EFFICIENT DETECTION AND LOCALIZATION ON GRAPH STRUCTURED DATA

Manjesh Kumar Hanawal and Venkatesh Saligrama

Boston University, Massachusetts, USA

ABSTRACT

The problem of efficiently identifying regions of interest arises in the context of surveillance, monitoring and exploration of a large area or network involving social, sensor, communication network data. We formulate these problems in terms of locating optimum values of signals on graphs. In this perspective we associate features with nodes/edges of a graph where the maxima/minima of these features correspond to interest points. We develop an algorithm that adaptively probes local sub-collection of nodes (local regions) on the graph and sequentially refines the search space from noisy averaged returns from each probed region. The size of the region determines the cost of the probe with larger regions corresponding to lower cost. Our goal is to minimize regret after T rounds with minimal budget/cost. Under suitable smoothness conditions on the signal we show that after T rounds the cumulative regret scales optimally as $O(\sqrt{T})$ with significant cost gain over other state-of-art techniques.

1. INTRODUCTION

The problem of searching for a region of interest (ROI) in a large area or network can be both time and resource intensive. The goal of this paper is to develop a sequential learning mechanism that identifies the ROI in a given time with minimum cost. We formulate this problem as finding optimum of a signal defined on the nodes of a graph and develop an algorithm that narrows down the search operation around the optimal node using low cost search actions.

This problem arises in a number of applications involving surveillance, monitoring and exploration of a large area or network. In sensor network (SNET) surveillance, sensors usually have a limited sensing range [1] and can only reliably sense presence/absence of target within their immediate vicinity. Consequently, the sensed field decays smoothly with distance from the target. To account for limited energy budget, several papers have dealt with sleep/wake scheduling (see [2]). Here a group of sensors are woken up sequentially based on probable locations of target. Those sensors then in principle coherently beam-form their signals to the fusion center, which then receives an aggregated signal. There is an inherent energy-resolution tradeoff here. Larger the pool of sensors involved in beam-forming smaller the required SNR and hence lower the energy requirement [1, 3]. On the other hand larger the sensor pool, higher the target ambiguity. A similar cost-resolution tradeoff also arises in aerial reconnaissance. Larger areas can be surveilled at higher altitudes more quickly (lower cost) but suffers lower resolution.

We model these instances in terms of a graph with n nodes denoting locations, edges denoting neighboring locations and feature values denoting the signal associated with a location. The goal in these cases can be abstractly viewed as locating nodes with largest signals. Our approach is to sequentially probe larger regions (at low cost) in the initial phases and switch to high-cost high-resolution probes once a rough estimate has been obtained. Our goal is to develop strategies that result in small cumulative regret over T rounds (with $T \ll n$) while minimizing the cumulative cost of the probes. Regret is described as the difference between the largest signal value and the average signal value from the probed region. The noisy returned signal is commonly referred to as the reward.

In many applications such as in SNET surveillance, the signal around the interest point decays slowly. We can express signals in terms of the graph eigenvectors. A smooth graph signal is expressed as a linear combination of eigenvectors associated with the smallest eigenvalues, and the learning of graph signal can be posed as regression on the eigenvectors of the graph Laplacian [4], [5]. We use the framework of *linear bandits* [6], [7] to learn optimum of smooth reward function on a graph, where the set of arms correspond to choice of nodes and their neighbors. In our setting, the arms are themselves graph signals, and their cost is defined using *graph Fourier transforms*.

Related work: We briefly describe several works that exploit structural properties of graph signals in learning. The works in [8], [9], [10], [11], present a sampling perspective for reconstructing signals from samples collected from a subset of nodes. In contrast our goal is to adaptively determine a region-based sampling scheme to identify maxima of the signal. Our work is most related to [12]. That paper develops a bandit approach to learning maxima and minima for smooth functions on a graph. It proposes the so called SpectralUCB algorithm and presents regret guarantees of order $d\sqrt{T}$, where T is the number of rounds and d is effective dimension that depends on T. Both T and d can in general be much smaller than n (the number of nodes on the graph). Other related

This research has been partially supported by NSF grant 1330008.

works include [13], where each node is assumed to be a linear bandit with unknown parameters that are smooth on the graph. In [14] the authors assume that the node rewards are correlated. They exploit the fact that observing rewards from a node reveals information of its neighbors. For recent works that consider cost in bandits see [15] and [16] as well.

Our contributions: We provide a setup using linear bandits for searching on large area that can be represented as graphs. In contrast to the above approaches we describe cost as well as regret in terms of arms of the bandit and the underlying signal. Both the signal and the cost is then related to the spectral properties of the graph. We develop an algorithm that aims to maximize the rewards collected from the arms while minimizing the cost. We show that our algorithm not only guarantees regret bound of the order $d\sqrt{T}$, but also guarantees reduction in cost that is of order T when compared to the SpectralUCB.

The paper is organized as follows. In Section 2, we give problem formulation and setup the notations. In Section 3, we present our algorithm and regret analysis. In Section 4, we demonstrate performance of our algorithm on two synthetic datasets. Finally, in Section 5 we conclude and discuss future work.

2. PROBLEM SETUP

Let $\mathcal{G} = (V, E)$ denote an undirected graph with number of nodes $|V| = n$. Let $\mathbf{s} : V \to \mathcal{R}$ denote a signal on $\mathcal{G}$, and $\mathcal{S}$ the set of all possible signals on $\mathcal{G}$. Let $\mathcal{L} = D - W$ denote the unnormalized Laplacian of the graph $\mathcal{G}$, where $W = w_{ij}$ is the weight matrix and D is the diagonal matrix with entries $d_{ii} = \sum_j w_{ij}$. We denote the eigenvalues of $\mathcal{L}$ as $0 = \lambda_1^{\mathcal{L}} \leq \lambda_2^{\mathcal{L}} \leq \cdots \leq \lambda_n^{\mathcal{L}}$, and the corresponding eigenvectors as $\mathbf{q_1}, \mathbf{q_2}, \cdots, \mathbf{q_n}$. Equivalently, we write $\mathcal{L} = \mathbf{Q}\mathbf{\Lambda}_{\mathcal{L}}\mathbf{Q}'$, where $\mathbf{\Lambda}_{\mathcal{L}} = diag(\lambda_1^{\mathcal{L}}, \lambda_2^{\mathcal{L}}, \cdots, \lambda_n^{L})$ 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.

We define a reward function on $\mathcal{G}$ as a linear combination of the eigenvectors. For a given weight vector $\alpha \in \mathcal{R}^n$, the reward function is defined as

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

The parameter α denotes the smoothness of the graph. If α is such that large coefficients correspond to the eigenvectors of the smaller eigenvalues and vice versa, then f_α is a smooth function of $\mathcal{G}$. We denote the parameter that defines the true reward function as α^* and is unknown.

Signal Space: In the literature on sampling theory on graphs, a function defined on the nodes is referred to as a signal. We deviate from this convention and refer to f_α^* as a reward function and weights on the nodes as signals. Let $\mathcal{S} \subset \{\mathbf{s} \in [0,1]^n : \sum_{i=1}^n s_i = 1\}$ denote the set of signals. Each $\mathbf{s} \in \mathcal{S}$ signal is of the form $s_i = 1/\text{supp}(\mathbf{s})$, for all $i = 1, 2, \cdots, n$, where $\text{supp}(\mathbf{s})$ denotes the number of positive elements in $\mathbf{s}$. The inner product of $f_{\boldsymbol{\alpha}^*}$ and a signal $\mathbf{s}$ gives average of $\text{supp}(\mathbf{s})$ number of nodes. Note that $|\mathcal{S}| = 2^n - 1$. For all $0 < w \leq n$, let $\tilde{\mathcal{S}}_w = \{\mathbf{s} \in \mathcal{S} : \text{supp}(\mathbf{s}) = w\}$ denote the set of signals of width w. For a given $w > 0$, we will be interested in a subset $\mathcal{S}_w \subset \tilde{\mathcal{S}}_w$ with n elements, one corresponding to each node of the graph. We denote the element in $\mathcal{S}_w$ associated with node i as $\boldsymbol{s}_i^w$. Let node i has neighbors at $\{j_1, j_2, \cdots j_{w-1}\}$, then $\boldsymbol{s}_i^w$ is of the form

$$\mathbf{s}_{ik}^w = \begin{cases} 1/w & \text{if } k = i \\ 1/w & \text{if } k = j_i, \;\; i = 1, 2, \cdots, w-1 \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

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 smaller than degree of node i, in defining s_i^w we only consider neighbors with larger edge weights. If all the weights are the same, then we select w neighbors arbitrarily. In the following we use the phrase 'probing with signal $\mathbf{s}$' to mean that signal $\mathbf{s}$ is used to observe reward from nodes.

Signal cost: Let $\tilde{\mathbf{s}}$ denote the *graph Fourier transform* (GFT) of signal $\mathbf{s} \in \mathcal{S}$. Analogous to Fourier transforms of a continuous function, GFT gives amplitudes associated with graph frequencies. The GFT coefficient of a signal 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 $\tilde{\mathbf{s}_i}, i = 1, 2, \cdots, n$ is the GFT coefficient associated with frequency λ_i. We define cost of probing the graph $\mathcal{G}$ with a signal as sum of squares of its GFT coefficients weighted by the corresponding frequency. Let $C : \mathcal{S} \to \mathcal{R}_+$ denote the cost function[1]. Then,

$$C(\mathbf{s}) = \sum_{i=}^{n} \lambda_i \tilde{\mathbf{s}}_i^2 = \tilde{\mathbf{s}}'\mathbf{\Lambda}_{\mathcal{L}}\tilde{\mathbf{s}} = \mathbf{s}'\mathcal{L}\mathbf{s}.$$

For a given matrix V, we denote V-norm of a vector $\mathbf{a}$ as $\| \mathbf{a} \|_V = \sqrt{\mathbf{a}'V\mathbf{a}}$. Then $C(\mathbf{s}) = \| \mathbf{s} \|_{\mathcal{L}}^2$. As graph Laplacian is a difference operator we can also write $C(\mathbf{s}) = \sum_{i \sim j}(s(i) - s(j))^2$, where the summation is over all the unordered node pairs $\{i, j\}$ for which node i is adjacent to node j. For signal $\mathbf{s}_i^w$ the above expression can be written after simplification as

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

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

We assume that by probing the graph with a signal yields a reward/information[2] that is proportional to the inner product

[1] In defining the cost we set $W = A$, where A is the adjacency matrix, and made the graph symmetric by adding self loops on the nodes to make their degree n

[2] In radar applications, this is through returned signal strength. In SNETs, this is average measurement from the sensors

of the signal used and the graph reward function f_{α^*}. Let $F_{\mathcal{G}} : \mathcal{S} \to \mathcal{R}$ defined as

$$F_{\mathcal{G}}(\mathbf{s}) =< \mathbf{s}, Q\alpha^* >=< \tilde{\mathbf{s}}, \alpha^* >$$

denote the reward obtained from signal $\mathbf{s}$. Thus, each signal is associated with linear reward and quadratic cost.

Let $\mathbf{s}^* = \arg\max_{\mathbf{s}\in\mathcal{S}} F_{\mathcal{G}}(\mathbf{s})$ denote the signal that gives the maximum. This is a straightforward linear optimization problem if the function parameter α^* is known. When α^* is unknown we can learn the function through a sequence of measurements.

2.1. Learning Setting and Performance Metrics

The learning setting is the following. The recommender uses a policy $\pi : \{1, 2, \cdots, T\} \to \mathcal{S}$ that assigns at step $t \leq T$, signal $\pi(t)$. In each step t, the recommender obtains a noisy reward such that

$$r_t = F_{\mathcal{G}}(\pi(t)) + \epsilon_t,$$

where ϵ_t is assumed to be R-sub Gaussian for any t. The goal of the recommender is to learn a policy π that minimizes the cumulative (pseudo) regret with respect to a policy that always picks the best signal with respect to the parameter α^* keeping the total cost incurred as low as possible.

The cumulative regret and the total cost of policy π is defined, respectively, as

$$R_T = TR(\mathbf{s}^*) - \sum_{t=1}^{T} R(\pi(t)) \quad (3)$$

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

The goal of the recommender is to learn a policy that minimizes R_T while keeping the C_T as small as possible.

Remark 1 *If we restrict the signal space to $\mathcal{S} = \{\mathbf{e}_i : i = 1, 2, \cdots, n\}$, where $\mathbf{e}_i$ denotes a binary signal with i^{th} component set to 1 and all the other components set to 0, then only one node is probed in each step. In this setting the cost is the same for all the signals, i.e., $C(\mathbf{e_i}) = 1$ for all i. This special case is studied in [12] where $C_T = T$. We take this setting as a benchmark to compare performance and cost of our algorithm.*

2.2. Assumptions:

We assume that the reward function satisfies the following smoothness properties.

Global smoothness : As in [12] we assume that $\mathbf{\Lambda}$-norm of α^* characterizes the smoothness of graph and is bounded. I.e.,

$$\exists \ c > 0 \ \text{ such that } \ \| \boldsymbol{\alpha}^* \|_{\mathbf{\Lambda}} \leq c \quad (4)$$

Here $\mathbf{\Lambda} = \mathbf{\Lambda}_{\boldsymbol{\mathcal{L}}} + \lambda I$, and $\lambda > 0$ is used to make $\mathbf{\Lambda}_{\boldsymbol{\mathcal{L}}}$ invertible. Before we state our next assumption we recall the definition of effective dimension. Let $\lambda = \lambda_1 \leq \lambda_2 \cdots \leq \lambda_n$ denote the diagonal elements of Λ.

Definition 1 (Effective dimension [12]) *Given T, effective dimension is the largest d such that:*

$$\lambda_d(d-1) \leq \frac{T}{\log(T/\lambda + 1)}.$$

Note that effective dimension depends on T. It characterizes the number of non-negligible dimensions in which solution of penalized least-squares (used in the alogrithm) may lie.

Local smoothness: The global assumption implies that the rewards of the neighbors are similar. We localize this notion around the optimal node and assume that the average reward of neighbors of the optimal node varies smoothly with respect to the reward of the optimal node. Let $\mathbf{s}^*_w \in \mathcal{S}_w$ denote $w-$width signal associated with the optimal node.

$$\forall \ w \leq g, \ |F_{\mathcal{G}}(\mathbf{s}^*_w) - F_{\mathcal{G}}(\mathbf{s}^*)| \leq c/(\lambda_{d+1} - w), \quad (5)$$

where λ_{d+1} is the eigenvalue corresponding to the effective dimension and g denotes degree of the optimal node. Note that Assumption (5) is made only for the optimal node. The average reward from a larger neighborhood of optimal node can degrade considerably and often not a good choice.

3. ALOGRITHM

We present an algortihm similar to LinUCB [17] and SpectralUCB [12] for regret minimization. The main difference between our algorithm and these algorithms comes from the enlarged signal space, which allows us to observe average reward from a subset of neighbors of each node in each time step. As we noted earlier, a single node probe provides more information about the node than a multi-node probe. On the other hand a multi-node probe is less expensive. Our goal is not only to minimize the regret but also to minimize the total cost. We tradeoff this conflicting requirement by switching from multi-node to single node signals as the learning process progresses. In particular, we split the time horizon into J stages, and as we move from state j to $j+1$ we use more costly signals, which corresponds to using signals of smaller widths. The algorithm uses the signals of different widths in each stage as follows: Stage $j = 1 \cdots, J$ consists of time steps from 2^{j-1} to $2^j - 1$ and uses j-width signals only.

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

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

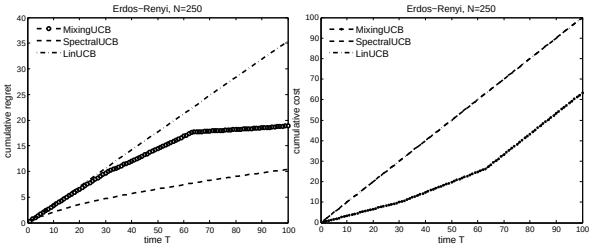

Fig. 1. ER graph: Cumulative regret and cost gain

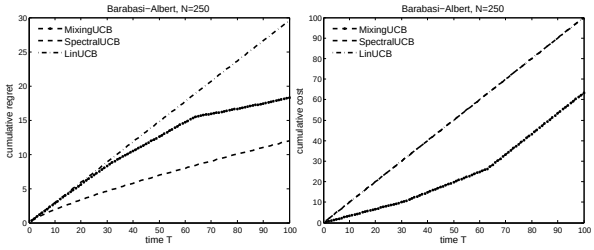

Fig. 2. BA graph: Cumulative regret and cost gain

Algorithm 1 MixingUCB

Input:
$\mathcal{G}$: the graph, T: number of steps
λ, δ: regularization and confidence parameters
R, c: Upper bound on noise and norm of α
Initialization:
$d \to \arg\max\{d : (d-1)\lambda_d \leq T/\log(1+T/\lambda)\}$
$V_0 \leftarrow \Lambda_{\mathcal{L}} + \lambda I, S_0 \leftarrow 0, r_0 \leftarrow 0$
for $j = 1 \to J$ **do**
  for $t = 2^{j-1} \to \min\{2^j - 1, T\}$ **do**
    $S_t \leftarrow S_{t-1} + r_{t-1}\tilde{\mathbf{s}}_{t-1}$
    $V_t \leftarrow V_{t-1} + \tilde{\mathbf{s}}_{t-1}\tilde{\mathbf{s}}'_{t-1}$
    $\hat{\boldsymbol{\alpha}}_t \leftarrow V_t^{-1} S_t$
    $\beta_t \leftarrow 2R\sqrt{d\log(1+t/\lambda) + 2\log(1/\delta)} + c$
    $\mathbf{s}_t \leftarrow \arg\max_{\mathbf{s} \in \mathcal{S}_{J-j+1}} \tilde{\mathbf{s}}'\hat{\alpha}_t + \beta_t \parallel \tilde{\mathbf{s}} \parallel_{V_t^{-1}}$
  end for
end for

We have the following result for our strategy of progressively switching from inexpensive to expensive probing signals.

Theorem 1 *Set $J = \lceil \log T \rceil$ in the algorithm. Let λ be the smallest eigenvalue of Λ. If $< \tilde{\mathbf{s}}, \alpha^* > \in [-1, 1]$ for all $\mathbf{s} \in \mathcal{S}$ and assumptions (4) and (5) hold. Then, the cumulative regret of the algorithm is with probability at least $1 - \delta$ bounded as:*

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

where d is the effective dimension. Further, the total cost is bounded as

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

Proof Sketch: To prove the bound on the cumulative regret, we first obtain a bound on instantaneous regret at time t that involves sum of two parts. The first part is the difference between reward obtained from the signal of width w used at time t and the reward from signal of width w corresponding to the optimal node. The second part is the upper bound in Assumption (5). We then bound the cumulative regret by again bounding the sums of each part. We use the same arguments as in the proof of Theorem 1 in [12] and the definition of effective dimension and the fact that number of stages in our algorithm is $\log T$ to bound both the summations.

To bound the cumulative cost, first note that the cost of any signal of width w is upper bounded by $1/w$. Then, the bound follows from simple combinatorial arguments.

Remark 2 *Compared to the SpectralUCB algortihm, regret bound of our algorithm increases by an amount $cd\log^2(T/\lambda + 1)$, but still it is of the same order as $d\sqrt{T}$. However, the total cost in our algorithm is smaller than that in SpectralUCB by an amount of at least $T/4 + 1/2$, i.e., cost reduction is of the order T is achieved by our algorithm.*

4. EXPERIMENTS

We evaluate and compare our algorithm with the SpectralUCB which is the state-of-art and outperforms its competitors such as LinUCB on graphs with large number of nodes. We set $\delta = 0.001$, $R = 0.01$ and $\lambda = 0.01$.

We generated two graph models that are widely used to analyze connectivity in social networks. First, we generated *Erdős-Rényi*(ER) graph with each edge sampled with probability 0.02 independent of others. Second, we generated *Barabási-Albert*(BA) graph with degree parameter 3. On the edges of these graphs we assigned weights uniformly random.

We randomly generated a spare vector α^* with a small $k << n$ and use it to linearly combine the eigenvectors of the graph Laplacian to obtain the reward function $f = Q\alpha^*$, where 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 numerical plots displayed we used $n = 250$, $T = 150$ and $k = 5$. We repeated the experiments 100 times and took the average.

From figures 1 and 2 we see that cumulative regret performance of our algorithm is close to that of SpectralUCB, but the cost gain is significantly higher.

5. CONCLUSION

We studied the problem of identifying region of interest in a large area by formulating it as a bandit problem to learn the maximum value of a signal on a graph. The arms of the bandit are defined as graph signals which allows to observe rewards from nodes and their neighbors. We showed that by using the arms of different costs in a phased manner, where cheaper arms are used in the initial steps and costlier ones at later steps, the total cost of SpectralUCB can be significantly reduced without changing the order of scaling of the cumulative regret. We demonstrated that our algorithm provides cost saving of at least 30% on Erdős-Rényi and Barabási-Albert graph models without suffering much on cumulative regret..

6. REFERENCES

[1] Erhan Baki Ermis and Venkatesh Saligrama, "Distributed detection in sensor networks with limited range multimodal sensors," *IEEE Transactions on Signal Processing*, vol. 58, no. 2, pp. 843–858, 2010.

[2] Jason A. Fuemmeler and Venugopal V. Veeravalli, "Smart sleeping policies for energy efficient tracking in sensor networks.," *IEEE Transactions on Signal Processing*, vol. 56, no. 5, pp. 2091–2101, 2008.

[3] S. Aeron and V. Saligrama, "Wireless ad hoc networks: Strategies and scaling laws for the fixed SNR regime," *IEEE Transaction on Information Theory*, vol. 53, no. 6, pp. 2044–2059, June 2007.

[4] M. Belkin, I. Matveeva, and P. Niyogi, "Regularization and semi-supervised learning on large graphs," in *Proceeding of COLT*, 2004.

[5] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2008.

[6] V. Dani, T. P. Hayes, and S. M. Kakade, "Stochastic linear optimization under bandit feedback," in *Proceeding of COLT*, Helsinki, Finland, July 2008.

[7] P. Rusmevichientong and J. N. Tsitsiklis, "Linearly parameterized bandits," *INFORMS, Mathematics of Operations Research*, vol. 35, no. 2, pp. 395–411, May 2010.

[8] S. K. Narang, A. Gadde, and A. Ortega, "Signal processing techniques for interpolation in graph structured data," in *Proceedings of ICASSP*, May 2013.

[9] A. Anis anf A. Gadde and A. Ortega, "Towards a sampling theorem for signals on arbitrary graphs," in *Proceedings of ICASSP*, May 2014.

[10] X. Zhu and M. Rabbat, "Graph spectral compressed sensing for sensor networks," in *Proceedings of ICASSP*, May 2012.

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

[12] M. Valko, R. Munos, B. Kveton, and T. Kocak, "Spectral bandits for smooth graph functions," in *Proceeding of ICML*, Beijing, China, June 2014.

[13] N. Cesa-Bianchi, C. Gentile, and G. Zappella, "A gang of bandits," in *Proceedings of NIPS*, 2013.

[14] S. Caron, B. Kveton, M. Lelarge, and S. Bhagat, "Leveraging side observations in stochastic bandits," in *Proceedings of Uncertainty in Artificial Intelligence*, 2012.

[15] S. Agarwal and N.R. Devanur, "Bandits with concave rewards and convex knapsacks," in *Proceedings of EC*, 2014.

[16] A. Badanidiuuru, R. Kleinberg, and A. Slivkins, "Bandits with concave rewards and convex knapsacks," in *Proceedings of ACM Symp. on Economics and Computation*, 2014.

[17] L. Li, C. Wei, J. Langford, and R. E. Schapire, "A contextual-bandit approach to personalized news article recommendation," in *Proceeding of WWW*, NC, USA, April 2010.