DART: Adaptive Accept Reject Algorithm for Non-Linear Combinatorial Bandits

Mridul Agarwal[1], **Vaneet Aggarwal** [1], **Abhishek Kumar Umrawal**[1], **Chris Quinn** [2]

[1] Purdue University
[2] Iowa State University
agarw180@purdue.edu, vaneet@purdue.edu, aumrawal@purdue.edu, cjquinn@iastate.edu

Abstract

We consider the bandit problem of selecting K out of N arms at each time step. The reward can be a non-linear function of the rewards of the selected individual arms. The direct use of a multi-armed bandit algorithm requires choosing among $\binom{N}{K}$ options, making the action space large. To simplify the problem, existing works on combinatorial bandits typically assume feedback as a linear function of individual rewards. In this paper, we prove the lower bound for top-K subset selection with bandit feedback with possibly correlated rewards. We present a novel algorithm for the combinatorial setting without using individual arm feedback or requiring linearity of the reward function. Additionally, our algorithm works on correlated rewards of individual arms. Our algorithm, aDaptive Accept RejecT (DART), sequentially finds good arms and eliminates bad arms based on confidence bounds. DART is computationally efficient and uses storage linear in N. Further, DART achieves a regret bound of $\tilde{\mathcal{O}}(K\sqrt{KNT})$ for a time horizon T, which matches the lower bound in bandit feedback up to a factor of $\sqrt{\log 2NT}$. When applied to the problem of cross selling optimization and maximizing the mean of individual rewards, the performance of the proposed algorithm surpasses that of state-of-the-art algorithms. We also show that DART significantly outperforms existing methods for both linear and non-linear joint reward environments.

Introduction

The problem of finding the best K out of N items to optimize a possibly non-linear function of reward of each item arises in a number of settings. For example, in the problem of erasure-coded storage (Xiang et al. 2016), the agent chooses K out of N servers to obtain the content for each request; the final reward is the negative of the time taken by the slowest server. A recommendation system agent may present a list of K items out of N items to user for a non-zero reward only if the user selects an item (Kveton et al. 2015a) from the list. Similarly, in cross-selling item selection, a retailer creates a bundle with K items, and the joint reward is a quadratic function of the selected items' individual rewards (Raymond Chi-Wing Wong, Ada Wai-Chee Fu, and Wang 2003). The problem of a daily advertising campaign is characterized by a set of sub-campaigns where the aggregate reward is the sum of the rewards of sub-campaigns (Zhang et al. 2012; Nuara et al. 2018). Combinatorial Multi-Armed Bandit (CMAB) algorithms can solve these problems in an online manner. For many CMAB algorithms, we can bound the regret, that is the loss incurred from accidentally selecting sub-optimal sets some of the time. We aim to find a space and time efficient CMAB algorithm that minimizes cumulative regret.

Existing algorithms for $K = 1$ that use Upper Confidence Bound (UCB) or Bayesian resampling methods (Auer 2002; Auer, Cesa-Bianchi, and Fischer 2002; Auer and Ortner 2010; Thompson 1933; Agrawal and Goyal 2012; Gopalan, Mannor, and Mansour 2014) can bound the regret by $\tilde{\mathcal{O}}(\sqrt{NT})$.These methods can be naturally extended to the combinatorial setting where K arms are chosen, treating each of the $\binom{N}{K}$ possible actions as a distinct 'arm'. Unfortunately, this approach has two significant drawbacks. First, the regret increases exponentially in K as the number of total actions to explore has grown from N to $\binom{N}{K}$. Second, the time and space complexities increase exponentially in K, requiring storage of values for all actions to find the action with highest UCB (Auer and Ortner 2010; Auer, Cesa-Bianchi, and Fischer 2002) or highest sampled rewards (Agrawal and Goyal 2012).

This paper addresses the said issues by proposing a novel algorithm called **aDaptive Accept RejecT** (DART). To estimate the "goodness" of an arm, we use the mean of the rewards obtained by playing actions containing that arm. In an adaptive manner, DART moves arms to "accept" or "reject" sets based on those estimates, reducing the number of arms that require further exploration. We assume that the expected joint reward of an arm i and other possible $K-1$ arms is better than the expected joint reward of arm j and other possible $K-1$ arms if arm i is better than arm j. This assumption is naturally satisfied in many reward setups such as click bandits (Kveton et al. 2015a). We then use Lipschitz continuity of the joint reward function to relate orderings between pairs of arms i and j to orderings between pairs of actions containing those arms. We construct a martingale sequence to analyze the regret bound of DART. Furthermore, DART achieves a space complexity of $\mathcal{O}(N)$ and a per-round time complexity of $\tilde{\mathcal{O}}(N)$.

The main contributions of this paper can be summarized

as follows:
(1). We propose DART - a time and space efficient algorithm for the non-linear CMAB problem with only the joint reward as feedback. We show that DART has per-step time complexity of $\tilde{\mathcal{O}}(N)$ and space complexity of $\mathcal{O}(N)$.
(2). We prove a lower bound of $\Omega(K\sqrt{NKT})$ for top-K subset identification problem in linear setup where the joint rewards are possibly correlated.
(3). We prove that DART achieves a regret of $\tilde{\mathcal{O}}(K\sqrt{NKT})$ over a time horizon T and under certain assumptions. The regret bound matches the lower bound for the bandit setting where individual arm rewards are possibly correlated.

We also empirically evaluate the proposed algorithm DART, comparing it to other, state-of-the-art full-bandit feedback CMAB algorithms. We first consider a linear setting, where the joint reward as simply the mean of individual arm rewards. We also examine the setting where the joint reward is a quadratic function of individual arm rewards, based on the problem of cross-selling item selection (Raymond Chi-Wing Wong, Ada Wai-Chee Fu, and Wang 2003). Our algorithm significantly outperforms existing state of the art algorithms, while only using polynomial space and time complexity.

Related Works

(Dani, Hayes, and Kakade 2008; Dani, Kakade, and Hayes 2008; Cesa-Bianchi and Lugosi 2012; Audibert, Bubeck, and Lugosi 2014a; Abbasi-Yadkori, Pal, and Szepesvari 2011; Li et al. 2010) consider a linear bandit setup where at time t, the agent selects a vector x_t from the decision set $D_t \subset \mathbb{R}^N$ and observes a reward $\theta^T x_t$ for an unknown constant vector $\theta \in \mathbb{R}^N$. The algorithms proposed in these works use the linearity of the reward function to estimate rewards of individual arms and achieve a regret of $\mathcal{O}(\sqrt{NT})$. (Filippi et al. 2010; Jun et al. 2017; Li, Lu, and Zhou 2017) studied the problem of generalized linear models (GLM) where the reward r_t is a function $(f(z) : \mathbb{R} \to \mathbb{R})$ of $z = \theta^T x_t$ plus additive, arm-independent and iid noise. generalized linear model algorithms also obtain a regret bound of $\mathcal{O}(\sqrt{NT})$. The proposed algorithms can be naturally extended to our setup for linear joint reward functions. However, the space and time complexity remains exponential in K to store all possible $\binom{N}{K}$ actions.

When $K = 1$, Liau et al. (2018) reduces the space complexity for extremely large N from $\mathcal{O}(N)$ to $\mathcal{O}(1)$ at the cost of worse regret bounds. When extended to the combinatorial setting, treating each set of K arms as a distinct 'arm,' the regret bound becomes exponential in K. Recently, (Rejwan and Mansour 2020) bounded the regret by $\mathcal{O}(K\sqrt{NT})$ for identifying the best K subset, in the case when the joint reward is the sum of rewards of independent arms, using $\mathcal{O}(N)$ space and per-round time complexity. (Lin et al. 2014) considered the combinatorial bandit problem with a non-linear reward function and additional feedback, where the feedback is a linear combination of the rewards of the K arms. Such feedback allows for the recovery of individual rewards. (Agarwal and Aggarwal 2018) proposed a divide and conquer based algorithm for the best K subset problem with a non-linear joint reward. (Agarwal and Aggarwal 2018) is the most related work as the setup is similar to ours. Algorithms by (Lin et al. 2014; Agarwal and Aggarwal 2018) achieve $\mathcal{O}(T^{2/3})$ regret while the proposed algorithm in this paper achieves $\mathcal{O}(T^{1/2})$ regret. We achieve a better regret bound compared to (Agarwal and Aggarwal 2018) or (Lin et al. 2014) which use additional feedback.

Many works have studied the semi-bandit setting, where individual arm rewards are also available as feedback (Kveton et al. 2014; Chen, Wang, and Yuan 2013; Kveton et al. 2014; Lattimore et al. 2018; Gai, Krishnamachari, and Jain 2012, 2010). (Kveton et al. 2014) provides a UCB-type algorithm for matroid bandits, where the agent selects a maximal independent set of rank K to maximize the sum of individual arm rewards. (Chen, Wang, and Yuan 2013) considered the combinatorial semi-bandit problem with non-linear rewards using a UCB-type analysis. In contrast to these prior works, we consider the full-bandit setting where individual arm rewards are not available. (Kveton et al. 2015b; Lattimore et al. 2018) proved a lower bound of $\Omega(\sqrt{NKT})$ for semi-bandit problems where the joint reward is simply the sum of individual arm rewards. (Kalyanakrishnan et al. 2012) also provides a lower bound for the best K subset problem of $\Omega\left(\frac{N}{\epsilon^2}\log\left(\frac{K}{\delta}\right)\right)$ for any (ϵ, δ)-PAC algorithm playing single arm at each time. (Audibert, Bubeck, and Lugosi 2014b) obtained a lower bound of $\Omega(K\sqrt{NT})$ for bandit feedback and provide an algorithm with regret bound of $\Omega(K\sqrt{NKT})$ for linear bandits without assuming independence between arms. (Cohen, Hazan, and Koren 2017) obtain a tighter lower bound of $\Omega(K\sqrt{KNT})$ for a bandit setup where the rewards of individual arms are possibly correlated. In this paper we achieve the tighter lower bound (ignoring $\log$ terms) for bandit feedback with possibly correlated rewards.

Problem Formulation

We consider N "arms" labeled as $i \in [N] = \{1, 2, \cdots, N\}$. On playing arm i at time step t, it generates a reward $X_{i,t} \in [0, 1]$ which is a random variable. We assume that $X_{i,t}$ are independent across time, and for any arm the distribution is identical at all times. For simplicity, we will use X_i instead of $X_{i,t}$ for analysis that holds for any t. The distribution for each arm i's rewards $\{X_{i,t}\}_{t=1}^T$ could be discrete, continuous, or mixed.

The agent can only play an action $\boldsymbol{a} \in \mathcal{N}$ where $\mathcal{N} = \{\boldsymbol{a} \in [N]^K \mid \boldsymbol{a}(i) \neq \boldsymbol{a}(j)\ \forall\ i, j : 1 \leq i < j \leq K\}$ is the set of all K sized tuples created using arms in $[N]$. Thus, the cardinality of $\mathcal{N}$ is $\binom{N}{K}$. For an action $\boldsymbol{a}$, let $\boldsymbol{d}_{\boldsymbol{a},t} = (X_{\boldsymbol{a}(1),t}, X_{\boldsymbol{a}(2),t}, \cdots, X_{\boldsymbol{a}(K),t})$ be the column reward vector of individual arm rewards at time t from arms in action $\boldsymbol{a}$. The reward $r_{\boldsymbol{a}}(t)$ of an action $\boldsymbol{a}$ at time t is a bounded function $f : [0, 1]^K \to [0, 1]$ of the individual arm rewards

$$r_{\boldsymbol{a}}(t) = f(\boldsymbol{d}_{\boldsymbol{a},t}). \tag{1}$$

As $X_{i,t}$ are i.i.d. across time t, $\boldsymbol{d}_{\boldsymbol{a},t}$ are also i.i.d. across time t for all $\boldsymbol{a} \in \mathcal{N}$. Later in the text we will skip index t, for brevity, where it is unambiguous. We denote the expected reward of any action $\boldsymbol{a} \in \mathcal{N}$ as $\mu_{\boldsymbol{a}} = \mathbb{E}[r_{\boldsymbol{a}}]$. We assume that there is a unique "optimal" action $\boldsymbol{a}^*$ for which the expected reward $\mu_{\boldsymbol{a}^*}$ is highest among all actions,

$$\boldsymbol{a}^* = \arg\max_{\boldsymbol{a}\in\mathcal{N}} \mu_{\boldsymbol{a}}. \tag{2}$$

At time t, the agent plays an action $\boldsymbol{a}_t$ randomly sampled from an arbitrary distribution over $\mathcal{N}$ dependent on the history of played actions and observed rewards till time $t-1$. The agent aims to reduce the cumulative regret R over time horizon T, defined as the expected difference between the rewards of the best action in hindsight and the actions selected by the agent.

$$R = \mathbb{E}_{\boldsymbol{a}_1, r_{\boldsymbol{a}_1}(1), \cdots, \boldsymbol{a}_T, r_{\boldsymbol{a}_T}(T)} \left[\sum_{t=1}^{T} r_{\boldsymbol{a}^*}(t) - r_{\boldsymbol{a}_t}(t)\right] \tag{3}$$

$$= T\mu_{\boldsymbol{a}^*} - \mathbb{E}_{\boldsymbol{a}_1, \cdots, \boldsymbol{a}_T,} \left[\sum_{t=1}^{T} \mu_{\boldsymbol{a}_t}\right]. \tag{4}$$

We define the gap $\Delta_{i,j}$ between two arms i and j as the difference between the expected rewards of arm i and arm j,

$$\Delta_{i,j} = \mathbb{E}\left[X_i\right] - \mathbb{E}\left[X_j\right]. \tag{5}$$

We now mention the assumptions for this paper. We first assume that the joint reward function f is permutation invariant. Let Π denote the set of all permutation functions of a length K vector.

Assumption 1 (Symmetry)**.** *For any permutation $\pi \in \Pi$ of the vector $\mathbf{d}$ of individual arm rewards,*

$$f\left(\boldsymbol{d}\right) = f\left(\pi(\boldsymbol{d})\right) \tag{6}$$

We also assume that the expected reward of an action with a good arm is higher than the expected reward of action with a bad arm for all possible combinations of the remaining $K-1$ arms. Further, if two arms are equally good, they are indistinguishable in every action and will not contribute to regret. This assumption is similar to Assumption 4 of (Lattimore et al. 2018).

Assumption 2 (Good arms generate good actions)**.** *We assume that if the expected reward of arm i is higher than the expected reward of arm j (for any given $i \neq j$), then for any subset $\boldsymbol{S}$ of size $K-1$ arms chosen from the remaining $N-2$ arms (arms excluding i and j), the expected reward of $\boldsymbol{S} \cup \{i\}$ is higher than the expected reward of $\boldsymbol{S} \cup \{j\}$. More precisely, if $\mathbb{E}[X_i] \geq \mathbb{E}[X_j]$ then*

$$\mathbb{E}_{X_i, X_j, \boldsymbol{d_S}}\left[f\left(h\left(X_i, \boldsymbol{d_S}\right)\right) - f\left(h\left(X_j, \boldsymbol{d_S}\right)\right)\right] \geq 0 \tag{7}$$

for all $\boldsymbol{S}$, where $h : \mathbb{R} \times \mathbb{R}^{K-1} \rightarrow \mathbb{R}^K$ is an appending function[1] *and $\boldsymbol{d_S} \in [0,1]^{K-1}$ is a random vector of the rewards from arms in $\boldsymbol{S}$. Further the equality holds only if $\mathbb{E}[X_i] = \mathbb{E}[X_j]$.*

We note that the analysis also holds if (7) holds in the opposite direction for all $\boldsymbol{S}$, i, and j such that $\mathbb{E}[X_i] \geq \mathbb{E}[X_j]$, by transforming the reward function as $\tilde{f}(\boldsymbol{d}) = 1 - f(\boldsymbol{d})$.

We also assume that $f(\cdot)$ is Bi-Lipschitz continuous (in an expected sense).

Assumption 3 (Continuity of expected rewards)**.** *The expected value of $f(\cdot)$ is Bi-Lipschitz continuous with respect to the expected value of the rewards obtained by the individual arms, meaning*

$$\frac{1}{U}\left|\left|\mathbb{E}[\boldsymbol{d}_{\mathbf{a}_1}] - \pi(\mathbb{E}[\boldsymbol{d}_{\mathbf{a}_2}])\right|\right|_1 \leq \left|\mu_{\mathbf{a}_1} - \mu_{\mathbf{a}_2}\right| = \left|\mathbb{E}\left[f(\boldsymbol{d}_{\mathbf{a}_1})\right] - \mathbb{E}\left[f(\boldsymbol{d}_{\mathbf{a}_2})\right]\right| \leq U\left|\left|\mathbb{E}[\boldsymbol{d}_{\mathbf{a}_1}] - \pi(\mathbb{E}[\boldsymbol{d}_{\mathbf{a}_2}])\right|\right|_1 \tag{8}$$

for any pair of actions $\mathbf{a}_1, \mathbf{a}_2 \in \mathcal{N}$ and for any permutation π of $\mathbf{d}$ and for some $U \in [1, \infty)$.

Corollary 1. *The expected value of $f(\cdot)$ is inverse-Lipschitz continuous with respect to expected rewards X_i and X_j for all $i, j \in [K]$.*

$$\left|\mathbb{E}[X_i] - \mathbb{E}[X_j]\right| \leq U\left|\left(\mathbb{E}\left[f(h\left(X_i, \boldsymbol{d_S}\right))\right] - \mathbb{E}\left[f(h\left(X_j, \boldsymbol{d_S}\right))\right]\right)\right| \tag{9}$$

for random vectors $\boldsymbol{d_S}$ of reward of subset $\boldsymbol{S}$ of size $K-1$ from $N-2$ arms.

Proof. We obtain the result by choosing $\boldsymbol{d}_{\boldsymbol{a}_1} = h(X_i, \boldsymbol{d_S})$ and $\boldsymbol{d}_{\boldsymbol{a}_2} = h(X_j, \boldsymbol{d_S})$. □

Assumptions 1-3 are satisfied for many problem setups, such as in cascade model for clicks (Kveton et al. 2015a) where a user interacting with a list of documents clicks on the first documents the user likes. The joint reward is $r(t) = \max_i(X_{1,t}, \cdots, X_{K,t})$ for independent arm rewards and the corresponding form of Equation (7) is $1 - (\Pi_k(1-\mathbb{E}[X_k]))\,(1-\mathbb{E}[X_i]) \geq 1 - (\Pi_k(1-\mathbb{E}[X_k]))\,(1-\mathbb{E}[X_j])$ which holds when $\mathbb{E}[X_i] > \mathbb{E}[X_j]$, and the Bi-Lipschitz property in individual expected rewards holds too. The assumptions are also satisfied in cross-selling optimization (Raymond Chi-Wing Wong, Ada Wai-Chee Fu, and Wang 2003), where the reward is a quadratic function of the individual items sold in a bundle K. The assumption are also satisfied for joint rewards as sum or mean of individual rewards (Rejwan and Mansour 2020; Chen, Wang, and Yuan 2013).

Lower bound on Top-K Subset Identification

Given the formulation, we now prove a tight lower bound on the subset identification problem in linear setup with correlated rewards. For the linear setup, we define the reward function as $f(\boldsymbol{d}) = \sum_i^K d(i)$, where $d(i)$ is the i^{th} entry of $\boldsymbol{d}$. Further we consider a setup where the best subset $\boldsymbol{a}^* = \{1, 2, \cdots, K\}$ and the reward distribution of individual arm is $X'_{i,t} = 1/2 + \epsilon\mathbf{1}_{\{i\in\boldsymbol{a}^*\}} + Z_t$, where Z_t follows Gaussian distribution with mean 0 and variance σ^2.

Theorem 1. *Any deterministic player must suffer expected regret of at least $\Omega(\sigma K\sqrt{KNT})$ against an environment with rewards $X'_{i,t}$ for $t = 1, 2, \cdots, T$ for each arm $i \in [N]$.*

Proof. (Outline:) We note that if the algorithm plays against a setup where all the arms are identical, then the expected number of times it selects an arm $i \in \boldsymbol{a}^*$ is KT/N as the arms are not distinguishable. Using this and the proof of

[1] $h(x, \boldsymbol{z}) = (x, z_1, \cdots, z_{K-1})$ where $\boldsymbol{z} = (z_1, \cdots, z_{K-1})$.

Lemma 4 from (Cohen, Hazan, and Koren 2017) we obtain the required result. A detailed proof is reconstructed in our technical report (Agarwal et al. 2020). □

Proposed DART Algorithm

We first state a relevant lemma to motivate the proposed algorithm. We note that Assumption 2 allows us to order arms without observing their individual expected rewards.

Lemma 1. *Let $\mathcal{N}(i)$ and $\mathcal{N}(j)$ be the set of all actions that contains arms i and j respectively ($|\mathcal{N}(i)| = |\mathcal{N}(j)| = \binom{N-1}{K-1}$). If the actions are uniformly randomly selected from the sets $\mathcal{N}(i)$ and $\mathcal{N}(j)$, then the following holds.*

$$\mathbb{E}\left[X_i\right] \leq \mathbb{E}\left[X_j\right]$$
$$\Leftrightarrow \mathbb{E}_{\boldsymbol{a}_i \sim \mathbb{U}(\mathcal{N}(i))}\left[\mu_{\boldsymbol{a}_i}\right] \leq \mathbb{E}_{\boldsymbol{a}_j \sim \mathbb{U}(\mathcal{N}(j))}\left[\mu_{\boldsymbol{a}_j}\right] \tag{10}$$

where $\mathbb{U}(\cdot)$ is the uniform distribution.

Proof Sketch:. We take expectation on all the actions over set $\mathcal{N}(i)$ in the left and over set $\mathcal{N}(j)$ in the right hand side of the Equation (7) from Assumption 2 to obtain the required result. A detailed proof is presented in (Agarwal et al. 2020). □

From Lemma 1, we note that if we create uniformly random partitions, the expected reward of action containing arm i will be higher than the expected reward of the action containing arm j if arm i is better than arm j. We use this idea to create the proposed DART algorithm in Algorithm 1.

The algorithm initializes $\hat{\mu}_i$ as the estimated mean for actions that contain arm i and n_i as the number of times an action containing arm i is played. The algorithm proceeds in epochs, indexed by e, and maintains three different sets at each epoch. The first set, $\mathcal{A}_e$, contains "good" arms which belong to the top-K arms found till epoch e. The second set, $\mathcal{N}_e$, contains the arms which the algorithm is still exploring at epoch e. The third set, $\mathcal{R}_e$, contains the arms that are "rejected" and do not belong in the top-K arms. We let K_e be the variable that contains the number of spots to fill in the top-K subset at epoch e. The algorithm maintains a decision variable Δ as the concentration bound and a parameter variable n as the minimum number of samples required for achieving the concentration bound Δ. Lastly, the algorithm maintains a hyper parameter λ tuned for the value of T, N, and K. λ is the minimum gap between any two arms the algorithm can resolve within time horizon T.

In Line 5, the algorithm selects a permutation of $\mathcal{N}_e$ uniformly at random and partitions it into sets of size K_e. If K_e does not divide $|\mathcal{N}_e|$, we repeat arms in the last group (cyclically, so that the last group has K_e distinct arms). To simplify the bookkeeping, $\hat{\mu}_i$ and n_i are not updated if arm i is repeated in the last group. The algorithm then creates an action $\boldsymbol{a}_t$ from the partitioned groups and the arms in the good set and plays it to obtain a reward $r_{\boldsymbol{a}_t}(t)$ at time t (Line 8-9). DART then updates the estimated mean for all arms played in $\boldsymbol{a}_t$ with the observed reward and increments the number of counts for the arms played (Line 10-12).

In lines 15-16, the algorithm moves an arm $i \in \mathcal{N}_e$ to $\mathcal{A}_e$ if estimated mean of actions that contain arm i, $\hat{\mu}_i$, is Δ more than the estimated mean of actions that contain arm at $(K+1)^{th}$ rank, $\hat{\mu}_{K+1}$. Similarly, the algorithm moves an arm $i \in \mathcal{N}_e$ to $\mathcal{R}_e$ if estimated mean of actions that contain arm i, $\hat{\mu}_i$, is Δ less than the estimated mean of actions that contain arm at K^{th} rank, $\hat{\mu}_K$.

Algorithm 1 DART(T, N, K)

```
1: Initialize $\hat{\mu}_i = 0, n_i = 0$ for $i \in \{1, 2, \cdots, N\}$; $t = 0$; $e = 0, \lambda = \sqrt{\frac{720NK\log 2NT}{T}}$
2: $\mathcal{A}_e = \phi, \mathcal{R}_e = \phi, \mathcal{N}_e = [N]$  ▷ Initialize parameters for rounds
3: $\Delta = 1, n = \frac{288\log(NT)}{\Delta^2}, K_e = K - |\mathcal{A}_e|$
4: while $t < T$ do
5:     Choose a permutation of $\mathcal{N}_e$ uniformly at random and partition it into sets of size $K_e$: $\mathcal{N}_{e,1}, \mathcal{N}_{e,2}, \cdots, \mathcal{N}_{e,|\mathcal{N}_e|/K_e}$
6:     $e = e + 1; \ell = 1$
7:     while $\ell \leq |\mathcal{N}_e|/K_e$ and $t < T$ do
8:         $\boldsymbol{a}_t = \mathcal{A}_e \cup \mathcal{N}_{e,\ell}$  ▷ Create action from arms in $\mathcal{A}_e \cup \mathcal{N}_{e,\ell}$
9:         Play action $\boldsymbol{a}_t$ and obtain reward $r_{\boldsymbol{a}_t}(t)$
10:        for all arm $i \in \mathcal{N}_{e,\ell}$ do
11:            $\hat{\mu}_i = \frac{n_i\hat{\mu}_i + r_{\boldsymbol{a}_t}(t)}{n_i + 1}; n_i = n_i + 1$
12:        $t = t + 1; \ell = \ell + 1$
13:    Sort $\mathcal{A}_e \cup \mathcal{N}_e \cup \mathcal{R}_e$ according to $\hat{\mu}_{(1)} \geq \hat{\mu}_{(2)} \geq \cdots \geq \hat{\mu}_{(N)}$
14:    $\bar{\mathcal{A}} = \{i \in \mathcal{N}_e | \hat{\mu}_i > \hat{\mu}_{(K+1)} + \Delta\}$; $\bar{\mathcal{R}} = \{i \in \mathcal{N}_e | \hat{\mu}_i < \hat{\mu}_{(K)} - \Delta\}$
15:    $\mathcal{A}_{e+1} = \mathcal{A}_e \cup \bar{\mathcal{A}}$; $\mathcal{R}_{e+1} = \mathcal{R}_e \cup \bar{\mathcal{R}}$; $\mathcal{N}_{e+1} = \mathcal{N}_e \setminus \left(\bar{\mathcal{A}} \cup \bar{\mathcal{R}}\right)$; $K_{e+1} = K - |\mathcal{A}_{e+1}|$
16:    if $e \geq n$ then
17:        $\Delta = \frac{\Delta}{2}, n = \frac{288\log(NT)}{\Delta^2}$
18:    if $\Delta < \lambda$ or $|\mathcal{A}_e \cup \mathcal{N}_e| == K$ then
19:        break while loop
20: Sort $\mathcal{A}_e \cup \mathcal{N}_e \cup \mathcal{R}_e$ according to $\hat{\mu}_{(1)} \geq \hat{\mu}_{(2)} \geq \cdots \geq \hat{\mu}_{(N)}$
21: $\boldsymbol{a} = \mathcal{A}_e \cup \{i \in \mathcal{N}_e | \hat{\mu}_i > \hat{\mu}_{(K+1)}\}$
22: while $t < T$ do
23:     Play action $\boldsymbol{a}; t = t + 1$
```

The proposed DART algorithm uses a random permutation of $\mathcal{N}_e$. The random permutation can be generated in $\mathcal{O}(N)$ steps. Also after each round, the algorithm finds the K^{th} and $(K+1)^{th}$ ranked arms. This operation can be completed in $\tilde{\mathcal{O}}(N)$ time complexity using sorting $\{\hat{\mu}_i\}_{i=1}^N$. Also going over each arm in $\mathcal{N}_e$ is of linear time complexity. Hence, the per-step time complexity of the algorithm comes out to be $\tilde{\mathcal{O}}(N)$. Also, the proposed DART algorithm only stores the estimates $\hat{\mu}_i$ for each arm $i \in [N]$. The resulting storage complexity is $\mathcal{O}(N)$ for maintaining the estimates. To find the top-K and the top-$(K+1)$ means, the algorithm may use additional space of $\mathcal{O}(N)$ to maintain a heap. Thus, the overall space complexity of the algorithm is only $\mathcal{O}(N)$.

Regret Analysis

We now analyse the sample complexity and regret of the proposed DART algorithm. To bound the regret, we first bound the number of samples required to move an arm in $\mathcal{N}_e$ to either of $\mathcal{A}_e$ or $\mathcal{R}_e$. Then, we bound the regret from including a sub-optimal arm in the played actions. For the analysis, without loss of generality, we assume that the expected rewards of arms are ranked as $\mathbb{E}[X_1] > \mathbb{E}[X_2] > \cdots > \mathbb{E}[X_N]$. If the arms are not in the said order, we relabel the arms to obtain the required order. From Assumption 2, we have $\boldsymbol{a}^* = \{1, 2, \cdots, K\}$. We refer to arms $1, \cdots, K$ as optimal arms and arms $K+1, \cdots, N$ as sub-optimal arms.

Number of samples to move an arm in $\mathcal{N}_e$ to either of $\mathcal{A}_e$ or $\mathcal{R}_e$

We call two arms $i, j \in \mathcal{N}_e,\ i < j$ separated if the algorithm has high confidence that $\mathbb{E}[X_i] > \mathbb{E}[X_j]$. We first analyze the general conditions to separate any two arms $i, j \in \mathcal{N}_e$ such that $\mathbb{E}[X_i] > \mathbb{E}[X_j]$. Let the epoch where arm i and arm j are separated and the epoch of Algorithm 1 be e. We define a filtration $\mathcal{F}_e$ as the history observed by the algorithm till epoch e.

For any $u \in \mathcal{N}_e$, let $\mathcal{N}_e(u) = \{\boldsymbol{a} \in [N]^K : u \in \boldsymbol{a}, \mathcal{A}_e \subseteq \boldsymbol{a}, \mathcal{R}_e \cap \boldsymbol{a} = \phi, \boldsymbol{a}(i) \neq \boldsymbol{a}(j) \forall i, j : 1 \leq i < j \leq K\}$.

We now define a random variable $Z_{i,j}(e)$ for $i, j \in \mathcal{N}_e$, which denotes the difference between the reward observed from playing an uniform random action from $\mathcal{N}_e(i)$ and an uniform random action from $\mathcal{N}_e(j)$. In other words,

$$Z_{i,j}(e) = r_{\boldsymbol{a}_i}(e) - r_{\boldsymbol{a}_j}(e), \tag{11}$$

where $\boldsymbol{a}_i \sim \mathbb{U}(\mathcal{N}_e(i))$, $\boldsymbol{a}_j \sim \mathbb{U}(\mathcal{N}_e(j))$ and $\mathbb{U}(\cdot)$ denotes the uniform distribution. Also, $r_{\boldsymbol{a}_i}(e)$ is the reward observed by playing $\boldsymbol{a}_i$ and $r_{\boldsymbol{a}_j}(e)$ is the reward obtained by playing $\boldsymbol{a}_j$ at epoch e. Hence, the randomness of $Z_{i,j}(e)$ comes from both the random selection of $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$, and from the reward generated by playing $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$. Let $\mathbb{P}_{Z_{i,j}(e)}$ denote the probability distribution of $Z_{i,j}(e)$. We now mention a lemma for bounding the expected value of $Z_{i,j}(e)$ for all epochs e.

Lemma 2. *Let $i, j \in [N]$ be two arms such that $\mathbb{E}[X_i] > \mathbb{E}[X_j]$. Let $Z_{i,j}(e)$ be a random variable denoting the difference between the reward obtained on playing a uniform random action $\boldsymbol{a}_i \sim \mathbb{U}(\mathcal{N}_e(i))$ containing arm i and a randomly selected action $\boldsymbol{a}_j \sim \mathbb{U}(\mathcal{N}_e(j))$ containing arm j. Then the expected value of $Z_{i,j}(e)$ is upper bounded by $U\Delta_{i,j}$, and lower bounded by 0, or,*

$$\frac{\Delta_{i,j}}{KU} \leq \mathbb{E}[Z_{i,j}(e)] \leq U\Delta_{i,j} \tag{12}$$

Proof Sketch. We first show the upper bound. The cardinality of both $\mathcal{N}_e(i)$ and $\mathcal{N}_e(j)$ is $\binom{|\mathcal{N}_e|-1}{K_e-1}$ as we have fixed one of the K_e places for arm i and now we can fill only $K_e - 1$ places from the available $|\mathcal{N}_e| - 1$ arms. Algorithm 1 partitions a random, uniformly distributed permutation over $\mathcal{N}_e$, so all actions $\boldsymbol{a} \in \mathcal{N}_e(i)$ are equally likely, and likewise for $\boldsymbol{a} \in \mathcal{N}_e(j)$. Taking the expectation over the actions played and the reward obtained, we get the expected value of $Z_{i,j}(e)$ as

$$\begin{aligned}
\mathbb{E}\left[Z_{i,j}(e)\right] &= \mathbb{E}\left[r_{\boldsymbol{a}_i}(e) - r_{\boldsymbol{a}_j}(e)\right] && (13)\\
&= \frac{1}{\binom{|\mathcal{N}_e|-1}{K_e-1}}\left(\sum_{\boldsymbol{a}\in\mathcal{N}_e(i)} \mu_{\boldsymbol{a}} - \sum_{\boldsymbol{a}\in\mathcal{N}_e(j)} \mu_{\boldsymbol{a}}\right) && (14)\\
&\leq \frac{1}{\binom{|\mathcal{N}_e|-1}{K_e-1}}\binom{|\mathcal{N}_e|-2}{K_e-1} U\Delta_{i,j} && (15)\\
&= \frac{|\mathcal{N}_e| - K_e}{|\mathcal{N}_e| - 1} U\Delta_{i,j} \leq U\Delta_{i,j}. && (16)
\end{aligned}$$

Equation (14) is obtained by linearity of expectation and taking the expectation over rewards of uniformly distributed actions $\boldsymbol{a}_i$ and $\boldsymbol{a}_j$. Equation (15) is obtained by noting that there exist exactly $\binom{|\mathcal{N}_e|-2}{K_e-1}$ actions where arm i is replaced by arm j. From Assumption 3 of Lipschitz continuity, the difference between the expected reward of those actions is bounded by $U\Delta_{i,j}$. The remaining actions contain both arms i and j, thus are in both $\mathcal{N}_e(i)$ and $\mathcal{N}_e(j)$, and so cancel out. Equation (16) comes from simplifying the fraction with binomial and noticing that $K_e \geq 1$. This proves the upper bound.

Similarly we obtain the lower bound using Assumption 1

$$\begin{aligned}
\mathbb{E}\left[Z_{i,j}(e)\right] &= \mathbb{E}\left[r_{\boldsymbol{a}_i}(e) - r_{\boldsymbol{a}_j}(e)\right] && (17)\\
&= \frac{1}{\binom{|\mathcal{N}_e|-1}{K_e-1}}\left(\sum_{\boldsymbol{a}\in\mathcal{N}_e(i)} \mu_{\boldsymbol{a}} - \sum_{\boldsymbol{a}\in\mathcal{N}_e(j)} \mu_{\boldsymbol{a}}\right) && (18)\\
&\geq \frac{\Delta_{i,j}}{U} \frac{1}{\binom{|\mathcal{N}_e|-1}{K_e-1}}\binom{|\mathcal{N}_e|-2}{K_e-1} && (19)\\
&= \frac{\Delta_{i,j}}{U} \frac{|\mathcal{N}_e| - K_e}{|\mathcal{N}_e| - 1} \geq \frac{\Delta_{i,j}}{K_e U} \geq \frac{\Delta_{i,j}}{KU}. && (20)
\end{aligned}$$

Equation (15) is obtained from Assumption 1. The difference between the expected reward of the actions are lower bounded by $\frac{\Delta_{i,j}}{U}$. Equation (20) comes from noting that $K_e(|\mathcal{N}_e| - K_e) \geq |\mathcal{N}_e| - 1$. This proves the lower bound. □

The sequence of random variables $Z_{i,j}(e), e = 1, 2, \cdots$ are not independent as the sets $\mathcal{A}_i(e)$ and $\mathcal{A}_j(e)$ are updated as the algorithm proceeds. Hence, we cannot apply Hoeffding's concentration inequality (Hoeffding 1994, Theorem 2) for analysis. To use Azuma-Hoeffding's inequality (Bercu, Delyon, and Rio 2015, Chapter 3), we need to construct a martingale. For each pair of arms $i, j \in [N]$ with $\mathrm{E}[X_i] > \mathrm{E}[X_j]$, we define $Y_{i,j}$ as a martingale with respect to filtration $\mathcal{F}_e$,

$$Y_{i,j}(e) = \sum_{e'=1}^{e} \left(Z_{i,j}(e') - \mathbb{E}_{e'-1}\left[Z_{i,j}(e')\right]\right) \tag{21}$$

where $\mathbb{E}_{e'-1}[\cdot] = \mathbb{E}[\cdot|\mathcal{F}_{e'-1}]$. $Y_{i,j}(e)$ is a martingale with zero-mean, and $|Y_{i,j}(e) - Y_{i,j}(e-1)| \leq 2$, and hence we can apply Azuma-Hoeffding's inequality to $Y_{i,j}(e)$ for all $i, j \in [N]$.

After obtaining the concentration of $Y_{i,j}(e)$ with respect to the e^{th} iteration of sample of action with arm i and arm j, we now obtain the value of e for which we can consider arm i and j to be separated with probability $1-\delta$.

Lemma 3. *Let arms $i, j \in [N]$ be two arms such that $\mathbb{E}[X_i] > \mathbb{E}[X_j]$. Let Δ be such that $\Delta < \hat{\mu}_i - \hat{\mu}_j \leq 2\Delta$. Then, with probability at least $1-\delta$, arm i and arm j are separable at epoch e and $\frac{32\log 2/\delta}{\Delta^2} \leq e \leq \mathcal{O}\left(\frac{288K^2U^2\log 2/\delta}{\Delta_{i,j}^2}\right)$.*[2]

Proof Sketch:. At epoch e, $\hat{\mu}_i - \hat{\mu}_j = \sum_{e'=1}^{e} Z_{i,j}(e')/e$. Using this relation and Azuma-Hoeffding's inequality on $Y_{i,j}(e)$, we get the required result. A detailed proof is provided in (Agarwal et al. 2020). □

We can now bound the number of samples required to move each arm from $\mathcal{N}_e$ to either the "accept" set $\mathcal{A}_e$ or the "reject" set $\mathcal{R}_e$. In the algorithm, arm i will be moved to the accept set $\mathcal{A}_e$ when its empirical mean $\hat{\mu}_i$ is sufficiently larger than that of the $K+1$ ranked arm. Consider an arm i in the optimal action $a^* = \{1, \ldots, K\}$. By Lemma 3, with probability $1-\delta$, arms i and $K+1$ will be separable by epoch

$$e \leq \frac{288U^2K^2\log(2/\delta)}{\Delta_{i,K+1}^2}. \tag{22}$$

Similarly, arm i will be moved to the reject set $\mathcal{R}_e$ when its empirical mean $\hat{\mu}_i$ is sufficiently less than that of the Kth ranked arm. Consider an arm $i \in \{K+1, \ldots, N\}$. By Lemma 3, with probability $1-\delta$, arms i and K will be separable by epoch

$$\frac{288U^2K^2\log(2/\delta)}{\Delta_{K,i}^2}. \tag{23}$$

Regret from sampling sub-optimal arms

We first bound the regret of playing any action $\boldsymbol{a} \in \mathcal{N}$ using Assumption 3.

Lemma 4. *Let $\boldsymbol{a} = (a_1, a_2, \cdots, a_K)$ be any action. The expected regret suffered from playing action $\boldsymbol{a}$ instead of action $\boldsymbol{a}^* = (1, 2, \cdots, K)$ is bounded as*

$$\left|\, \mu_{\mathbf{a}} - \mu_{\mathbf{a}^*} \right| \leq \; U \sum_{i=1}^{K} \left| \mathbb{E}[X_{a_i}] - \mathbb{E}[X_{\pi(i)}] \right|, \tag{24}$$

for any permutation π of $\{1, \cdots, K\}$ for which $\pi(i) = a_i$ if $a_i \leq K$.

Proof Sketch:. From Assumption 3, we first find a tight upper bound. We finish the proof by using the fact that Assumption 3 selects the permutation which minimizes the bound, hence any other permutation also gives a valid upper bound. A detailed proof is provided in (Agarwal et al. 2020). □

We now bound the regret incurred by playing an action $\boldsymbol{a_t}$ at time t containing sub-optimal arm $i \in \{K+1, \cdots, N\}$ replacing an optimal arm $j \in \{1, \cdots, K\}$ in Lemma 5.

Lemma 5. *For any sub-optimal action, the regret it can accumulate by replacing an optimal arm $j \in \{1, \cdots, K\}$ by an arm $i \in K+1, \cdots, N$ is bounded by*

$$\frac{1440K^2U^3\log(2/\delta)}{\Delta_{K,i}} \tag{25}$$

Proof Sketch. The agent suffers from regret if it an action that contains at least one sub-optimal arm. To bound the regret from a sub-optimal action, we use the proof technique of (Rejwan and Mansour 2020) to divide the optimal arms $j \in \{1, \cdots, K\}$ into two groups: first group with $\Delta_{j,K+1} > \Delta_{K,i}$ and second group with $\Delta_{j,K+1} \leq \Delta_{K,i}$. We show that regret from both groups is bounded by $\mathcal{O}\left(\frac{1}{\Delta_{K,i}}\right)$ The detailed proof is provided in (Agarwal et al. 2020). □

After calculating the regret from individual arms, we now calculate the total regret of the DART algorithm in the following theorem

Theorem 2. *For $\lambda = U\sqrt{\frac{720NK\log 2NKT}{T}}$, the distribution free regret incurred by* DART *algorithm is bounded by*

$$R \leq \mathcal{O}\left(U^2K\sqrt{NKT\log 2NT}\right) \tag{26}$$

Proof Sketch. We use the standard proof technique of bounding regret accumulated while eliminating arms to reject set of a confidence bound based algorithm to tune λ and calculate the regret. A detailed proof is provided in (Agarwal et al. 2020). □

We note that the regret bound of DART matches matches the lower bound in Theorem 1 upto the factor of $\log(2NT)$ for bandits with joint reward as sum of rewards of individual arms in an action.

We note that there may be scenarios where an agent does not know the value of U and cannot tune λ accordingly. In such a case, the agent increases its regret because of not knowing the joint reward function. For a value of $\lambda = \sqrt{\frac{720NK\log 2NT}{T}}$, which does not use U, the regret of the algorithm is bounded as

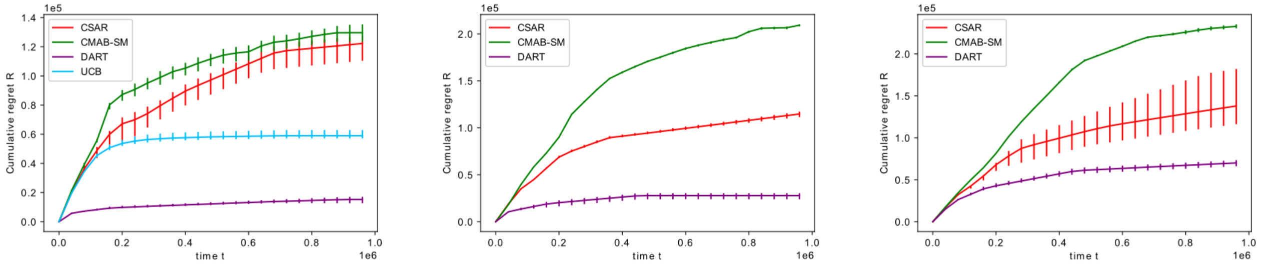
$$\mathcal{O}\left(\left(U^3 + U\right)K\sqrt{NKT\log 2NT}\right). \tag{27}$$

Additionally, we note that we can convert DART to an anytime algorithm using doubling trick of restarting algorithm at $T_l = 2^l \; l = 1, 2, \cdots$ until the unknown time horizon T is reached (Auer and Ortner 2010). Using analysis from (Besson and Kaufmann 2018, Theorem 4), we show that DART for unknown T achieves a regret bound of $\tilde{\mathcal{O}}(\sqrt{T})$. We present the complete proof in (Agarwal et al. 2020).

Experiments

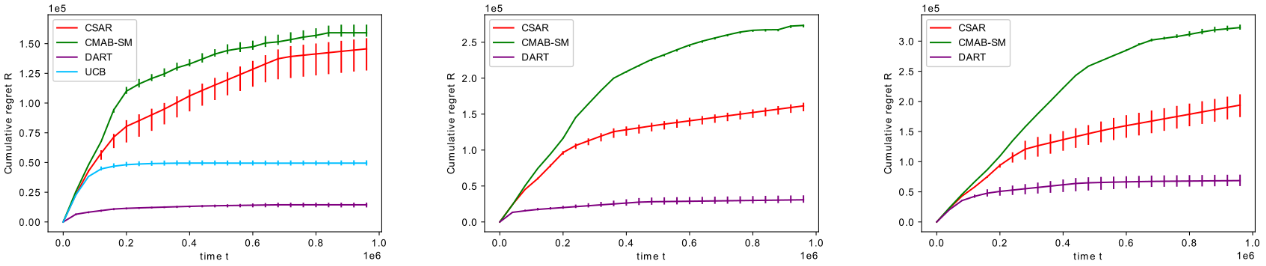
We now present comparison results of DART with CSAR (Rejwan and Mansour 2020) and CMAB-SM (Agarwal and

[2] For the particular case of $K = 1$, the upper bound reduces to $\mathcal{O}\left(\frac{288\log 2/\delta}{\Delta_{i,j}^2}\right)$



(a) $K = 2$ (b) $K = 4$ (c) $K = 8$

Figure 1: Regret plots for joint rewards as mean of individual arm rewards



(a) $K = 2$ (b) $K = 4$ (c) $K = 8$

Figure 2: Regret plots for joint rewards as quadratic function of individual arm rewards

Aggarwal 2018) and UCB (Auer and Ortner 2010) algorithms. We used $N = 45$ and $T = 10^6$ for simulations. We chose $K = 2, 4, 8$ for easy construction of Hadamard matrices for CSAR algorithm. We compare for two different reward setups. First setup has joint reward as linear function of individual rewards. Second setup has joint reward as a quadratic function of individual rewards. In each setup, individual arm rewards follows Bernoulli distribution with means sampled from $\mathbb{U}([0, 1])$. We run 25 independent iterations to plot average regret and the maximum and minimum values of the regret of each algorithm.

In the first setup, we have the joint reward of the form $r(t) = \boldsymbol{\theta}^T \boldsymbol{d}_{\boldsymbol{a}_t}$, where $\boldsymbol{\theta} \in \mathbb{R}^K$ is a vector with all entries as $1/K$. From Figure 1, we note that the performance of DART is significantly better than both CSAR and CMAB-SM for joint reward as the mean of the individual arm rewards. For CSAR algorithm in (Rejwan and Mansour 2020), this is because after updating Δ, it generates fresh $\frac{K^2}{\Delta^2}$ samples instead of using previous samples to improve estimates. We only compare with UCB (Auer and Ortner 2010) for $K = 2$ as the action space became too large for $K = 4, 8$. Also, we note that LinUCB algorithm (Li et al. 2010) for linear bandits runs extremely slow even for $K = 2$ and we show comparison for $N = 15, K = 2$ in (Agarwal et al. 2020).

We also simulate the joint reward of the form $r(t) = \boldsymbol{d}_{\boldsymbol{a}_t}^T \boldsymbol{A} \boldsymbol{d}_{\boldsymbol{a}_t}$, where $\boldsymbol{A} \in \mathbb{R}^{K \times K}$ is an upper triangular matrix with all entries as $2/K(K+1)$. A quadratic reward function is used in cross-selling optimization to quantify the total profit from selling a bundle of items compared to the profit from selling the items in the bundle separately (Raymond Chi-Wing Wong, Ada Wai-Chee Fu, and Wang 2003). From Figure 2, we note that DART significantly outperforms CSAR and CMAB-SM algorithm for quadratic function of individual rewards as well. We note that CSAR, though designed for linear setup, is able to model the ranking of the arms from quadratic rewards and beat CMAB-SM algorithm. However, we note that CSAR is not able to outperform when the joint reward is $\max$ of individual rewards as noted in (Agarwal et al. 2020).

Conclusion

We considered the problem of combinatorial multi-armed bandits with non-linear rewards, where the agent chooses K out of N arms in each time-step and receives an aggregate reward. We obtained a lower bound of $\Omega(K\sqrt{NKT})$ for the linear case with possibly correlated rewards. We proposed a novel algorithm, called DART, which is computationally efficient and has a space complexity which is linear in number of base arms. We analyzed the algorithm in terms of regret bound, and show that it is upper bounded by $\tilde{\mathcal{O}}(K\sqrt{NKT})$, which matches the lower bound of $\Omega(K\sqrt{NKT})$ for bandit setup with correlated rewards. DART works efficiently for large N and K and outperforms existing methods empirically. Based on the contributions, finding lower bound for Bi-Lipschitz reward function bandits and extending the proposed algorithm to a more general class of functions are some of the potential future works.

References

Abbasi-Yadkori, Y.; Pal, D.; and Szepesvari, C. 2011. Improved Algorithms for Linear Stochastic Bandits. In *Advances in Neural Information Processing Systems 24*, 2312–2320.

Agarwal, M.; and Aggarwal, V. 2018. Regret bounds for stochastic combinatorial multi-armed bandits with linear space complexity. *arXiv preprint arXiv:1811.11925* .

Agarwal, M.; Aggarwal, V.; Quinn, C. J.; and Umrawal, A. 2020. DART: aDaptive Accept RejecT for non-linear top-K subset identification. *arXiv preprint arXiv:2011.07687* .

Agrawal, S.; and Goyal, N. 2012. Analysis of Thompson sampling for the multi-armed bandit problem. In *Conference on Learning Theory*, 39–1.

Audibert, J.-Y.; Bubeck, S.; and Lugosi, G. 2014a. Regret in Online Combinatorial Optimization. *Math. Oper. Res.* 39(1): 31–45. ISSN 0364-765X. doi:10.1287/moor.2013.0598. URL http://dx.doi.org/10.1287/moor.2013.0598.

Audibert, J.-Y.; Bubeck, S.; and Lugosi, G. 2014b. Regret in Online Combinatorial Optimization. *Mathematics of Operations Research* 39(1): 31–45.

Auer, P. 2002. Using Confidence Bounds for Exploitation-Exploration Trade-offs. *Journal of Machine Learning Research* 3: 397–422.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47(2-3): 235–256.

Auer, P.; and Ortner, R. 2010. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica* 61(1-2): 55–65.

Bercu, B.; Delyon, B.; and Rio, E. 2015. *Concentration inequalities for sums and martingales*. Springer.

Besson, L.; and Kaufmann, E. 2018. What Doubling Tricks Can and Can't Do for Multi-Armed Bandits .

Cesa-Bianchi, N.; and Lugosi, G. 2012. Combinatorial Bandits. *J. Comput. Syst. Sci.* 78(5): 1404–1422. ISSN 0022-0000. doi:10.1016/j.jcss.2012.01.001. URL http://dx.doi.org/10.1016/j.jcss.2012.01.001.

Chen, W.; Wang, Y.; and Yuan, Y. 2013. Combinatorial Multi-Armed Bandit: General Framework and Applications. In Dasgupta, S.; and McAllester, D., eds., *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, 151–159. Atlanta, Georgia, USA: PMLR.

Cohen, A.; Hazan, T.; and Koren, T. 2017. Tight Bounds for Bandit Combinatorial Optimization. In *Conference on Learning Theory*, 629–642.

Dani, V.; Hayes, T.; and Kakade, S. 2008. Stochastic Linear Optimization under Bandit Feedback. In *Proceedings of the 21st Annual Conference on Learning Theory*, 355–366.

Dani, V.; Kakade, S. M.; and Hayes, T. P. 2008. The Price of Bandit Information for Online Optimization. In Platt, J. C.; Koller, D.; Singer, Y.; and Roweis, S. T., eds., *Advances in Neural Information Processing Systems 20*, 345–352. Curran Associates, Inc.

Filippi, S.; Cappe, O.; Garivier, A.; and Szepesvari, C. 2010. Parametric Bandits: The Generalized Linear Case. In *Advances in Neural Information Processing Systems 23*, 586–594.

Gai, Y.; Krishnamachari, B.; and Jain, R. 2010. Learning multiuser channel allocations in cognitive radio networks: A combinatorial multi-armed bandit formulation. In *New Frontiers in Dynamic Spectrum, 2010 IEEE Symposium on*, 1–9. IEEE.

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

Gopalan, A.; Mannor, S.; and Mansour, Y. 2014. Thompson Sampling for Complex Online Problems. In *Proceedings of the 31st International Conference on Machine Learning*, 100–108.

Hoeffding, W. 1994. Probability inequalities for sums of bounded random variables. In *The Collected Works of Wassily Hoeffding*, 409–426. Springer.

Jun, K.-S.; Bhargava, A.; Nowak, R.; and Willett, R. 2017. Scalable Generalized Linear Bandits: Online Computation and Hashing. In *Advances in Neural Information Processing Systems 30*, 98–108.

Kalyanakrishnan, S.; Tewari, A.; Auer, P.; and Stone, P. 2012. PAC Subset Selection in Stochastic Multi-armed Bandits. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*. icml.cc / Omnipress. URL http://icml.cc/2012/papers/359.pdf.

Kveton, B.; Szepesvari, C.; Wen, Z.; and Ashkan, A. 2015a. Cascading bandits: Learning to rank in the cascade model. In *International Conference on Machine Learning*, 767–776.

Kveton, B.; Wen, Z.; Ashkan, A.; Eydgahi, H.; and Eriksson, B. 2014. Matroid bandits: Fast combinatorial optimization with learning. *arXiv preprint arXiv:1403.5045* .

Kveton, B.; Wen, Z.; Ashkan, A.; and Szepesvari, C. 2015b. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*, 535–543.

Lattimore, T.; Kveton, B.; Li, S.; and Szepesvari, C. 2018. Toprank: A practical algorithm for online stochastic ranking. In *Advances in Neural Information Processing Systems*, 3945–3954.

Li, L.; Chu, W.; Langford, J.; and Schapire, R. E. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, 661–670.

Li, L.; Lu, Y.; and Zhou, D. 2017. Provably Optimal Algorithms for Generalized Linear Contextual Bandits. In *Pro-*

ceedings of the 34th International Conference on Machine Learning, 2071–2080.

Liau, D.; Song, Z.; Price, E.; and Yang, G. 2018. Stochastic Multi-armed Bandits in Constant Space. In Storkey, A.; and Perez-Cruz, F., eds., *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, 386–394. Playa Blanca, Lanzarote, Canary Islands: PMLR. URL http://proceedings.mlr.press/v84/liau18a.html.

Lin, T.; Abrahao, B.; Kleinberg, R.; Lui, J.; and Chen, W. 2014. Combinatorial Partial Monitoring Game with Linear Feedback and Its Applications. In Xing, E. P.; and Jebara, T., eds., *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, 901–909. Bejing, China: PMLR.

Nuara, A.; Trovo, F.; Gatti, N.; Restelli, M.; et al. 2018. A Combinatorial-Bandit Algorithm for the Online Joint Bid/Budget Optimization of Pay-per-Click Advertising Campaigns. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 1840–1846.

Raymond Chi-Wing Wong; Ada Wai-Chee Fu; and Wang, K. 2003. MPIS: maximal-profit item selection with cross-selling considerations. In *Third IEEE International Conference on Data Mining*, 371–378.

Rejwan, I.; and Mansour, Y. 2020. Top-k Combinatorial Bandits with Full-Bandit Feedback. In *Algorithmic Learning Theory*, 752–776.

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

Xiang, Y.; Lan, T.; Aggarwal, V.; and Chen, Y.-F. R. 2016. Joint latency and cost optimization for erasure-coded data center storage. *IEEE/ACM Transactions on Networking (TON)* 24(4): 2443–2457.

Zhang, W.; Zhang, Y.; Gao, B.; Yu, Y.; Yuan, X.; and Liu, T.-Y. 2012. Joint optimization of bid and budget allocation in sponsored search. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1177–1185. ACM.