Sec. 4 relates the above regret guarantee to structural properties of the graph by proposing a polynomial time computable *greedy embedding scheme* (based on graph colorings). In Sec. 5, we derive a possible lower bound for our problem setup. Finally we study the comparative performances of our proposed algorithms with state-of-the-art methods on several synthetic and real world datasets in Sec. 6. Sec. 7 concludes the paper with some future directions.

2 Preliminaries and Problem Settings

Preliminaries. We denote $[n] = \{1, 2, \ldots n\}$, for any $n \in \mathbb{N}$. For any graph $G(V, E)$, let its vertex set $V(G) = \{v_1, v_2, \ldots v_N\}$ and edge set $E(G) \subseteq V \times V$. Clearly $|V| = N$. The Laplacian of graph G is defined by $\mathbf{L}_G = \mathcal{D}_G - \mathbf{A}_G$, $\mathbf{A}_G$ being the adjacency matrix of G and $\mathcal{D}_G$ is a diagonal matrix with $\mathcal{D}_G(i,i)$ being the degree of v_i. $\chi(G)$ and $\alpha(G)$ respectively denotes the chromatic and independence number of G. For any real, square, symmetric matrix $\mathbf{A} \in \mathbb{R}^{m\times m}$, we denote its eigenvalues by $\lambda_m(\mathbf{A}) \geq \cdots \geq \lambda_1(\mathbf{A})$, rank by $r(\mathbf{A})$, determinant by $|\mathbf{A}|$ and trace by $Tr(\mathbf{A})$. $\mathbb{S}_+^n$ denotes the family of $n \times n$ symmetric positive semi-definite matrices, $\mathbf{A}^\dagger$ the pseudo inverse of $\mathbf{A}$. $\mathbf{I}_n$ denotes the n-dimensional identity matrix.

The MAB problem [6, 7]. The problem of multiarmed bandit (MAB) consists of a learner presented with a set of N arms, with each arm $i \in [N]$ being associated to an unknown reward $f_i \in \mathbb{R}$. At each round $t \in [T]$, the learner's task is to select an arm $i_t \in [N]$ from $[N]$, upon which the nature provides a noisy reward $r_t \in \mathbb{R}$ with $\mathbf{E}[r_t] = f_{i_t}$. The objective is to minimize the expected regret in T rounds, with respect to the 'best arm' $i^* = \underset{i\in[n]}{\text{argmax}} f_i$, defined as:

$$R_T := \sum_{t=1}^{T}(f_{i^*} - f_{i_t}). \tag{1}$$

2.1 Problem setting: *Bandit on Graphs*

We define the problem of *"Bandits on Graphs"*, which is special case of famously studied multiarmed bandit problem (MAB) with additional knowledge of arm dependencies modeled through a graph structure. The formal problem statement is defined as follows:

Bandit on Graphs. Given a simple undirected graph $G([N], E)$ defined on the set of N bandit arms, the problem assumes an unknown reward vector $\boldsymbol{f} = \mathbf{S}\tilde{\boldsymbol{\alpha}} \in \mathbb{R}^N$, over the arm set $[N]$, where $\tilde{\boldsymbol{\alpha}} \in \mathbb{R}^N$ and $\mathbf{S} \in \mathbb{R}^{N\times N}$ represents a similarity matrix that models pairwise similarity $S(i,j)$ between any two items $i, j \in [N]$ as a function of their edge information $E(i,j)$; in particular, $S(i,j) = 0$, if $(i,j) \notin E$. Thus above implies that for any node i, $f_i = \sum_{j\in\mathcal{N}(i)} S(i,j)\tilde{\alpha}_j$, $\mathcal{N}(i) = \{j \in [N] \mid (i,j) \in E\}$ denotes the neighboring nodes of i.

Locality Assumptions on Rewards. Above implies two arms with similar structure (i.e. neighborhood) would have similar rewards – an intuitive *locality property* over the arm rewards. Formally we assume $\boldsymbol{f}$ to be such that $\|f_i - f_j\| \leq b$, $b \in \mathbb{R}_+$ being a small constant.

Objective. Similar to the setting of MAB, the goal of the learner is to play an arm $i_t \in [N]$ at each round t, upon which a noisy reward feedback is observed:

$$r_t = f_{i_t} + \eta_t, \tag{2}$$

where η_t is a zero-mean R-sub-Gaussian noise, i.e. $\mathbf{E}[\eta_t] = 0$, and $\forall a \in \mathbb{R}, \mathbf{E}\left[e^{a\eta_t}\right] \leq \exp\left(\frac{a^2R^2}{2}\right)$. As before, the objective of the learner is to minimize the cumulative regret of T rounds as defined in (1).

2.2 Finding a graph Embedding: Key Intuition

Note that, denoting $\mathbf{s}_i$ as the i^{th} row of $\mathbf{S}$ we can alternatively write $f_i = \mathbf{s}_i^\top \tilde{\boldsymbol{\alpha}}$, $\forall i \in [n]$. Thus our problem would have reduced to *linear bandits* if the arm features $\mathbf{s}_i$ were known [33, 39, 11]. However recall that unlike *linear bandits* the learner has only access to the G and not the underlying embedding matrix $\mathbf{S}$. Then how to proceed with a solution and how much error do we incur for not having the knowledge of $\mathbf{S}$?

The key to our approach is to *find a suitable graph embedding that 'best fits' the underlying matrix S*. We choose to work with the class of *orthonormal representations of graphs* for this purpose since it is rich enough to represent any $\boldsymbol{f}$ in its range space. Moreover the properties of any such embedding closely resembles that of $\mathbf{S}$, as evident from its definition:

Definition 1 (Orthonormal Representation of Graphs.). *An orthonormal representation of $G = ([N], E)$ is given by a matrix $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_N] \in \mathbb{R}^{d\times N}$, such that $\mathbf{u}_i^\top \mathbf{u}_j = 0$ whenever $(i,j) \notin E$ and $\|\mathbf{u}_i\|_2 = 1\ \forall i \in [N]$. Let the set $Lab(G)$ denotes all orthonormal representations of G, i.e. $Lab(G) = \{\mathbf{U} \mid \mathbf{U}$ is an Orthonormal Representation$\}$, and consider the set $\mathcal{K}(G) := \{\mathbf{K} \in \mathbb{S}_+^N \mid K_{ii} = 1, \forall i \in [N];\ K_{ij} = 0, \forall (i,j) \notin E\}$. [21] showed that the two sets are equivalent i.e. for each $\mathbf{U} \in Lab(G)$, we can have $\mathbf{K} \in \mathcal{K}(G)$ and vice-versa. We will term $\mathcal{K}(G)$ as the set of orthonormal embedding kernels of G.*

Now note that for a fixed $\mathbf{K} \in \mathcal{K}(G)$, any matrix $\mathbf{S} \in \mathbb{R}^{N\times N}$ can always be decomposed as $\mathbf{S} = \mathbf{K}\hat{\mathbf{S}} + \mathbf{E}$, where $\mathbf{K}\mathbf{E} = \mathbf{0}_{N\times N}$ and $\hat{\mathbf{S}} = (\mathbf{K}^\top\mathbf{K})^\dagger(\mathbf{K}\mathbf{S})$ (as $\mathbf{K} \in \mathbb{S}_+^N$, $\mathbf{K}^\top = \mathbf{K}$). Or equivalently we can decompose $\boldsymbol{f} = (\mathbf{K}\hat{\mathbf{S}} + \mathbf{E})\tilde{\boldsymbol{\alpha}} = \mathbf{K}\boldsymbol{\beta} + \mathbf{e}$, where $\boldsymbol{\beta} = \hat{\mathbf{S}}\tilde{\boldsymbol{\alpha}}$, and $\mathbf{K}\mathbf{e} = \mathbf{0}$. Assume $\|\mathbf{e}\|_\infty = \epsilon$ and $\mathbf{k}_i$ denote the i^{th} column (equivalently row) of matrix $\mathbf{K}$. From (1), we can then further derive (see details in Appendix A):

$$R_T \le \sum_{t=1}^{T} (\mathbf{k}_{i_t}^\top \boldsymbol{\beta} - \mathbf{k}_{i^*}^\top \boldsymbol{\beta}) + 2\epsilon T \qquad (3)$$

Remark 2 (Tradeoff between $\mathbf{K}$ vs ϵ). *If we find a $\mathbf{K}$ such that $\epsilon = o(\frac{1}{T})$, we can still hope to have a sublinear regret as long as the first term of* (3) *is sublinear in T. E.g. if we choose $\mathbf{K} = \mathbf{I}_N$, ($\mathbf{I}_N$ being the N-Identity matrix. Note $\mathbf{I}_N \in \mathcal{K}(G)$ for any graph on $[N]$), clearly $\epsilon = 0$. However the 'complexity' of the embedding is very high since rank $r(\mathbf{K}) = N$. On the contrary, setting a very low rank $\mathbf{K}$ will poorly approximate $\boldsymbol{f}$ resulting a high ϵ and the second term of* (3) *dominates.*

Thus our goal here is quantity the above tradeoff, i.e. how the first term of (3) varies with different choices of $\mathbf{K}$. For this, we henceforth assume $f = \mathbf{K}\boldsymbol{\beta}$ with $\epsilon = 0$, to analyze the regret dependency on $\mathbf{K}$.

3 Proposed Algorithms

Similar to classical UCB-algorithm for MAB [6], the main idea of our proposed algorithms is to keep an estimate of $\boldsymbol{f}$ (or equivalently $\boldsymbol{\beta}$, as we assumed $\boldsymbol{f} = \mathbf{K}\boldsymbol{\beta}$, for some $\mathbf{K} \in \mathcal{K}(G)$, or since $\exists \mathbf{U} \in Lab(G)$, such that $\mathbf{K} = \mathbf{U}^\top\mathbf{U}$, equivalently $\boldsymbol{f} = \mathbf{U}^\top\boldsymbol{\alpha}$, where $\boldsymbol{\beta} = \mathbf{U}^\dagger\boldsymbol{\alpha}$) with high confidence and pick the arms optimistically at each round based on that. Further we also assume $\|\boldsymbol{\alpha}\| \le B$. Our algorithms and their regret analysis are presented next. We use ridge regression to estimate $\boldsymbol{f}$, see (4). We denote the reward estimate obtained from the observations of rounds s to t by $\hat{\boldsymbol{f}}_{s,t}$ (or $\hat{\boldsymbol{\alpha}}_{s,t}$), $1 \le s < t \le T$, estimated as:

$$\hat{\boldsymbol{f}}_{s,t} = \underset{\mathbf{g}\in\mathbf{K}\hat{\boldsymbol{\beta}}}{\operatorname{argmin}} \left(\sum_{\tau=s}^{t-1} (g_{v_\tau} - r_\tau)^2 + \gamma \mathbf{g}^\top \mathbf{K}^\dagger \mathbf{g} \right), \text{ or}$$

$$\hat{\boldsymbol{\beta}}_{\boldsymbol{s,t}} = \underset{\boldsymbol{\beta}'\in\mathbb{R}^N}{\operatorname{argmin}} \left(\|\mathbf{K}_{s,t}^\top \boldsymbol{\beta}' - \mathbf{r}_{s,t}\|_2^2 + \gamma \boldsymbol{\beta}'^\top \mathbf{K} \boldsymbol{\beta}' \right),$$

$$\hat{\boldsymbol{\alpha}}_{s,t} = \underset{\boldsymbol{\alpha}'\in\mathbb{R}^N}{\operatorname{argmin}} \left(\|\mathbf{X}_{s,t}^\top \boldsymbol{\alpha}' - \mathbf{r}_{s,t}\|_2^2 + \gamma \|\boldsymbol{\alpha}'\|^2 \right) \qquad (4)$$

where $\mathbf{r}_{s,t} = [r_s\ r_{s+1} \ldots r_{t-1}]^\top \in \mathbb{R}^{t-s}$ being the vector of observed rewards in rounds s to t, $\mathbf{X}_{s,t} = [\mathbf{x}_s\ \mathbf{x}_{s+1} \ldots \mathbf{x}_{t-1}] \in \mathbb{R}^{N\times t-s}$, $\mathbf{x}_\tau = \mathbf{u}_{v_\tau} \in \mathbb{R}^N$ represents the arm v_τ, played at round $\tau \in \{s, s+1, \ldots t-1\}$. $\gamma > 0$ denotes the regularization parameter.

Remark 3 (Choice of regularization ($\mathbf{g}^\top\mathbf{K}^\dagger\mathbf{g}$)). *Recall our* regularity assumptions *on $\boldsymbol{f}$ implies the arm rewards to vary "smoothly" over the graph: i.e., if $(i,j) \in E$ then $f_i \approx f_j$, $\forall i,j \in [N]$, which is precisely ensured by our above choice of regularization. This is because $\boldsymbol{f} = \mathbf{K}\boldsymbol{\beta}$, $\boldsymbol{f}$ lies in the* Reproducing Kernel Hilbert Space (RKHS) *of the kernel matrix* $\mathbf{K}$. *Now, it is well known from the literature of kernel methods that the above smoothness condition equivalently implies $\boldsymbol{f}$ to be bounded in terms of the RKHS norm $\|\boldsymbol{f}\|_{\mathbf{K}} = \boldsymbol{f}^\top\mathbf{K}^\dagger\boldsymbol{f}$, say $\|\boldsymbol{f}\|_{\mathbf{K}} \le B$, for some small constant $B > 0$. (A detailed discussion on smoothness of RKHS functions is given in Appendix B).*

Lemma 4. *The least square estimate of $\boldsymbol{\alpha}$ in* (4) *is given by $\hat{\boldsymbol{\alpha}}_{s,t} = (\mathbf{X}_{s,t}\mathbf{X}_{s,t}^\top + \gamma\mathbf{I}_N)^{-1}\mathbf{X}_{s,t}\mathbf{r}_{s,t}$, which gives the following estimated reward vector:*

$$\hat{\boldsymbol{f}}_{s,t}(v) = \mathbf{u}_v^\top \hat{\boldsymbol{\alpha}}_{s,t} = \hat{\boldsymbol{k}}_{s,t}^{v} (\hat{\boldsymbol{K}}_{s,t} + \gamma\mathbf{I}_{t-s})^{-1} \mathbf{r}_{s,t}\ \forall v \in [N]$$

where $\hat{\boldsymbol{K}}_{s,t} \in \mathbb{R}^{t-s\times t-s}$ is such that $\hat{K}_{s,t}(\tau,\tau') = K(v_\tau, v_{\tau'})$, $\forall \tau,\tau' \in \{s, s+1, \ldots t-1\}$. $\hat{\boldsymbol{k}}_{s,t}^{v} = [K(v,v_s)\ K(v,v_{s+1}) \ldots K(v,v_{t-1})]$. For ease of notations we also denote $\mathbf{M}_{s,t} = (\hat{\boldsymbol{K}}_{s,t} + \gamma\mathbf{I}_{t-s})^{-1}$ and using block matrix inversion rule, we get:

$$\mathbf{M}_{s,t+1} = \begin{bmatrix} \hat{\boldsymbol{K}}_{s,t} + \gamma\mathbf{I}_{t-s} & \hat{\boldsymbol{k}}_{s,t} \\ (\hat{\boldsymbol{k}}_{s,t})^\top & K(v_t,v_t) + \gamma \end{bmatrix}^{-1}$$

$$= \begin{bmatrix} \mathbf{M}_{s,t} + z\mathbf{M}_{s,t}\hat{\boldsymbol{k}}_{s,t}\hat{\boldsymbol{k}}_{s,t}^\top\mathbf{M}_{s,t} & z\mathbf{M}_{s,t}\hat{\boldsymbol{k}}_{s,t} \\ z\hat{\boldsymbol{k}}_{s,t}^\top\mathbf{M}_{s,t} & z \end{bmatrix}, \qquad (5)$$

where $z = 1/(1 + \gamma - \hat{\boldsymbol{k}}_{s,t}^\top\mathbf{M}_{s,t}\hat{\boldsymbol{k}}_{s,t})$, and as $K(v_t,v_t) = 1$. Also we abbreviate $\hat{\boldsymbol{k}}_{s,t} = \hat{\boldsymbol{k}}_{s,t}^{v_t} = [K(v_s,v_t)\ K(v_{s+1},v_t) \ldots K(v_{t-1},v_t)]^\top$.

Our first algorithm is developed on the idea of estimating the reward vector $\boldsymbol{f}$ (alternatively $\boldsymbol{\alpha}$) using ridge-regression on the entire past observations and select the arms optimistically based on their upper confidence estimates. Formally, at each round t, *OUCB* estimates the reward of each arm $v \in [N]$ using $\hat{\boldsymbol{f}}_{1,t}(v)$ (Eqn. (4)), along with a confidence term V_t^v, and plays the arm with highest estimated reward (line 4). Thus the noise on the reward feedback observed at round t is not independent of the earlier rewards till time $t-1$. Due to this, we use self-normalized martingale inequalities [39], to obtain confidence width on the estimated reward $\hat{\boldsymbol{f}}_{1,t}$. Details of *OUCB* is given in Algorithm 1.

Algorithm 1 *OUCB*

```
input : $\mathbf{K} \in \mathcal{K}(G)$ : Embedding kernel
    $T, \delta$ Time horizon and confidence parameter
    $B, R$: Upper bound on $\|\boldsymbol{\alpha}\|$ and noise $\eta_t$ respectively.
    $\gamma$ : Regularization parameter.
init : $\mathbf{M}_{1,1} = 1$
for each round $t = 1, 2, \cdots, T$ do
    $\mathbf{r}_{1,t} = [r_1, \cdots, r_{t-1}]^T$
    $B_t = 2R\sqrt{r(\mathbf{K})\log(1+\frac{T}{\gamma}) + 2\log(1/\delta)} + \gamma^{\frac{1}{2}}B$
    Play $v_t = \underset{v \in [N]}{\operatorname{argmax}} \left(\hat{\boldsymbol{k}}^v_{1,t}\mathbf{M}_{1,t}\mathbf{r}^\top_{1,t} + V^v_t\right)$,
        where $V^v_t = B_t\left(K(v,v) - (\hat{\boldsymbol{k}}^v_{1,t})^\top\mathbf{M}_{1,t}\hat{\boldsymbol{k}}^v_{1,t}\right)$
    Observe the reward $r_t$.
    Update $\mathbf{M}_{1,t+1}$ using equation (5).
end for
```

At any round $t \in [T]$, *OUCB* gives the following confidence bound on the estimated reward:

Lemma 5. *Let $\delta \in (0,1)$, and B_t is same as defined in OUCB . Then any $v \in [N]$, we have*

$$Pr\left(|\hat{\boldsymbol{f}}_{1,t}(v) - \boldsymbol{f}(v)| \le B_t\right) \ge 1-\delta,$$

where $\hat{\boldsymbol{f}}_{1,t}(v) = \hat{\boldsymbol{k}}^v_{1,t}\mathbf{M}_{1,t}\mathbf{r}^\top_{1,t}$ is the reward estimate of arm v at round t (as obtained using Lemma 4).

Using Lemma 5, we further get:

Theorem 6. *Given any $\delta \in (0,1)$ and $\boldsymbol{f} \in [-1,1]^N$, with probability at least $1-\delta$, the regret of OUCB algorithm with embedding kernel $\mathbf{K} \in \mathcal{K}(G)$ is:*

$$R_T \le 2\left(2R\sqrt{r(\mathbf{K})\log(1+\frac{T}{\gamma}) + 2\log(1/\delta)} + \gamma^{\frac{1}{2}}B\right) \times \sqrt{2r(\mathbf{K})T\log\left(1+\frac{T}{\gamma}\right)} = O(r(\mathbf{K})\log T\sqrt{T})$$

Our second algorithm, *SupOUCB* , is in spirit similar to our first algorithm *OUCB* except it divides the T rounds into $\lceil \log T \rceil$ phases, and at each round, the arms are chosen independent of past rewards within that phase. This ensures the independence of the observed rewards in the successive rounds within a phase which allows to obtain confidence bounds on the arms' estimated rewards using tail-inequality on sub-Gaussian quadratic forms [12]. *SupOUCB* adopts its key idea from Sup-LinUCB [11] algorithm for linear bandits. The advantage of above trick gives a $\sqrt{\log T}$ factor improvement on the regret of *SupOUCB* compared to *OUCB* .

Algorithm 2 *SupOUCB*

```
input : $\mathbf{K} \in \mathcal{K}(G)$ : Embedding kernel
    $T, \delta$ Time horizon and confidence parameter
    $B, R$: Upper bound on $\|\boldsymbol{\alpha}\|$ and noise $\eta_t$ respectively.
    $\gamma$ : Regularization parameter.
init : $\mathcal{A}_1 \leftarrow [N]$, and $B' = \gamma^{-1}\Big(B\max\left(1, \frac{1}{\sqrt{\gamma}}\right)$
        $+\sqrt{R\left(r(\mathbf{K}) + 2\sqrt{r(\mathbf{K})\log\frac{1}{\delta}} + 2\log\frac{1}{\delta}\right)}\Big)$
 1: for each $j = 1, 2, \ldots, \lceil \log T \rceil$ do
 2:     $s_j = 2^{j-1}$, $t_j = \min(2^j - 1, T)$, $\mathbf{M}_{l_j,l_j} = 1$
 3:     for each round $t = l_j$ to $t_j$ do
 4:     Play $v_t = \underset{v \in \mathcal{A}_j}{\operatorname{argmax}}\Big[K(v,v) - (\hat{\boldsymbol{k}}^v_{s_j,t})^T\mathbf{M}_{s_j,t}\hat{\boldsymbol{k}}^v_{s_j,t}\Big]$
 5:         Observe the reward $r_t$
 6:         Compute $\mathbf{M}_{s_j,t+1}$ using equation (5)
 7:     end for
 8:     $\mathbf{r}_j = [r_{s_j}, \cdots, r_{t_j}]^T$
 9:     $\hat{\boldsymbol{k}}^v_j = [K(v_{s_j}, v), \ldots K(v_{t_j}, v)]^T, \forall v \in [N]$
10:     Eliminate nodes that are not promising:
        $p = \max_{v \in \mathcal{A}_j}\left((\hat{\boldsymbol{k}}^v_j)^T\mathbf{M}_{s_j,t_j+1}\mathbf{r}_j - V^v_j\right)$
        $\mathcal{A}_{j+1} = \{v \in \mathcal{A}_j \mid (\hat{\boldsymbol{k}}^v_j)^T\mathbf{M}_{s_j,t_j+1}\mathbf{r}_j + V^v_j \ge p\}$,
          where $V^v_j = B'(K(v,v) - (\hat{\boldsymbol{k}}^v_j)^\top\mathbf{M}_{s_j,t_j+1}\hat{\boldsymbol{k}}^v_j)$.
11: end for
```

More formally, here each phase $j \in \lceil \log T \rceil$ has 2^{j-1} rounds, which begins at round $s_j = 2^{j-1}$ and ends at $t_j = \min(2^j - 1, T)$. At each round t, *SupOUCB* plays the arm with largest confidence (line 4). At the end of each phase, we eliminate the arms $v \in [N]$ that are not promising in terms of their optimistic estimated reward — the ridge estimate $\hat{\boldsymbol{f}}_{s_j,t_j+1}$ added with a confidence term V^v_j (line 10). Algorithm 2 describes *SupOUCB* .

SupOUCB gives the following confidence bound on the estimated reward per phase j:

Lemma 7. *Let $\delta \in (0,1)$, and is as defined in SupOUCB . Then at any phase $j \in [\lceil \log T \rceil]$, for any $v \in \mathbb{R}^N$, we have $Pr\left(|\hat{\boldsymbol{f}}_{s_j,t_j+1}(v) - \boldsymbol{f}(v)| \le B'\left(1 - (\hat{\boldsymbol{k}}^v_j)^\top\mathbf{M}_{s_j,t_j+1}\hat{\boldsymbol{k}}^v_j\right)\right) \ge 1-\delta$, $\hat{\boldsymbol{f}}_{s_j,t_j+1}(v) = (\hat{\boldsymbol{k}}^v_j)^T\mathbf{M}_{s_j,t_j+1}\mathbf{r}_j$ being the reward estimate of arm v at the end of phase j (from Lem. 4).*

Theorem 8. *Given any $\delta \in (0,1)$, and $\boldsymbol{f} \in [-1,1]^N$, with probability at least $1-\delta$, the regret of SupOUCB algorithm with embedding kernel $\mathbf{K} \in \mathcal{K}(G)$ is:*

$$R_T \le 8\left(R\sqrt{\left(r(\mathbf{K}) + 2\sqrt{r(\mathbf{K})\log\frac{1}{\delta}} + 2\log\frac{1}{\delta}\right)} + B\max\left(1, \frac{1}{\sqrt{\gamma}}\right)\right) \times \sqrt{r(\mathbf{K})T\log\left(1+\frac{T}{\gamma}\right)} = O(r(\mathbf{K})\sqrt{T\log T})$$

Remark 9. *Given a fixed choice of embedding $\mathbf{K}$, thus our regret bound depends of embedding rank $r(\mathbf{K})$. The question that still remains is to actually find an embedding with lowest possible rank that optimally fits the reward vector $\boldsymbol{f}$ with small enough ϵ $\big(= O(\frac{1}{T})\big)$ that leads to a sublinear regret. This answers tradeoff of Remark 2.*

4 Towards Interpretable Bounds

In this section we explore the relationship of the regret bounds, obtained in the previous section, with the structural properties of the underlying graph G. In particular we show the optimal regret can be linked to the Chromatic number of its complement graph $\chi(\overline{G})$. We start with the observation that our regret bounds in Thm. 6 and 8 suggests that the best possible regret can be achieved with the orthonormal embedding of minimum rank defined as:

Definition 10. Orthonormal Rank [32]. *Given any graph G, its orthonormal rank is defined as $d^* = \min\{r(\mathbf{U}) \mid \mathbf{U} \in Lab(G)\}$. We denote the embedding corresponding to d^* by $\mathbf{U}^*$.*

Corollary 11. *Given a graph G, for any $\delta \in [0,1]$, with probability at least $(1-\delta)$,*

1. *The regret of SupOUCB with embedding kernel $\mathbf{K}^* = \mathbf{U}^{*\top}\mathbf{U}^*$ is given by, $R_T = O\left(\chi(\overline{G})\sqrt{T\log T}\right)$.*
2. *Computing the kernel $\mathbf{K}^*$ is NP-complete problem.*

4.1 Implication of our Regret Bound

In this section we analyze what improvement does our new regret guarantee (Cor. 11) offer over the existing results (specifically the $\tilde{O}\Big(\sqrt{\tilde{d}(G,T)T}\Big)$ regret of [29] due to having a very similar problem setting), though some specific graph instances:

Example-1: Union of cliques. Consider the graph $G([N],E)$ to be a union of c cliques, each with $\frac{N}{c}$ vertices, $c > 0$ being some constant. E.g. $c = 1$ for Complete graph, $c = 10$ for union of 10 cliques etc.

Note that, for above family of graphs, $\chi(\overline{G}) = c$. Also if $\mathbf{L}$ denotes the graph Laplacian, the eigenvalues of $\mathbf{L}$ are: $\lambda_i(\mathbf{L}) = 0$, $\forall i \in [c]$, and $\lambda_i(\mathbf{L}) = \frac{N}{c}$, $\forall i > c$. Now the effective dimension $\tilde{d}(G,T)$ [29] of the problem is defined to be the largest k such that: $(k-1)\lambda_k(\mathbf{L}) \le \frac{T}{\log(1+\frac{T}{\lambda})}$, $\lambda > 0$ being some constant. Clearly for large enough T, $\tilde{d}(G,T)$ can shoot up to N as it is an increasing function of T. Let us fix $c = 10$, $N = 100$, and $\lambda = 1$. Then for any time iteration $T > 300$, we have $\frac{T}{\log(1+\frac{T}{\lambda})} > \frac{300}{\log(301)} \approx 121.031 > 12 * 10$, which implies it has to be the case that $\tilde{d}(G,T) > 13$ as $\lambda_k(\mathbf{L}) = \frac{N}{c} = 10$ for any $k > 10$. Similarly for any $T > 5000$, $\tilde{d}(G,T)$ reaches 100 which becomes is orderwise larger than $\chi(\overline{G})$. Whereas $\chi(\overline{G}) = 10$ is a constant through out, independent of T. Thus the regret of $\tilde{O}(\sqrt{\tilde{d}(G,T)T})$ becomes $\tilde{O}(\sqrt{NT})$ for large T whereas our proposed methods give just $\tilde{O}(\sqrt{T})$ regret (Cor. 11).

Example-2: Regular graphs Consider a r-regular graph $G([N],E)$ with each node of degree $r \in [N]$. Clearly $\chi(\overline{G}) = n - r$. Let us construct a dense regular graph with $r = 90$, $N = 100$, and assume $\lambda = 1$. It can be shown that for this graph the largest eigenvalue of the Laplacian is $\lambda_N(\mathbf{L}) = r$. From a similar calculation as of Example 1, we here get that as T reaches $O(10^4)$, $\tilde{d}(G,T) \to N$ leading to a regret of $\tilde{O}(\sqrt{NT})$, whereas $\chi(\overline{G})$ being only $n - r = 10$, we are done with just $\tilde{O}(\sqrt{T})$ regret.

Example-3: Complement of Planar graphs. Consider the graph $G([N],E)$ to be any planar graph, then we know that $\chi(\overline{G}) \le 4$ (Four-color theorem [5]). But similarly in this case too, for large enough T, $\tilde{d}(G,T)$ becomes N.

We can show similar results on more graph families, Sec. 4.3 shows few more examples.

4.2 Greedy Graph Embeddings

Cor. 11 shows the existence of an optimal embedding kernel $\mathbf{K}^*$ which leads to $O\left(\chi(\overline{G})\sqrt{T\log T}\right)$ regret, however finding such $\mathbf{K}^*$ is NP-hard [30]. Hence we propose a polynomial time embedding scheme, namely *Greedy Graph Embedding*, which has small rank for a large family of graphs, and can be shown to perform optimally (same as $\mathbf{K}^*$) on certain specific graph families.

Algorithm 3 Coloring based Orthogonal Embedding

```
input  Coloring function of $G$, $\mathcal{C} : V(\overline{G}) \mapsto \mathbb{N}$
output $\mathbf{U}_c \in \mathbb{R}^{N\times|V|}$ s.t. $\mathbf{U}_c \in Lab(G)$, $r(\mathbf{U}_c) = |\mathcal{C}|$.
for color classes $C_k \in \mathcal{C}(\overline{G}) = \{C_1, \ldots, C_{|\mathcal{C}|}\}$ do
    Embed each node $i \in C_k$ by $\mathbf{e}_k$, i.e. set $\mathbf{U}_{ci} = \mathbf{e}_k$,
    $\mathbf{e}_k \in \{0,1\}^N$ being the $k^{th}$ standard basis of $\mathbb{R}^N$.
end for
```

A coloring of a graph $G(V,E)$ can be defined as a function $\mathcal{C} : V(G) \to \mathbb{N}$ where $\mathcal{C}(i) \ne \mathcal{C}(j)$ if $(i,j) \notin E$, for any $i, j \in V$. We define (with a slight abuse of notation) $|\mathcal{C}| := |\{\mathcal{C}(v) \mid v \in V\}|$ as the number of colors used by $\mathcal{C}$ to color the nodes of G. The graph is said to be c colorable if there exists a coloring function $\mathcal{C}$ such that $c = |\mathcal{C}|$. Also given a coloring $\mathcal{C}$, its color classes, denoted by $\mathcal{C}(G)$, are obtained by clustering the nodes of same color together, i.e $\mathcal{C}(G) = \{C_1, \cdots, C_c\}$, where $C_k = \{i \in V \mid \mathcal{C}(i) = k\}$. Clearly $\cup_{i=1}^{c} C_i = V$ and $C_i \cap C_j = \emptyset$. It is easy to see that given $\mathcal{C}$, one can derive it color classes $\mathcal{C}(G)$ in $O(|V|)$ time. Given a graph $G(V,E)$, we below give an algorithm to construct an orthogonal embedding $\mathbf{U} \in Lab(G)$, with embedding rank $r(\mathbf{U}) = |\mathcal{C}(\overline{G})|$, provided any coloring function its complement graph $\overline{G}$, say $\mathcal{C} : V(\overline{G}) \mapsto \mathbb{N}$. (Alg. 3).

Lemma 12. *The embedding $\mathbf{U}_c$ returned by Algorithm 3 belongs to the class of orthogonal embedding $Lab(G)$ with rank $|\mathcal{C}|$. Thus if $\mathbf{K}_c = \mathbf{U}_c^\top \mathbf{U}_c$, then $\mathbf{K} \in \mathcal{K}(G)$ such that $r(\mathbf{K}_c) = |\mathcal{C}|$. Further Algorithm 3 runs in $O(N)$ time, just linear in number of nodes.*

Corollary 13. *Given a graph $G(V,E)$, and a coloring function $\mathcal{C} : V(\overline{G}) \mapsto \mathbb{N}$ of $\overline{G}$, one can find an embedding kernel $\mathbf{K}_c \in \mathcal{K}(G)$ in $poly(|V|)$ time, such that for any $\delta \in (0,1)$, with probability at least $(1-\delta)$, SupOUCB achieves regret $R_T = O(|\mathcal{C}|\sqrt{T\log T})$, using $\mathbf{K}_c$ as the embedding kernel (where the true reward $f \in \mathbb{R}^N$ lies in the column space of $\mathbf{K}^*$, i.e. $f = \mathbf{K}^*\boldsymbol{\beta}$).*

However note that, in Corollary 13, *SupOUCB* requires the knowledge of the actual color classes $\mathcal{C}(\overline{G})$, and it leads to least regret when $|\mathcal{C}| = O(\chi(\overline{G}))$. But this requires an optimal graph coloring algorithm which is known to be a NP-Hard problem in general. One can potentially use any of the existing approximate graph coloring algorithm for the purpose, e.g. using greedy graph coloring [19]. The algorithm is described below:

Algorithm 4 Greedy Graph Coloring Algorithm

```
input : Graph $G(V,E)$, and an ordered list of vertices
    $\mathbf{W} = (v_1, v_2, \cdots, v_{|V|})$, $v_i \in V, \forall i \in [|V|]$.
output A coloring function of graph $G$, $\mathcal{C}_g : V \mapsto \mathbb{N}$.
Initialize coloring $\mathcal{C}_g(v_i) = 0,\ \forall v_i \in \mathbf{W}$
for $i = 1$ to $|V|$ do
    $\mathcal{C}_g(v_i) \leftarrow \min[|V|] - \cup_{\{j|(i,j)\in E\}}\mathcal{C}_g(v_j)$
end for
```

Theorem 14. *[19] Given any graph G', the number of colors used by Alg. 4 is at most $d_{max}(G') + 1$.*

Above theorem along with Cor. 13 immediately leads to:

Corollary 15. *Given a graph $G(V,E)$, if $\mathcal{C}_g(\overline{G})$ denotes the coloring function of $\overline{G}$ obtained using greedy coloring algorithm, and $\mathbf{U}_g$ be the embedding returned by Algorithm 3 upon $\mathcal{C}_g(\overline{G})$ as the input, then for any $\delta \in (0,1)$, with high probability $(1-\delta)$, SupOUCB achieves the regret $R_T = O(d_{\max}(\overline{G})\sqrt{T\log T})$ upon using $\mathbf{K}_g = \mathbf{U}^\top\mathbf{U}$ as the embedding kernel, $d_{\max}(\overline{G})$ being the maximum degree of graph $\overline{G}$.*

Remark Similarly one can derive results similar to Cor. 13 and 15 for our other algorithm *OUCB* as well.

4.3 Be Greedy: Graph Families where Greedy Embedding Performs Optimally

We now show that how our proposed algorithms, *OUCB* and *SupOUCB* perform optimally with greedy embedding $\mathbf{K}_g$ (i.e. same as that of knowing true $\mathbf{K}^*$) on large family of graphs. Due to Cor. 15, we see that for any graph G such that maximum degree of $\bar{G}$ is constant, greedy embeddings yields a regret guarantee of just $O(\sqrt{T\log T})$. Let us study some specific graph families: (1). Complete or isolated graphs, note that $\chi(\overline{G})$ is equal to 1 and N respectively, and $r(\mathbf{K}_g) = \chi(\overline{G})$. (2). When G is complement of a q-regular graphs, $\chi(\overline{G}) = q+1 = r(\mathbf{K}_g)$ (last equality follows from Thm. 14). Thus our proposed algorithms with greedy kernel embedding leads to the regret of $O(q\sqrt{T\log T})$. (3). If G is a k-ary tree, similarly we can show that $\chi(\overline{G}) = k+1 = r(\mathbf{K}_g)$ which implies $O(k\sqrt{T\log T})$ regret. (4). For G to be union of k disconnected cliques, $\chi(\overline{G}) = k$, and again Thm. 14 shows that $r(\mathbf{K}_g) = k$ which leads to regret guarantee of $O(k\sqrt{T\log T})$. (5). For complement of planar graphs one can obtain a regret of $O(\sqrt{T\log T})$ as greedy algorithm colors any planar graph with at most 6 colors, i.e. $r(\mathbf{K}_g) \le 6$ [13]. (6). For *Erdős Réyni random $G(n,p)$ graphs* (with constant $p \in [0,1]$), we also have $r(\mathbf{K}_g) \le 2\chi(\overline{G})$ as for almost all $G(n,p)$ graphs greedy gives a two factor approximation of the chromatic number [18], which again implies optimal learning rate. Our experimental results also shows the advantages of *greedy embeddings* on synthetic and real world graphs (Sec. 6).

5 Lower Bound

In this section, we prove a matching lower bound of the regret guarantee as derived in (Cor. 11, Sec. 4).

Theorem 16. *For any online learning algorithm $\mathcal{A}$, there exists a graph $G([N],E)$, and a reward assignment $\boldsymbol{f} \in \mathbb{R}^N$, such that regret incurred by $\mathcal{A}$ on our problem setup is atleast $\Omega\Big(\sqrt{\chi(\overline{G})T}\Big)$, given any time horizon $T > 0$.*

Proof. (sketch). The main idea is to construct a graph composed of $\chi(\overline{G})$ almost disconnected components such that nodes within a same component has identical rewards, and reduce it to standard N-armed MAB setup for which the lower bound is known to be $\Omega(\sqrt{NT})$ [7]. □

Remark 17. *Thm. 16 does not give a lower bound for any general graph, but we show a family of graphs where $R(T) = \Omega\Big(\sqrt{\chi(\overline{G})T}\Big)$, and thus the performance of our proposed algorithms (Cor. 11) are tight for these cases (up to a factor of $\sqrt{\chi(\overline{G})\log T}$).*

6 Experiments

We run experiments on both synthetic and real datasets to compare our algorithms, *OUCB* and *SupOUCB* (with *greedy graph embedding*, Sec. 4.2), against the state of the art SpectralUCB [29], LinUCB [24] and KernelUCB [35]. All the results reported are averaged across 50 runs. For all experiments, we set the confidence parameter $\delta = 0.001$, upper bound on noise $R = 0.01$, and B, the upper bound on $\|\boldsymbol{\alpha}\|$, as $B = \log T$ if $T < N$ else clamped as $B = \log N$ if $T >= N$ (as suggested in [29]). The regularization parameter γ is set using the best value from the range $[10^{-3}, 10]$ separated by a multiplicative gap of 0.1 and report the performances at which algorithms converge with the smallest regret. For KernelUCB, parameter $\eta = \sqrt{\frac{\log(2TK/\delta)}{2\lambda}}$ was set as suggested in [35].

The different experimental setups are described below:

6.1 Experiments on Synthetic Datasets

Recall that our problem formulation requires the knowledge of the underlying graph $G(V, E)$, and considers an unknown assignment of the reward vector $\boldsymbol{f} \in \mathbb{R}^N$ over the N arms (Sec. 2.1). For the experiments, we simulate the above in the following ways:

Type of graphs. We consider four types of synthetic graphs, with $N = 500$ nodes. For each graph we compute the coloring number $\mathcal{C}_g(\overline{G})$ by greedy algorithm or $\chi(\overline{G})$ (if it is easy to compute): (1) **Erdos-Reyni** ($G(N,p)$) graphs with $p = 0.03$ and for the generated graph $|\mathcal{C}_g(\overline{G})| = 234$ i.e., the estimated coloring number given by greedy algorithm, (2) **Random k-Regular** graphs with degree $k = 400$ with $|\mathcal{C}_g(\overline{G})| = 30$, (3) **Union of Disconnected cliques** with total 25 cliques each clique containing 20 nodes, and hence $\chi(\overline{G}) = 25$, and (4) **Barabasi-Albert(BA)** graphs [1] with the parameter *Connectivity* (degree) **cp** $= 3$ and $|\mathcal{C}_g(\overline{G})| = 271$.

Reward models. We experimented on the following two reward models: **1. Laplacian based rewards.** This is the reward model used in the state of the art (see Sec. 2 and 6.1 of [29]) i.e., $\boldsymbol{\alpha}$ (as used in [29]) is generated randomly with sparsity factor $k = 5$, such that $\boldsymbol{\alpha}$ is bounded by some constant C as $||\boldsymbol{\alpha}||_{\boldsymbol{\Sigma}} = \boldsymbol{\alpha}^\top \mathbf{L}\boldsymbol{\alpha} \leq C$, where $\mathbf{L} = \boldsymbol{Q\Sigma Q}^T$ (the eigenvalue decomposition of the Laplacian $\mathbf{L}$), C is set same as B described above. **2. Orthogonal embedding based rewards.** Here we use our proposed reward model (Eqn. (2)) with orthogonal embedding (Sec. 2.2), and use greedy labellings $\mathbf{K}_g$ of the corresponding graphs to generate $\mathbf{K}$ (computed using Algorithm 3).

6.1.1 Performance on Synthetic Graphs

We run our algorithms, *OUCB* and *SupOUCB* for greedy coloring(Sec. 4.2), and plot the averaged regret of all five algorithms with varying T i.e., until one of the algorithms converged, for all the four types of graph models. The comparative results for Laplacian and orthogonal embedding based rewards are respectively shown in Figure 1 and 2. Both LinUCB and KernelUCB are run with feature vectors used to generate the rewards as context vectors (i.e. depending on reward model either with Laplacian eigenvectors [29], or Orthogonal embedding (Sec. 2.2).

Discussion. Our results in Figure 1 and 2 show a vast improvement of *OUCB* over the baseline algorithms, and converges much earlier on both the reward settings and even for higher values of $|\mathcal{C}_g(\overline{G})|$. *SupOUCB* generally outperforms all the baselines but defeats to *OUCB* , in spite of having an $O(\sqrt{\log T})$-factor better regret guarantee. This behaviour corroborates with other similar algorithms, e.g. SupLinUCB [11] or Spectral Eliminator [29] was shown to fare poorly compared to LinUCB or SpectralUCB respectively, although later ones excel theoretically in similar way. KernelUCB generally performs poorly due to not being able to exploit the influence of the underlying graph on the arm rewards (locality property).

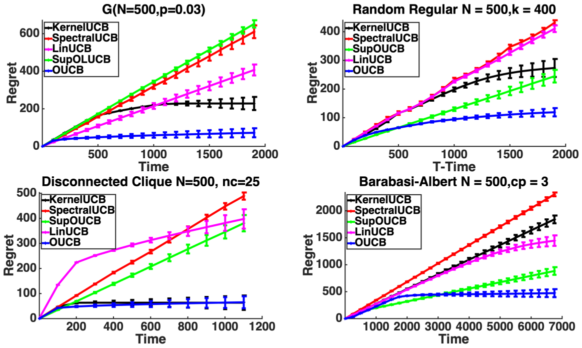

Figure 1: Regret performances on Laplacian rewards

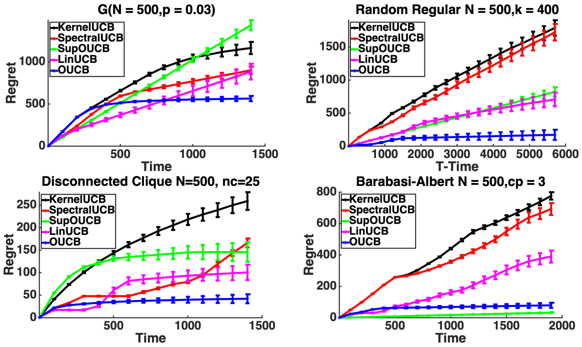

Figure 2: Performance of algorithms on orthogonal embedding based rewards, as defined in (2)

Runtime performance. We also plot the run-time performances of both *SupOUCB* and *OUCB* and compare it with that of baselines over graph families of $G(n, p)$ and disconnected cliques for $T = 2000$ across varying sizes N. We observe that runtimes are better than baselines. We see an expected increase with N across all algorithms except *SupOUCB* as it eliminates arms periodically. The performance of SpectralUCB and LinUCB are worst since they require to perform a matrix inversion at each round, whereas KernelUCB performs almost as good as our algorithms *SupOUCB* and *OUCB* .

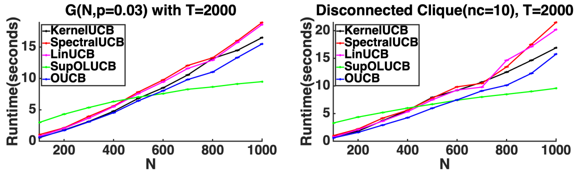

Figure 3: Running time for T rounds

6.1.2 Regret vs node size (N) and $\chi(\bar{G})$

We also run two experiments for analyzing the true effect of the graph properties on the regret of our proposed algorithms. The underlying graph is chosen to be disconnected cliques, since we have a direct handle on $\chi(\overline{G})$ value of these graphs by simply controlling its number of disconnected components. We use *OUCB* algorithm with greedy coloring embedding on our proposed reward model of (2). The first experiment compares the regret of *OUCB* for a fixed $\chi(\overline{G}) = 20$, varying N in range $1100 - 2000$ (increments of 100). On the contrary, the second experiment run for a fixed $N = 500$ with varying $\chi(\overline{G}) = 5, 10, 20, 40, 50, 100, 125, 250$. In both cases, the algorithms are run until one of them has converged.

Discussion. The results are shown in Figure 4. As expected, the regret of *OUCB* is seen to be varying only with $\chi(\overline{G})$ (right plot), and remains constant with varying N as long as $\chi(\overline{G})$ is fixed (left plot), rightfully justifying its theoretical guarantee (Cor. 11). In both cases, *OUCB* converges the fastest and its regret significantly less compared to the all other baselines due to inherent ability to exploit the dependency of the graph on the arm rewards through suitable embeddings, which others can not.

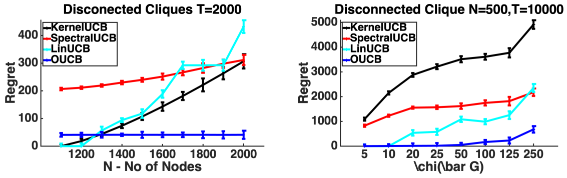

Figure 4: Regret of *OUCB* with (left) varying N fixed $\chi(\bar{G})$ and (right) varying $\chi(\bar{G})$ with fixed N

6.2 Experiments on Real-World Datasets

We use two popular real world dataset for the purpose: (1) MovieLens and (2) Flixster. For the sake of fair comparisons, we mimick the same experimental setup of [29] including graph construction and imputation of the missing ratings as discussed below. The values of the parameters are set to values as discussed earlier. Same as Sec. 6.1 LinUCB and KernelUCB are again run using the feature vectors used to construct the graphs (as described below).

MovieLens [25] It has $6k$ users who rated 1 million movies. We split the dataset into two equal parts, on one of them, we used OptSpace algorithm [31] for performing low-rank matrix factorization [2] to impute the missing ratings. On the other split, we again perform matrix factorization and using the latent vectors obtained for movies we build a similarity graph for the movies. The graph contains an edge between movie i and j if the j is one of the 10 nearest neighbors of the movie i(Euclidean distance). Note that here each user defines one independent instance of the problem: We use a random sample of 50 users, and evaluate the algorithms on each, for $T = 2000$ rounds.

Flixster [20] This dataset has 1 million users on 49000 movies with 8.2 million ratings. We extracted a subset of popular movies and active users, where each movie has at least 1000 ratings and each user rated at least 300 movies. This resulted in a dataset of 1712 movies and 8465 users. As with the MovieLens, the dataset is imputed and similarity graph is built (on 10 nearest neighbours same as Movielens) by splitting dataset and carrying out low-rank matrix factorization. A random sample of 50 users are used to evaluate the algorithms, for $T = 3000$ rounds.

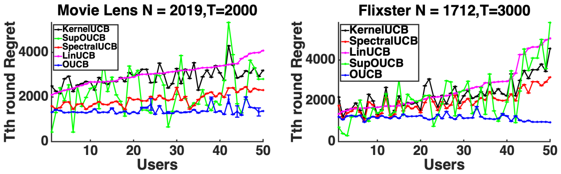

Figure 5: Performance of algorithms on real datasets

Discussion. The results are given in Figure 5, which again shows the superiority of *OUCB* over other baselines, and consistently outperforms others across different users, for both the datasets. This reflects the practicability of our reward model, as well as the effectiveness of our greedy embedding based algorithms for real world scenarios.

7 Conclusion and Future works

We address the problem of linear bandit on graphs, where arm rewards follow a locality property according to a given graph structure $G([N],E)$. For any general *orthogonal embedding* based rewards we show a regret of $\tilde{O}(r\sqrt{T})$ in terms of rank of the underlying graph embedding r. Above bound further boils down to $\tilde{O}(\chi(\bar{G})\sqrt{T})$ under minimum rank orthogonal embedding of G, which immediately relates the inherent problem complexity in terms of the structure of graph G–a faster learning rate for denser graphs–as also intuitive due to the graph-based locality assumption on rewards. However, we show computing the above minimum rank orthogonal embedding is NP-Hard in general, towards which we propose an $O(N + |E|)$ time embedding scheme–*greedy coloring*–with which our proposed algorithms are shown to perform optimally on a large family of graphs. Moreover, our experimental results reveal that our proposed *greedy embedding based algorithms* also perform well in practice on standard benchmark real datasets, and outperforms the state-of-the-art methods, both in terms of regret and runtime performances. Our findings open up new directions for exploiting graph structures on regret complexity of bandits. In future it would be interesting to explore other class of graph embeddings, and dependence of the regret on G. Analysing our problem setup for weighted graphs also remains a matter of future investigation.

Acknowledgements. We thank Samik Datta, Flipkart and Amazon CSR grant for supporting this work.

[2] `https://github.com/scheinerman/matgraph`

References

[1] R. Albert Albert-László Barabási. Emergence of scaling in random networks. *Science*, 1999.

[2] Noga Alon, Nicolo Cesa-Bianchi, Ofer Dekel, and Tomer Koren. Online learning with feedback graphs: Beyond bandits. In *JMLR WORKSHOP AND CONFERENCE PROCEEDINGS*, volume 40. Microtome Publishing, 2015.

[3] Noga Alon, Nicolo 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.

[4] Rie K Ando and Tong Zhang. Learning on graph with Laplacian regularization. In *Advances in neural information processing systems*, pages 25–32, 2007.

[5] Kenneth Appel and Wolfgang Haken. Every planar map is four colorable. *Mathematical Solitaires & Games*, 1980.

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

[7] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 2002.

[8] Swapna Buccapatnam, Atilla Eryilmaz, and Ness B Shroff. Stochastic bandits with side observations on networks. *ACM SIGMETRICS Performance Evaluation Review*, 42(1):289–300, 2014.

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

[10] Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. *arXiv:1704.00445*, 2017.

[11] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 208–214, 2011.

[12] T. Zhang D. Hsu, S. Kakade. A tail inequality for quadratic forms of subgaussian random vectors. *Electronic Communications in Probability*, 2012.

[13] Margaret M Fleck. Planar graphs, 2011.

[14] Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. In *International Conference on Machine Learning*, pages 757–765, 2014.

[15] Alison L Gibbs and Francis Edward Su. On choosing and bounding probability metrics. *International statistical review*, 70(3):419–435, 2002.

[16] Manjesh Hanawal, Venkatesh Saligrama, Michal Valko, and Rémi Munos. Cheap bandits. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2133–2142, 2015.

[17] Mark Herbster, Massimiliano Pontil, and Lisa Wainer. Online learning over graphs. In *Proceedings of the 22nd international conference on Machine learning*, pages 305–312. ACM, 2005.

[18] Thore Husfeldt. Graph colouring algorithms. *arXiv preprint arXiv:1505.05825*, 2015.

[19] Thore Husfeldt. *Graph Colouring Algorithms, Ch.13 of Topics in Chromatic Graph Theory*. Cambridge University Press, 2015.

[20] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *ACM - Recommender systems*, 2010.

[21] Vinay Jethava, Anders Martinsson, Chiranjib Bhattacharyya, and Devdatt Dubhashi. Lovász ϑ function, svms and finding dense subgraphs. *The Journal of Machine Learning Research*, 14(1):3495–3536, 2013.

[22] Tomas Kocak, Gergely Neu, and Michal Valko. Online learning with noisy side observations. In *AISTATS*, pages 1186–1194, 2016.

[23] Tomas Kocak, Gergely Neu, Michal Valko, and Rémi Munos. Efficient learning by implicit exploration in bandit problems with side observations. In *Advances in Neural Information Processing Systems*, pages 613–621, 2014.

[24] J. Langford L. Li, W. Chu and R.E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *WWW*, 2010.

[25] S. Lam and J. Herlocker. Movielens 1m dataset. https://grouplens.org/datasets/movielens/1m/, 2003.

[26] John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. In *Advances in neural information processing systems*, pages 817–824, 2008.

[27] Tyler Lu, Dávid Pál, and Martin Pál. Contextual multi-armed bandits. In *Proceedings of the Thirteenth international conference on Artificial Intelligence and Statistics*, pages 485–492, 2010.

[28] Shie Mannor and Ohad Shamir. From bandits to experts: On the value of side-observations. In *Advances in Neural Information Processing Systems*, pages 684–692, 2011.

[29] Branislav Kveton Michal Valko, Remi Munos and Tomas Kocak. Spectral bandits for smooth graph functions. In *Proceedings of International Conference on Machine Learning*, 2014.

[30] Rene Peeters. Orthogonal representations over finite fields and the chromatic number of graphs. 1994.

[31] S. Oh R. Keshavan and Montanari. A. matrix completion from a few entries. In *IEEE International Symposium on Information Theory*, 2009.

[32] Geoff Tims. *Haemers' Minimum Rank*. PhD thesis, Department of Mathematics, Iowa State University, 2013.

[33] T.P. Hayes V. Dani and S.M. Kakade. Stochastic linear linear optimization under bandit feedback. In *The Annual Conference on Learning Theory*, 2008.

[34] Michal Valko. *Bandits on graphs and structures*. PhD thesis, Ecole normale superieure de Cachan-ENS, 2016.

[35] Michal Valko, Nathaniel Korda, Rémi Munos, Ilias Flaounas, and Nelo Cristianini. Finite-time analysis of kernelised contextual bandits. *arXiv preprint arXiv:1309.6869*, 2013.

[36] Sharan Vaswani, Mark Schmidt, and Laks VS Lakshmanan. Horde of bandits using gaussian markov random fields. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

[37] Joannes Vermorel and Mehryar Mohri. Multi-armed bandit algorithms and empirical evaluation. In *European conference on machine learning*, pages 437–448. Springer, 2005.

[38] Xuefeng Xu. Generalization of the sherman–morrison–woodbury formula involving the schur complement. *Applied Mathematics and Computation*, 309:183–191, 2017.

[39] D. Pal Y. Abbasi-Yadkori and C. Szepesvari. Improved algorithms for linear stochastic bandits. In *Neural Information Processing Systems*, 2011.