Learning to Act Greedily: Polymatroid Semi-Bandits

Branislav Kveton BRANISLAV.KVETON@TECHNICOLOR.COM
Technicolor Labs
175 S San Antonio Road, Suite 200, Los Altos, CA 94022, USA

Zheng Wen ZHENGWEN@YAHOO-INC.COM
Yahoo Labs
701 1st Ave, Sunnyvale, CA 94089, United States

Azin Ashkan AZIN.ASHKAN@TECHNICOLOR.COM
Technicolor Labs
175 S San Antonio Road, Suite 200, Los Altos, CA 94022, USA

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

Editor:

Abstract

Many important optimization problems, such as the minimum spanning tree and minimum-cost flow, can be solved optimally by a greedy method. In this work, we study a learning variant of these problems, where the model of the problem is unknown and has to be learned by interacting repeatedly with the environment in the bandit setting. We formalize our learning problem quite generally, as learning how to maximize an unknown *modular function* on a known *polymatroid*. We propose a computationally efficient algorithm for solving our problem and bound its expected cumulative regret. Our gap-dependent upper bound is tight up to a constant and our gap-free upper bound is tight up to polylogarithmic factors. Finally, we evaluate our method on three problems and demonstrate that it is practical.

Keywords: bandits, combinatorial optimization, matroids, polymatroids, submodularity

1. Introduction

Many important combinatorial optimization problems, such as the minimum-cost flow (Megiddo, 1974) and minimum spanning tree (Papadimitriou and Steiglitz, 1998), can be solved optimally by a greedy algorithm. These problems can be solved efficiently because they can be viewed as optimization on *matroids* (Whitney, 1935) or *polymatroids* (Edmonds, 1970). More specifically, they can be formulated as finding the maximum of a modular function on the polytope of a submodular function. In this work, we study a learning variant of this problem where the modular function is unknown.

Our learning problem is sequential and divided into episodes. In each episode, the learning agent chooses a feasible solution to our problem, the basis of a polymatroid; observes noisy weights of all items with non-zero contributions in the basis; and receives the dot product between the basis and the weights as a payoff. The goal of the learning agent is to maximize its expected cumulative return over time, or equivalently to minimize its expected cumulative regret. Many practical problems can be formulated in our setting, such as learning a routing network (Oliveira and Pardalos, 2005) where the delays on the links of the network are stochastic and initially unknown. In this problem, the bases are spanning trees, the observed weights are the the delays on the links of the spanning tree, and the cost is the sum of the observed delays.

This paper makes three contributions. First, we bring together the concepts of bandits (Lai and Robbins, 1985; Auer et al., 2002a) and polymatroids (Edmonds, 1970), and propose *polymatroid bandits*, a new class of stochastic learning problems. A multi-armed bandit (Lai and Robbins, 1985) is a framework for solving online learning problems that require exploration. The framework has been successfully applied to a variety of problems, including those in combinatorial optimization (Gai et al., 2012; Cesa-Bianchi and Lugosi, 2012; Audibert et al., 2014). In this paper, we extend bandits to the combinatorial optimization problems that can be solved greedily.

Second, we propose a simple algorithm for solving our problem, which explores based on the optimism in the face of uncertainty. We refer to our algorithm as *Optimistic Polymatroid Maximization (*`OPM`*)*. `OPM` has two key properties. First, it is computationally efficient because the basis in each episode is chosen greedily. Second, `OPM` is also sample efficient. In particular, we derive a gap-dependent upper bound on the expected cumulative regret of `OPM` and show that it is tight up to a constant, and we also derive a gap-free upper bound and show that it is tight up to polylogarithmic factors. Our upper bounds exploit the structural properties of polymatroids and improve over general-purpose bounds for stochastic combinatorial semi-bandits.

Finally, we evaluate `OPM` on three problems. The first problem is a synthetic flow network and we use it to demonstrate that our gap-dependent upper bound is quite practical, an order of magnitude larger than the observed regret. The second problem is learning of a routing network for an Internet service provider (ISP). The last problem is learning how to recommend diverse movies. All three problems can be solved efficiently in our framework. This demonstrates that `OPM` is practical and can solve a wide range of problems.

We adopt the following notation. We write $A + e$ instead of $A \cup \{e\}$, and $A + B$ instead of $A \cup B$. We also write $A - e$ instead of $A \setminus \{e\}$, and $A - B$ instead of $A \setminus B$.

2. Polymatroids

In this section, we first introduce polymatroids and then illustrate them on practical problems. A *polymatroid* (Edmonds, 1970) is a polytope associated with a submodular function. More specifically, a polymatroid is a pair $M = (E, f)$, where $E = \{1, \dots, L\}$ is a *ground set* of L items and $f : 2^E \to \mathbb{R}^+$ is a function from the power set of E to non-negative real numbers. The function f is *monotonic*, $\forall X \subseteq Y \subseteq E : f(X) \leq f(Y)$; *submodular*, $\forall X, Y \subseteq E : f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y)$; and $f(\emptyset) = 0$. Since f is monotonic, $f(E)$ is one of its maxima. We refer to $f(E)$ as the *rank* of a polymatroid and denote it by K. Without loss of generality, we assume that $f(e) \leq 1$ for all items $e \in E$. Because f is submodular, we indirectly assume that $f(X + e) - f(X) \leq 1$ for all $X \subseteq E$.

The *independence polyhedron* P_M associated with polymatroid M is a subset of $\mathbb{R}^L$ defined as:

$$P_M = \left\{\mathbf{x} : \mathbf{x} \in \mathbb{R}^L,\ \mathbf{x} \geq 0,\ \forall X \subseteq E : \textstyle\sum_{e \in X} \mathbf{x}(e) \leq f(X)\right\}, \tag{1}$$

where $\mathbf{x}(e)$ is the e-th entry of vector $\mathbf{x}$. The vector $\mathbf{x}$ is *independent* if $\mathbf{x} \in P_M$. The *base polyhedron* B_M is a subset of P_M defined as:

$$B_M = \left\{\mathbf{x} : \mathbf{x} \in P_M,\ \textstyle\sum_{e \in E} \mathbf{x}(e) = K\right\}. \tag{2}$$

The vector $\mathbf{x}$ is a *basis* if $\mathbf{x} \in B_M$. In other words, $\mathbf{x}$ is independent and its entries sum up to K.

2.1 Optimization on Polymatroids

A *weighted polymatroid* is a polymatroid associated with a vector of weights $\mathbf{w} \in (\mathbb{R}^+)^L$. The e-th entry of $\mathbf{w}$, $\mathbf{w}(e)$, is the weight of item e. A classical problem in polyhedral optimization is to find a *maximum-weight basis* of a polymatroid:

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in B_M} \langle \mathbf{w}, \mathbf{x} \rangle = \arg\max_{\mathbf{x} \in P_M} \langle \mathbf{w}, \mathbf{x} \rangle. \tag{3}$$

This basis can be computed greedily (Algorithm 1). The greedy algorithm works as follows. First, the items E are sorted in decreasing order of their weights, $\mathbf{w}(e_1) \geq \ldots \geq \mathbf{w}(e_L)$. We assume that the ties are broken by an arbitrary but fixed rule. Second, $\mathbf{x}^*$ is computed as $\mathbf{x}^*(e_i) = f(\{e_1, \dots, e_i\}) - f(\{e_1, \dots, e_{i-1}\})$ for

Algorithm 1 `Greedy`: Edmond's algorithm for computing the maximum-weight basis of a polymatroid.

Input: Polymatroid $M = (E, f)$, weights $\mathbf{w}$

Let $e_1, \dots, e_L$ be an ordering of items such that:
  $\mathbf{w}(e_1) \geq \dots \geq \mathbf{w}(e_L)$
$\mathbf{x} \leftarrow$ All-zeros vector of length L
for all $i = 1, \dots, L$ **do**
  $\mathbf{x}(e_i) \leftarrow f(\{e_1, \dots, e_i\}) - f(\{e_1, \dots, e_{i-1}\})$
end for

Output: Maximum-weight basis $\mathbf{x}$

all i. Note that the *minimum-weight basis* of a polymatroid with weights $\mathbf{w}$ is the maximum-weight basis of the same polymatroid with weights $\max_{e \in E} \mathbf{w}(e) - \mathbf{w}$:

$$\arg\min_{\mathbf{x} \in B_M} \langle \mathbf{w}, \mathbf{x} \rangle = \arg\max_{\mathbf{x} \in B_M} \langle \max_{e \in E} \mathbf{w}(e) - \mathbf{w}, \mathbf{x} \rangle. \tag{4}$$

So the minimization problem is mathematically equivalent to the maximization problem (3), and all results in this paper straightforwardly generalize to the minimization.

Many existing problems can be viewed as optimization on a polymatroid (3). For instance, polymatroids generalize *matroids* (Whitney, 1935), a notion of independence in combinatorial optimization that is closely related to computational efficiency. In particular, let $M = (E, \mathcal{I})$ be a matroid, where $E = \{1, \dots, L\}$ is its ground set, $\mathcal{I} \subseteq 2^E$ are its independent sets, and:

$$f(X) = \max_{Y : Y \subseteq X, Y \in \mathcal{I}} |Y| \tag{5}$$

is its *rank function*. Let $\mathbf{w} \in (\mathbb{R}^+)^L$ be a vector of non-negative weights. Then the maximum-weight basis of a matroid:

$$A^* = \arg\max_{A \in \mathcal{I}} \sum_{e \in A} \mathbf{w}(e) \tag{6}$$

can be also derived as $A^* = \{e : \mathbf{x}^*(e) = 1\}$, where $\mathbf{x}^*$ is the maximum-weight basis of the corresponding polymatroid. The basis is $\mathbf{x}^* \in \{0, 1\}^L$ because the rank function is a monotonic submodular function with zero-one increments (Fujishige, 2005).

Our optimization problem can be written as a *linear program (LP)* (Bertsimas and Tsitsiklis, 1997):

$$\max_{\mathbf{x}} \sum_{e \in E} \mathbf{w}(e)\mathbf{x}(e) \qquad \text{subject to:} \sum_{e \in X} \mathbf{x}(e) \leq f(X) \quad \forall X \subseteq E, \tag{7}$$

where $\mathbf{x} \in (\mathbb{R}^+)^L$ is a vector of L optimized variables. This LP has exponentially many constraints, one for each subset $X \subseteq E$. Therefore, it cannot be solved directly. Nevertheless, `Greedy` can solve the problem in $O(L \log L)$ time. Therefore, our problem is a very efficient form of linear programming.

Many combinatorial optimization concepts, such as *flows* and *entropy* (Fujishige, 2005), are submodular. Therefore, optimization on these concepts involves polymatroids. A well-known problem in this class is the *minimum-cost flow* (Megiddo, 1974). This problem can be formulated as follows. The ground set E are the source nodes of a flow network, $f(X)$ is the maximum flow through source nodes $X \subseteq E$, and $\mathbf{w}(e)$ is the cost of a unit flow through source node e. The minimum-weight basis of this polymatroid is the maximum flow with the minimum cost (Fujishige, 2005), which we refer to as the minimum-cost flow.

The problem of recommending diverse items can be also cast as optimization on a polymatroid (Ashkan et al., 2014a,b). Let E be a set of recommendable items, $f(X)$ be the number of topics covered by items X, and $\mathbf{w}$ be a weight vector such that $\mathbf{w}(e)$ is the popularity of item e. Then $\mathbf{x}^* = \texttt{Greedy}(M, \mathbf{w})$ is a vector such that $\mathbf{x}^*(e) > 0$ if and only if item e is the most popular item in at least one topic covered by item e. We illustrate this concept on a simple example. Let the ground set E be a set of 3 movies:

e	Movie title	Popularity $\mathbf{w}(e)$	Movie genres
1	Inception	0.8	Action
2	Grown Ups 2	0.5	Comedy
3	Kindergarten Cop	0.6	Action Comedy

Let $f(X)$ be the number of movie genres covered by movies X. Then f is submodular and defined as:

$$\begin{aligned} f(\emptyset) &= 0, & f(\{2\}) &= 1, & f(\{1,2\}) &= 2, & f(\{2,3\}) &= 2, \\ f(\{1\}) &= 1, & f(\{3\}) &= 2, & f(\{1,3\}) &= 2, & f(\{1,2,3\}) &= 2. \end{aligned} \tag{8}$$

The maximum-weight basis of polymatroid $M = (E, f)$ is $\mathbf{x}^* = (1, 0, 1)$, and $\{e : \mathbf{x}^*(e) > 0\} = \{1, 3\}$ is the minimal set of movies that cover each movie genre by the most popular movie in that genre.

2.2 Combinatorial Optimization on Polymatroids

In this paper, we restrict our attention to the feasible solutions:

$$\Theta = \left\{\mathbf{x} : \left(\exists \mathbf{w} \in (\mathbb{R}^+)^L : \mathbf{x} = \texttt{Greedy}(M, \mathbf{w})\right)\right\} \tag{9}$$

that can be computed greedily for some weight vector $\mathbf{w}$ and define our objective as finding:

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in \Theta} \langle \mathbf{w}, \mathbf{x} \rangle. \tag{10}$$

The set Θ are the vertices of B_M and we prove this formally in Lemma 8 in Appendix.

Our choice is motivated by three reasons. First, we study the problem of learning to act greedily. So we are only interested in the bases that can be computed greedily. Second, many optimization problems of our interest (Section 2.1) are combinatorial in nature and only the bases in Θ are suitable feasible solutions. For instance, in a graphic matroid, Θ is a set of spanning trees. In a linear matroid, Θ is a set of maximal sets of linearly independent vectors. The bases in $B_M - \Theta$ do not have this interpretation. Another example is our recommendations problem in Section 2.1. In this problem, for any $\mathbf{x} = \texttt{Greedy}(M, \mathbf{w})$, $\{e : \mathbf{x}(e) > 0\}$ is a minimal set of items that cover each topic by the most popular item according to $\mathbf{w}$. The bases in $B_M - \Theta$ cannot be interpreted in this way. Finally, we note that our choice does not have any impact on the notion of optimality. In particular, let $\mathbf{x}$ be optimal for some $\mathbf{w}$. Then $\mathbf{x}^g = \texttt{Greedy}(M, \mathbf{w})$ is also optimal and since $\mathbf{x}^g \in \Theta$, it follows that:

$$\max_{\mathbf{x} \in B_M} \langle \mathbf{w}, \mathbf{x} \rangle = \max_{\mathbf{x} \in \Theta} \langle \mathbf{w}, \mathbf{x} \rangle. \tag{11}$$

3. Polymatroid Bandits

The maximum-weight basis of a polymatroid cannot be computed when the weights $\mathbf{w}$ are unknown. This may happen in practice. For instance, suppose that we want to recommend a diverse set of popular movies (Section 2.1) but the popularity of these movies is initially unknown, perhaps because the movies are newly released. In this work, we study a learning variant of maximizing a modular function on a polymatroid that can solve this type of problems.

3.1 Model

We formalize our learning problem as a polymatroid bandit. A *polymatroid bandit* is a pair (M, P), where M is a polymatroid and P is a probability distribution over the weights $\mathbf{w} \in \mathbb{R}^L$ of items E in M. The e-th entry of $\mathbf{w}$, $\mathbf{w}(e)$, is the weight of item e. We assume that the weights $\mathbf{w}$ are drawn i.i.d. from P and that P is unknown. Without loss of generality, we assume that P is a distribution over the unit cube $[0,1]^L$. Other than that, we do not assume anything about P. We denote the expected weights of the items by $\bar{\mathbf{w}} = \mathbb{E}[\mathbf{w}]$. By our assumptions on P, $\bar{\mathbf{w}}(e) \geq 0$ for all items e.

Each item e is associated with an *arm* and each feasible solution $\mathbf{x} \in \Theta$ is associated with a set of arms $A = \{e : \mathbf{x}(e) > 0\}$. The arms A are the items with non-zero contributions in $\mathbf{x}$. After the arms are *pulled*, the learning agent receives a *payoff* of $\langle \mathbf{w}, \mathbf{x} \rangle$ and *observes* $\{(e, \mathbf{w}(e)) : \mathbf{x}(e) > 0\}$, the weights of all items with non-zero contributions in $\mathbf{x}$. This feedback model is known as *semi-bandit* (Audibert et al., 2014). The solution to our problem is a maximum-weight basis in expectation:

$$\mathbf{x}^* = \arg\max_{\mathbf{x} \in \Theta} \mathbb{E}_{\mathbf{w}}[\langle \mathbf{w}, \mathbf{x} \rangle] = \arg\max_{\mathbf{x} \in \Theta} \langle \bar{\mathbf{w}}, \mathbf{x} \rangle. \tag{12}$$

This problem is equivalent to problem (10) and therefore can be solved greedily, $\mathbf{x}^* = \texttt{Greedy}(M, \bar{\mathbf{w}})$.

We choose our observation model for several reasons. First, the model is a natural generalization of that in matroid bandits (Kveton et al., 2014a). In matroid bandits, the bases are of the form $\mathbf{x} \in \{0,1\}^L$ and the learning agents observes the weights of all chosen items e, $\mathbf{x}(e) = 1$. In this case, $\mathbf{x}(e) = 1$ is equivalent to $\mathbf{x}(e) > 0$. Second, our observation model is suitable for our motivating examples (Section 2.1). Specifically, in the minimum-cost flow problem, we assume that the learning agent observes the costs of all source nodes that contribute to the maximum flow. In the movie recommendation problem, the agent observes individual movies chosen by the user, from a set of recommended movies. Finally, our observation model allows us to derive similar regret bounds to those in matroid bandits (Kveton et al., 2014a).

Our learning problem is *episodic*. Let $(\mathbf{w}_t)_{t=1}^n$ be an i.i.d. sequence of weights drawn from distribution P. In episode t, the learning agent chooses basis $\mathbf{x}_t$ based on its prior actions $\mathbf{x}_1, \ldots, \mathbf{x}_{t-1}$ and observations of $\mathbf{w}_1, \ldots, \mathbf{w}_{t-1}$; gains $\langle \mathbf{w}_t, \mathbf{x}_t \rangle$; and observes $\{(e, \mathbf{w}_t(e)) : \mathbf{x}_t(e) > 0\}$, the weights of all items with non-zero contributions in $\mathbf{x}_t$. The agent interacts with the environment in n episodes. The goal of the agent is to maximize its expected cumulative return, or equivalently to minimize its *expected cumulative regret*:

$$R(n) = \mathbb{E}_{\mathbf{w}_1, \ldots, \mathbf{w}_n}\left[\sum_{t=1}^n R(\mathbf{x}_t, \mathbf{w}_t)\right], \tag{13}$$

where $R(\mathbf{x}, \mathbf{w}) = \langle \mathbf{w}, \mathbf{x}^* \rangle - \langle \mathbf{w}, \mathbf{x} \rangle$ is the regret associated with basis $\mathbf{x}$ and weights $\mathbf{w}$.

3.2 Algorithm

Our learning algorithm is designed based on the *optimism in the face of uncertainty* principle (Auer et al., 2002a). In particular, it is a greedy method for finding a maximum-weight basis of a polymatroid where the expected weight $\bar{\mathbf{w}}(e)$ of each item is substituted with its optimistic estimate $U_t(e)$. We refer to our method as *Optimistic Polymatroid Maximization (*`OPM`*)*.

The pseudocode of `OPM` is given in Algorithm 2. In each episode t, the algorithm works as follows. First, we compute an *upper confidence bound* (UCB) on the expected weight of each item e:

$$U_t(e) = \hat{\mathbf{w}}_{T_{t-1}(e)}(e) + c_{t-1, T_{t-1}(e)}, \tag{14}$$

where $\hat{\mathbf{w}}_{T_{t-1}(e)}(e)$ is our estimate of the expected weight $\bar{\mathbf{w}}(e)$ in episode t, $c_{t-1, T_{t-1}(e)}$ is the radius of the confidence interval around this estimate, and $T_{t-1}(e)$ denotes the number of times that item e is selected in the first $t-1$ episodes, $\mathbf{x}_i(e) > 0$ for $i < t$. Second, we compute the maximum-weight basis with respect to U_t using `Greedy`. Finally, we select the basis, observe the weights of all items e where $\mathbf{x}_t(e) > 0$, and then update our model $\hat{\mathbf{w}}$ of the environment. The radius:

$$c_{t,s} = \sqrt{\frac{2 \log t}{s}} \tag{15}$$

Algorithm 2 `OPM`: Optimistic polymatroid maximization.

Input: Polymatroid $M = (E, f)$

Observe $\mathbf{w}_0 \sim P$ ▷ Initialization
$\hat{\mathbf{w}}_1(e) \leftarrow \mathbf{w}_0(e) \quad \forall e \in E$
$T_0(e) \leftarrow 1 \quad \forall e \in E$

for all $t = 1, \dots, n$ **do**
  $U_t(e) \leftarrow \hat{\mathbf{w}}_{T_{t-1}(e)}(e) + c_{t-1, T_{t-1}(e)} \quad \forall e \in E$ ▷ Compute UCBs
  $\mathbf{x}_t \leftarrow \mathtt{Greedy}(M, U_t)$ ▷ Find a maximum-weight basis
  Observe $\{(e, \mathbf{w}_t(e)) : \mathbf{x}_t(e) > 0\}$, where $\mathbf{w}_t \sim P$ ▷ Choose the basis

  $T_t(e) \leftarrow T_{t-1}(e) \quad \forall e \in E$ ▷ Update statistics
  $T_t(e) \leftarrow T_t(e) + 1 \quad \forall e : \mathbf{x}_t(e) > 0$
  $\hat{\mathbf{w}}_{T_t(e)}(e) \leftarrow \dfrac{T_{t-1}(e)\hat{\mathbf{w}}_{T_{t-1}(e)}(e) + \mathbf{w}_t(e)}{T_t(e)} \quad \forall e : \mathbf{x}_t(e) > 0$
end for

is designed such that each UCB is a high-probability upper bound on the corresponding weight $\hat{\mathbf{w}}_s(e)$. The UCBs encourage exploration of items that have not been observed sufficiently often. As the number of past episodes increases, we get better estimates of the weights $\bar{\mathbf{w}}$, all confidence intervals shrink, and `OPM` starts exploiting most rewarding items. The $\log(t)$ term increases with time and enforces continuous exploration.

For simplicity of exposition, we assume that `OPM` is initialized by observing each item once. In practice, this initialization step can be implemented efficiently in the first L episodes. In particular, in episode $t \leq L$, `OPM` chooses first item t and then all other items, in an arbitrary order. The corresponding regret is bounded by KL because $\langle \bar{\mathbf{w}}, \mathbf{x} \rangle \in [0, K]$ for any $\bar{\mathbf{w}}$ (Section 3.1) and basis $\mathbf{x}$ (Section 2).

`OPM` is a greedy method and therefore is extremely computationally efficient. In particular, suppose that the function f is an oracle that can be queried in $O(1)$ time. Then the time complexity of `OPM` in episode t is $O(L \log L)$, comparable to that of sorting L numbers. The design of `OPM` is not very surprising and it draws on prior work (Kveton et al., 2014a; Gai et al., 2012).

Our major contribution is that we derive a tight upper bound on the regret of `OPM`. Our analysis is novel and is a significant improvement over Kveton et al. (2014a), who analyze the regret of `OPM` in the context of matroids. Roughly speaking, the analysis of Kveton et al. (2014a) leverages the augmentation property of a matroid. Our analysis is based on the submodularity of a polymatroid.

4. Analysis

This section is organized as follows. First, we propose a novel decomposition of the regret of `OPM` in a single episode (Section 4.1). Loosely speaking, we decompose the regret as a sum of its parts, the fractional gains of individual items in the optimal and suboptimal bases. This part of the proof relies heavily on the structure of a polymatroid and is a major contribution. Second, we apply the regret decomposition to bound the regret of `OPM` (Section 4.2). Third, we compare our regret bounds to existing upper bounds (Section 4.3) and prove matching lower bounds (Section 4.4). Finally, we summarize our results (Section 4.5).

4.1 Regret Decomposition

The key step in our analysis is that we bound the expected regret in episode t for any basis $\mathbf{x}_t$, $R(\mathbf{x}_t, \bar{\mathbf{w}})$. In rest of this section, we fix the basis $\mathbf{x}_t$ and drop indexing by time t to simplify our notation.

Without loss of generality, we assume that the items in the ground set E are ordered such that $\bar{\mathbf{w}}(1) \geq \ldots \geq \bar{\mathbf{w}}(L)$. So the optimal basis $\mathbf{x}^*$ is defined as:

$$\mathbf{x}^*(i) = f(A_i^*) - f(A_{i-1}^*) \qquad i = 1, \ldots, L; \tag{16}$$

where $A_i^* = \{1, \ldots, i\}$ are the first i items in E. Let U_t be the vector of UCBs in episode t and $a_1, \ldots, a_L$ be the ordering of items such that $U_t(a_1) \geq \ldots \geq U_t(a_L)$. Then the basis $\mathbf{x}$ in episode t is defined as:

$$\mathbf{x}(a_k) = f(A_k) - f(A_{k-1}) \qquad k = 1, \ldots, L; \tag{17}$$

where $A_k = \{a_1, \ldots, a_k\}$. The hardness of discriminating items e and e^* is measured by a *gap* between the expected weights of the items:

$$\Delta_{e,e^*} = \bar{\mathbf{w}}(e^*) - \bar{\mathbf{w}}(e). \tag{18}$$

For each item e, we define $\rho(e)$, the largest index such that $\bar{\mathbf{w}}(\rho(e)) > \bar{\mathbf{w}}(e)$ and $\mathbf{x}^*(\rho(e)) > 0$, the expected weight of item $\rho(e)$ is larger than that of item e and the item contributes to $\mathbf{x}^*$. For simplicity of exposition, we assume that item 1 contributes to the optimal basis $\mathbf{x}^*$, $\mathbf{x}^*(1) > 0$. This guarantees that $\rho(e)$ is properly defined for all items but item 1. We assume that $\rho(1) = 0$.

Our regret decomposition is based on rewriting the difference in the expected returns of bases $\mathbf{x}^*$ and $\mathbf{x}$ as the sum of the differences in the returns of intermediate solutions, which are obtained by interleaving the bases. We refer to these solutions as *augmentations*. A *k-augmentation* is a vector $\mathbf{y}_k \in [0, 1]^L$ such that:

$$\mathbf{y}_k(i) = \left\{ \begin{array}{ll} f(A_j) - f(A_{j-1}) & i \in A_k \text{ and } a_j = i \\ f(A_k + A_i^*) - f(A_k + A_{i-1}^*) & i \notin A_k \end{array} \right| \qquad i = 1, \ldots, L. \tag{19}$$

It can be also viewed as a basis generated by `Greedy`, which first selects k suboptimal items $a_1, \ldots, a_k$ and then the remaining $L - K$ items, ordered from 1 to L. Now we prove our first lemma.

Lemma 1 *For any k, the difference of two consecutive augmentations $\mathbf{y}_{k-1}$ and $\mathbf{y}_k$ satisfies:*

$$\mathbf{y}_{k-1}(i) - \mathbf{y}_k(i) \; \left\{ \begin{array}{ll} = 0 & i \in A_{k-1} \\ \leq 0 & i = a_k \\ \geq 0 & i \notin A_k \end{array} \right| \qquad i = 1, \ldots, L.$$

Proof First, let $i = a_j \in A_{k-1}$. Then by definition (19):

$$\mathbf{y}_{k-1}(i) - \mathbf{y}_k(i) = f(A_j) - f(A_{j-1}) - (f(A_j) - f(A_{j-1})) = 0. \tag{20}$$

Second, let $i = a_k$. Then:

$$\begin{aligned} \mathbf{y}_{k-1}(i) - \mathbf{y}_k(i) &= f(A_{k-1} + A_i^*) - f(A_{k-1} + A_{i-1}^*) - (f(A_k) - f(A_{k-1})) \\ &= f(A_k + A_{i-1}^*) - f(A_{k-1} + A_{i-1}^*) - (f(A_k) - f(A_{k-1})) \\ &\leq 0. \end{aligned} \tag{21}$$

The first equality is due to definition (19). The second equality follows from the assumption that $i = a_k$. The inequality is due to the submodularity of f. Finally, let $i \notin A_k$. Then:

$$\begin{aligned} \mathbf{y}_{k-1}(i) - \mathbf{y}_k(i) &= f(A_{k-1} + A_i^*) - f(A_{k-1} + A_{i-1}^*) - (f(A_k + A_i^*) - f(A_k + A_{i-1}^*)) \\ &\geq 0. \end{aligned} \tag{22}$$

The equality is due to definition (19). The inequality is due to the submodularity of f. ■

Lemma 1 says that $\mathbf{y}_{k-1}(a_k) - \mathbf{y}_k(a_k)$ is the only non-positive entry in $\mathbf{y}_{k-1} - \mathbf{y}_k$. Since $\mathbf{y}_{k-1}$ and $\mathbf{y}_k$ are bases, $\sum_{e=1}^{L} \mathbf{y}_{k-1}(i) = \sum_{e=1}^{L} \mathbf{y}_k(i) = K$, it follows that $\mathbf{y}_{k-1}(a_k) - \mathbf{y}_k(a_k) = -\sum_{i \notin A_k} [\mathbf{y}_{k-1}(i) - \mathbf{y}_k(i)]$. The quantity $\mathbf{y}_{k-1}(i) - \mathbf{y}_k(i)$ can be viewed as a fraction of item i in $\mathbf{y}_{k-1}$ exchanged for item a_k in $\mathbf{y}_k$. In the rest of our analysis, we represented these fractions as a vector:

$$\delta(a_k, i) = \max\left\{\mathbf{y}_{k-1}(i) - \mathbf{y}_k(i), 0\right\}. \tag{23}$$

Lemma 2 *For any k, the difference in the expected returns of augmentations $\mathbf{y}_{k-1}$ and $\mathbf{y}_k$ is bounded as:*

$$\langle \bar{\mathbf{w}}, \mathbf{y}_{k-1} - \mathbf{y}_k \rangle \leq \sum_{e^*=1}^{\rho(a_k)} \Delta_{a_k, e^*} \delta(a_k, e^*).$$

Proof The claim is proved as:

$$\begin{aligned}
\langle \bar{\mathbf{w}}, \mathbf{y}_{k-1} - \mathbf{y}_k \rangle &= \sum_{e^* \notin A_k} \bar{\mathbf{w}}(e^*) \delta(a_k, e^*) - \bar{\mathbf{w}}(a_k) \sum_{e^* \notin A_k} \delta(a_k, e^*) \\
&= \sum_{e^* \notin A_k} \underbrace{(\bar{\mathbf{w}}(e^*) - \bar{\mathbf{w}}(a_k))}_{\Delta_{a_k, e^*}} \delta(a_k, e^*) \\
&\leq \sum_{e^*=1}^{a_k - 1} \mathbb{1}\{e^* \notin A_k\} \, \Delta_{a_k, e^*} \delta(a_k, e^*) \\
&= \sum_{e^*=1}^{\rho(a_k)} \Delta_{a_k, e^*} \delta(a_k, e^*).
\end{aligned} \tag{24}$$

The first two steps follow from Lemma 1 and the subsequent discussion. Then we neglect the negative gaps. Finally, because f is monotonic and submodular, $\delta(a_k, e^*) = 0$ for any $e^* \notin A_k$ such that $\mathbf{x}^*(e^*) = 0$. As a result, we can restrict the scope of the summation over e^* to between 1 and $\rho(a_k)$. ■

Now we are ready to prove our main lemma.

Theorem 3 *The expected regret of choosing any basis $\mathbf{x}$ in episode t is bounded as:*

$$R(\mathbf{x}, \bar{\mathbf{w}}) \leq \sum_{e=1}^{L} \sum_{e^*=1}^{\rho(e)} \Delta_{e, e^*} \delta(e, e^*),$$

where $\delta(e, e^*)$ is the fraction of item e^* exchanged for item e in episode t, and is defined in (23)*. Moreover, when $\delta(e, e^*) > 0$,* `OPM` *observes the weight of item e and $U_t(e) \geq U_t(e^*)$. Finally:*

$$\forall t : \sum_{e=1}^{L} \sum_{e^*=1}^{\rho(e)} \delta(e, e^*) \leq K, \qquad \forall t, e \in E : \sum_{e^*=1}^{\rho(e)} \delta(e, e^*) \leq 1.$$

Proof The first claim is proved as follows:

$$R(\mathbf{x}, \bar{\mathbf{w}}) = \langle \bar{\mathbf{w}}, \mathbf{x}^* - \mathbf{x} \rangle = \sum_{k=1}^{L} \langle \bar{\mathbf{w}}, \mathbf{y}_{k-1} - \mathbf{y}_k \rangle \leq \sum_{k=1}^{L} \sum_{e^*=1}^{\rho(a_k)} \Delta_{a_k, e^*} \delta(a_k, e^*). \tag{25}$$

First, we rewrite the regret $\langle \bar{\mathbf{w}}, \mathbf{x}^* - \mathbf{x} \rangle$ as the sum of the differences in $(L+1)$ k-augmentations, from $\mathbf{y}_0$ to $\mathbf{y}_L$. Note that $\mathbf{y}_0 = \mathbf{x}^*$ and $\mathbf{y}_L = \mathbf{x}$. Second, we bound each term in the sum using Lemma 2. Finally, we replace the sum over all indices k by the sum over all items e.

The second claim is proved as follows. Let $\delta(e, e^*) > 0$. Then OPM is guaranteed to observe the weight of item e because $\mathbf{x}(e) \geq \delta(e, e^*) > 0$. Furthermore, let $U_t(e) < U_t(e^*)$. Then OPM chooses item e^* before item e, and $\delta(e, e^*) = 0$ by Lemma 1. This is a contradiction because $\delta(e, e^*) > 0$ by our assumption. As a result, it must be true that $\delta(e, e^*) > 0$ implies $U_t(e) \geq U_t(e^*)$.

The last two inequalities follow from two observations. First, $\sum_{e^*=1}^{\rho(e)} \delta(e, e^*) \leq \mathbf{x}(e)$ for any basis $\mathbf{x}$ and item e, the sum of the contributions from items e^* to e cannot be larger than the total contribution of item e in $\mathbf{x}$. Second, by the definitions in Section 2, $\mathbf{x}(e) \leq 1$ and $\sum_{e \in E} \mathbf{x}(e) = K$ for any basis $\mathbf{x}$ and item e. ■

Note that $\delta(e, e^*)$ is a random variable that depends on the basis $\mathbf{x}_t$ in episode t. To stress this dependence, we denote it by $\delta_t(e, e^*)$ in the rest of our analysis.

4.2 Upper Bounds

Our first result is a gap-dependent bound. We prove a gap-free bound in sequel.

Theorem 4 (gap-dependent bound) *The expected cumulative regret of* OPM *is bounded as:*

$$R(n) \leq \sum_{e=1}^{L} \frac{16}{\Delta_{e,\rho(e)}} \log n + \sum_{e=1}^{L} \sum_{e^*=1}^{\rho(e)} \Delta_{e,e^*} \frac{4}{3}\pi^2.$$

Proof First, we bound the expected regret in episode t using Theorem 3:

$$\begin{aligned} R(n) &= \sum_{t=1}^{n} \mathbb{E}_{\mathbf{w}_1,\dots,\mathbf{w}_{t-1}}[\mathbb{E}_{\mathbf{w}_t}[R(\mathbf{x}_t, \mathbf{w}_t)]] \\ &\leq \sum_{t=1}^{n} \mathbb{E}_{\mathbf{w}_1,\dots,\mathbf{w}_{t-1}}\left[\sum_{e=1}^{L} \sum_{e^*=1}^{\rho(e)} \Delta_{e,e^*} \delta_t(e, e^*)\right] \\ &= \sum_{e=1}^{L} \sum_{e^*=1}^{\rho(e)} \Delta_{e,e^*} \mathbb{E}_{\mathbf{w}_1,\dots,\mathbf{w}_n}\left[\sum_{t=1}^{n} \delta_t(e, e^*)\right]. \end{aligned} \tag{26}$$

Second, we bound the regret associated with each item e. The key idea is to decompose $\delta_t(e, e^*)$ as:

$$\delta_t(e, e^*) = \delta_t(e, e^*)\mathbb{1}\{T_{t-1}(e) \leq \ell_{e,e^*}\} + \delta_t(e, e^*)\mathbb{1}\{T_{t-1}(e) > \ell_{e,e^*}\} \tag{27}$$

and then select ℓ_{e,e^*} appropriately. By Lemma 9 in Appendix, the regret corresponding to $\mathbb{1}\{T_{t-1}(e) > \ell_{e,e^*}\}$ is bounded as:

$$\sum_{e^*=1}^{\rho(e)} \Delta_{e,e^*} \mathbb{E}_{\mathbf{w}_1,\dots,\mathbf{w}_n}\left[\sum_{t=1}^{n} \delta_t(e, e^*)\mathbb{1}\{T_{t-1}(e) > \ell_{e,e^*}\}\right] \leq \sum_{e^*=1}^{\rho(e)} \Delta_{e,e^*} \frac{4}{3}\pi^2 \tag{28}$$

when $\ell_{e,e^*} = \left\lfloor \frac{8}{\Delta_{e,e^*}^2} \log n \right\rfloor$. At the same time, the regret corresponding to $\mathbb{1}\{T_{t-1}(e) \leq \ell_{e,e^*}\}$ is bounded as:

$$\begin{aligned} &\sum_{e^*=1}^{\rho(e)} \Delta_{e,e^*} \mathbb{E}_{\mathbf{w}_1,\dots,\mathbf{w}_n}\left[\sum_{t=1}^{n} \delta_t(e, e^*)\mathbb{1}\{T_{t-1}(e) \leq \ell_{e,e^*}\}\right] \leq \\ &\qquad \max_{\mathbf{w}_1,\dots,\mathbf{w}_n} \left[\sum_{t=1}^{n} \sum_{e^*=1}^{\rho(e)} \Delta_{e,e^*} \delta_t(e, e^*)\mathbb{1}\left\{T_{t-1}(e) \leq \frac{8}{\Delta_{e,e^*}^2} \log n\right\}\right]. \end{aligned} \tag{29}$$

The next step is due to three observations. First, the gaps Δ_{e,e^*} are ordered such that $\Delta_{e,1} \geq \ldots \geq \Delta_{e,\rho(e)}$. Second, by Theorem 3, $T_{t-1}(e)$ increases by one when $\delta_t(e, e^*) > 0$, because this event implies that item e is observed. Finally, by Theorem 3, $\sum_{e^*=1}^{\rho(e)} \delta_t(e, e^*) \leq 1$ for all e and t. Based on these facts, two of which follow from the structure of a polymatroid, the bound in (29) can be bounded from above by:

$$\left[\Delta_{e,1}\frac{1}{\Delta_{e,1}^2} + \sum_{e^*=2}^{\rho(e)} \Delta_{e,e^*}\left(\frac{1}{\Delta_{e,e^*}^2} - \frac{1}{\Delta_{e,e^*-1}^2}\right)\right] 8\log n. \tag{30}$$

By Lemma 10 in Appendix, the above quantity is further bounded by $\frac{16}{\Delta_{e,\rho(e)}}\log n$. Finally, we combine all of our inequalities and get:

$$\sum_{e^*=1}^{\rho(e)} \Delta_{e,e^*}\mathbb{E}_{\mathbf{w}_1,\ldots,\mathbf{w}_n}\left[\sum_{t=1}^{n} \delta_t(e, e^*)\right] \leq \frac{16}{\Delta_{e,\rho(e)}}\log n + \sum_{e^*=1}^{\rho(e)} \Delta_{e,e^*}\frac{4}{3}\pi^2. \tag{31}$$

Our main claim is obtained by summing over all items e. ■

Theorem 5 (gap-free bound) *The expected cumulative regret of* `OPM` *is bounded as:*

$$R(n) \leq 8\sqrt{KLn\log n} + \frac{4}{3}\pi^2 L^2.$$

Proof The main idea is to decompose the expected cumulative regret of `OPM` into two parts, where the gaps are larger than ε and at most ε. We analyze each part separately and then select ε to get the desired result.

Let $\rho_\varepsilon(e)$ be the number of items whose expected weight is higher than that of item e by more than ε and:

$$Z_{e,e^*}(n) = \mathbb{E}_{\mathbf{w}_1,\ldots,\mathbf{w}_n}\left[\sum_{t=1}^{n} \delta_t(e, e^*)\right]. \tag{32}$$

Then for any ε, the regret of `OPM` can be decomposed as:

$$R(n) = \sum_{e=1}^{L}\sum_{e^*=1}^{\rho_\varepsilon(e)} \Delta_{e,e^*}Z_{e,e^*}(n) + \sum_{e=1}^{L}\sum_{e^*=\rho_\varepsilon(e)+1}^{\rho(e)} \Delta_{e,e^*}Z_{e,e^*}(n). \tag{33}$$

The first term can be bounded similarly to (31):

$$\begin{aligned}\sum_{e=1}^{L}\sum_{e^*=1}^{\rho_\varepsilon(e)} \Delta_{e,e^*}Z_{e,e^*}(n) &\leq \sum_{e=1}^{L}\frac{16}{\Delta_{e,\rho_\varepsilon(e)}}\log n + \sum_{e=1}^{L}\sum_{e^*=\rho_\varepsilon(e)+1}^{\rho(e)} \Delta_{e,e^*}\frac{4}{3}\pi^2 \\ &\leq \frac{16}{\varepsilon}L\log n + \frac{4}{3}\pi^2 L^2. \end{aligned} \tag{34}$$

The second term is bounded trivially as:

$$\sum_{e=1}^{L}\sum_{e^*=\rho_\varepsilon(e)+1}^{\rho(e)} \Delta_{e,e^*}Z_{e,e^*}(n) \leq \varepsilon Kn \tag{35}$$

because $\sum_{e=1}^{L}\sum_{e^*=1}^{\rho(e)} \delta_t(e, e^*) \leq K$ in any episode t (Theorem 3) and $\Delta_{e,e^*} \leq \varepsilon$. Finally, we get:

$$R(n) \leq \frac{16}{\varepsilon}L\log n + \varepsilon Kn + \frac{4}{3}\pi^2 L^2 \tag{36}$$

and choose $\varepsilon = 4\sqrt{\frac{L\log n}{Kn}}$. This concludes our proof. ■

4.3 Improvement over General Upper Bounds

`OPM` is an instance of the UCB algorithm by Gai et al. (2012) for combinatorial semi-bandits (Section 6). So it is natural to ask if our upper bounds on the regret of `OPM` (Section 4.2) are tighter than those in stochastic combinatorial semi-bandits (Chen et al., 2013; Kveton et al., 2014b). In this section, we show that this is the case, by comparing our upper bounds to Kveton et al. (2014b).

Our $O(\sqrt{KLn \log n})$ gap-free upper bound (Theorem 5) has the same dependence on K, L, and n as the upper bound of Kveton et al. (2014b) (Theorem 6). The only notable improvement in our analysis is that we reduce the constant at the $\sqrt{n \log n}$ term from 47 to 8.

Our $O(L(1/\Delta) \log n)$ upper bound (Theorem 4) is tighter by a factor of K than the $O(KL(1/\Delta) \log n)$ upper bound of Kveton et al. (2014b) (Theorem 5). However, our notion of the gap, item-based (18), differs from that of Kveton et al. (2014b), solution-based. So hypothetically, the improvement in our bound may be solely due to a different notion of the gap. In the rest of this section, we argue that this is not the case.

Specifically, we consider the following *uniform matroid bandit*. Let $E = \{1, \dots, L\}$ be a set of L items and the family of independent sets be defined as:

$$\mathcal{I} = \{I \subseteq E : |I| \leq K\} , \tag{37}$$

which means that any set of up to K items is feasible. Then $M = (E, \mathcal{I})$ is a rank-K *uniform matroid*. Let P be a distribution over the weights of the items, where the weight of each item is distributed independently of the other items. The weight of item e is drawn i.i.d. from a Bernoulli distribution with mean:

$$\bar{\mathbf{w}}(e) = \begin{cases} 0.5 & e \leq K \\ 0.5 - \Delta & \text{otherwise}, \end{cases} \tag{38}$$

where $0 < \Delta < 0.5$. Then $B_{\text{unif}} = (M, P)$ is our uniform matroid bandit. The optimal solution to B_{unif} is $A^* = \{1, \dots, K\}$, the first K items with the largest weights.

The key property of B_{unif} is that our gaps coincide with those of Kveton et al. (2014b). In particular, for any suboptimal item $e \notin A^*$, the difference between the returns of A^* and the best suboptimal solution that contains item e is Δ; the same as the difference between the returns of item e and any optimal item $e^* \in A^*$ (18). Because the gaps Δ are the same, our $O(L(1/\Delta) \log n)$ bound is indeed a factor of K tighter than the $O(KL(1/\Delta) \log n)$ bound of Kveton et al. (2014b).

4.4 Lower Bounds

We prove gap-dependent and gap-free lower bounds on the regret in polymatroid bandits. These bounds are derived on a class of polymatroid bandits that are equivalent to K independent Bernoulli bandits.

Specifically, we consider the following *partition matroid bandit*. Let $E = \{1, \dots, L\}$ be a set of L items and $B_1, \dots, B_K$ be a partition of this set such that $|B_i| = L/K$, where L/K is an integer. Let the family of independent sets be defined as:

$$\mathcal{I} = \{I \subseteq E : (\forall k : |\mathcal{I} \cap B_k| \leq 1)\} . \tag{39}$$

Then $M = (E, \mathcal{I})$ is a *partition matroid* of rank K. Let P be a probability distribution over the weights of the items, where the weight of each item is distributed independently of the other items. The weight of item e is drawn i.i.d. from a Bernoulli distribution with mean:

$$\bar{\mathbf{w}}(e) = \begin{cases} 0.5 & \exists k : e = \min_{i \in B_k} i \\ 0.5 - \Delta & \text{otherwise}, \end{cases} \tag{40}$$

where $0 < \Delta < 0.5$. Then $B_{\text{part}} = (M, P)$ is our partition matroid bandit. The key property of B_{part} is that it is equivalent to K independent Bernoulli bandits with L/K arms each. The optimal item in each bandit is the item with the smallest index. So the optimal solution is $A^* = \{e : (\exists k : e = \min_{i \in B_k} i)\}$. We also note that all gaps (18) are Δ.

To formalize our gap-dependent lower bound, we introduce the notion of *consistent algorithms*. We say that the algorithm is consistent if for any partition matroid bandit, any $e \notin A^*$, and any $\alpha > 0$, $\mathbb{E}[T_n(e)] = o(n^\alpha)$, where $T_n(e)$ is the number of times that item e is observed in n episodes. In the rest of our analysis, we focus only on consistent algorithms. This is without loss of generality. In particular, by the definition of consistency, inconsistent algorithms perform poorly on some instances of our problem and therefore cannot achieve logarithmic regret on all instances.

Proposition 6 *For any L and K such that L/K is an integer, and any Δ such that $0 < \Delta < 0.5$, the regret of any consistent algorithm on partition matroid bandit B_{part} is bounded from below as:*

$$\liminf_{n \to \infty} \frac{R(n)}{\log n} \geq \frac{L - K}{4\Delta}.$$

Proof The theorem is proved as follows:

$$\begin{aligned}
\liminf_{n \to \infty} \frac{R(n)}{\log n} &\geq \sum_{k=1}^{K} \sum_{e \in B_k - A^*} \frac{\Delta}{\mathrm{kl}(0.5 - \Delta, 0.5)} \\
&= \frac{(L - K)\Delta}{\mathrm{kl}(0.5 - \Delta, 0.5)} \\
&\geq \frac{L - K}{4\Delta},
\end{aligned} \tag{41}$$

where $\mathrm{kl}(0.5 - \Delta, 0.5)$ is the KL divergence between two Bernoulli variables with means $0.5 - \Delta$ and 0.5. The first inequality is due to an existing lower bound for Bernoulli bandits (Lai and Robbins, 1985), which is applied separately to each part B_k. The second inequality is due to $\mathrm{kl}(p, q) \leq \frac{(p-q)^2}{q(1-q)}$, where $p = 0.5 - \Delta$ and $q = 0.5$. ■

Now we prove a gap-free lower bound.

Proposition 7 *For any L and K such that L/K is an integer, and any $n > 0$, the regret of any algorithm on partition matroid bandit B_{part} is bounded from below as:*

$$R(n) \geq \frac{1}{20} \min(\sqrt{KLn}, Kn).$$

Proof The matroid bandit B_{part} can be viewed as K independent Bernoulli bandits with L/K arms each. By Theorem 5.1 of Auer et al. (2002b), for any time horizon n, the gap Δ can be chosen such that the regret of any algorithm on any of the K bandits is at least $\frac{1}{20} \min(\sqrt{(L/K)n}, n)$. So the regret due to all bandits is at least:

$$K \frac{1}{20} \min\left\{\sqrt{(L/K)n}, n\right\} = \frac{1}{20} \min\left\{\sqrt{KLn}, Kn\right\}. \tag{42}$$

Note that the bound of Auer et al. (2002b) is stated for the adversarial setting. However, because the worst-case environment in the proof is stochastic, it also applies to our problem. ■

4.5 Discussion of Theoretical Results

We prove two upper bounds on the expected cumulative regret of `OPM`:

$$\begin{aligned}
\text{Theorem 4 :} &\qquad O(L(1/\Delta)\log n), \\
\text{Theorem 5 :} &\qquad O(\sqrt{KLn \log n}),
\end{aligned}$$

where the gap is $\Delta = \min_{e} \min_{e^* \leq \rho(e)} \Delta_{e,e^*}$. Both bounds are at most linear in K and L, and sublinear in n. In other words, they scale favorably with all quantities of interest and therefore we expect them to be practical. Our $O(L(1/\Delta)\log n)$ upper bound matches the lower bound in Proposition 6 up to a constant and therefore is tight. It is also a factor of K tighter than the $O(KL(1/\Delta)\log n)$ upper bound of Kveton et al. (2014b) for a more general class of problems, stochastic combinatorial semi-bandits (Section 4.3). Our $O(\sqrt{KLn\log n})$ upper bound matches the lower bound in Proposition 7 up to a factor of $\sqrt{\log n}$.

Our gap-dependent upper bound has the same form as the bound of Auer et al. (2002a) for multi-armed bandits. This suggests that the sample complexity of learning the maximum-weight basis of a polymatroid is similar to that of the multi-armed bandit problem. The only major difference is in the definitions of the gaps. In other words, learning in polymatroids is extremely sample efficient.

The key step in our analysis is showing that the difference in the expected returns of bases $\mathbf{x}^*$ and $\mathbf{x}_t$ can be expressed as the sum of the differences in the expected returns of intermediate solutions, all of which are bases such that the difference in the gains of any two consecutive bases has at most one negative entry. This decomposition is highly non-trivial and is derived based on the submodularity of our problem. An important aspect of our analysis is that the terms $\delta_t(e, e^*)$ (Theorem 3) are not bounded until it is necessary, such as in (30). Therefore, our upper bounds are tight and do not contain quantities that are not native to our problem, such as the maximum number of non-zero entries in $\mathbf{x} \in \Theta$. In fact, under the assumption that $f(e) \leq 1$ for all items $e \in E$ (Section 2), our notion of complexity $K = f(E)$ is never larger than the maximum number of non-zero entries in any feasible solution $\mathbf{x} \in \Theta$, a common quantity in the regret bounds of combinatorial bandits (Gai et al., 2012; Kveton et al., 2014b; Cesa-Bianchi and Lugosi, 2012; Audibert et al., 2014).

5. Experiments

We conduct three experiments. In Section 5.1, we evaluate the tightness of our regret bounds on a synthetic problem. In Section 5.2, we apply `OPM` to the problem of learning routing networks. Finally, in Section 5.3, we evaluate `OPM` on the problem of recommending diverse movies.

All experiments are episodic. In each episode, `OPM` chooses a basis $\mathbf{x}_t$, observes the weights of all items that contribute to $\mathbf{x}_t$, and updates its model of the world. In Sections 5.2 and 5.3, the performance of `OPM` is measured by the *expected per-step return* in n episodes:

$$\frac{1}{n}\mathbb{E}_{\mathbf{w}_1,\dots,\mathbf{w}_n}\left[\sum_{t=1}^{n}\langle \mathbf{w}_t, \mathbf{x}_t\rangle\right], \tag{43}$$

which is the expected cumulative return in n episodes divided by n. We choose this metric because we want to report the quality of solutions and not just their regret, the difference from the optimal solution.

We compare `OPM` to two baselines. The first baseline is the maximum-weight basis $\mathbf{x}^*$ (12). This is our notion of optimality. The second baseline is an ε-greedy policy. The policy is implemented similarly to `OPM`. In particular, it is Algorithm 2 that is modified as follows. In each episode, $U_t(e)$ is set to $\hat{\mathbf{w}}_{T_{t-1}(e)}(e)$ for all items e with probability $1 - \varepsilon$. With probability ε, $U_t(e)$ is chosen randomly for all items e. The exploration rate is set as $\varepsilon = 0.1$. In all of our experiments, this is the best performing ε-greedy policy from the class of ε-greedy policies where $\varepsilon \in \{0, 0.1, \dots, 1\}$.

5.1 Minimum-Cost Flow

In the first experiment, we evaluate `OPM` on a synthetic problem of learning minimum-cost flows. The experiment shows that our $O(L(1/\Delta)\log n)$ gap-dependent upper bound is practical. We experiment with larger values of Δ. In this setting, our gap-dependent upper bound is tighter than the gap-free one.

We experiment with a flow network with L source nodes and one sink node. The network is illustrated in Figure 1. The network is defined by three constraints. First, the maximum flow through any source node is 1. Second, the maximum flow through any two consecutive source nodes, e and $e + 1$ where $e = 2i - 1$ for $i \in \{1, \dots, L/2\}$, is $\frac{3}{2}$. Third, the maximum flow is K. We assume that K is an integer multiple of $\frac{3}{2}$. The

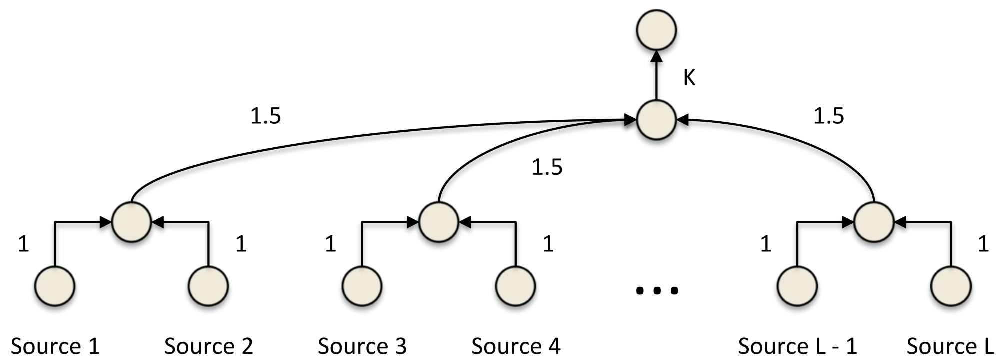


Figure 1: The flow network in Section 5.1. The network contains L source nodes and the maximum flow is K. The capacity of the link is shown next to the link.

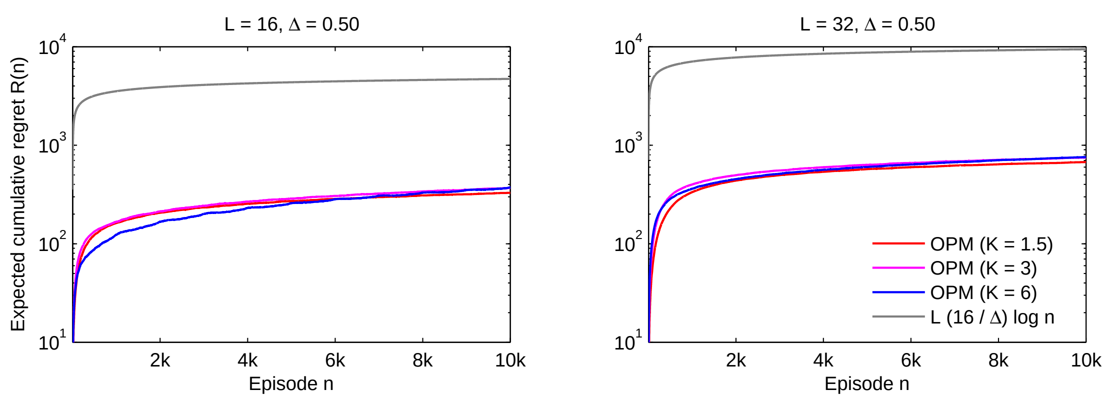


Figure 2: The regret of `OPM` as a function of the number of episodes n. The regret is averaged over 100 runs.

cost of the flow from source node e is a Bernoulli random variable with mean:

$$\bar{\mathbf{w}}(e) = \begin{cases} 0.5 - \Delta/2 & e \leq \frac{4}{3}K \\ 0.5 + \Delta/2 & \text{otherwise.} \end{cases} \tag{44}$$

Our problem is parametrized by K, L, and Δ. The optimal solution to the problem is to pass the maximum flow through the first $\frac{4}{3}K$ source nodes.

Our problem can be formulated as minimizing a modular function on a polymatroid. The ground set E are L source nodes. The submodular function f captures the structure of the network and is defined as:

$$f(X) = \min\left\{\sum_{i=1}^{L/2} \min\left\{\mathbb{1}\{(2i-1) \in X\} + \mathbb{1}\{2i \in X\}, \frac{3}{2}\right\}, K\right\}. \tag{45}$$

Note that $f(X)$ can be computed in $O(L)$ time, by summing up L indicators. The weight of item e is drawn i.i.d. from a Bernoulli distribution with mean $\bar{\mathbf{w}}(e)$ in (44), independently of the other items.

In Figure 2, we report the regret of `OPM` as a function of the number of episodes n for various settings of K and L. The gap is $\Delta = 0.5$. We observe three major trends. First, the regret grows on the order of $\log n$, as suggested by our $O(L(1/\Delta)\log n)$ upper bound. Second, the regret does not change much with K. This is consistent with the fact that our bound is independent of K. Finally, we note that the bound is surprisingly tight. In particular, for larger values of n, it is only about 10 times larger than the actual regret.

L	K	Δ	Regret $R(n)$	$L(16/\Delta)\log n$	Δ	Regret $R(n)$	$L(16/\Delta)\log n$
16	1.50	0.5	329.1 ± 2.5	4,716	0.25	577.6 ± 4.1	9,431
16	3.00	0.5	368.6 ± 3.4	4,716	0.25	599.7 ± 4.3	9,431
16	6.00	0.5	373.3 ± 4.8	4,716	0.25	546.1 ± 5.6	9,431
32	1.50	0.5	675.5 ± 3.0	9,431	0.25	$1,182.6 \pm 6.2$	18,863
32	3.00	0.5	748.8 ± 3.9	9,431	0.25	$1,356.1 \pm 6.0$	18,863
32	6.00	0.5	759.6 ± 4.3	9,431	0.25	$1,299.3 \pm 6.9$	18,863

Table 1: The regret of `OPM` as a function of K, L, and Δ. The regret is averaged over 100 runs.

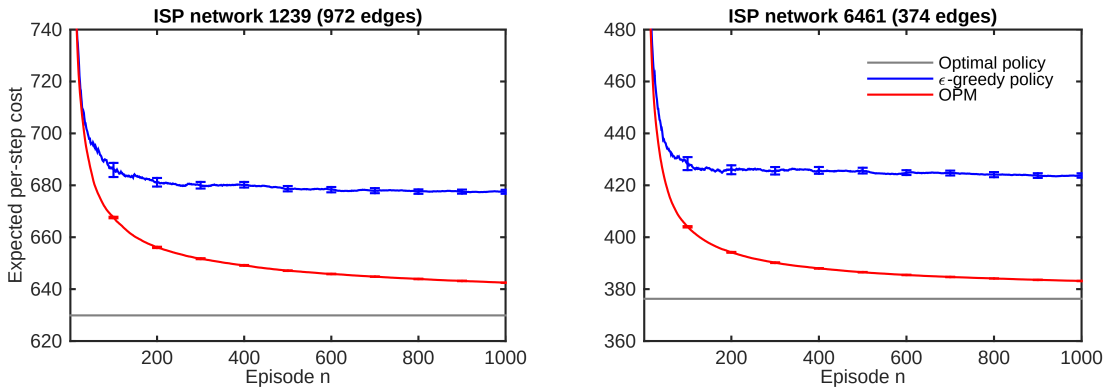


Figure 3: The expected per-step cost of building two minimum spanning trees in up to 10^3 episodes.

In Table 1, we report the regret of `OPM` in 10k episodes for various settings of K, L, and Δ. We observe that the regret depends on L and Δ as suggested by our $O(L(1/\Delta)\log n)$ upper bound. In particular, it does not change much with K, and it doubles as we double L or halve Δ. We note again that our upper bound is surprisingly tight, never more than 20 times larger than the actual regret.

5.2 Minimum Spanning Tree

In the second experiment, we apply `OPM` to the problem of learning routing networks for an Internet service provider (ISP). The routing network is a spanning tree (Oliveira and Pardalos, 2005). Our goal is to identify the spanning tree that has the lowest expected latency on its edges. Note that this is a minimization problem. Therefore, we refer to the return of a policy as its *cost*.

Our problem can be formulated as a *graphic matroid bandit* (Kveton et al., 2014a), which is a form of a polymatroid bandit. The ground set E are the edges of the graph that represents the topology of the network. We experiment with 6 networks from the *RocketFuel* dataset (Spring et al., 2004), with up to 300 nodes and 10^3 edges (Table 2). A set of edges is *independent* if it forms a forest. The corresponding rank function f is defined as:

$$f(X) = |\text{largest subset of } X \text{ that is a forest}|\,. \tag{46}$$

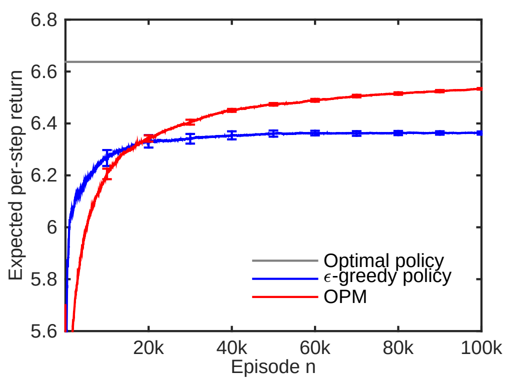
The value of $f(X)$ can be computed naively by a greedy algorithm in $O(|X|^2)$ time. The latency of edge e in episode t is:

$$\mathbf{w}_t(e) = \bar{\mathbf{w}}(e) - 1 + \varepsilon, \tag{47}$$

where $\bar{\mathbf{w}}(e)$ is the expected latency, which is recorded in our dataset; and $\varepsilon \sim \mathrm{Exp}(1)$ is exponential noise. The latency $\bar{\mathbf{w}}(e)$ ranges from 1 to 64 milliseconds. Our noise model is motivated by the observation that the latency in ISP networks can be mostly explained by geographical distances (Choi et al., 2004), the expected

ISP network	Number of nodes	Number of edges	Minimum latency	Maximum latency	Average latency	Optimal policy	ε-greedy policy	OPM
1221	108	153	1	17	2.78	305.00	307.26 ± 0.06	305.82 ± 0.07
1239	315	972	1	64	3.20	629.88	677.51 ± 0.77	642.49 ± 0.09
1755	87	161	1	31	2.91	192.81	199.56 ± 0.13	194.92 ± 0.07
3257	161	328	1	47	4.30	550.85	570.21 ± 0.40	560.11 ± 0.08
3967	79	147	1	44	5.19	306.80	321.21 ± 0.22	308.66 ± 0.05
6461	141	374	1	45	6.32	376.27	423.78 ± 0.85	383.15 ± 0.08

Table 2: The description of 6 ISP networks from our experiments and the expected per-step costs of building minimum spanning trees on these networks in 10^3 episodes. All latencies and costs are reported in milliseconds.



Movie title	Movie genres
American Beauty	**Comedy Drama**
Jurassic Park	**Action Adventure Sci-Fi**
Saving Private Ryan	Action Drama **War**
The Matrix	Action Sci-Fi **Thriller**
Fargo	**Crime** Drama Thriller
Shakespeare in Love	Comedy **Romance**
L.A. Confidential	Crime **Film-Noir Mystery** Thriller
E.T.	**Children's** Drama **Fantasy** Sci-Fi
Ghostbusters	Comedy **Horror**
Toy Story	**Animation** Children's Comedy

Figure 4: **Left**. The return of three movie recommendation policies in up to 10^5 episodes. **Right**. Ten most popular movies in the optimal solution, $\{e : \mathbf{x}^*(e) > 0\}$. These movies are shown in the order of decreasing popularity. The movie genre is highlighted if the associated movie is the most popular movie in that genre.

latency $\bar{\mathbf{w}}(e)$. The noise tends to be small, on the order of a few hundred microseconds, and it is unlikely to cause high latency.

In Figure 3, we report our results on two largest ISP networks. We observe two trends. First, the cost of OPM approaches that of the optimal solution $\mathbf{x}^*$ as the number of episodes increases. Second, OPM performs better than the ε-greedy policy in less than 10 episodes. The costs of all policies on all networks are reported in Table 2. We observe that OPM outperforms the ε-greedy policy, typically by a large margin.

OPM learns quickly because our networks are sparse. In particular, the number of edges in each network is never more than four times larger than the number of edges in its spanning tree. So theoretically, each edge can be observed at least once in four episodes and OPM learns quickly the mean latency of each edge.

5.3 Diverse recommendations

In the third experiment, OPM is evaluated as a movie recommender. The recommender is used repeatedly by simulated users and the goal is to learn how to recommended diverse movies that maximize the satisfaction of an average user (Section 2.1). A system like this could be used in practice to identify trending movies.

We experiment with the *MovieLens* dataset (Lam and Herlocker, 2013), a dataset of 6 thousand people who assigned one million ratings to 4 thousand movies. The ground set E are 100 movies, 50 most and least rated movies in our dataset. The movies cover 18 genres. The submodular function f is defined as:

$$f(X) = |\text{movie genres covered by movies } X| \,. \tag{48}$$

The value of $f(X)$ can be computed in $O(|X|)$ time. The weight of movie e in episode t is:

$$\mathbf{w}_t(e) = \mathbb{1}\{\text{user in episode } t \text{ watches movie } e\} \,. \tag{49}$$

The user in episode t is chosen randomly from our pool of 6k users. We assume that the user watches movie e if that movie is rated by that user in our dataset. The expected weight $\bar{\mathbf{w}}(e)$ is the probability that movie e is watched by a randomly chosen user.

Our results are reported in Figure 4. As in Section 5.2, we observe two major trends. First, the return of `OPM` approaches that of the optimal solution $\mathbf{x}^*$ as the number of episodes increases. Second, `OPM` performs better than the ε-greedy policy after about 20k episodes. The optimal solution $\mathbf{x}^*$ is visualized in Figure 4. It contains 13 movies and therefore is extremely sparse.

6. Related Work

Polymatroids (Edmonds, 1970) are a generalization of matroids (Whitney, 1935). Therefore, our work can be viewed as a generalization of matroid semi-bandits (Kveton et al., 2014a). We significantly extend the work of Kveton et al. (2014a) and essentially show that the problem of maximizing a modular function subject to a submodular constraint can be learned efficiently. Our generalization is by far non-trivial. For instance, the key part of our analysis is a novel regret decomposition (Section 4.1), which leverages the submodularity of our constraint. This structure is not apparent in the work of Kveton et al. (2014a).

Our problem is an instance of a stochastic combinatorial semi-bandit (Gai et al., 2012). Gai et al. (2012) proposed and analyzed a UCB-like algorithm for solving this problem. Chen et al. (2013) and Kveton et al. (2014b) proved $O(K^2 L (1/\Delta) \log n)$ and $O(KL(1/\Delta) \log n)$ upper bounds on the regret of this algorithm, respectively. The latter is tight (Kveton et al., 2014b). `OPM` is an instance of the UCB-like algorithm where the combinatorial optimization oracle is greedy. Our optimization problem is on a polymatroid and therefore we can derive a factor of K tighter gap-dependent regret bound (Theorem 4) than Kveton et al. (2014b). We note that our gap-free regret bound is of the same magnitude.

COMBAND (Cesa-Bianchi and Lugosi, 2012), follow-the-perturbed-leader (FPL) with geometric resampling (Neu and Bartok, 2013), and online stochastic mirror descent (OSMD) (Audibert et al., 2014) are three recently proposed algorithms for adversarial combinatorial semi-bandits. FPL does not achieve the optimal regret, but it is computationally efficient when the offline variant of the combinatorial optimization problem can be solved efficiently (Audibert et al., 2014). OSMD achieves the optimal regret, but it is not guaranteed to be computationally efficient if the projection on the convex hull of the feasible set cannot be implemented efficiently. In our problem, the convex hull is B_M (2) and the projection can be implemented in $O(L^6)$ time (Suehiro et al., 2013). So the time complexity of a single step of OSMD is $O(L^6)$. This is several orders of magnitude higher than the time complexity of `OPM`, $O(L \log L)$; and not very practical for large values of L. Finally, COMBAND is not guaranteed to be computationally efficient. Based on Section 5.4 of Cesa-Bianchi and Lugosi (2012), even on the problem of learning the minimum spanning tree, an instance of maximizing a modular function on a polymatroid.

Several recent papers studied the problem of learning how to maximize a submodular function (Guillory and Bilmes, 2011; Yue and Guestrin, 2011; Gabillon et al., 2013; Wen et al., 2013; Gabillon et al., 2014). These papers are only loosely related to our work because they study a different problem, which is learning how to maximize an *unknown submodular function* subject to a cardinality constraint. Our learning problem is maximizing an *unknown modular function* subject to a *known submodular constraint*.

7. Conclusions

In this work, we study the problem of learning to act greedily. We formulate the problem as learning how to maximize an unknown modular function on a known polymatroid in the bandit setting. Our formulation is quite general and includes many popular problems, such as learning variants of the minimum spanning tree and minimum-cost flow. We propose a computationally-efficient method for solving the problem and prove two upper bounds on its regret. Our $O(L(1/\Delta)\log n)$ gap-dependent upper bound is tight up to a constant and our $O(\sqrt{KLn\log n})$ gap-free upper bound is tight up to a factor of $\sqrt{\log n}$. We evaluate our method on three problems, and show that it can learn near-optimal policies computationally and sample efficiently.

We leave open several questions of interest. For instance, our $O(\sqrt{KLn\log n})$ upper bound matches the $\Omega(\sqrt{KLn})$ lower bound only up to a factor of $\sqrt{\log n}$. We strongly believe that this factor can be eliminated by modifying the confidence radius in (15) as in Audibert and Bubeck (2009). We leave this for future work.

Thompson sampling (Thompson, 1933) often performs better in practice than `UCB1` (Auer et al., 2002a). We believe that it is relatively straightforward to propose a Thompson-sampling variant of `OPM`, by replacing the UCBs in Algorithm 2 with sampling from the posterior on the mean weights (Wen et al., 2014). We also believe that the regret of this algorithm is bounded and this can be proved. The reason is that the frequentist analysis of Thompson sampling (Agrawal and Goyal, 2012) resembles that of `UCB1` (Auer et al., 2002a). As a result, it is likely that the analysis of Thompson-sampling `OPM` can be carried out similarly to this paper.

In this work, we study one particular problem, maximization of a modular function on a polymatroid, in one particular learning setting, stochastic semi-bandits. It is an open question whether the ideas in our paper generalize to other polymatroid problems, such as maximizing a modular function on the intersection of two matroids (Papadimitriou and Steiglitz, 1998); and other learning variants of our problem, such as learning in the adversarial setting (Auer et al., 2002b) or with the full-bandit feedback (Dani et al., 2008).

References

Shipra Agrawal and Navin Goyal. Analysis of Thompson sampling for the multi-armed bandit problem. In *Proceeding of the 25th Annual Conference on Learning Theory*, pages 39.1–39.26, 2012.

Azin Ashkan, Branislav Kveton, Shlomo Berkovsky, and Zheng Wen. Diversified utility maximization for recommendations. In *Poster Proceedings of the 8th ACM Conference on Recommender Systems*, 2014a.

Azin Ashkan, Branislav Kveton, Shlomo Berkovsky, and Zheng Wen. DUM: Diversity-weighted utility maximization for recommendations. *CoRR*, abs/1411.3650, 2014b.

Jean-Yves Audibert and Sebastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *Proceedings of the 22nd Annual Conference on Learning Theory*, 2009.

Jean-Yves Audibert, Sebastien Bubeck, and Gabor Lugosi. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 39(1):31–45, 2014.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002a.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal of Computing*, 32(1):48–77, 2002b.

Dimitris Bertsimas and John Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, MA, 1997.

Nicolo Cesa-Bianchi and Gabor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework, results and applications. In *Proceedings of the 30th International Conference on Machine Learning*, pages 151–159, 2013.

Baek-Young Choi, Sue Moon, Zhi-Li Zhang, Konstantina Papagiannaki, and Christophe Diot. Analysis of point-to-point packet delay in an operational network. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, 2004.

Varsha Dani, Thomas Hayes, and Sham Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of the 21st Annual Conference on Learning Theory*, pages 355–366, 2008.

Jack Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Structures and Their Applications: Proceedings of the Calgary International Conference on Combinatorial Structures and Their Applications*, pages 69–87. 1970.

Satoru Fujishige. *Submodular Functions and Optimization*. Elsevier, Amsterdam, The Netherlands, 2005.

Victor Gabillon, Branislav Kveton, Zheng Wen, Brian Eriksson, and S. Muthukrishnan. Adaptive submodular maximization in bandit setting. In *Advances in Neural Information Processing Systems 26*, pages 2697–2705, 2013.

Victor Gabillon, Branislav Kveton, Zheng Wen, Brian Eriksson, and S. Muthukrishnan. Large-scale optimistic adaptive submodularity. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2014.

Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking*, 20(5):1466–1478, 2012.

Andrew Guillory and Jeff Bilmes. Online submodular set cover, ranking, and repeated active learning. In *Advances in Neural Information Processing Systems 24*, pages 1107–1115, 2011.

Branislav Kveton, Zheng Wen, Azin Ashkan, Hoda Eydgahi, and Brian Eriksson. Matroid bandits: Fast combinatorial optimization with learning. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*, pages 420–429, 2014a.

Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. *CoRR*, abs/1410.0949, 2014b.

T. L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

Shyong Lam and Jon Herlocker. MovieLens 1M Dataset. http://www.grouplens.org/node/12, 2013.

Nimrod Megiddo. Optimal flows in networks with multiple sources and sinks. *Mathematical Programming*, 7(1):97–107, 1974.

Gergely Neu and Gabor Bartok. An efficient algorithm for learning with semi-bandit feedback. In *24th International Conference on Algorithmic Learning Theory*, volume 8139 of *Lecture Notes in Computer Science*, pages 234–248. 2013.

Carlos Oliveira and Panos Pardalos. A survey of combinatorial optimization problems in multicast routing. *Computers and Operations Research*, 32(8):1953–1981, 2005.

Christos Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization*. Dover Publications, Mineola, NY, 1998.

Neil Spring, Ratul Mahajan, and David Wetherall. Measuring ISP topologies with Rocketfuel. *IEEE / ACM Transactions on Networking*, 12(1):2–16, 2004.

Daiki Suehiro, Kohei Hatano, Shuji Kijima, Eiji Takimoto, and Kiyohito Nagano. Online prediction under submodular constraints. In *23rd International Conference on Algorithmic Learning Theory*, volume 7568 of *Lecture Notes in Computer Science*, pages 260–274. 2013.

William. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3-4):285–294, 1933.

Zheng Wen, Branislav Kveton, Brian Eriksson, and Sandilya Bhamidipati. Sequential Bayesian search. In *Proceedings of the 30th International Conference on Machine Learning*, pages 226–234, 2013.

Zheng Wen, Azin Ashkan, Hoda Eydgahi, and Branislav Kveton. Efficient learning in large-scale combinatorial semi-bandits. *CoRR*, abs/1406.7443, 2014.

Hassler Whitney. On the abstract properties of linear dependence. *American Journal of Mathematics*, 57(3): 509–533, 1935.

Yisong Yue and Carlos Guestrin. Linear submodular bandits and their application to diversified retrieval. In *Advances in Neural Information Processing Systems 24*, pages 2483–2491, 2011.

Appendix A. Technical Lemmas

Lemma 8 *Let $M = (E, f)$ be a polymatroid, V be the vertices of base polyhedron B_M in* (2)*, and Θ be the feasible solutions in* (9)*. Then $V = \Theta$.*

Proof The key observation is that B_M is a convex polytope because it is an intersection of convex polytope P_M (1) and hyperplane $\sum_{e \in E} \mathbf{x}(e) = K$. Therefore, any vector $\mathbf{x} \in B_M$ is a convex combination of V. We prove $V = \Theta$ by proving that $V \subseteq \Theta$ and $\Theta \subseteq V$.

First, we prove that $V \subseteq \Theta$. By contradiction, suppose that there exists $\mathbf{x} \in V$ such that $\mathbf{x} \notin \Theta$. Since $\mathbf{x}$ is a vertex of a convex polytope, there must exist a weight vector $\mathbf{w}$ such that $\mathbf{x}$ is a unique optimum in (3). By the definition of `Greedy`, $\mathbf{x} = \texttt{Greedy}(M, \mathbf{w})$ and therefore $\mathbf{x} \in \Theta$. This is clearly a contradiction.

Second, we prove that $\Theta \subseteq V$. By contradiction, suppose that there exists $\mathbf{x} \in \Theta$ such that $\mathbf{x} \notin V$. Since B_M is a convex polytope, the solution $\mathbf{x}$ can be expressed as a convex combination of the vertices in V. For simplicity of exposition, suppose that $\mathbf{x} = \alpha \mathbf{x}_1 + (1 - \alpha)\mathbf{x}_2$, where $\{\mathbf{x}_1, \mathbf{x}_2\} \subset V$ and $\alpha \in (0, 1)$. Let e_i be the first item in `Greedy` where $\mathbf{x}_1(e_i) < \mathbf{x}_2(e_i)$. Since $\mathbf{x}$ is generated by `Greedy`, and f is a monotonic and submodular function, $\mathbf{x}(e_i) \geq \mathbf{x}_2(e_i)$. This is clearly a contradiction since $\mathbf{x}(e_i) < (1 - \alpha)\mathbf{x}_2(e_i)$. The case where $\mathbf{x}_1(e_i) > \mathbf{x}_2(e_i)$ is proved similarly. Finally, suppose that $\mathbf{x}(e_i) \neq \mathbf{x}_2(e_i)$ does not happen for any e_i. Then $\mathbf{x}_1 = \mathbf{x}_2$, which is also a contradiction. ■

Lemma 9 *For all items e and $e^* \leq \rho(e)$:*

$$\mathbb{E}_{\mathbf{w}_1, \dots, \mathbf{w}_n}\left[\sum_{t=1}^{n} \delta_t(e, e^*) \mathbb{1}\{T_{t-1}(e) > \ell\}\right] \leq \frac{4}{3}\pi^2$$

when $\ell = \left\lfloor \frac{8}{\Delta_{e,e^*}^2} \log n \right\rfloor$.

Proof First, we note that $\delta_t(e, e^*) \leq 1$. Moreover, by Theorem 3, the event $\delta_t(e, e^*) > 0$ implies that we observe the weight of item e and $U_t(e) \geq U_t(e^*)$. Based on these facts, it follows that:

$$\begin{aligned} \sum_{t=1}^{n} \delta_t(e, e^*) \mathbb{1}\{T_{t-1}(e) > \ell\} &\leq \sum_{t=1}^{n} \mathbb{1}\{\delta_t(e, e^*) > 0,\ T_{t-1}(e) > \ell\} \\ &\leq \sum_{t=\ell+1}^{n} \mathbb{1}\{U_t(e) \geq U_t(e^*),\ T_{t-1}(e) > \ell\} \\ &\leq \sum_{t=\ell+1}^{n} \sum_{s=1}^{t} \sum_{s_e=\ell+1}^{t} \mathbb{1}\{\hat{\mathbf{w}}_{s_e}(e) + c_{t-1,s_e} \geq \hat{\mathbf{w}}_s(e^*) + c_{t-1,s}\} \\ &= \sum_{t=\ell}^{n-1} \sum_{s=1}^{t+1} \sum_{s_e=\ell+1}^{t+1} \mathbb{1}\{\hat{\mathbf{w}}_{s_e}(e) + c_{t,s_e} \geq \hat{\mathbf{w}}_s(e^*) + c_{t,s}\}\,. \end{aligned} \tag{50}$$

When $\hat{\mathbf{w}}_{s_e}(e) + c_{t,s_e} \geq \hat{\mathbf{w}}_s(e^*) + c_{t,s}$, at least one of the following events must happen:

$$\hat{\mathbf{w}}_s(e^*) \leq \bar{\mathbf{w}}(e^*) - c_{t,s} \tag{51}$$

$$\hat{\mathbf{w}}_{s_e}(e) \geq \bar{\mathbf{w}}(e) + c_{t,s_e} \tag{52}$$

$$\bar{\mathbf{w}}(e^*) < \bar{\mathbf{w}}(e) + 2c_{t,s_e}. \tag{53}$$

We bound the probability of the first two events, (51) and (52), using Hoeffding's inequality:

$$P(\hat{\mathbf{w}}_s(e^*) \leq \bar{\mathbf{w}}(e^*) - c_{t,s}) \leq \exp[-4 \log t] = t^{-4} \tag{54}$$

$$P(\hat{\mathbf{w}}_{s_e}(e) \geq \bar{\mathbf{w}}(e) + c_{t,s_e}) \leq \exp[-4 \log t] = t^{-4}. \tag{55}$$

When $s_e \geq \frac{8}{\Delta_{e,e^*}^2} \log n$, the third event (53) cannot happen because:

$$\bar{\mathbf{w}}(e^*) - \bar{\mathbf{w}}(e) - 2c_{t,s_e} = \Delta_{e,e^*} - 2\sqrt{\frac{2\log t}{s_e}} \geq 0. \tag{56}$$

This is guaranteed when $\ell = \left\lfloor \frac{8}{\Delta_{e,e^*}^2} \log n \right\rfloor$. Finally, we combine all of our claims and get:

$$\begin{aligned}
\mathbb{E}_{\mathbf{w}_1,\ldots,\mathbf{w}_n}\left[\sum_{t=1}^{n} \delta_t(e,e^*)\mathbb{1}\{T_{t-1}(e) > \ell\}\right] &\leq \sum_{t=\ell}^{n-1}\sum_{s=1}^{t+1}\sum_{s_e=\ell+1}^{t+1} [P(\hat{\mathbf{w}}_s(e^*) \leq \bar{\mathbf{w}}(e^*) - c_{t,s}) + \\
&\qquad\qquad\qquad P(\hat{\mathbf{w}}_{s_e}(e) \geq \bar{\mathbf{w}}(e) + c_{t,s_e})] \\
&\leq \sum_{t=1}^{\infty} 2(t+1)^2 t^{-4} \\
&\leq \sum_{t=1}^{\infty} 8t^{-2} \\
&= \frac{4}{3}\pi^2.
\end{aligned} \tag{57}$$

The last equality follows from the fact that $\sum_{t=1}^{\infty} t^{-2} = \frac{\pi^2}{6}$. ■

Lemma 10 (Kveton et al. (2014a)) *Let $\Delta_1 \geq \ldots \geq \Delta_K$ be a sequence of K positive numbers. Then:*

$$\left[\Delta_1 \frac{1}{\Delta_1^2} + \sum_{k=2}^{K} \Delta_k \left(\frac{1}{\Delta_k^2} - \frac{1}{\Delta_{k-1}^2}\right)\right] \leq \frac{2}{\Delta_K}.$$

Proof First, we note that:

$$\left[\Delta_1 \frac{1}{\Delta_1^2} + \sum_{k=2}^{K} \Delta_k \left(\frac{1}{\Delta_k^2} - \frac{1}{\Delta_{k-1}^2}\right)\right] = \sum_{k=1}^{K-1} \frac{\Delta_k - \Delta_{k+1}}{\Delta_k^2} + \frac{1}{\Delta_K}. \tag{58}$$

Second, by our assumption, $\Delta_k \geq \Delta_{k+1}$ for all $k < K$. Therefore:

$$\begin{aligned}
\sum_{k=1}^{K-1} \frac{\Delta_k - \Delta_{k+1}}{\Delta_k^2} + \frac{1}{\Delta_K} &\leq \sum_{k=1}^{K-1} \frac{\Delta_k - \Delta_{k+1}}{\Delta_k \Delta_{k+1}} + \frac{1}{\Delta_K} \\
&= \sum_{k=1}^{K-1} \left[\frac{1}{\Delta_{k+1}} - \frac{1}{\Delta_k}\right] + \frac{1}{\Delta_K} \\
&= \frac{2}{\Delta_K} - \frac{1}{\Delta_1} \\
&< \frac{2}{\Delta_K}.
\end{aligned} \tag{59}$$

This concludes our proof. ■