Submodular Bandit Problem Under Multiple Constraints

Sho Takemori, Masahiro Sato, Takashi Sonoda, Janmajay Singh, Tomoko Ohkuma
Fuji Xerox Co., Ltd., Yokohama, Japan.
{takemori.sho, sato.masahiro, takashi.sonoda, janmajay.singh, ohkuma.tomoko}@fujixerox.co.jp

Abstract

The *linear submodular bandit problem* was proposed to simultaneously address diversified retrieval and online learning in a recommender system. If there is no uncertainty, this problem is equivalent to a submodular maximization problem under a cardinality constraint. However, in some situations, recommendation lists should satisfy additional constraints such as *budget constraints*, other than a cardinality constraint. Thus, motivated by diversified retrieval considering budget constraints, we introduce a submodular bandit problem under the intersection of l *knapsacks* and a k-*system constraint*. Here k-system constraints form a very general class of constraints including cardinality constraints and the intersection of k *matroid* constraints. To solve this problem, we propose a non-greedy algorithm that adaptively focuses on a standard or modified upper-confidence bound. We provide a high-probability upper bound of an *approximation regret*, where the approximation ratio matches that of a fast offline algorithm. Moreover, we perform experiments under various combinations of constraints using a synthetic and two real-world datasets and demonstrate that our proposed method outperforms the existing baselines.

1 INTRODUCTION

The *multi-armed bandit (MAB) problem* has been widely used for practical applications. Examples include interactive recommender systems, Internet advertising, portfolio selection, and clinical trials. In a typical MAB problem, the agent selects one arm in each round. However, in practice, it is more convenient to select more than one arm in each round. Such a problem is called a *combinatorial bandit problem* (W. Chen et al. 2013). For example, in (Yue and Guestrin 2011; Radlinski et al. 2008), they considered the problem where in each round, the agent proposes multiple news articles or web documents to a user.

When recommending multiple items to a user, agents should select *well-diversified* items to maximize coverage of the information the user finds interesting (Yue and Guestrin 2011) or to reduce item similarity in the list (Ziegler et al. 2005). Recommending redundant items leads to *diminishing returns* in terms of utility (Yu et al. 2016). It is well-known that properties such as diversity or diminishing returns are well captured by *submodular set functions* (Krause and Golovin 2014). To simultaneously address diversified retrieval and online learning in a recommender system, Yue and Guestrin (2011) proposed a combinatorial bandit problem (or more specifically a semi-bandit problem), called the *linear submodular bandit problem*, where in each round a sequence rewards are generated by an unknown submodular function.

For a real-world application, recommendation lists should satisfy several constraints. We explain this by using a news article recommendation example. For a comfortable user experience while selecting news articles from a recommendation list, the length of the list should not be excessively long, which implies that the list should satisfy a cardinality constraint. Furthermore, a user may not wish to spend more than a certain amount of time by reading news articles. This can be modeled as a *knapsack constraint*. With only a knapsack constraint, a system can recommend a long list of short (or low cost) news articles. However, due to the space constraint of the web site, such a list cannot be displayed. Therefore, it is necessary to consider a submodular bandit problem under the intersection of the knapsack and cardinality constraints.

Proceedings of the 36th Conference on Uncertainty in Artificial Intelligence (UAI), PMLR volume 124, 2020.

Yue and Guestrin (2011) introduced a submodular bandit problem under a cardinality constraint and proposed an algorithm called LSBGreedy. Later, Yu et al. (2016) considered a submodular bandit problem under a knapsack constraint and proposed two greedy algorithms called MCSGreedy and CGreedy. However, such existing algorithms fail to properly optimize the objective function under complex constraints. In fact, we theoretically and empirically show that such simple greedy algorithms can perform poorly.

Under a simple constraint such as a cardinality or a knapsack constraint, there is a simple rule to select elements. This rule is called the *upper confidence bound (UCB) rule* or the *modified UCB rule* if the constraint is a cardinality or a knapsack constraint, respectively (Yu et al. 2016). For example, with the UCB rule, the algorithm selects the element with the largest UCB sequentially in each round. Considering that our problem is a generalization of both the problems, we should generalize both the rules.

In this study, we solve the problem under a more generalized constraint, i.e., the intersection of *l knapsacks* and *k-system constraints*. Here, the k-system constraints form a very general class of constraints, including cardinality constraints and the intersection of *k matroid* constraints. For example, when recommending news articles, we can restrict the number of news articles from each topic with a k-system constraint. To solve the problem, we propose a non-greedy algorithm that adaptively focuses on the UCB and modified UCB rules. Since the submodular maximization problem is NP-hard, we theoretically evaluate our method by an *α-approximation regret*, where $\alpha \in (0, 1)$ is an approximation ratio. In this study, we provide an upper bound of the α-approximation regret in the case when $\alpha = \frac{1}{(1+\varepsilon)(k+2l+1)}$, where ε is a parameter of the algorithm. We note that the approximation ratio matches that of an *offline* algorithm (Badanidiyuru and Vondrák 2014). To the best of our knowledge, no known offline algorithm achieves a better approximation ratio than α above and better computational complexity than the offline algorithm, simultaneously [1]. More precisely, our contributions are stated as follows:

OUR CONTRIBUTIONS

1. We propose a submodular bandit problem with semi-bandit feedback under the intersection of l knapsacks and k-system constraints (Section 4). This is the first attempt to solve the submodular bandit problem under such complex constraints. The problem is new even when the k-system constraint is a cardinality constraint.
2. We propose a novel algorithm called *AFSM-UCB* that Adaptively Focuses on a Standard or Modified Upper Confidence Bound (Section 5).
3. We provide a high-probability upper bound of an approximation regret for AFSM-UCB (Section 6). We prove that the α-approximation regret $\mathrm{Reg}_\alpha(T)$ is given by $O(\sqrt{mT}\ln(mT/\delta))$ with probability in least $1-\delta$ and the computational complexity in each round is given as $O(m|\mathcal{N}|\ln|\mathcal{N}|/\ln(1+\varepsilon))$, where $\alpha = \frac{1}{(1+\varepsilon)(k+2l+1)}$, ε is a parameter of the algorithm, T is the time horizon, m is the cardinality of a maximal feasible solution, and $\mathcal{N}$ is the ground set (e.g., the set of all news articles in the news recommendation example). We note that no known offline fast[2] algorithm achieves a better approximation ratio than above.[3]
4. We empirically prove the effectiveness of our proposed method by comprehensively evaluating it on a synthetic and two real-world datasets. We show that our proposed method outperforms the existing greedy baselines such as LSBGreedy and CGreedy.

2 RELATED WORK

2.1 SUBMODULAR MAXIMIZATION

Although submodular maximization has been studied over four decades, we introduce only recent results relevant to our work. Badanidiyuru and Vondrák (2014) provided a maximization algorithm for a non-negative, monotone submodular function with l knapsack constraints and a k-system constraint that achieves $\frac{1}{(1+\varepsilon)(k+2l+1)}$-approximation solution. Based on this work and Gupta et al. (2010), Mirzasoleiman et al. (2016) proposed a maximization algorithm called FANTOM under the same constraint in the case when the objective function is not necessarily monotone. Our proposed method is inspired by these two offline algorithms. However, because of uncertainty due to semi-bandit feedback, we need a nontrivial modification. Some algorithms (Sarpatwar et al. 2019; Chekuri, Vondrak, et al. 2010; Chekuri, Vondrák, et al. 2014) achieves better approximation ratios than that of (Badanidiyuru and Vondrák 2014) under narrower classes of constraints (e.g., a matroid + l knapsacks). However, these algorithms are not "fast" because their computational com-

[1] After we submitted this paper to the conference, Li and Shroff (2020) have updated their preprint. They improved the approximation ratio of (Badanidiyuru and Vondrák 2014) to $1/(k + 7l/4 + 1) - \varepsilon$.

[2] We refer to Section 2.1 for the meaning of "fast".

[3] See footnote 1.

plexity is $O(\mathrm{poly}(|\mathcal{N}|))$ with a polynomial of high degree, while that of (Badanidiyuru and Vondrák 2014) is $O(\frac{|\mathcal{N}|}{\varepsilon^2} \ln^2 \frac{|\mathcal{N}|}{\varepsilon})$. For example, the computational complexity of the algorithm provided in (Sarpatwar et al. 2019) is $\widetilde{O}(|\mathcal{N}|^6)$ when $k = 1$. We refer to (Sarpatwar et al. 2019; Mirzasoleiman et al. 2016) for further comparison with respect to an approximation ratio and computational complexity.

2.2 SUBMODULAR BANDIT PROBLEMS

Yue and Guestrin (2011) introduced the linear submodular bandit problem to solve a diversification problem in a retrieval system and proposed a greedy algorithm called LSBGreedy. Later, Yu et al. (2016) considered a variant of the problem, that is, the linear submodular bandit problem with a knapsack constraint and proposed two greedy algorithms called MCSGreedy and CGreedy. L. Chen et al. (2017) generalized the linear submodular bandit problem to an infinite dimensional case, i.e., in the case where the marginal gain of the score function belongs to a *reproducing kernel Hilbert space (RKHS)* and has a bounded norm in the space. Then, they proposed a greedy algorithm called SM-UCB. Recently, Hiranandani et al. (2019) studied a model combining linear submodular bandits with a *cascading model* (Craswell et al. 2008). Strictly speaking, their objective function is not a submodular function. Table 2.2 shows a comparison with other submodular bandit problems with respect to constraints.

Table 1: Comparison of other submodular bandit algorithms with respect to constraints.

Methods	Cardinality	Knapsack	k-system
LSBGreedy	✓		
CGreedy		✓	
SM-UCB	✓		
Our method	✓	✓	✓

3 DEFINITION

In this section, we provide definitions of terminology used in this paper. Throughout this paper, we fix a finite set $\mathcal{N}$ called a ground set that represents the set of the entire news articles in the news article recommendation example.

3.1 SUBMODULAR FUNCTION

In this subsection, we define submodular functions. We refer to (Krause and Golovin 2014) for an introduction to this subject.

We denote by $2^{\mathcal{N}}$ the set of subsets of $\mathcal{N}$. For $e \in \mathcal{N}$ and $S \subseteq \mathcal{N}$, we write $S + e = S \cup \{e\}$. Let $f : 2^{\mathcal{N}} \to \mathbb{R}$ be a set function. We call f a *submodular function* if f satisfies $\Delta f(e|A) \geq \Delta f(e|B)$ for any $A, B \in 2^{\mathcal{N}}$ with $A \subseteq B$ and for any $e \in \mathcal{N} \setminus B$. Here, $\Delta f(e|A)$ is the marginal gain when e is added to A and defined as $f(A+e) - f(A)$. We note that a linear combination of submodular functions with non-negative coefficients is also submodular. A submodular function f on $2^{\mathcal{N}}$ is called monotone if $f(B) \geq f(A)$ for any $A, B \in 2^{\mathcal{N}}$ with $A \subseteq B$. A set function f on $2^{\mathcal{N}}$ is called non-negative if $f(S) \geq 0$ for any $S \subseteq \mathcal{N}$. Although non-monotone submodular functions have important applications (Mirzasoleiman et al. 2016), we consider only non-negative, monotone submodular functions in this study as in the preceding work (Yue and Guestrin 2011; Yu et al. 2016; L. Chen et al. 2017).

3.2 MATROID, k-SYSTEM, AND KNAPSACK CONSTRAINTS

For succinctness, we omit formal definitions of the matroid and k-system. Instead, we introduce examples of matroids and remark that the intersection of k matroids is a k-system. For definitions of these notions, we refer to (Calinescu et al. 2011).

First, we provide an important example of a matroid. Let $\mathcal{N}_i \subseteq \mathcal{N}$ $(i = 1, \dots, n)$ be a partition of $\mathcal{N}$, that is $\mathcal{N}$ is the disjoint union of these subsets. For $1 \leq i \leq n$, we fix a non-negative integer d_i and let $\mathcal{P} = \left\{S \in 2^{\mathcal{N}} \mid |S \cap \mathcal{N}_i| \leq d_i, \forall i\right\}$. Then, the pair $(\mathcal{N}, \mathcal{P})$ is an example of a matroid and called a *partition matroid*. Let d be a non-negative integer and put $\mathcal{U} = \left\{S \in 2^{\mathcal{N}} \mid |S| \leq d\right\}$. Then $(\mathcal{N}, \mathcal{U})$ is a special case of partition matroids and called a *uniform matroid*. Let $(\mathcal{N}, \mathcal{M}_i)$ for $1 \leq i \leq k$ be k matroids, where $\mathcal{M}_i \subseteq 2^{\mathcal{N}}$. The intersection of matroids $(\mathcal{N}, \cap_{i=1}^{k} \mathcal{M}_i)$ is not necessarily a matroid but a *k-system* (or more specifically it is a k-extendible system) (Calinescu et al. 2011; Mestre 2006; Mestre 2015). In particular, any matroid is a 1-system. For a k-system $(\mathcal{N}, \mathcal{I})$ with $\mathcal{I} \subseteq 2^{\mathcal{N}}$ and a subset $S \subseteq \mathcal{N}$, we say that S satisfies the k-system constraint if and only if $S \in \mathcal{I}$. Trivially, a uniform matroid constraint is equivalent to a cardinality constraint.

Next, we provide a definition of knapsack constraint. Let $c : \mathcal{N} \to \mathbb{R}_{>0}$ be a function. For $e \in \mathcal{N}$, we suppose $c(e)$ represents the cost of e. Let $b \in \mathbb{R}_{>0}$ be a budget and $S \subseteq \mathcal{N}$ a subset. We say that S satisfies the *knapsack constraint* with the budget b if $c(S) := \sum_{e \in S} c(e) \leq b$. Without loss of generality, it is sufficient to consider the unit budget case, i.e., $b = 1$.

4 PROBLEM FORMULATION

Throughout this paper, we consider the following intersection of l knapsacks and k-system constraints:

$$c_j(S) \leq 1 \ (1 \leq \forall j \leq l) \quad \text{and } S \in \mathcal{I} \qquad (1)$$

Here for $1 \leq j \leq l$, $c_j : \mathcal{N} \to \mathbb{R}_{>0}$ is a cost and $(\mathcal{N}, \mathcal{I})$ is a k-system.

In this study, we consider the following sequential decision-making process for times steps $t = 1, \dots, T$.

(i) The algorithm selects a list $S_t = \left\{e_t^{(1)}, \dots, e_t^{(m_t)}\right\} \subseteq \mathcal{N}$ satisfying the constraints (1).

(ii) The algorithm receives noisy rewards $y_t^{(1)}, \dots, y_t^{(m_t)}$ as follows:

$$y_t^{(i)} = \Delta f\left(e_t^{(i)} \mid S_t^{(1:i-1)}\right) + \varepsilon_t^{(i)}, \text{ for } i = 1, \dots, m_t,$$

Here f is a submodular function *unknown* to the algorithm, $S_t^{(1:i-1)} = \left\{e_t^{(1)}, \dots, e_t^{(i-1)}\right\}$ and $\varepsilon_t^{(i)}$ is a noise. We regard $S_t^{(1:i-1)}$, $e_t^{(i-1)}$ and $\varepsilon_t^{(i)}$ as random variables. The objective of the algorithm is to maximize the sum of rewards $\sum_{t=1}^{T} f(S_t)$.

Following (Yue and Guestrin 2011), we explain this problem by using the news article recommendation example. In each round, the user scans the list of the recommended items $S_t = \left\{e_t^{(1)}, \dots, e_t^{(m_t)}\right\}$ one-by-one in top-down fashion, where m_t is the cardinality of S_t at round t. We assume that the marginal gain $\Delta f(e_t^{(i)} \mid S_t^{(1:i-1)})$ represents the new information covered by $e_t^{(i)}$ and not covered by $S_t^{(1:i-1)}$. The noisy rewards $y_t^{(1)}, \dots, y_t^{(m_t)}$ are binary random variables and the user likes $e_t^{(i)}$ with probability $\Delta f(e_t^{(i)} \mid S_t^{(1:i-1)})$.

4.1 ASSUMPTIONS ON THE SCORE FUNCTION f

Following (Yue and Guestrin 2011), we assume that there exist d known submodular functions $f_1, \dots, f_d$ on $2^{\mathcal{N}}$ that are linearly independent and the objective submodular function f can be written as a linear combination $f = \sum_{i=1}^{d} w_i f_i$, where the coefficients $w_1, \dots, w_d$ are non-negative and *unknown* to the algorithm. We fix a parameter $B > 0$ and assume that $\sqrt{\sum_{i=1}^{d} w_i^2} \leq B$. We also assume that for some $A > 0$, the L^2-norm of vector $[\Delta f_i(e \mid S)]_{i=1}^{d}$ is bounded above by $\sqrt{A}$ for any $e \in \mathcal{N}$ and $S \in 2^{\mathcal{N}}$.

We note that this can be generalized to an infinite dimensional case as in (L. Chen et al. 2017). We discuss this setting more in detail in the supplemental material and provide a theoretical result in this setting.

4.2 ASSUMPTIONS ON NOISE STOCHASTIC PROCESS

We assume that there exists $m \in \mathbb{Z}_{>0}$ such that $m_t \leq m$ for all t and consider the lexicographic order on the set $\{(t, i) \mid t = 1, \dots,\ 1 \leq i \leq m\}$, i.e., $(t, i) \leq (t', i')$ if and only if either $t < t'$ or $t = t'$ and $i \leq i'$. Then, we can identify the set with the set of natural numbers (as ordered sets) and can regard $\{\varepsilon_t^{(i)}\}_{t,i}$ as a sequence. We assume that the stochastic process $\left\{\varepsilon_t^{(i)}\right\}_{t,i}$ is *conditionally R-sub-Gaussian* for a fixed constant $R \geq 0$, i.e., $\mathbf{E}\left[\exp\left(\xi \varepsilon_t^{(i)}\right) \mid \mathcal{F}_{t,i}\right] \leq \exp\left(\frac{\xi^2 R^2}{2}\right)$, for any (t, i) and any $\xi \in \mathbb{R}$. Here, $\mathcal{F}_{t,i}$ is the σ-algebra generated by $\left\{S_u^{(1:j)} \mid (u, j) < (t, i)\right\}$ and $\left\{\varepsilon_u^{(j)} \mid (u, j) < (t, i)\right\}$. This is a standard assumption on the noise sequence (Chowdhury and Gopalan 2017; Abbasi-Yadkori et al. 2011). For example, if $\{\varepsilon_t^{(i)}\}$ is a martingale difference sequence and $|\varepsilon_t^{(i)}| \leq R$ or $\{\varepsilon_t^{(i)}\}$ is conditionally Gaussian with zero mean and variance R^2, then the condition is satisfied (Lattimore and Szepesvári 2019).

4.3 APPROXIMATION REGRET

As usual in the combinatorial bandit problem, we evaluate bandit algorithms by a regret called α-*approximation regret* (or α-*regret* in short), where $\alpha \in (0, 1)$. The approximation regret is necessary for meaningful evaluation. Even if the submodular function f is completely known, it has been proved that no algorithm can achieve the optimal solution by evaluating f in polynomial time (Nemhauser and Wolsey 1978).

We denote by OPT the optimal solution, i.e., $OPT = \operatorname{argmax}_S f(S)$, where S runs over $2^{\mathcal{N}}$ satisfying the constraint (1). We define the α-regret as follows:

$$\mathrm{Reg}_{\alpha}(T) = \sum_{t=1}^{T} \left\{\alpha f(OPT) - f(S_t)\right\}.$$

This definition is slightly different from that given in (Yue and Guestrin 2011) because our definition does not include noise as in (Chowdhury and Gopalan 2017). In either case, one can prove a similar upper bound. For the proof in the cardinality constraint case, we refer to Lemma 4 in the supplemental material of (Yue and Guestrin 2011).

In this study, we take the same approximation ratio $\alpha = \frac{1}{(1+\varepsilon)(k+2l+1)}$ as that of a fast algorithm in the offline setting (Badanidiyuru and Vondrák 2014, Theorem 6.1). As mentioned in Section 2, there exist offline algorithms that achieve better approximation ratios than above, but they have high computational complexity. Later, we remark that our proposed method is also "fast".

5 ALGORITHM

In this section, following (Yue and Guestrin 2011; Yu et al. 2016), we first define a *UCB score* of the marginal gain $\Delta f(e \mid S)$ and introduce a *modified UCB score*. With a UCB score, one can balance the exploitation and exploration tradeoff with bandit feedback. Then, we propose a non-greedy algorithm (Algorithm 2) that adaptively focuses on the UCB score and modified UCB score.

5.1 UCB SCORES

For $e \in \mathcal{N}$ and $S \in 2^{\mathcal{N}}$, we define a column vector $x(e \mid S)$ by $(\Delta f_i(e \mid S))_{i=1}^{d} \in \mathbb{R}^d$ and put $x_t^{(i)} = x\left(e_t^{(i)} \mid S_t^{(1:i-1)}\right)$. Here, we use the same notation as in Section 4. We define $b_t, w_t \in \mathbb{R}^d$ and $M_t \in \mathbb{R}^{d \times d}$ as follows:

$$b_t := \sum_{s=1}^{t} \sum_{i=1}^{m_s} y_s^{(i)} x_s^{(i)},$$

$$M_t := \lambda I + \sum_{s=1}^{t} \sum_{i=1}^{m_s} x_s^{(i)} \otimes x_s^{(i)}, \quad w_t := M_t^{-1} b_t,$$

Here, $\lambda > 0$ is a parameter of the model and for a column vector $x \in \mathbb{R}^d$, we denote by $x \otimes x \in \mathbb{R}^{d \times d}$ the Kronecker product of x and x.

For $e \in \mathcal{N}$ and $S \in 2^{\mathcal{N}}$, we define $\mu(e \mid S) := w_t \cdot x(e \mid S)$ and $\sigma(e \mid S) := x(e|S)^{\mathrm{T}} M_t^{-1} x(e|S$. Then, we define a UCB score of the marginal gain by

$$\mathrm{ucb}_t(e \mid S) = \mu_{t-1}(e \mid S) + \beta_{t-1}\sigma_{t-1}(e \mid S),$$

and a modified UCB score by $\mathrm{ucb}_t(e \mid S)/c(e)$. Here, $\beta_t := B + R\sqrt{\ln\det(\lambda^{-1}M_t) + 2 + 2\ln(1/\delta)}$ and $c(e) := \sum_{j=1}^{l} c_j(e)$. It is well-known that $\mathrm{ucb}_t(e \mid \phi, S)$ is an upper confidence bound for $\Delta f_{\phi}(e \mid S)$. More precisely, we have the following result.

Proposition 1. *We assume there exists $m \in \mathbb{Z}_{\geq 1}$ such that $m_t \leq m$ for all $1 \ \leq t \leq T$. We also assume that $1 < \lambda/A \leq 1 + 2/(mT)$. Then, with probability at least $1 - \delta$, the following inequality holds:*

$$|\mu_{t-1}(e \mid S) - \Delta f(e \mid S)| \leq \beta_{t-1}\sigma_{t-1}(e \mid S),$$

for any t, S, and e.

Proposition 1 follows from the proof of (Chowdhury and Gopalan 2017, Theorem 2). We note that this theorem is a more generalized result than the statement above (they do not assume that the objective function is linear but belongs to an RKHS). In the linear kernel case, an equivalent result to Proposition 1 was proved in (Abbasi-Yadkori et al. 2011).

We also define the UCB score for a list $S = \left\{e^{(1)}, \ldots, e^{(m)}\right\}$ by $\mathrm{ucb}_t(S) = \mu_{t-1}(S) + 3\beta_{t-1}\sigma_{t-1}(S)$. Here $\mu_t(S)$ and $\sigma_t(S)$ are defined as $\sum_{i=1}^{m} \mu_t(e^{(i)} \mid S^{(1:i-1)})$ and $\sum_{i=1}^{m} \sigma_t(e^{(i)} \mid S^{(1:i-1)})$, respectively. The factor 3 in the definition of $\mathrm{ucb}_t(S)$ is due to a technical reason as clarified by the proof of Lemma 1 in the supplemental material.

5.2 AFSM-UCB

```
Input  : Threshold $\rho$, round $t$
Output: A list $S$ satisfying the constraints (1)
Set $S = \emptyset$, $i = 1$
while True do
    $\mathcal{N}_S =$
      $\{e \in \mathcal{N} \mid S + e$ satisfies the constraint (1)$\}$.
    $\mathcal{N}_{S,\geq\rho} = \left\{e \in \mathcal{N}_S \mid \substack{\mathrm{ucb}_t(e|S)/c(e)\geq\rho \text{ and} \\ \mathrm{ucb}_t(e|\emptyset)/c(e)\geq\rho}\right\}$.
    if $\mathcal{N}_{S,\geq\rho} = \emptyset$ then
        break;
    $e_i = \mathrm{argmax}_{e\in\mathcal{N}_{S,\geq\rho}} \mathrm{ucb}_t(e \mid S)$.
    Add $e_i$ to $S$. Set $i \leftarrow i + 1$
end
Return $S$
```

Algorithm 1: GM-UCB (Sub-algorithm)

```
Input  : Parameters $B, R, \lambda, \delta, \nu, \nu', \varepsilon$
Output: A list $S$ satisfying the constraints (1)
for $t = 1, \ldots, T$ do
    $U = \emptyset$, $r = \frac{2}{k+2l+1}$, $\rho = r(1+\varepsilon)^{-1}\nu$
    while $\rho \leq r\nu'|\mathcal{N}|$ do
        $S = \mathrm{Algorithm1}(\rho, t)$
        Add $S$ to $U$
        Set $\rho \leftarrow (1+\epsilon)\rho$
    end
    Select $S_t = \mathrm{argmax}_{S\in U} \mathrm{ucb}_t(S)$
    Receive rewards $y_t^{(1)}, \ldots, y_t^{(m_t)}$
end
```

Algorithm 2: AFSM-UCB (Main Algorithm)

In this subsection, we propose a UCB-type algorithm for our problem. We call our proposed method AFSM-UCB and its pseudo code is outlined in Algorithm 2. Algorithm 2 calls a sub-algorithm called GM-UCB (an algorithm that Greedily selects elements with Modified UCB scores larger than a threshold, outlined in Algorithm 1). Algorithm 1 takes a threshold ρ as a parameter and returns a list of elements satisfying the constraint 1. Algorithm 1 selects elements greedily from the elements whose modified UCB scores $\mathrm{ucb}_t(e \mid S)/c(e)$ and $\mathrm{ucb}_t(e \mid \emptyset)/c(e)$ are larger or equal to the threshold ρ. If the threshold ρ is small, then this algorithm is almost the same as a greedy algorithm, such as LSBGreedy

(Yue and Guestrin 2011). If the threshold ρ is large, then the elements with large modified UCB scores will be selected. Thus, the threshold ρ controls the importance of the standard and modified scores. The main algorithm 2 calls Algorithm 1 repeatedly by changing the threshold ρ and returns a list with the largest UCB score. We prove that there exists a good list among these candidates lists.

As remarked before, Algorithm 2 is inspired by submodular maximization algorithms in the the offline setting (Badanidiyuru and Vondrák 2014; Mirzasoleiman et al. 2016). However, we need a nontrivial modification since the diminishing return property does not hold for $\mathrm{ucb}_t(e \mid S)$ unlike the marginal gain $\Delta f(e \mid S)$. We note that $\mathrm{ucb}_t(e \mid S)$ can be large not only when the estimated value of $\Delta f(e \mid S)$ is large but also if the uncertainty in adding e to S is high. Therefore, we need additional filter conditions to ensure that e is a "good" element. Natural candidates for the condition are that $\mathrm{ucb}_t(e \mid S^{(1:i)})/c(e) \geq \rho$ for some indices i. In Algorithm 2, we require $\mathrm{ucb}_t(e \mid \emptyset)/c(e) \geq \rho$ in addition to $\mathrm{ucb}_t(e \mid S)/c(e) \geq \rho$.

In the algorithm, we introduce parameters ν and ν'. The parameter ν (resp. ν') is used for defining the initial (resp. terminal) value of the threshold ρ. In the next section, for a theoretical guarantee, we assume that $\nu \leq \max_{e \in \mathcal{N}} f(\{e\}) \leq \nu'$. If the upper bound of the reward is known, then we can take ν' as the known upper bound. In practice, it is plausible that most users are interested in at least one item in the entire item set $\mathcal{N}$, which implies $\max_{e \in \mathcal{N}} f(\{e\})$ is not very small. In addition, the number of iterations in the while loop in Algorithm 2 is given by $O(\ln(\nu'|\mathcal{N}|/\nu))$. Therefore, taking a very small ν does not increase the number of iterations as much.

5.3 COMPUTATIONAL COMPLEXITY

We discuss the computational complexity of Algorithm 2 and that of existing methods. We consider a greedy algorithm by applying LSBGreedy to our problem; i.e., we consider a greedy algorithm that selects the element with the largest UCB score until the constraint is satisfied. By abuse of terminology, we call this algorithm LSBGreedy. Similarly, when we apply CGreedy (resp. MCSGreedy) to our problem, we also call this algorithm CGreedy (resp. MCSGreedy). In each round, the expected number of times to compute $\mathrm{ucb}_t(e \mid S)$ in Algorithm 2 is given by $O(m|\mathcal{N}|\ln(\nu'|\mathcal{N}|/\nu)/\ln(1+\epsilon))$, while that of LSBGreedy is given by $O(m|\mathcal{N}|)$. The computational complexity of MCSGreedy and CGreedy is given as $O(|\mathcal{N}|^3)$ and $O(m|\mathcal{N}|)$ respectively. Therefore, ignoring unimportant parameters, our algorithms incur an additional factor $\ln|\mathcal{N}|/\ln(1+\varepsilon)$ compared to that of LSBGreedy and CGreedy.

6 MAIN RESULTS

The main challenge of this paper is to provide a strong theoretical result for AFSM-UCB. In this section, under the assumptions stated as in the previous section, we provide an upper bound for the approximation regret of AFSM-UCB and give a sketch of the proof. We also show that existing greedy methods incur linear approximation regret in the worst case for our problem.

6.1 STATEMENT OF THE MAIN RESULTS

THEOREM 1. *Let the notation and assumptions be as previously mentioned. We also assume that $1 < \lambda/A \leq 1+2/(mT)$. We let $\alpha = \frac{1}{(1+\varepsilon)(k+2l+1)}$. Then, with probability at least $1-\delta$, the proposed algorithm achieves the following α-regret bound:*

$$\mathrm{Reg}_\alpha(T) \leq 4A\beta_T\sqrt{2(mT+2)\ln\det(\lambda^{-1}M_T)}.$$

In particular, ignoring A, B, R, we have $\mathrm{Reg}_\alpha(T) = O(d\sqrt{mT}\ln\frac{mT}{\delta})$ with probability at least $1-\delta$.

REMARK 1.

1. There is a tradeoff between the approximation ratio and computational complexity. As discussed in Section 5.3, the computational complexity of the algorithm is given as $O(m|\mathcal{N}|\ln(|\mathcal{N}|)/\ln(1+\epsilon))$ in each round, while the approximation of the algorithm is given as $\frac{1}{(1+\varepsilon)(k+2l+1)}$.
2. We assume the score function f is a linear combination of known submodular functions. We can relax the assumption to the case when the function $(e, S) \to \Delta f(e|S)$ belongs to an RKHS and has a bounded norm in the space as in (L. Chen et al. 2017). We discuss this setting more in detail and provide a generalized result in the supplemental material.

In the setting of (Yue and Guestrin 2011; Yu et al. 2016), greedy methods have good theoretical properties. However, we show that for any $\alpha > 0$, these greedy methods incur linear α-regret in the worst case for our problem. We denote by $\mathrm{Reg}_{\alpha,\mathrm{MCS}}(T)$ and $\mathrm{Reg}_{\alpha,\mathrm{LSB}}(T)$ the α-regret of MCSGreedy and that of LSBGreedy, respectively. Then the following proposition holds.

PROPOSITION 2. *For any $\alpha > 0$, there exists cost c_1, k-system $\mathcal{I}$, a submodular function f, $T_0 > 0$ and a constant $C > 0$ such that with probability at least $1-\delta$,*

$$\mathrm{Reg}_{\alpha,\mathrm{MCS}}(T) > CT,$$

for any $T > T_0$. Moreover, the same statement holds for $\mathrm{Reg}_{\alpha,\mathrm{LSB}}(T)$.

We provide the proof in the supplemental material.

6.2 SKETCH OF THE PROOF OF THEOREM 1

We provide a sketch of the proof of Theorem 1 and provide a detailed and generalized proof in the supplemental material. Throughout the proof, we fix the event $\mathcal{F}$ on which the inequality in Proposition 1 holds.

We evaluate the solution S_t by AFSM-UCB in each round t. The following is a key result for our proof of Theorem 1.

PROPOSITION 3. *Let $C \subseteq \mathcal{N}$ be any set satisfying the constraint* (1)*. Let S be a set returned by GM-UCB at time step t. Then, on the event $\mathcal{F}$, we have* $f(S) + 2\beta_{t-1}\sigma_{t-1}(S) \geq \min\left\{\frac{\rho}{2}, \frac{1}{k+1}f(S \cup C) - \frac{l\rho}{k+1}\right\}.$

sketch of proof. This can be proved in a similar way to the proof of (Badanidiyuru and Vondrák 2014, Theorem 6.1) or (Mirzasoleiman et al. 2016, Theorem 5.1). However, because of uncertainty and lack of diminishing property of the UCB score, we need further analysis. We divide the proof into two cases.

Case One. This is the case when GM-UCB terminates because there exists an element e such that $\mathrm{ucb}_t(e \mid S) \geq \rho c(e)$ and $\mathrm{ucb}_t(e \mid \emptyset) \geq \rho c(e)$, but any element e satisfying $\mathrm{ucb}_t(e \mid S), \mathrm{ucb}_t(e \mid \emptyset) \geq \rho c(e)$ does not satisfy the knapsack constraints, i.e., $c_j(S + e) > 1$ for some $1 \leq j \leq l$. We fix an element e' satisfying $\mathrm{ucb}_t(e' \mid S), \mathrm{ucb}_t(e' \mid \emptyset) \geq \rho c(e')$. Because any element of S has enough modified UCB score, by Proposition 1, we have $f(S) + 2\beta_{t-1}\sigma_{t-1}(S) \geq \rho c(S)$. By the definition of e', we also have $\mathrm{ucb}_t(e' \mid \emptyset) \geq \rho c(e')$. Because $f(S) + 2\beta_{t-1}\sigma_{t-1}(S) \geq \mathrm{ucb}_t(e' \mid \emptyset) \geq \rho c(e')$ and $S + e'$ does not satisfy the knapsack constraint, we have $f(S) + 2\beta_{t-1}\sigma_{t-1}(S) \geq \frac{\rho}{2}c(S + e') \geq \rho/2$.

Case Two. This is the case when GM-UCB terminates because for any element e satisfying $\mathrm{ucb}_t(e \mid S), \mathrm{ucb}_t(e \mid \emptyset) \geq \rho c(e)$, e satisfies the knapsack constraints but $S + e$ does not satisfy the k-system constraint. We note that this case includes the case when there does not exist an element e satisfying $\mathrm{ucb}_t(e \mid S), \mathrm{ucb}_t(e \mid \emptyset) \geq \rho c(e)$.

We define a set $C_{<\rho}$ as

$$\left\{e \in C \mid \exists i \text{ such that } \mathrm{ucb}_t(e \mid S^{(1:i)}) < \rho c(e)\right\},$$

and $C_{\geq\rho} = C \setminus C_{<\rho}$. Let $e \in C_{<\rho}$. Then on the event $\mathcal{F}$, by Proposition 1 and submodularity, we have

$$\Delta f(C_{<\rho} \mid S) \leq \sum_{e \in C_{<\rho}} \Delta f(e \mid S) \leq \sum_{e \in C_{<\rho}} \rho c(e) \leq l\rho. \tag{2}$$

Next, we consider $C_{\geq\rho}$. Running the greedy algorithm (with respect to the UCB score) on $S \cup C_{\geq\rho}$ under only the k-system constraint, we obtain S by the assumption of this case. Then, it can be proved that $f(S) + 2\beta_{t-1}\sigma_{t-1}(S) \geq \frac{1}{k+1}f(S \cup C_{\geq\rho})$. We note that this is a variant of the result proved in (Calinescu et al. 2011, Appendix B). By this inequality, inequality (2), and submodularity, we can derive the desired result. □

Using Proposition 3, we can bound the approximation regret above by the sum of uncertainty $\beta_{t-1}\sigma_{t-1}(S_t)$. Because the algorithm selects S_t and obtain feedbacks for S_t, the sum of uncertainty can be bounded above by a sub-linear function of T.

7 EXPERIMENTAL ANALYSIS

In this section, we empirically evaluate our methods by a synthetic dataset that simulates an environment for news article recommendation and two real-world datasets (MovieLens100K (Grouplens 1998) and the Million Song Dataset (Bertin-Mahieux et al. 2011)).

We compare our proposed algorithm to the following baselines:

- RANDOM. In each round, this algorithm selects elements uniform randomly until no element satisfies the constraints.
- LSBGreedy. This was proposed in (Yue and Guestrin 2011) to solve the submodular bandit problem under a cardinality constraint. In the linear kernel case, SM-UCB (L. Chen et al. 2017) is equivalent to LSBGreedy.
- CGreedy. This is an algorithm for a submodular bandit problem under a knapsack constraint and was proposed in (Yu et al. 2016). They also proposed an algorithm called MCSGreedy. However because MCSGreedy is computationally expensive (in each round it calls functions $f_1, \ldots, f_d$ for $O(|\mathcal{N}|^3)$ times) and their experimental results show that both algorithms have a similar empirical performance, we do not add MCSGreedy to the baselines.

In Proposition 2, we showed that these greedy algorithms incur linear approximation regret in the worst case. However, even without theoretical guarantee, it is empirically known that a greedy algorithm achieve a good experimental performance. In this section, we demonstrate that our algorithm outperforms these greedy algorithms under various combinations of constraints. As a special case, such constraints include the case when there is a sufficiently large budget for knapsack constraints and the

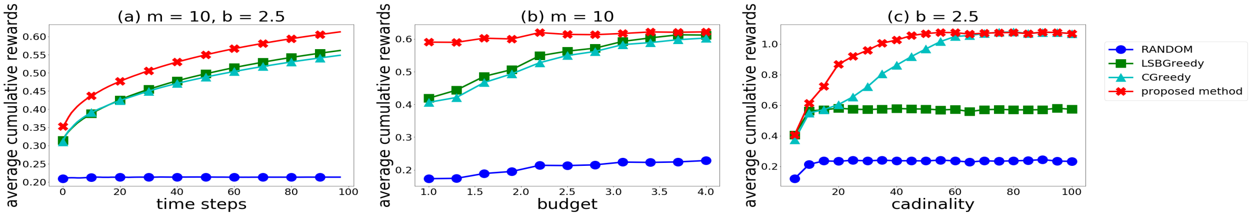

Figure 1: Cumulative average rewards on the synthetic news article recommendation dataset

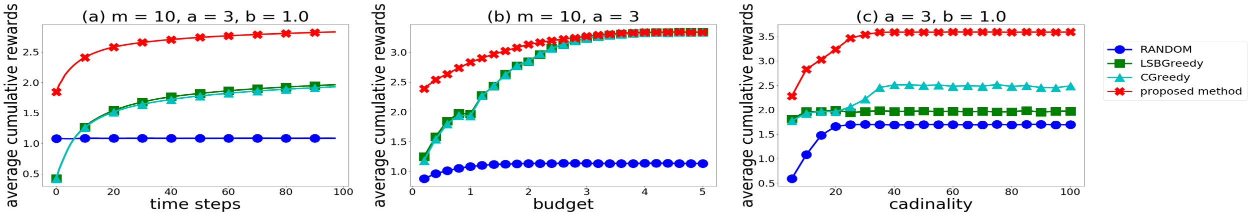

Figure 2: Cumulative average rewards on the MovieLens dataset

case when the k-system constraint is sufficiently mild. The greedy algorithms are algorithms for such cases. We also show that our proposed method performs no worse than the baselines even in these cases.

As in the preceding work (Yue and Guestrin 2011), we assume the score function f is a linear combination of known probabilistic coverage functions. We assume there exists a set of topics (or genres) $\mathcal{G}$ with $|\mathcal{G}| = d$ and for each item $e \in \mathcal{N}$, there is a feature vector $x(e) := (P_g(e))_{g \in \mathcal{G}} \in \mathbb{R}^d$ that represents the information coverage on different genres. For each genre g, we define the probabilistic coverage function $f_g(S)$ by $1 - \prod_{e \in S}(1 - P_g(e))$ and we assume $f = \sum_i w_i f_i$ with unknown linear coefficients w_i. The vector $w := [w_1, \dots, w_d]$ represents user preference on genres. We assume that the noisy rewards $y_t^{(i)}$ are sampled by $y_t^{(i)} \sim \mathrm{Ber}\left(\Delta f(e_t^{(i)} \mid S_t^{(1:i-1)})\right)$. Below, we define these feature vectors $x(e)$, w, and constraints explicitly. We note that in the experiments, we use an un-normalized knapsack constraint $c(S) \leq b$. In the following experiments, using 100 users (100 vectors w), we compute cumulative average rewards for each algorithm. When taking the average, we repeated this experiment 10 times for each user.

7.1 NEWS ARTICLE RECOMMENDATION

In this synthetic dataset, we assume $d = 15$ and $|\mathcal{N}| = 1000$. We define $x(e)$ and costs for a knapsack constraint in a similar manner in (Yu et al. 2016). We sample each entry of $x(e)$ from two types of uniform distributions. We assume that for each item e, the number of genres that have high information coverage is limited to two. More precisely, we randomly select two indices of $x(e)$ and sample entries from $U(0.5, 0.8)$ and sample other entries from $U(0.0, 0.01)$. We generate 100 user preference vectors w in a similar way to $x(e)$. We also sample the costs of items uniform randomly from $U(0.0, 1.0)$. In this dataset, we consider the intersection of a cardinality constraint and a knapsack constraint. The result is shown in Figure 1.

7.2 MOVIE RECOMMENDATION

We perform a similar experiment in (Mirzasoleiman et al. 2016) but with a semi-bandit feedback. In MovieLens100K, there are 943 users and 1682 movies. We take $\mathcal{N}$ as the set of 1682 movies in the dataset. There are $d = 18$ genres in this dataset. First, we fill the ratings for all the user-item pairs using matrix factorization (Koren et al. 2009) and we normalized the ratings r so that $r \in [0, 1]$. For each movie $e \in \mathcal{N}$, we denote by $r_e \in [0, 1]$ the mean of the ratings of the movie for all users. We define $P(g \mid e) = r_e / |\mathcal{G}_e|$ if $g \in \mathcal{G}_e$, otherwise we define $P(g \mid e) = 0$. We normalize $P(g \mid e)$ as previously mentioned, because if $w_i = 1$ for all i, then we have $P(\{e\}) = r_e$.

We define a similar knapsack, cardinality, and matroid constraints to those of (Mirzasoleiman et al. 2016). For $e \in \mathcal{N}$, the cost $c(e)$ is defined as $c(e) = F_{\mathrm{Beta}(10,2)}(r_e)$, where $F_{\mathrm{Beta}(10,2)}$ is the cumulative distribution function of the $\mathrm{Beta}(10, 2)$. For a budget $b \in \mathbb{R}_{>0}$, we consider a knapsack constraint $c(S) \leq b$. The beta distribution lets us differentiate the highly rated movies from those with lower ratings (Mirzasoleiman et al. 2016). We

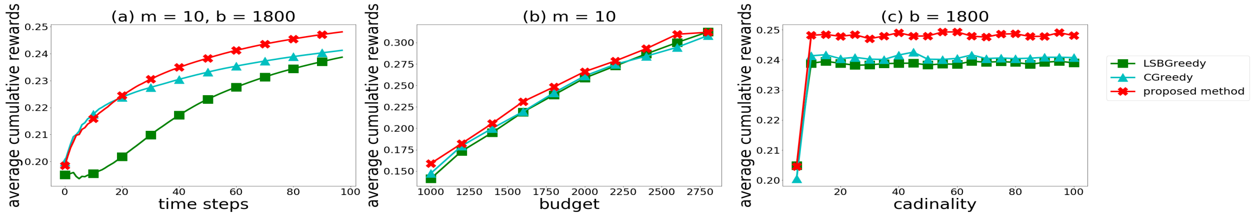

Figure 3: Cumulative average rewards on the Million Song Dataset

generate 100 user preference vectors w in a similar way to the news article recommendation example. In this dataset, we consider the following constraints on genres in addition to the knapsack $c(S) \leq b$ and cardinality $|S| \leq m$ constraints, There are k genres in MovieLens100K, where $k = d = 18$. For each genre g, we fix a non-negative integer a and consider the constraint $|\{e \in S \mid e \text{ has genre } g\}| \leq a$ for $S \subseteq \mathcal{N}$. This can be regarded as a partition matroid constraint. Therefore, the intersection of the constraints for all genres is a k-system constraint. One can prove that the intersection of this k-system constraint and a cardinality constraint is also a k-system constraint. The results are displayed in Figure 2 in the case of the matroid limit $a = 3$.

7.3 MUSIC RECOMMENDATION

From the Million Song Dataset, we select 1000 most popular songs and 30 most popular genres. Thus, we have $|\mathcal{N}| = 1000$ and $d = 30$. For active 100 users, we compute $P_g(e)$ and user preference vector w in almost the same way as $\overline{w}(e, g)$ and θ^* in (Hiranandani et al. 2019) respectively. They assume that a user likes a song e if the user listened to the song at least five times, however, we assume that a user likes the song if the user listened to the song at least two times. We consider the intersection of a cardinality and a knapsack constraint $c(S) \leq b$. We define a cost c for the knapsack constraint by the length (in seconds) of the song in the dataset. The costs represent the length of time spent by users before they decide to listen to the song and we assume that it is proportional to the length of the song [4]. The results are displayed in Figure 3. We do not show the performance of RANDOM in the figure since it achieves only very low rewards.

7.4 RESULTS

In Figures 1a, 2a, 3a, we plot the cumulative average rewards for each algorithm up to time step $T = 100$. In Figures 1b, 2b, and, 3b (resp. 1c, 2c, and, 3c), we show the cumulative average rewards at the final round by changing the budget b (resp. by changing the cardinality limit m) and fixing the cardinality limit m (resp. fixing the budget b). These results shows that overall our proposed method outperforms the baselines. We note that Figure 3 shows different tendency as compared to other datasets since popular items in the Million Song Dataset have high information coverage for multiple genres and about 47 % of the items have low information coverage (less than 0.01) for all genres. Figures 1b, 2b, and 3b also show the results for the case when the budget is sufficiently large. This is the case when LSBGreedy performs well and our experimental results show that even in this case, our method have comparable performance to greedy algorithms. Moreover, Figures 1c, 2c, and 3c also show the results in the case when the cardinality constraints are sufficiently mild. In this case, CGreedy performs well since the constraints are almost same as a knapsack constraint. The experimental results show that our method tends to have better performance than that of CGreedy even in this case.

8 CONCLUSIONS

In this study, motivated by diversified retrieval considering cost of items, we introduced the submodular bandit problem under the intersection of a k-system and knapsack constraints. Then, we proposed a non-greedy algorithm to solve the problem and provide a strong theoretical guarantee. We demonstrated our proposed method outperforms the greedy baselines using synthetic and two real-world datasets.

A possible generalization of this work is a generalization to the full bandit setting. In this setting, a leaner observes only a value $f(S_t) + \epsilon$ in each round. Since it needs much work to derive a theoretical guarantee, we leave this setting for future work.

References

Abbasi-Yadkori, Yasin, Dávid Pál, and Csaba Szepesvári (2011). “Improved algorithms for linear stochastic

[4] We can also assume that users listen to the song and give feedbacks later.

bandits”. In: *Advances in Neural Information Processing Systems*, pp. 2312–2320.

Badanidiyuru, Ashwinkumar and Jan Vondrák (2014). “Fast algorithms for maximizing submodular functions”. In: *ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, pp. 1497–1514.

Bertin-Mahieux, Thierry et al. (2011). “The Million Song Dataset”. In: *International Conference on Music Information Retrieval (ISMIR 2011)*.

Calinescu, Gruia et al. (2011). “Maximizing a monotone submodular function subject to a matroid constraint”. In: *SIAM Journal on Computing* 40.6, pp. 1740–1766.

Chekuri, Chandra, Jan Vondrak, and Rico Zenklusen (2010). “Dependent randomized rounding via exchange properties of combinatorial structures”. In: *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. IEEE, pp. 575–584.

Chekuri, Chandra, Jan Vondrák, and Rico Zenklusen (2014). “Submodular function maximization via the multilinear relaxation and contention resolution schemes”. In: *SIAM Journal on Computing* 43.6, pp. 1831–1879.

Chen, Lin, Andreas Krause, and Amin Karbasi (2017). “Interactive submodular bandit”. In: *Advances in Neural Information Processing Systems*, pp. 141–152.

Chen, Wei, Yajun Wang, and Yang Yuan (2013). “Combinatorial multi-armed bandit: General framework and applications”. In: *International Conference on Machine Learning*, pp. 151–159.

Chowdhury, Sayak Ray and Aditya Gopalan (2017). “On kernelized multi-armed bandits”. In: *International Conference on Machine Learning*. supplementary material, pp. 844–853.

Craswell, Nick et al. (2008). “An experimental comparison of click position-bias models”. In: *International Conference on Web Search and Data Mining*. ACM, pp. 87–94.

Grouplens (1998). “MovieLens100k dataset”. In: URL: `https://grouplens.org/datasets/movielens/100k/`.

Gupta, Anupam et al. (2010). “Constrained non-monotone submodular maximization: Offline and secretary algorithms”. In: *International Workshop on Internet and Network Economics*. Springer, pp. 246–257.

Hiranandani, Gaurush et al. (2019). “Cascading linear submodular bandits: Accounting for position bias and diversity in online learning to rank”. In: *Uncertainty in Artificial Intelligence*.

Koren, Yehuda, Robert Bell, and Chris Volinsky (2009). “Matrix factorization techniques for recommender systems”. In: *Computer* 8, pp. 30–37.

Krause, Andreas and Daniel Golovin (2014). *Submodular function maximization.* In Tractability: Practical Approaches to Hard Problems, Cambridge University Press.

Lattimore, Tor and Csaba Szepesvári (2019). *Bandit algorithms*. Cambridge University Press, Forthcoming. URL: `https://tor-lattimore.com/downloads/book/book.pdf`.

Li, Wenxin and Ness Shroff (2020). *Efficient algorithms and lower bound for submodular maximization*. arXiv preprint, arXiv:1804.08178v3.

Mestre, Julián (2006). “Greedy in approximation algorithms”. In: *European Symposium on Algorithms*. Springer, pp. 528–539.

– (2015). “On the intersection of independence systems”. In: *Operations Research Letters* 43.1, pp. 7–9.

Mirzasoleiman, Baharan, Ashwinkumar Badanidiyuru, and Amin Karbasi (2016). “Fast constrained submodular maximization: Personalized data summarization.” In: *International Conference on Machine Learning*, pp. 1358–1367.

Nemhauser, George L and Laurence A Wolsey (1978). “Best algorithms for approximating the maximum of a submodular set function”. In: *Mathematics of Operations Research* 3.3, pp. 177–188.

Radlinski, Filip, Robert Kleinberg, and Thorsten Joachims (2008). “Learning diverse rankings with multi-armed bandits”. In: *International Conference on Machine Learning*, pp. 784–791.

Sarpatwar, Kanthi K, Baruch Schieber, and Hadas Shachnai (2019). “Constrained submodular maximization via greedy local search”. In: *Operations Research Letters* 47.1, pp. 1–6.

Yu, Baosheng, Meng Fang, and Dacheng Tao (2016). “Linear submodular bandits with a knapsack constraint”. In: *AAAI Conference on Artificial Intelligence*.

Yue, Yisong and Carlos Guestrin (2011). “Linear submodular bandits and their application to diversified retrieval”. In: *Advances in Neural Information Processing Systems*, pp. 2483–2491.

Ziegler, Cai-Nicolas et al. (2005). “Improving recommendation lists through topic diversification”. In: *International Conference on World Wide Web*. ACM, pp. 22–32.