Combinatorial Multi-Armed Bandit with General Reward Functions

Wei Chen[*] **Wei Hu**[†] **Fu Li**[‡] **Jian Li**[§] **Yu Liu**[¶] **Pinyan Lu**[‖]

Abstract

In this paper, we study the stochastic combinatorial multi-armed bandit (CMAB) framework that allows a general nonlinear reward function, whose expected value may not depend only on the means of the input random variables but possibly on the entire distributions of these variables. Our framework enables a much larger class of reward functions such as the $\max()$ function and nonlinear utility functions. Existing techniques relying on accurate estimations of the means of random variables, such as the upper confidence bound (UCB) technique, do not work directly on these functions. We propose a new algorithm called *stochastically dominant confidence bound (SDCB)*, which estimates the distributions of underlying random variables and their stochastically dominant confidence bounds. We prove that SDCB can achieve $O(\log T)$ distribution-dependent regret and $\tilde{O}(\sqrt{T})$ distribution-independent regret, where T is the time horizon. We apply our results to the K-MAX problem and expected utility maximization problems. In particular, for K-MAX, we provide the first polynomial-time approximation scheme (PTAS) for its offline problem, and give the first $\tilde{O}(\sqrt{T})$ bound on the $(1-\epsilon)$-approximation regret of its online problem, for any $\epsilon > 0$.

1 Introduction

Stochastic multi-armed bandit (MAB) is a classical online learning problem typically specified as a player against m machines or arms. Each arm, when pulled, generates a random reward following an unknown distribution. The task of the player is to select one arm to pull in each round based on the historical rewards she collected, and the goal is to collect cumulative reward over multiple rounds as much as possible. In this paper, unless otherwise specified, we use MAB to refer to stochastic MAB.

MAB problem demonstrates the key tradeoff between exploration and exploitation: whether the player should stick to the choice that performs the best so far, or should try some less explored alternatives that may provide better rewards. The performance measure of an MAB strategy is its cumulative *regret*, which is defined as the difference between the cumulative reward obtained by always playing the arm with the largest expected reward and the cumulative reward achieved by the learning strategy. MAB and its variants have been extensively studied in the literature, with classical results such as tight $\Theta(\log T)$ distribution-dependent and $\Theta(\sqrt{T})$ distribution-independent upper and lower bounds on the regret in T rounds [19, 2, 1].

An important extension to the classical MAB problem is combinatorial multi-armed bandit (CMAB). In CMAB, the player selects not just one arm in each round, but a subset of arms or a combinatorial

[*]Microsoft Research, email: weic@microsoft.com. The authors are listed in alphabetical order.
[†]Princeton University, email: huwei@cs.princeton.edu.
[‡]The University of Texas at Austin, email: fuli.theory.research@gmail.com.
[§]Tsinghua University, email: lapordge@gmail.com.
[¶]Tsinghua University, email: liuyujyyz@gmail.com.
[‖]Shanghai University of Finance and Economics, email: lu.pinyan@mail.shufe.edu.cn.

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

object in general, referred to as a super arm, which collectively provides a random reward to the player. The reward depends on the outcomes from the selected arms. The player may observe partial feedbacks from the selected arms to help her in decision making. CMAB has wide applications in online advertising, online recommendation, wireless routing, dynamic channel allocations, etc., because in all these settings the action unit is a combinatorial object (e.g. a set of advertisements, a set of recommended items, a route in a wireless network, and an allocation between channels and users), and the reward depends on unknown stochastic behaviors (e.g. users' click through behaviors, wireless transmission quality, etc.). Therefore CMAB has attracted a lot of attention in online learning research in recent years [12, 8, 22, 15, 7, 16, 18, 17, 23, 9].

Most of these studies focus on linear reward functions, for which the expected reward for playing a super arm is a linear combination of the expected outcomes from the constituent base arms. Even for studies that do generalize to non-linear reward functions, they typically still assume that the expected reward for choosing a super arm is a function of the expected outcomes from the constituent base arms in this super arm [8, 17]. However, many natural reward functions do not satisfy this property. For example, for the function $\max()$, which takes a group of variables and outputs the maximum one among them, its expectation depends on the full distributions of the input random variables, not just their means. Function $\max()$ and its variants underly many applications. As an illustrative example, we consider the following scenario in auctions: the auctioneer is repeatedly selling an item to m bidders; in each round the auctioneer selects K bidders to bid; each of the K bidders independently draws her bid from her private valuation distribution and submits the bid; the auctioneer uses the first-price auction to determine the winner and collects the largest bid as the payment.[1] The goal of the auctioneer is to gain as high cumulative payments as possible. We refer to this problem as the K-MAX bandit problem, which cannot be effectively solved in the existing CMAB framework.

Beyond the K-MAX problem, many expected utility maximization (EUM) problems are studied in stochastic optimization literature [27, 20, 21, 4]. The problem can be formulated as maximizing $\mathbb{E}[u(\sum_{i\in S} X_i)]$ among all feasible sets S, where X_i's are independent random variables and $u(\cdot)$ is a utility function. For example, X_i could be the random delay of edge e_i in a routing graph, S is a routing path in the graph, and the objective is maximizing the utility obtained from any routing path, and typically the shorter the delay, the larger the utility. The utility function $u(\cdot)$ is typically nonlinear to model risk-averse or risk-prone behaviors of users (e.g. a concave utility function is often used to model risk-averse behaviors). The non-linear utility function makes the objective function much more complicated: in particular, it is no longer a function of the means of the underlying random variables X_i's. When the distributions of X_i's are unknown, we can turn EUM into an online learning problem where the distributions of X_i's need to be learned over time from online feedbacks, and we want to maximize the cumulative reward in the learning process. Again, this is not covered by the existing CMAB framework since only learning the means of X_i's is not enough.

In this paper, we generalize the existing CMAB framework with semi-bandit feedbacks to handle general reward functions, where the expected reward for playing a super arm may depend more than just the means of the base arms, and the outcome distribution of a base arm can be arbitrary. This generalization is non-trivial, because almost all previous works on CMAB rely on estimating the expected outcomes from base arms, while in our case, we need an estimation method and an analytical tool to deal with the whole distribution, not just its mean. To this end, we turn the problem into estimating the cumulative distribution function (CDF) of each arm's outcome distribution. We use *stochastically dominant confidence bound (SDCB)* to obtain a distribution that stochastically dominates the true distribution with high probability, and hence we also name our algorithm `SDCB`. We are able to show $O(\log T)$ distribution-dependent and $\tilde{O}(\sqrt{T})$ distribution-independent regret bounds in T rounds. Furthermore, we propose a more efficient algorithm called `Lazy-SDCB`, which first executes a discretization step and then applies `SDCB` on the discretized problem. We show that `Lazy-SDCB` also achieves $\tilde{O}(\sqrt{T})$ distribution-independent regret bound. Our regret bounds are tight with respect to their dependencies on T (up to a logarithmic factor for distribution-independent bounds). To make our scheme work, we make a few reasonable assumptions, including boundedness, monotonicity and Lipschitz-continuity[2] of the reward function, and independence among base arms. We apply our algorithms to the K-MAX and EUM problems, and provide efficient solutions with concrete regret bounds. Along the way, we also provide the first polynomial time approximation

[1]We understand that the first-price auction is not truthful, but this example is only for illustrative purpose for the $\max()$ function.

[2]The Lipschitz-continuity assumption is only made for `Lazy-SDCB`. See Section 4.

scheme (PTAS) for the offline K-MAX problem, which is formulated as maximizing $\mathbb{E}[\max_{i \in S} X_i]$ subject to a cardinality constraint $|S| \leq K$, where X_i's are independent nonnegative random variables.

To summarize, our contributions include: (a) generalizing the CMAB framework to allow a general reward function whose expectation may depend on the entire distributions of the input random variables; (b) proposing the SDCB algorithm to achieve efficient learning in this framework with near-optimal regret bounds, even for arbitrary outcome distributions; (c) giving the first PTAS for the offline K-MAX problem. Our general framework treats any offline stochastic optimization algorithm as an oracle, and effectively integrates it into the online learning framework.

Related Work. As already mentioned, most relevant to our work are studies on CMAB frameworks, among which [12, 16, 18, 9] focus on linear reward functions while [8, 17] look into non-linear reward functions. In particular, Chen et al. [8] look at general non-linear reward functions and Kveton et al. [17] consider specific non-linear reward functions in a conjunctive or disjunctive form, but both papers require that the expected reward of playing a super arm is determined by the expected outcomes from base arms.

The only work in combinatorial bandits we are aware of that does not require the above assumption on the expected reward is [15], which is based on a general Thompson sampling framework. However, they assume that the joint distribution of base arm outcomes is from a known parametric family within known likelihood function and only the parameters are unknown. They also assume the parameter space to be finite. In contrast, our general case is non-parametric, where we allow arbitrary bounded distributions. Although in our known finite support case the distribution can be parametrized by probabilities on all supported points, our parameter space is continuous. Moreover, it is unclear how to efficiently compute posteriors in their algorithm, and their regret bounds depend on complicated problem-dependent coefficients which may be very large for many combinatorial problems. They also provide a result on the K-MAX problem, but they only consider Bernoulli outcomes from base arms, much simpler than our case where general distributions are allowed.

There are extensive studies on the classical MAB problem, for which we refer to a survey by Bubeck and Cesa-Bianchi [5]. There are also some studies on adversarial combinatorial bandits, e.g. [26, 6]. Although it bears conceptual similarities with stochastic CMAB, the techniques used are different.

Expected utility maximization (EUM) encompasses a large class of stochastic optimization problems and has been well studied (e.g. [27, 20, 21, 4]). To the best of our knowledge, we are the first to study the online learning version of these problems, and we provide a general solution to systematically address all these problems as long as there is an available offline (approximation) algorithm. The K-MAX problem may be traced back to [13], where Goel et al. provide a constant approximation algorithm to a generalized version in which the objective is to choose a subset S of cost at most K and maximize the expectation of a certain knapsack profit.

2 Setup and Notation

Problem Formulation. We model a combinatorial multi-armed bandit (CMAB) problem as a tuple $(E, \mathcal{F}, D, R)$, where $E = [m] = \{1, 2, \ldots, m\}$ is a set of m (base) arms, $\mathcal{F} \subseteq 2^E$ is a set of subsets of E, D is a probability distribution over $[0,1]^m$, and R is a reward function defined on $[0,1]^m \times \mathcal{F}$. The arms produce stochastic outcomes $X = (X_1, X_2, \ldots, X_m)$ drawn from distribution D, where the i-th entry X_i is the outcome from the i-th arm. Each feasible subset of arms $S \in \mathcal{F}$ is called a *super arm*. Under a realization of outcomes $x = (x_1, \ldots, x_m)$, the player receives a reward $R(x, S)$ when she chooses the super arm S to play. Without loss of generality, we assume the reward value to be nonnegative. Let $K = \max_{S \in \mathcal{F}} |S|$ be the maximum size of any super arm.

Let $X^{(1)}, X^{(2)}, \ldots$ be an i.i.d. sequence of random vectors drawn from D, where $X^{(t)} = (X_1^{(t)}, \ldots, X_m^{(t)})$ is the outcome vector generated in the t-th round. In the t-th round, the player chooses a super arm $S_t \in \mathcal{F}$ to play, and then the outcomes from all arms in S_t, i.e., $\{X_i^{(t)} \mid i \in S_t\}$, are revealed to the player. According to the definition of the reward function, the reward value in the t-th round is $R(X^{(t)}, S_t)$. The expected reward for choosing a super arm S in any round is denoted by $r_D(S) = \mathbb{E}_{X \sim D}[R(X, S)]$.

We also assume that for a fixed super arm $S \in \mathcal{F}$, the reward $R(x, S)$ only depends on the revealed outcomes $x_S = (x_i)_{i \in S}$. Therefore, we can alternatively express $R(x, S)$ as $R_S(x_S)$, where R_S is a function defined on $[0,1]^S$.[3]

A learning algorithm $\mathcal{A}$ for the CMAB problem selects which super arm to play in each round based on the revealed outcomes in all previous rounds. Let $S_t^{\mathcal{A}}$ be the super arm selected by $\mathcal{A}$ in the t-th round.[4] The goal is to maximize the expected cumulative reward in T rounds, which is $\mathbb{E}\left[\sum_{t=1}^{T} R(X^{(t)}, S_t^{\mathcal{A}})\right] = \sum_{t=1}^{T} \mathbb{E}\left[r_D(S_t^{\mathcal{A}})\right]$. Note that when the underlying distribution D is known, the optimal algorithm $\mathcal{A}^*$ chooses the optimal super arm $S^* = \operatorname{argmax}_{S \in \mathcal{F}}\{r_D(S)\}$ in every round. The quality of an algorithm $\mathcal{A}$ is measured by its *regret* in T rounds, which is the difference between the expected cumulative reward of the optimal algorithm $\mathcal{A}^*$ and that of $\mathcal{A}$:

$$\mathsf{Reg}_D^{\mathcal{A}}(T) = T \cdot r_D(S^*) - \sum_{t=1}^{T} \mathbb{E}\left[r_D(S_t^{\mathcal{A}})\right].$$

For some CMAB problem instances, the optimal super arm S^* may be computationally hard to find even when the distribution D is known, but efficient approximation algorithms may exist, i.e., an α-approximate ($0 < \alpha \le 1$) solution $S' \in \mathcal{F}$ which satisfies $r_D(S') \ge \alpha \cdot \max_{S \in \mathcal{F}}\{r_D(S)\}$ can be efficiently found given D as input. We will provide the exact formulation of our requirement on such an *α-approximation computation oracle* shortly. In such cases, it is not fair to compare a CMAB algorithm $\mathcal{A}$ with the optimal algorithm $\mathcal{A}^*$ which always chooses the optimal super arm S^*. Instead, we define the *α-approximation regret* of an algorithm $\mathcal{A}$ as

$$\mathsf{Reg}_{D,\alpha}^{\mathcal{A}}(T) = T \cdot \alpha \cdot r_D(S^*) - \sum_{t=1}^{T} \mathbb{E}\left[r_D(S_t^{\mathcal{A}})\right].$$

As mentioned, almost all previous work on CMAB requires that the expected reward $r_D(S)$ of a super arm S depends only on the expectation vector $\mu = (\mu_1, \ldots, \mu_m)$ of outcomes, where $\mu_i = \mathbb{E}_{X \sim D}[X_i]$. This is a strong restriction that cannot be satisfied by a general nonlinear function R_S and a general distribution D. The main motivation of this work is to remove this restriction.

Assumptions. Throughout this paper, we make several assumptions on the outcome distribution D and the reward function R.

Assumption 1 (Independent outcomes from arms)**.** *The outcomes from all m arms are mutually independent, i.e., for $X \sim D$, $X_1, X_2, \ldots, X_m$ are mutually independent. We write D as $D = D_1 \times D_2 \times \cdots \times D_m$, where D_i is the distribution of X_i.*

We remark that the above independence assumption is also made for past studies on the offline EUM and K-MAX problems [27, 20, 21, 4, 13], so it is not an extra assumption for the online learning case.

Assumption 2 (Bounded reward value)**.** *There exists $M > 0$ such that for any $x \in [0,1]^m$ and any $S \in \mathcal{F}$, we have $0 \le R(x, S) \le M$.*

Assumption 3 (Monotone reward function)**.** *If two vectors $x, x' \in [0,1]^m$ satisfy $x_i \le x'_i$ ($\forall i \in [m]$), then for any $S \in \mathcal{F}$, we have $R(x, S) \le R(x', S)$.*

Computation Oracle for Discrete Distributions with Finite Supports. We require that there exists an α-approximation computation oracle ($0 < \alpha \le 1$) for maximizing $r_D(S)$, when each D_i ($i \in [m]$) has a *finite support*. In this case, D_i can be fully described by a finite set of numbers (i.e., its support $\{v_{i,1}, v_{i,2}, \ldots, v_{i,s_i}\}$ and the values of its cumulative distribution function (CDF) F_i on the supported points: $F_i(v_{i,j}) = \Pr_{X_i \sim D_i}\left[X_i \le v_{i,j}\right]$ ($j \in [s_i]$)). The oracle takes such a representation of D as input, and can output a super arm $S' = \mathsf{Oracle}(D) \in \mathcal{F}$ such that $r_D(S') \ge \alpha \cdot \max_{S \in \mathcal{F}}\{r_D(S)\}$.

3 SDCB Algorithm

[3] $[0,1]^S$ is isomorphic to $[0,1]^{|S|}$; the coordinates in $[0,1]^S$ are indexed by elements in S.

[4] Note that $S_t^{\mathcal{A}}$ may be random due to the random outcomes in previous rounds and the possible randomness used by $\mathcal{A}$.

Algorithm 1 SDCB (Stochastically dominant confidence bound)

1: Throughout the algorithm, for each arm $i \in [m]$, maintain: (i) a counter T_i which stores the number of times arm i has been played so far, and (ii) the empirical distribution $\hat{D}_i$ of the observed outcomes from arm i so far, which is represented by its CDF $\hat{F}_i$

2: // Initialization
3: **for** $i = 1$ **to** m **do**
4:     // Action in the i-th round
5:     Play a super arm S_i that contains arm i
6:     Update T_j and $\hat{F}_j$ for each $j \in S_i$
7: **end for**

8: **for** $t = m+1, m+2, \ldots$ **do**
9:     // Action in the t-th round
10:     For each $i \in [m]$, let $\underline{D}_i$ be a distribution whose CDF $\underline{F}_i$ is

$$\underline{F}_i(x) = \begin{cases} \max\{\hat{F}_i(x) - \sqrt{\frac{3\ln t}{2T_i}}, 0\}, & 0 \le x < 1 \\ 1, & x = 1 \end{cases}$$

11:     Play the super arm $S_t \leftarrow \mathsf{Oracle}(\underline{D})$, where $\underline{D} = \underline{D}_1 \times \underline{D}_2 \times \cdots \times \underline{D}_m$
12:     Update T_j and $\hat{F}_j$ for each $j \in S_t$
13: **end for**

We present our algorithm *stochastically dominant confidence bound (SDCB)* in Algorithm 1. Throughout the algorithm, we store, in a variable T_i, the number of times the outcomes from arm i are observed so far. We also maintain the empirical distribution $\hat{D}_i$ of the observed outcomes from arm i so far, which can be represented by its CDF $\hat{F}_i$: for $x \in [0,1]$, the value of $\hat{F}_i(x)$ is just the fraction of the observed outcomes from arm i that are no larger than x. Note that $\hat{F}_i$ is always a step function which has "jumps" at the points that are observed outcomes from arm i. Therefore it suffices to store these discrete points as well as the values of $\hat{F}_i$ at these points in order to store the whole function $\hat{F}_i$. Similarly, the later computation of stochastically dominant CDF $\underline{F}_i$ (line 10) only requires computation at these points, and the input to the offline oracle only needs to provide these points and corresponding CDF values (line 11).

The algorithm starts with m initialization rounds in which each arm is played at least once[5] (lines 2-7). In the t-th round ($t > m$), the algorithm consists of three steps. First, it calculates for each $i \in [m]$ a distribution $\underline{D}_i$ whose CDF $\underline{F}_i$ is obtained by lowering the CDF $\hat{F}_i$ (line 10). The second step is to call the α-approximation oracle with the newly constructed distribution $\underline{D} = \underline{D}_1 \times \cdots \times \underline{D}_m$ as input (line 11), and thus the super arm S_t output by the oracle satisfies $r_{\underline{D}}(S_t) \ge \alpha \cdot \max_{S \in \mathcal{F}}\{r_{\underline{D}}(S)\}$. Finally, the algorithm chooses the super arm S_t to play, observes the outcomes from all arms in S_t, and updates T_j's and $\hat{F}_j$'s accordingly for each $j \in S_t$.

The idea behind our algorithm is the *optimism in the face of uncertainty* principle, which is the key principle behind UCB-type algorithms. Our algorithm ensures that with high probability we have $\underline{F}_i(x) \le F_i(x)$ simultaneously for all $i \in [m]$ and all $x \in [0,1]$, where F_i is the CDF of the outcome distribution D_i. This means that each $\underline{D}_i$ has *first-order stochastic dominance* over D_i.[6] Then from the monotonicity property of $R(x, S)$ (Assumption 3) we know that $r_{\underline{D}}(S) \ge r_D(S)$ holds for all $S \in \mathcal{F}$ with high probability. Therefore $\underline{D}$ provides an "optimistic" estimation on the expected reward from each super arm.

Regret Bounds. We prove $O(\log T)$ distribution-dependent and $O(\sqrt{T \log T})$ distribution-independent upper bounds on the regret of SDCB (Algorithm 1).

[5]Without loss of generality, we assume that each arm $i \in [m]$ is contained in at least one super arm.

[6]We remark that while $\underline{F}_i(x)$ is a numerical lower confidence bound on $F_i(x)$ for all $x \in [0,1]$, at the distribution level, $\underline{D}_i$ serves as a "stochastically dominant (upper) confidence bound" on D_i.

We call a super arm S *bad* if $r_D(S) < \alpha \cdot r_D(S^*)$. For each super arm S, we define

$$\Delta_S = \max\{\alpha \cdot r_D(S^*) - r_D(S), 0\}.$$

Let $\mathcal{F}_{\mathrm{B}} = \{S \in \mathcal{F} \mid \Delta_S > 0\}$, which is the set of all *bad* super arms. Let $E_{\mathrm{B}} \subseteq [m]$ be the set of arms that are contained in at least one *bad* super arm. For each $i \in E_{\mathrm{B}}$, we define

$$\Delta_{i,\min} = \min\{\Delta_S \mid S \in \mathcal{F}_{\mathrm{B}}, i \in S\}.$$

Recall that M is an upper bound on the reward value (Assumption 2) and $K = \max_{S \in \mathcal{F}} |S|$.

Theorem 1. *A distribution-dependent upper bound on the α-approximation regret of `SDCB` (Algorithm 1) in T rounds is*

$$M^2 K \sum_{i \in E_{\mathrm{B}}} \frac{2136}{\Delta_{i,\min}} \ln T + \left(\frac{\pi^2}{3} + 1\right) \alpha M m,$$

and a distribution-independent upper bound is

$$93 M \sqrt{mKT \ln T} + \left(\frac{\pi^2}{3} + 1\right) \alpha M m.$$

The proof of Theorem 1 is given in Appendix A.1. The main idea is to reduce our analysis on general reward functions satisfying Assumptions 1-3 to the one in [18] that deals with the *summation* reward function $R(x, S) = \sum_{i \in S} x_i$. Our analysis relies on the Dvoretzky-Kiefer-Wolfowitz inequality [10, 24], which gives a uniform concentration bound on the empirical CDF of a distribution.

Applying Our Algorithm to the Previous CMAB Framework. Although our focus is on general reward functions, we note that when `SDCB` is applied to the previous CMAB framework where the expected reward depends only on the means of the random variables, it can achieve the same regret bounds as the previous *combinatorial upper confidence bound (`CUCB`)* algorithm in [8, 18].

Let $\mu_i = \mathbb{E}_{X \sim D}[X_i]$ be arm i's mean outcome. In each round `CUCB` calculates (for each arm i) an upper confidence bound $\bar{\mu}_i$ on μ_i, with the essential property that $\mu_i \le \bar{\mu}_i \le \mu_i + \Lambda_i$ holds with high probability, for some $\Lambda_i > 0$. In `SDCB`, we use $\underline{D}_i$ as a stochastically dominant confidence bound of D_i. We can show that $\mu_i \le \mathbb{E}_{Y_i \sim \underline{D}_i}[Y_i] \le \mu_i + \Lambda_i$ holds with high probability, with the same interval length Λ_i as in `CUCB`. (The proof is given in Appendix A.2.) Hence, the analysis in [8, 18] can be applied to `SDCB`, resulting in the same regret bounds.We further remark that in this case we *do not* need the three assumptions stated in Section 2 (in particular the independence assumption on X_i's): the summation reward case just works as in [18] and the nonlinear reward case relies on the properties of monotonicity and bounded smoothness used in [8].

4 Improved SDCB Algorithm by Discretization

In Section 3, we have shown that our algorithm `SDCB` achieves near-optimal regret bounds. However, that algorithm might suffer from large running time and memory usage. Note that, in the t-th round, an arm i might have been observed $t-1$ times already, and it is possible that all the observed values from arm i are different (e.g., when arm i's outcome distribution D_i is continuous). In such case, it takes $\Theta(t)$ space to store the empirical CDF $\hat{F}_i$ of the observed outcomes from arm i, and both calculating the stochastically dominant CDF $\underline{F}_i$ and updating $\hat{F}_i$ take $\Theta(t)$ time. Therefore, the worst-case space usage of `SDCB` in T rounds is $\Theta(T)$, and the worst-case running time is $\Theta(T^2)$ (ignoring the dependence on m and K); here we do not count the time and space used by the offline computation oracle.

In this section, we propose an improved algorithm `Lazy-SDCB` which reduces the worst-case memory usage and running time to $O(\sqrt{T})$ and $O(T^{3/2})$, respectively, while preserving the $O(\sqrt{T \log T})$ distribution-independent regret bound. To this end, we need an additional assumption on the reward function:

Assumption 4 (Lipschitz-continuous reward function). *There exists $C > 0$ such that for any $S \in \mathcal{F}$ and any $x, x' \in [0,1]^m$, we have $|R(x, S) - R(x', S)| \le C \|x_S - x'_S\|_1$, where $\|x_S - x'_S\|_1 = \sum_{i \in S} |x_i - x'_i|$.*

Algorithm 2 `Lazy-SDCB` with known time horizon

Input: time horizon T

$s \leftarrow \lceil \sqrt{T} \rceil$

$I_j \leftarrow \begin{cases} [0, \frac{1}{s}], & j = 1 \\ (\frac{j-1}{s}, \frac{j}{s}], & j = 2, \ldots, s \end{cases}$

Invoke `SDCB` (Algorithm 1) for T rounds, with the following change: whenever observing an outcome x (from any arm), find $j \in [s]$ such that $x \in I_j$, and regard this outcome as $\frac{j}{s}$

Algorithm 3 `Lazy-SDCB` without knowing the time horizon

$q \leftarrow \lceil \log_2 m \rceil$

In rounds $1, 2, \ldots, 2^q$, invoke Algorithm 2 with input $T = 2^q$

for $k = q, q+1, q+2, \ldots$ **do**

  In rounds $2^k + 1, 2^k + 2, \ldots, 2^{k+1}$, invoke Algorithm 2 with input $T = 2^k$

end for

We first describe the algorithm when the time horizon T is known in advance. The algorithm is summarized in Algorithm 2. We perform a *discretization* on the distribution $D = D_1 \times \cdots \times D_m$ to obtain a discrete distribution $\tilde{D} = \tilde{D}_1 \times \cdots \times \tilde{D}_m$ such that (i) for $\tilde{X} \sim \tilde{D}$, $\tilde{X}_1, \ldots, \tilde{X}_m$ are also mutually independent, and (ii) every $\tilde{D}_i$ is supported on a set of equally-spaced values $\{\frac{1}{s}, \frac{2}{s}, \ldots, 1\}$, where s is set to be $\lceil \sqrt{T} \rceil$. Specifically, we partition $[0,1]$ into s intervals: $I_1 = [0, \frac{1}{s}], I_2 = (\frac{1}{s}, \frac{2}{s}], \ldots, I_{s-1} = (\frac{s-2}{s}, \frac{s-1}{s}], I_s = (\frac{s-1}{s}, 1]$, and define $\tilde{D}_i$ as

$$\Pr_{\tilde{X}_i \sim \tilde{D}_i}[\tilde{X}_i = j/s] = \Pr_{X_i \sim D_i}[X_i \in I_j], \qquad j = 1, \ldots, s.$$

For the CMAB problem $([m], \mathcal{F}, D, R)$, our algorithm "pretends" that the outcomes are drawn from $\tilde{D}$ instead of D, by replacing any outcome $x \in I_j$ by $\frac{j}{s}$ ($\forall j \in [s]$), and then applies `SDCB` to the problem $([m], \mathcal{F}, \tilde{D}, R)$. Since each $\tilde{D}_i$ has a known support $\{\frac{1}{s}, \frac{2}{s}, \ldots, 1\}$, the algorithm only needs to maintain the number of occurrences of each support value in order to obtain the empirical CDF of all the observed outcomes from arm i. Therefore, all the operations in a round can be done using $O(s) = O(\sqrt{T})$ time and space, and the total time and space used by `Lazy-SDCB` are $O(T^{3/2})$ and $O(\sqrt{T})$, respectively.

The discretization parameter s in Algorithm 2 depends on the time horizon T, which is why Algorithm 2 has to know T in advance. We can use the doubling trick to avoid the dependency on T. We present such an algorithm (without knowing T) in Algorithm 3. It is easy to see that Algorithm 3 has the same asymptotic time and space usages as Algorithm 2.

Regret Bounds. We show that both Algorithm 2 and Algorithm 3 achieve $O(\sqrt{T \log T})$ distribution-independent regret bounds. The full proofs are given in Appendix B. Recall that C is the coefficient in the Lipschitz condition in Assumption 4.

Theorem 2. *Suppose the time horizon T is known in advance. Then the α-approximation regret of Algorithm 2 in T rounds is at most*

$$93M\sqrt{mKT \ln T} + 2CK\sqrt{T} + \left(\frac{\pi^2}{3} + 1\right)\alpha M m.$$

Proof Sketch. The regret consists of two parts: (i) the regret for the discretized CMAB problem $([m], \mathcal{F}, \tilde{D}, R)$, and (ii) the error due to discretization. We directly apply Theorem 1 for the first part. For the second part, a key step is to show $|r_D(S) - r_{\tilde{D}}(S)| \le CK/s$ for all $S \in \mathcal{F}$ (see Appendix B.1). $\square$

Theorem 3. *For any time horizon $T \geq 2$, the α-approximation regret of Algorithm 3 in T rounds is at most*

$$318M\sqrt{mKT \ln T} + 7CK\sqrt{T} + 10\alpha M m \ln T.$$

5 Applications

We describe the K-MAX problem and the class of expected utility maximization problems as applications of our general CMAB framework.

The K-MAX Problem. In this problem, the player is allowed to select at most K arms from the set of m arms in each round, and the reward is the maximum one among the outcomes from the selected arms. In other words, the set of feasible super arms is $\mathcal{F} = \{S \subseteq [m] \mid |S| \leq K\}$, and the reward function is $R(x, S) = \max_{i \in S} x_i$. It is easy to verify that this reward function satisfies Assumptions 2, 3 and 4 with $M = C = 1$.

Now we consider the corresponding offline K-MAX problem of selecting at most K arms from m independent arms, with the largest expected reward. It can be implied by a result in [14] that finding the exact optimal solution is NP-hard, so we resort to approximation algorithms. We can show, using submodularity, that a simple greedy algorithm can achieve a $(1 - 1/e)$-approximation. Furthermore, we give the first PTAS for this problem. Our PTAS can be generalized to constraints other than the cardinality constraint $|S| \leq K$, including s-t simple paths, matchings, knapsacks, etc. The algorithms and corresponding proofs are given in Appendix C.

Theorem 4. *There exists a PTAS for the offline K-MAX problem. In other words, for any constant $\epsilon > 0$, there is a polynomial-time $(1 - \epsilon)$-approximation algorithm for the offline K-MAX problem.*

We thus can apply our `SDCB` algorithm to the K-MAX bandit problem and obtain $O(\log T)$ distribution-dependent and $\tilde{O}(\sqrt{T})$ distribution-independent regret bounds according to Theorem 1, or can apply `Lazy-SDCB` to get $\tilde{O}(\sqrt{T})$ distribution-independent bound according to Theorem 2 or 3.

Streeter and Golovin [26] study an *online submodular maximization* problem in the oblivious adversary model. In particular, their result can cover the stochastic K-MAX bandit problem as a special case, and an $O(K\sqrt{mT \log m})$ upper bound on the $(1 - 1/e)$-regret can be shown. While the techniques in [26] can only give a bound on the $(1 - 1/e)$-approximation regret for K-MAX, we can obtain the first $\tilde{O}(\sqrt{T})$ bound on the $(1 - \epsilon)$-approximation regret for any constant $\epsilon > 0$, using our PTAS as the offline oracle. Even when we use the simple greedy algorithm as the oracle, our experiments show that `SDCB` performs significantly better than the algorithm in [26] (see Appendix D).

Expected Utility Maximization. Our framework can also be applied to reward functions of the form $R(x, S) = u(\sum_{i \in S} x_i)$, where $u(\cdot)$ is an increasing utility function. The corresponding offline problem is to maximize the expected utility $\mathbb{E}[u(\sum_{i \in S} x_i)]$ subject to a feasibility constraint $S \in \mathcal{F}$. Note that if u is nonlinear, the expected utility may not be a function of the means of the arms in S. Following the celebrated von Neumann-Morgenstern expected utility theorem, nonlinear utility functions have been extensively used to capture risk-averse or risk-prone behaviors in economics (see e.g., [11]), while linear utility functions correspond to risk-neutrality.

Li and Deshpande [20] obtain a PTAS for the expected utility maximization (EUM) problem for several classes of utility functions (including for example increasing concave functions which typically indicate risk-averseness), and a large class of feasibility constraints (including cardinality constraint, s-t simple paths, matchings, and knapsacks). Similar results for other utility functions and feasibility constraints can be found in [27, 21, 4]. In the online problem, we can apply our algorithms, using their PTASs as the offline oracle. Again, we can obtain the first tight regret bounds on the $(1 - \epsilon)$-approximation regret for any $\epsilon > 0$, for the class of online EUM problems.

Acknowledgments

Wei Chen was supported in part by the National Natural Science Foundation of China (Grant No. 61433014). Jian Li and Yu Liu were supported in part by the National Basic Research Program of China grants 2015CB358700, 2011CBA00300, 2011CBA00301, and the National NSFC grants 61033001, 61361136003. The authors would like to thank Tor Lattimore for referring to us the DKW inequality.

References

[1] Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *COLT*, pages 217–226, 2009.

[2] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

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

[4] Anand Bhalgat and Sanjeev Khanna. A utility equivalence theorem for concave functions. In *IPCO*, pages 126–137. Springer, 2014.

[5] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.

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

[7] Shouyuan Chen, Tian Lin, Irwin King, Michael R. Lyu, and Wei Chen. Combinatorial pure exploration of multi-armed bandits. In *NIPS*, 2014.

[8] Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *Journal of Machine Learning Research*, 17(50):1–33, 2016.

[9] Richard Combes, M. Sadegh Talebi, Alexandre Proutiere, and Marc Lelarge. Combinatorial bandits revisited. In *NIPS*, 2015.

[10] Aryeh Dvoretzky, Jack Kiefer, and Jacob Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, pages 642–669, 1956.

[11] P. C. Fishburn. *The foundations of expected utility*. Dordrecht: Reidel, 1982.

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

[13] Ashish Goel, Sudipto Guha, and Kamesh Munagala. Asking the right questions: Model-driven optimization using probes. In *PODS*, pages 203–212. ACM, 2006.

[14] Ashish Goel, Sudipto Guha, and Kamesh Munagala. How to probe for an extreme value. *ACM Transactions on Algorithms (TALG)*, 7(1):12:1–12:20, 2010.

[15] Aditya Gopalan, Shie Mannor, and Yishay mansour. Thompson sampling for complex online problems. In *ICML*, pages 100–108, 2014.

[16] Branislav Kveton, Zheng Wen, Azin Ashkan, Hoda Eydgahi, and Brian Eriksson. Matroid bandits: Fast combinatorial optimization with learning. In *UAI*, pages 420–429, 2014.

[17] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvári. Combinatorial cascading bandits. In *NIPS*, 2015.

[18] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvári. Tight regret bounds for stochastic combinatorial semi-bandits. In *AISTATS*, pages 535–543, 2015.

[19] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[20] Jian Li and Amol Deshpande. Maximizing expected utility for stochastic combinatorial optimization problems. In *FOCS*, pages 797–806, 2011.

[21] Jian Li and Wen Yuan. Stochastic combinatorial optimization via poisson approximation. In *STOC*, pages 971–980, 2013.

[22] Tian Lin, Bruno Abrahao, Robert Kleinberg, John Lui, and Wei Chen. Combinatorial partial monitoring game with linear feedback and its applications. In *ICML*, pages 901–909, 2014.

[23] Tian Lin, Jian Li, and Wei Chen. Stochastic online greedy learning with semi-bandit feedbacks. In *NIPS*, 2015.

[24] Pascal Massart. The tight constant in the dvoretzky-kiefer-wolfowitz inequality. *The Annals of Probability*, pages 1269–1283, 1990.

[25] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions – I. *Mathematical Programming*, 14(1):265–294, 1978.

[26] Matthew Streeter and Daniel Golovin. An online algorithm for maximizing submodular functions. In *NIPS*, 2008.

[27] Jiajin Yu and Shabbir Ahmed. Maximizing expected utility over a knapsack constraint. *Operations Research Letters*, 44(2):180–185, 2016.

Appendix

A Missing Proofs from Section 3

A.1 Proof of Theorem 1

We present the proof of Theorem 1 in four steps. In Section A.1.1, we review the L_1 distance between two distributions and present a property of it. In Section A.1.2, we review the Dvoretzky-Kiefer-Wolfowitz (DKW) inequality, which is a strong concentration result for empirical CDFs. In Section A.1.3, we prove some key technical lemmas. Then we complete the proof of Theorem 1 in Section A.1.4.

A.1.1 The L_1 Distance between Two Probability Distributions

For simplicity, we only consider discrete distributions with finite supports – this will be enough for our purpose.

Let P be a probability distribution. For any x, let $P(x) = \Pr_{X\sim P}[X = x]$. We write $P = P_1 \times P_2 \times \cdots \times P_n$ if the (multivariate) random variable $X \sim P$ can be written as $X = (X_1, X_2, \ldots, X_n)$, where $X_1, \ldots, X_n$ are mutually independent and $X_i \sim P_i$ ($\forall i \in [n]$).

For two distributions P and Q, their L_1 *distance* is defined as

$$L_1(P, Q) = \sum_x |P(x) - Q(x)|,$$

where the summation is taken over $x \in \mathsf{supp}(P) \cup \mathsf{supp}(Q)$.

The L_1 distance has the following property. It is a folklore result and we provide a proof for completeness.

Lemma 1. *Let $P = P_1 \times P_2 \times \cdots \times P_n$ and $Q = Q_1 \times Q_2 \times \cdots \times Q_n$ be two probability distributions. Then we have*

$$L_1(P, Q) \le \sum_{i=1}^{n} L_1(P_i, Q_i). \tag{1}$$

Proof. We prove (1) by induction on n.

When $n = 2$, we have

$$\begin{aligned}
L_1(P, Q) &= \sum_x \sum_y |P(x, y) - Q(x, y)| \\
&= \sum_x \sum_y |P_1(x)P_2(y) - Q_1(x)Q_2(y)| \\
&\le \sum_x \sum_y (|P_1(x)P_2(y) - P_1(x)Q_2(y)| + |P_1(x)Q_2(y) - Q_1(x)Q_2(y)|) \\
&= \sum_x P_1(x) \sum_y |P_2(y) - Q_2(y)| + \sum_y Q_2(y) \sum_x |P_1(x) - Q_1(x)| \\
&= 1 \cdot L_1(P_2, Q_2) + 1 \cdot L_1(P_1, Q_1) \\
&= \sum_{i=1}^{2} L_1(P_i, Q_i).
\end{aligned}$$

Here the summation is taken over $x \in \mathsf{supp}(P_1) \cup \mathsf{supp}(Q_1)$ and $y \in \mathsf{supp}(P_2) \cup \mathsf{supp}(Q_2)$.

Suppose (1) is proved for $n = k - 1$ ($k \ge 3$). When $n = k$, using the results for $n = k - 1$ and $n = 2$, we get

$$L_1(P, Q) \le \sum_{i=1}^{k-2} L_1(P_i, Q_i) + L_1(P_{k-1} \times P_k, Q_{k-1} \times Q_k)$$

$$\leq \sum_{i=1}^{k-2} L_1(P_i, Q_i) + L_1(P_{k-1}, Q_{k-1}) + L_1(P_k, Q_k)$$
$$= \sum_{i=1}^{k} L_1(P_i, Q_i).$$

This completes the proof. □

A.1.2 The DKW Inequality

Consider a distribution D with CDF $F(x)$. Let $\hat{F}_n(x)$ be the empirical CDF of n i.i.d. samples $X_1, \ldots, X_n$ drawn from D, i.e., $\hat{F}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\{X_i \leq x\}$ $(x \in \mathbb{R})$.[7] Then we have:

Lemma 2 (Dvoretzky-Kiefer-Wolfowitz inequality [10, 24])**.** *For any $\epsilon > 0$ and any $n \in \mathbb{Z}_+$, we have*

$$\Pr\left[\sup_{x\in\mathbb{R}} \left|\hat{F}_n(x) - F(x)\right| \geq \epsilon\right] \leq 2e^{-2n\epsilon^2}.$$

Note that for any fixed $x \in \mathbb{R}$, from the Chernoff bound we have $\Pr\left[\left|\hat{F}_n(x) - F(x)\right| \geq \epsilon\right] \leq 2e^{-2n\epsilon^2}$. The DKW inequality states a stronger guarantee that the Chernoff concentration holds simultaneously for all $x \in \mathbb{R}$.

A.1.3 Technical Lemmas

The following lemma describes some properties of the expected reward $r_P(S) = \mathbb{E}_{X\sim P}[R(X, S)]$.

Lemma 3. *Let $P = P_1 \times \cdots \times P_m$ and $P' = P'_1 \times \cdots \times P'_m$ be two probability distributions over $[0,1]^m$. Let F_i and F'_i be the CDFs of P_i and P'_i, respectively ($i = 1, \ldots, m$). Suppose each P_i ($i \in [m]$) is a discrete distribution with finite support.*

(i) If for any $i \in [m], x \in [0,1]$ we have $F'_i(x) \leq F_i(x)$, then for any super arm $S \in \mathcal{F}$, we have

$$r_{P'}(S) \geq r_P(S).$$

(ii) If for any $i \in [m], x \in [0,1]$ we have $F_i(x) - F'_i(x) \leq \Lambda_i$ ($\Lambda_i > 0$), then for any super arm $S \in \mathcal{F}$, we have

$$r_{P'}(S) - r_P(S) \leq 2M \sum_{i\in S} \Lambda_i.$$

Proof. It is easy to see why (i) is true. If we have $F'_i(x) \leq F_i(x)$ for all $i \in [m]$ and $x \in [0,1]$, then for all i, P'_i has first-order stochastic dominance over P_i. When we change the distribution from P_i into P'_i, we are moving some probability mass from smaller values to larger values. Recall that the reward function $R(x, S)$ has a monotonicity property (Assumption 3): if x and x' are two vectors in $[0,1]^m$ such that $x_i \leq x'_i$ for all $i \in [m]$, then $R(x, S) \leq R(x', S)$ for all $S \in \mathcal{F}$. Therefore we have $r_P(S) \leq r_{P'}(S)$ for all $S \in \mathcal{F}$.

Now we prove (ii). Without loss of generality, we assume $S = \{1, 2, \ldots, n\}$ $(n \leq m)$. Let $P'' = P''_1 \times \cdots \times P''_m$ be a distribution over $[0,1]^m$ such that the CDF of P''_i is the following:

$$F''_i(x) = \begin{cases} \max\{F_i(x) - \Lambda_i, 0\}, & 0 \leq x < 1, \\ 1, & x = 1. \end{cases} \tag{2}$$

It is easy to see that $F''_i(x) \leq F'_i(x)$ for all $i \in [m]$ and $x \in [0,1]$. Thus from the result in (i) we have

$$r_{P'}(S) \leq r_{P''}(S). \tag{3}$$

[7] We use $\mathbb{1}\{\cdot\}$ to denote the indicator function, i.e., $\mathbb{1}\{\mathcal{H}\} = 1$ if an event $\mathcal{H}$ happens, and $\mathbb{1}\{\mathcal{H}\} = 0$ if it does not happen.

Let $\mathsf{supp}(P_i) = \{v_{i,1}, v_{i,2}, \dots, v_{i,s_i}\}$ where $0 \le v_{i,1} < \cdots < v_{i,s_i} \le 1$. Define $P_S = P_1 \times P_2 \times \cdots \times P_n$, and define P'_S and P''_S similarly. Recall that the reward function $R(x, S)$ can be written as $R_S(x_S) = R_S(x_1, \dots, x_n)$. Then we have

$$
\begin{aligned}
& r_{P''}(S) - r_P(S) \\
&= \sum_{x_1,\dots,x_n} R_S(x_1,\dots,x_n) P''_S(x_1,\dots,x_n) - \sum_{x_1,\dots,x_n} R_S(x_1,\dots,x_n) P_S(x_1,\dots,x_n) \\
&= \sum_{x_1,\dots,x_n} R_S(x_1,\dots,x_n) \cdot \left(P''_S(x_1,\dots,x_n) - P_S(x_1,\dots,x_n)\right) \\
&\le \sum_{x_1,\dots,x_n} M \cdot \left|P''_S(x_1,\dots,x_n) - P_S(x_1,\dots,x_n)\right| \\
&= M \cdot L_1(P''_S, P_S),
\end{aligned}
$$

where the summation is taken over $x_i \in \{v_{i,1}, \dots, v_{i,s_i}\}$ ($\forall i \in S$). Then using Lemma 1 we obtain

$$
r_{P''}(S) - r_P(S) \le M \cdot \sum_{i \in S} L_1(P''_i, P_i). \tag{4}
$$

Now we give an upper bound on $L_1(P''_i, P_i)$ for each i. Let $F_{i,j} = F_i(v_{i,j})$, $F''_{i,j} = F''_i(v_{i,j})$, and $F_{i,0} = F''_{i,0} = 0$. We have

$$
\begin{aligned}
L_1(P''_i, P_i) &= \sum_{j=1}^{s_i} \left|P''_i(v_{i,j}) - P_i(v_{i,j})\right| \\
&= \sum_{j=1}^{s_i} \left|(F''_{i,j} - F''_{i,j-1}) - (F_{i,j} - F_{i,j-1})\right| \\
&= \sum_{j=1}^{s_i} \left|(F_{i,j} - F''_{i,j}) - (F_{i,j-1} - F''_{i,j-1})\right|.
\end{aligned} \tag{5}
$$

In fact, for all $1 \le j < s_i$, we have $F_{i,j} - F''_{i,j} \ge F_{i,j-1} - F''_{i,j-1}$. To see this, consider two cases:

- If $F_{i,j} < \Lambda_i$, then we have $F_{i,j-1} \le F_{i,j} < \Lambda_i$. By definition (2) we have $F''_{i,j} = F''_{i,j-1} = 0$. Thus $F_{i,j} - F''_{i,j} = F_{i,j} \ge F_{i,j-1} = F_{i,j-1} - F''_{i,j-1}$.
- If $F_{i,j} \ge \Lambda_i$, then by definition (2) we have $F_{i,j} - F''_{i,j} = \Lambda_i \ge F_{i,j-1} - F''_{i,j-1}$.

Therefore (5) becomes

$$
\begin{aligned}
L_1(P''_i, P_i) &= \sum_{j=1}^{s_i - 1} \left((F_{i,j} - F''_{i,j}) - (F_{i,j-1} - F''_{i,j-1})\right) + \left|(1-1) - (F_{i,s_i-1} - F''_{i,s_i-1})\right| \\
&= F_{i,s_i-1} - F''_{i,s_i-1} + \left|F_{i,s_i-1} - F''_{i,s_i-1}\right| \\
&= 2\left(F_{i,s_i-1} - F''_{i,s_i-1}\right) \\
&\le 2\Lambda_i,
\end{aligned} \tag{6}
$$

where the last inequality is due to (2).

We complete the proof of the lemma by combining (3), (4) and (6):

$$
r_{P'}(S) - r_P(S) \le r_{P''}(S) - r_P(S) \le M \cdot \sum_{i \in S} L_1(P''_i, P_i) \le 2M \sum_{i \in S} \Lambda_i. \qquad \square
$$

The following lemma is similar to Lemma 1 in [18]. We will use some additional notation:

- For $t \ge m + 1$ and $i \in [m]$, let $T_{i,t}$ be the value of counter T_i right after the t-th round of `SDCB`. In other words, $T_{i,t}$ is the number of observed outcomes from arm i in the first t rounds.

- Let S_t be the super arm selected by SDCB in the t-th round.

Lemma 4. *Define an event in each round t ($m+1 \le t \le T$):*

$$\mathcal{H}_t = \left\{ 0 < \Delta_{S_t} \le 4M \cdot \sum_{i \in S_t} \sqrt{\frac{3 \ln t}{2 T_{i,t-1}}} \right\}. \tag{7}$$

Then the α-approximation regret of SDCB in T rounds is at most

$$\mathbb{E}\left[\sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{H}_t\} \Delta_{S_t} \right] + \left(\frac{\pi^2}{3} + 1 \right) \alpha M m.$$

Proof. Let F_i be the CDF of D_i. Let $\hat{F}_{i,l}$ be the empirical CDF of the first l observations from arm i. For $m+1 \le t \le T$, define an event

$$\mathcal{E}_t = \left\{ \text{there exists } i \in [m] \text{ such that } \sup_{x \in [0,1]} \left| \hat{F}_{i,T_{i,t-1}}(x) - F_i(x) \right| \ge \sqrt{\frac{3 \ln t}{2 T_{i,t-1}}} \right\},$$

which means that the empirical CDF $\hat{F}_i$ is not close enough to the true CDF F_i at the *beginning* of the t-th round.

Recall that we have $S^* = \operatorname{argmax}_{S \in \mathcal{F}}\{r_D(S)\}$ and $\Delta_S = \max\{\alpha \cdot r_D(S^*) - r_D(S), 0\}$ ($S \in \mathcal{F}$). We bound the α-approximation regret of SDCB as

$$\begin{aligned} \mathsf{Reg}_{D,\alpha}^{\mathtt{SDCB}}(T) &= \sum_{t=1}^{T} \mathbb{E}\left[\alpha \cdot r_D(S^*) - r_D(S_t)\right] \le \sum_{t=1}^{T} \mathbb{E}\left[\Delta_{S_t}\right] \\ &= \mathbb{E}\left[\sum_{t=1}^{m} \Delta_{S_t} \right] + \mathbb{E}\left[\sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{E}_t\} \Delta_{S_t} \right] + \mathbb{E}\left[\sum_{t=m+1}^{T} \mathbb{1}\{\neg\mathcal{E}_t\} \Delta_{S_t} \right], \end{aligned} \tag{8}$$

where $\neg\mathcal{E}_t$ is the complement of event $\mathcal{E}_t$.

We separately bound each term in (8).

(a) the first term

The first term in (8) can be trivially bounded as

$$\mathbb{E}\left[\sum_{t=1}^{m} \Delta_{S_t} \right] \le \sum_{t=1}^{m} \alpha \cdot r_D(S^*) \le m \cdot \alpha M. \tag{9}$$

(b) the second term

By the DKW inequality we know that for any $i \in [m], l \ge 1, t \ge m+1$ we have

$$\Pr\left[\sup_{x \in [0,1]} \left| \hat{F}_{i,l}(x) - F_i(x) \right| \ge \sqrt{\frac{3 \ln t}{2l}} \right] \le 2e^{-2l \cdot \frac{3 \ln t}{2l}} = 2e^{-3 \ln t} = 2t^{-3}.$$

Therefore

$$\begin{aligned} \mathbb{E}\left[\sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{E}_t\} \right] &\le \sum_{t=m+1}^{T} \sum_{i=1}^{m} \sum_{l=1}^{t-1} \Pr\left[\left| \hat{F}_{i,j,l} - F_{i,j} \right| \ge \sqrt{\frac{3 \ln t}{2l}} \right] \\ &\le \sum_{t=m+1}^{T} \sum_{i=1}^{m} \sum_{l=1}^{t-1} 2t^{-3} \\ &\le 2m \sum_{t=m+1}^{T} t^{-2} \\ &\le \frac{\pi^2}{3} m, \end{aligned}$$

and then the second term in (8) can be bounded as

$$\mathbb{E}\left[\sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{E}_t\}\Delta_{S_t}\right] \le \frac{\pi^2}{3} m \cdot (\alpha \cdot r_D(S^*)) \le \frac{\pi^2}{3}\alpha M m. \tag{10}$$

(c) the third term

We fix $t > m$ and first assume $\neg\mathcal{E}_t$ happens. Let $c_i = \sqrt{\frac{3\ln t}{2T_{i,t-1}}}$ for each $i \in [m]$. Since $\neg\mathcal{E}_t$ happens, we have

$$\left|\hat{F}_{i,T_{i,t-1}}(x) - F_i(x)\right| < c_i \qquad \forall i \in [m], x \in [0,1]. \tag{11}$$

Recall that in round t of `SDCB` (Algorithm 1), the input to the oracle is $\underline{D} = \underline{D}_1 \times \cdots \times \underline{D}_m$, where the CDF $\underline{F}_i$ of $\underline{D}_i$ is

$$\underline{F}_i(x) = \begin{cases} \max\{\hat{F}_{i,T_{i,t-1}}(x) - c_i, 0\}, & 0 \le x < 1, \\ 1, & x = 1. \end{cases} \tag{12}$$

From (11) and (12) we know that $\underline{F}_i(x) \le F_i(x) \le \underline{F}_i(x) + 2c_i$ for all $i \in [m], x \in [0,1]$. Thus, from Lemma 3 (i) we have

$$r_D(S) \le r_{\underline{D}}(S) \qquad \forall S \in \mathcal{F}, \tag{13}$$

and from Lemma 3 (ii) we have

$$r_{\underline{D}}(S) \le r_D(S) + 2M\sum_{i\in S} 2c_i \qquad \forall S \in \mathcal{F}. \tag{14}$$

Also, from the fact that the algorithm chooses S_t in the t-th round, we have

$$r_{\underline{D}}(S_t) \ge \alpha \cdot \max_{S\in\mathcal{F}}\{r_{\underline{D}}(S)\} \ge \alpha \cdot r_{\underline{D}}(S^*). \tag{15}$$

From (13), (14) and (15) we have

$$\alpha \cdot r_D(S^*) \le \alpha \cdot r_{\underline{D}}(S^*) \le r_{\underline{D}}(S_t) \le r_D(S_t) + 2M\sum_{i\in S_t} 2c_i,$$

which implies

$$\Delta_{S_t} \le 4M\sum_{i\in S_t} c_i.$$

Therefore, when $\neg\mathcal{E}_t$ happens, we always have $\Delta_{S_t} \le 4M\sum_{i\in S_t} c_i$. In other words,

$$\neg\mathcal{E}_t \Longrightarrow \left\{\Delta_{S_t} \le 4M\sum_{i\in S_t}\sqrt{\frac{3\ln t}{2T_{i,t-1}}}\right\}.$$

This implies

$$\{\neg\mathcal{E}_t, \Delta_{S_t} > 0\} \Longrightarrow \left\{0 < \Delta_{S_t} \le 4M\sum_{i\in S_t}\sqrt{\frac{3\ln t}{2T_{i,t-1}}}\right\} = \mathcal{H}_t.$$

Hence, the third term in (8) can be bounded as

$$\mathbb{E}\left[\sum_{t=m+1}^{T} \mathbb{1}\{\neg\mathcal{E}_t\}\Delta_{S_t}\right] = \mathbb{E}\left[\sum_{t=m+1}^{T} \mathbb{1}\{\neg\mathcal{E}_t, \Delta_{S_t} > 0\}\Delta_{S_t}\right] \le \mathbb{E}\left[\sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{H}_t\}\Delta_{S_t}\right]. \tag{16}$$

Finally, by combining (8), (9), (10) and (16) we have

$$\mathrm{Reg}_{D,\alpha}^{\mathtt{SDCB}}(T) \le \mathbb{E}\left[\sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{H}_t\}\Delta_{S_t}\right] + \left(\frac{\pi^2}{3} + 1\right)\alpha M m,$$

completing the proof of the lemma. □

A.1.4 Finishing the Proof of Theorem 1

Lemma 4 is very similar to Lemma 1 in [18]. We now apply the counting argument in [18] to finish the proof of Theorem 1.

From Lemma 4 we know that it remains to bound $\mathbb{E}\left[\sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{H}_t\}\Delta_{S_t}\right]$, where $\mathcal{H}_t$ is defined in (7).

Define two decreasing sequences of positive constants

$$\begin{aligned} 1 = \beta_0 > &\beta_1 > \beta_2 > \dots \\ &\alpha_1 > \alpha_2 > \dots \end{aligned}$$

such that $\lim_{k\to\infty} \alpha_k = \lim_{k\to\infty} \beta_k = 0$. We choose $\{\alpha_k\}$ and $\{\beta_k\}$ as in Theorem 4 of [18], which satisfy

$$\sqrt{6}\sum_{k=1}^{\infty} \frac{\beta_{k-1}-\beta_k}{\sqrt{\alpha_k}} \le 1 \tag{17}$$

and

$$\sum_{k=1}^{\infty} \frac{\alpha_k}{\beta_k} < 267. \tag{18}$$

For $t \in \{m+1, \dots, T\}$ and $k \in \mathbb{Z}_+$, let

$$m_{k,t} = \begin{cases} \alpha_k \left(\frac{2MK}{\Delta_{S_t}}\right)^2 \ln T & \Delta_{S_t} > 0, \\ +\infty & \Delta_{S_t} = 0, \end{cases}$$

and

$$A_{k,t} = \{i \in S_t \mid T_{i,t-1} \le m_{k,t}\}.$$

Then we define an event

$$\mathcal{G}_{k,t} = \{|A_{k,t}| \ge \beta_k K\},$$

which means "in the t-th round, at least $\beta_k K$ arms in S_t had been observed at most $m_{k,t}$ times."

Lemma 5. *In the t-th round ($m+1 \le t \le T$), if event $\mathcal{H}_t$ happens, then there exists $k \in \mathbb{Z}_+$ such that event $\mathcal{G}_{k,t}$ happens.*

Proof. Assume that $\mathcal{H}_t$ happens and that none of $\mathcal{G}_{1,t}, \mathcal{G}_{2,t}, \dots$ happens. Then $|A_{k,t}| < \beta_k K$ for all $k \in \mathbb{Z}_+$.

Let $A_{0,t} = S_t$ and $\bar{A}_{k,t} = S_t \setminus A_{k,t}$ for $k \in \mathbb{Z}_+ \cup \{0\}$. It is easy to see $\bar{A}_{k-1,t} \subseteq \bar{A}_{k,t}$ for all $k \in \mathbb{Z}_+$. Note that $\lim_{k\to\infty} m_{k,t} = 0$. Thus there exists $N \in \mathbb{Z}_+$ such that $\bar{A}_{k,t} = S_t$ for all $k \ge N$, and then we have $S_t = \bigcup_{k=1}^{\infty} \left(\bar{A}_{k,t} \setminus \bar{A}_{k-1,t}\right)$. Finally, note that for all $i \in \bar{A}_{k,t}$, we have $T_{i,t-1} > m_{k,t}$. Therefore

$$\begin{aligned} \sum_{i\in S_t} \frac{1}{\sqrt{T_{i,t-1}}} &= \sum_{k=1}^{\infty} \sum_{i\in \bar{A}_{k,t}\setminus \bar{A}_{k-1,t}} \frac{1}{\sqrt{T_{i,t-1}}} \le \sum_{k=1}^{\infty} \sum_{i\in \bar{A}_{k,t}\setminus \bar{A}_{k-1,t}} \frac{1}{\sqrt{m_{k,t}}} \\ &= \sum_{k=1}^{\infty} \frac{\left|\bar{A}_{k,t} \setminus \bar{A}_{k-1,t}\right|}{\sqrt{m_{k,t}}} = \sum_{k=1}^{\infty} \frac{\left|A_{k-1,t} \setminus A_{k,t}\right|}{\sqrt{m_{k,t}}} = \sum_{k=1}^{\infty} \frac{|A_{k-1,t}| - |A_{k,t}|}{\sqrt{m_{k,t}}} \\ &= \frac{|S_t|}{\sqrt{m_{1,t}}} + \sum_{k=1}^{\infty} |A_{k,t}| \left(\frac{1}{\sqrt{m_{k+1,t}}} - \frac{1}{\sqrt{m_{k,t}}}\right) \\ &< \frac{K}{\sqrt{m_{1,t}}} + \sum_{k=1}^{\infty} \beta_k K \left(\frac{1}{\sqrt{m_{k+1,t}}} - \frac{1}{\sqrt{m_{k,t}}}\right) \\ &= \sum_{k=1}^{\infty} \frac{(\beta_{k-1}-\beta_k)K}{\sqrt{m_{k,t}}}. \end{aligned}$$

Note that we assume $\mathcal{H}_t$ happens. Then we have

$$\Delta_{S_t} \leq 4M \cdot \sum_{i \in S_t} \sqrt{\frac{3 \ln t}{2T_{i,t-1}}} \leq 2M\sqrt{6 \ln T} \cdot \sum_{i \in S_t} \frac{1}{\sqrt{T_{i,t-1}}}$$
$$< 2M\sqrt{6 \ln T} \cdot \sum_{k=1}^{\infty} \frac{(\beta_{k-1} - \beta_k)K}{\sqrt{m_{k,t}}} = \sqrt{6} \sum_{k=1}^{\infty} \frac{\beta_{k-1} - \beta_k}{\sqrt{\alpha_k}} \cdot \Delta_{S_t} \leq \Delta_{S_t},$$

where the last inequality is due to (17). We reach a contradiction here. The proof of the lemma is completed. $\square$

By Lemma 5 we have

$$\sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{H}_t\}\Delta_{S_t} \leq \sum_{k=1}^{\infty} \sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{G}_{k,t}, \Delta_{S_t} > 0\}\Delta_{S_t}.$$

For $i \in [m], k \in \mathbb{Z}_+, t \in \{m+1, \ldots, T\}$, define an event

$$\mathcal{G}_{i,k,t} = \mathcal{G}_{k,t} \wedge \{i \in S_t, T_{i,t-1} \leq m_{k,t}\}.$$

Then by the definitions of $\mathcal{G}_{k,t}$ and $\mathcal{G}_{i,k,t}$ we have

$$\mathbb{1}\{\mathcal{G}_{k,t}, \Delta_{S_t} > 0\} \leq \frac{1}{\beta_k K} \sum_{i \in E_{\mathrm{B}}} \mathbb{1}\{\mathcal{G}_{i,k,t}, \Delta_{S_t} > 0\}.$$

Therefore

$$\sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{H}_t\}\Delta_{S_t} \leq \sum_{i \in E_{\mathrm{B}}} \sum_{k=1}^{\infty} \sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{G}_{i,k,t}, \Delta_{S_t} > 0\} \frac{\Delta_{S_t}}{\beta_k K}.$$

For each arm $i \in E_{\mathrm{B}}$, suppose i is contained in N_i *bad* super arms $S^{\mathrm{B}}_{i,1}, S^{\mathrm{B}}_{i,2}, \ldots, S^{\mathrm{B}}_{i,N_i}$. Let $\Delta_{i,l} = \Delta_{S^{\mathrm{B}}_{i,l}}$ ($l \in [N_i]$). Without loss of generality, we assume $\Delta_{i,1} \geq \Delta_{i,2} \geq \ldots \geq \Delta_{i,N_i}$. Note that $\Delta_{i,N_i} = \Delta_{i,\min}$. For convenience, we also define $\Delta_{i,0} = +\infty$, i.e., $\alpha_k \left(\frac{2MK}{\Delta_{i,0}}\right)^2 = 0$. Then we have

$$\sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{H}_t\}\Delta_{S_t}$$
$$\leq \sum_{i \in E_{\mathrm{B}}} \sum_{k=1}^{\infty} \sum_{t=m+1}^{T} \sum_{l=1}^{N_i} \mathbb{1}\{\mathcal{G}_{i,k,t}, S_t = S^{\mathrm{B}}_{i,l}\} \frac{\Delta_{S_t}}{\beta_k K}$$
$$\leq \sum_{i \in E_{\mathrm{B}}} \sum_{k=1}^{\infty} \sum_{t=m+1}^{T} \sum_{l=1}^{N_i} \mathbb{1}\{T_{i,t-1} \leq m_{k,t}, S_t = S^{\mathrm{B}}_{i,l}\} \frac{\Delta_{i,l}}{\beta_k K}$$
$$= \sum_{i \in E_{\mathrm{B}}} \sum_{k=1}^{\infty} \sum_{t=m+1}^{T} \sum_{l=1}^{N_i} \mathbb{1}\left\{T_{i,t-1} \leq \alpha_k \left(\frac{2MK}{\Delta_{i,l}}\right)^2 \ln T, S_t = S^{\mathrm{B}}_{i,l}\right\} \frac{\Delta_{i,l}}{\beta_k K}$$
$$= \sum_{i \in E_{\mathrm{B}}} \sum_{k=1}^{\infty} \sum_{t=m+1}^{T} \sum_{l=1}^{N_i} \sum_{j=1}^{l} \mathbb{1}\left\{\alpha_k \left(\frac{2MK}{\Delta_{i,j-1}}\right)^2 \ln T < T_{i,t-1} \leq \alpha_k \left(\frac{2MK}{\Delta_{i,j}}\right)^2 \ln T, S_t = S^{\mathrm{B}}_{i,l}\right\} \frac{\Delta_{i,l}}{\beta_k K}$$
$$\leq \sum_{i \in E_{\mathrm{B}}} \sum_{k=1}^{\infty} \sum_{t=m+1}^{T} \sum_{l=1}^{N_i} \sum_{j=1}^{l} \mathbb{1}\left\{\alpha_k \left(\frac{2MK}{\Delta_{i,j-1}}\right)^2 \ln T < T_{i,t-1} \leq \alpha_k \left(\frac{2MK}{\Delta_{i,j}}\right)^2 \ln T, S_t = S^{\mathrm{B}}_{i,l}\right\} \frac{\Delta_{i,j}}{\beta_k K}$$
$$\leq \sum_{i \in E_{\mathrm{B}}} \sum_{k=1}^{\infty} \sum_{t=m+1}^{T} \sum_{l=1}^{N_i} \sum_{j=1}^{N_i} \mathbb{1}\left\{\alpha_k \left(\frac{2MK}{\Delta_{i,j-1}}\right)^2 \ln T < T_{i,t-1} \leq \alpha_k \left(\frac{2MK}{\Delta_{i,j}}\right)^2 \ln T, S_t = S^{\mathrm{B}}_{i,l}\right\} \frac{\Delta_{i,j}}{\beta_k K}$$
$$\leq \sum_{i \in E_{\mathrm{B}}} \sum_{k=1}^{\infty} \sum_{t=m+1}^{T} \sum_{j=1}^{N_i} \mathbb{1}\left\{\alpha_k \left(\frac{2MK}{\Delta_{i,j-1}}\right)^2 \ln T < T_{i,t-1} \leq \alpha_k \left(\frac{2MK}{\Delta_{i,j}}\right)^2 \ln T\right\} \frac{\Delta_{i,j}}{\beta_k K}$$

$$\leq \sum_{i \in E_{\mathrm{B}}} \sum_{k=1}^{\infty} \sum_{j=1}^{N_i} \left(\alpha_k \left(\frac{2MK}{\Delta_{i,j}} \right)^2 \ln T - \alpha_k \left(\frac{2MK}{\Delta_{i,j-1}} \right)^2 \ln T \right) \frac{\Delta_{i,j}}{\beta_k K}$$
$$= 4M^2 K \left(\sum_{k=1}^{\infty} \frac{\alpha_k}{\beta_k} \right) \ln T \cdot \sum_{i \in E_{\mathrm{B}}} \sum_{j=1}^{N_i} \left(\frac{1}{\Delta_{i,j}^2} - \frac{1}{\Delta_{i,j-1}^2} \right) \Delta_{i,j}$$
$$\leq 1068 M^2 K \ln T \cdot \sum_{i \in E_{\mathrm{B}}} \sum_{j=1}^{N_i} \left(\frac{1}{\Delta_{i,j}^2} - \frac{1}{\Delta_{i,j-1}^2} \right) \Delta_{i,j},$$

where the last inequality is due to (18).

Finally, for each $i \in E_{\mathrm{B}}$ we have

$$\begin{aligned} \sum_{j=1}^{N_i} \left(\frac{1}{\Delta_{i,j}^2} - \frac{1}{\Delta_{i,j-1}^2} \right) \Delta_{i,j} &= \frac{1}{\Delta_{i,N_i}} + \sum_{j=1}^{N_i - 1} \frac{1}{\Delta_{i,j}^2} (\Delta_{i,j} - \Delta_{i,j+1}) \\ &\leq \frac{1}{\Delta_{i,N_i}} + \int_{\Delta_{i,N_i}}^{\Delta_{i,1}} \frac{1}{x^2} \, \mathrm{d}x \\ &= \frac{2}{\Delta_{i,N_i}} - \frac{1}{\Delta_{i,1}} \\ &< \frac{2}{\Delta_{i,\min}}. \end{aligned}$$

It follows that

$$\sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{H}_t\} \Delta_{S_t} \leq 1068 M^2 K \ln T \cdot \sum_{i \in E_{\mathrm{B}}} \frac{2}{\Delta_{i,\min}} = M^2 K \sum_{i \in E_{\mathrm{B}}} \frac{2136}{\Delta_{i,\min}} \ln T. \tag{19}$$

Combining (19) with Lemma 4, the distribution-dependent regret bound in Theorem 1 is proved.

To prove the distribution-independent bound, we decompose $\sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{H}_t\} \Delta_{S_t}$ into two parts:

$$\begin{aligned} \sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{H}_t\} \Delta_{S_t} &= \sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{H}_t, \Delta_{S_t} \leq \epsilon\} \Delta_{S_t} + \sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{H}_t, \Delta_{S_t} > \epsilon\} \Delta_{S_t} \\ &\leq \epsilon T + \sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{H}_t, \Delta_{S_t} > \epsilon\} \Delta_{S_t}, \end{aligned} \tag{20}$$

where $\epsilon > 0$ is a constant to be determined. The second term can be bounded in the same way as in the proof of the distribution-dependent regret bound, except that we only consider the case $\Delta_{S_t} > \epsilon$. Thus we can replace (19) by

$$\sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{H}_t, \Delta_{S_t} > \epsilon\} \Delta_{S_t} \leq M^2 K \sum_{i \in E_{\mathrm{B}}, \Delta_{i,\min} > \epsilon} \frac{2136}{\Delta_{i,\min}} \ln T \leq M^2 K m \frac{2136}{\epsilon} \ln T. \tag{21}$$

It follows that

$$\sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{H}_t\} \Delta_{S_t} \leq \epsilon T + M^2 K m \frac{2136}{\epsilon} \ln T.$$

Finally, letting $\epsilon = \sqrt{\frac{2136 M^2 K m \ln T}{T}}$, we get

$$\sum_{t=m+1}^{T} \mathbb{1}\{\mathcal{H}_t\} \Delta_{S_t} \leq 2\sqrt{2136 M^2 K m T \ln T} < 93 M \sqrt{m K T \ln T}.$$

Combining this with Lemma 4, we conclude the proof of the distribution-independent regret bound in Theorem 1. $\square$

Algorithm 4 `CUCB` [8, 18]

1: For each arm i, maintain: (i) $\hat{\mu}_i$, the average of all observed outcomes from arm i so far, and (ii) T_i, the number of observed outcomes from arm i so far.
2: // Initialization
3: **for** $i = 1$ **to** m **do**
4:     // Action in the i-th round
5:     Play a super arm S_i that contains arm i, and update $\hat{\mu}_i$ and T_i.
6: **end for**
7: **for** $t = m+1, m+2, \ldots$ **do**
8:     // Action in the t-th round
9:     $\bar{\mu}_i \leftarrow \min\{\hat{\mu}_i + \sqrt{\frac{3 \ln t}{2T_i}}, 1\} \qquad \forall i \in [m]$
10:     Play the super arm $S_t \leftarrow \text{Oracle}(\bar{\mu})$, where $\bar{\mu} = (\bar{\mu}_1, \ldots, \bar{\mu}_m)$.
11:     Update $\hat{\mu}_i$ and T_i for all $i \in S_t$.
12: **end for**

A.2 Analysis of Our Algorithm in the Previous CMAB Framework

We now give an analysis of `SDCB` in the previous CMAB framework, following our discussion in Section 3. We consider the case in which the expected reward only depends on the means of the random variables. Namely, $r_D(S)$ only depends on μ_i's ($i \in S$), where μ_i is arm i's mean outcome. In this case, we can rewrite $r_D(S)$ as $r_\mu(S)$, where $\mu = (\mu_1, \ldots, \mu_m)$ is the vector of means. Note that the offline computation oracle only needs a mean vector as input.

We no longer need the three assumptions (Assumptions 1-3) given in Section 2. In particular, we do not require independence among outcome distributions of all arms (Assumption 1). Although we cannot write D as $D = D_1 \times \cdots \times D_m$, we still let D_i be the outcome distribution of arm i. In this case, D_i is the marginal distribution of D in the i-th component.

We summarize the `CUCB` algorithm [8, 18] in Algorithm 4. It maintains the empirical mean $\hat{\mu}_i$ of the outcomes from each arm i, and stores the number of observed outcomes from arm i in a variable T_i. In each round, it calculates an upper confidence bound (UCB) $\bar{\mu}_i$ of μ_i, Then it uses the UCB vector $\bar{\mu}$ as the input to the oracle, and plays the super arm output by the oracle. In the t-th round ($t > m$), each UCB $\bar{\mu}_i$ has the key property that

$$\mu_i \le \bar{\mu}_i \le \mu_i + 2\sqrt{\frac{3 \ln t}{2T_{i,t-1}}} \tag{22}$$

holds with high probability. (Recall that $T_{i,t-1}$ is the value of T_i after $t-1$ rounds.) To see this, note that we have $|\mu_i - \hat{\mu}_i| \le \sqrt{\frac{3 \ln t}{2T_{i,t-1}}}$ with high probability (by Chernoff bound), and then (22) follows from the definition of $\bar{\mu}_i$ in line 9 of Algorithm 4.

We prove that the same property as (22) also holds for `SDCB`. Consider a fixed $t > m$, and let $\underline{D} = \underline{D}_1 \times \cdots \times \underline{D}_m$ be the input to the oracle in the t-th round of `SDCB`. Let $\nu_i = \mathbb{E}_{Y_i \sim \underline{D}_i}[Y_i]$. We can think that `SDCB` uses the mean vector $\nu = (\nu_1, \ldots, \nu_m)$ as the input to the oracle used by `CUCB`. We now show that for each i, we have

$$\mu_i \le \nu_i \le \mu_i + 2\sqrt{\frac{3 \ln t}{2T_{i,t-1}}} \tag{23}$$

with high probability.

To show (23), we first prove the following lemma.

Lemma 6. *Let P and P' be two distributions over $[0,1]$ with CDFs F and F', respectively. Consider two random variables $Y \sim P$ and $Y' \sim P'$.*

(i) If for all $x \in [0,1]$ we have $F'(x) \le F(x)$, then we have $\mathbb{E}[Y] \le \mathbb{E}[Y']$.

(ii) If for all $x \in [0,1]$ we have $F(x) - F'(x) \le \Lambda$ ($\Lambda > 0$), then we have $\mathbb{E}[Y'] \le \mathbb{E}[Y] + \Lambda$.

Proof. We have

$$\mathbb{E}[Y] = \int_0^1 x \,\mathrm{d}F(x) = (xF(x))\big|_0^1 - \int_0^1 F(x)\,\mathrm{d}x = 1 - \int_0^1 F(x)\,\mathrm{d}x.$$

Similarly, we have

$$\mathbb{E}[Y'] = 1 - \int_0^1 F'(x)\,\mathrm{d}x.$$

Then the lemma holds trivially. □

Now we prove (23). According to the DKW inequality, with high probability we have

$$F_i(x) - 2\sqrt{\frac{3\ln t}{2T_{i,t-1}}} \le \underline{F}_i(x) \le F_i(x) \tag{24}$$

for all $i \in [m]$ and $x \in [0,1]$, where $\underline{F}_i$ is the CDF of $\underline{D}_i$ used in round t of SDCB, and F_i is the CDF of D_i. Suppose (24) holds for all i, x, then for any i, the two distributions D_i and $\underline{D}_i$ satisfy the two conditions in Lemma 6, with $\Lambda = 2\sqrt{\frac{3\ln t}{2T_{i,t-1}}}$; then from Lemma 6 we know that $\mu_i \le \nu_i \le \mu_i + 2\sqrt{\frac{3\ln t}{2T_{i,t-1}}}$. Hence we have shown that (23) holds with high probability.

The fact that (23) holds with high probability means that the mean of $\underline{D}_i$ is also a UCB of μ_i with the same confidence as in CUCB. With this property, the analysis in [8, 18] can also be applied to SDCB, resulting in exactly the same regret bounds.

B Missing Proofs from Section 4

B.1 Analysis of the Discretization Error

The following lemma gives an upper bound on the error due to discretization. Refer to Section 4 for the definition of the discretized distribution $\tilde{D}$.

Lemma 7. *For any $S \in \mathcal{F}$, we have*

$$|r_D(S) - r_{\tilde{D}}(S)| \le \frac{CK}{s}.$$

To prove Lemma 7, we show a slightly more general lemma which gives an upper bound on the discretization error of the expectation of a Lipschitz continuous function.

Lemma 8. *Let $g(x)$ be a Lipschitz continuous function on $[0,1]^n$ such that for any $x, x' \in [0,1]^n$, we have $|g(x) - g(x')| \le C\|x - x'\|_1$, where $\|x - x'\|_1 = \sum_{i=1}^n |x_i - x'_i|$. Let $P = P_1 \times \cdots \times P_n$ be a probability distribution over $[0,1]^n$. Define another distribution $\tilde{P} = \tilde{P}_1 \times \cdots \times \tilde{P}_n$ over $[0,1]^n$ as follows: each $\tilde{P}_i$ ($i \in [n]$) takes values in $\{\frac{1}{s}, \frac{2}{s}, \ldots, 1\}$, and*

$$\Pr_{\tilde{X}_i \sim \tilde{P}_i}[\tilde{X}_i = j/s] = \Pr_{X_i \sim P_i}[X_i \in I_j], \qquad j \in [s],$$

where $I_1 = [0, \frac{1}{s}], I_2 = (\frac{1}{s}, \frac{2}{s}], \ldots, I_{s-1} = (\frac{s-2}{s}, \frac{s-1}{s}], I_s = (\frac{s-1}{s}, 1]$. Then

$$\left|\mathbb{E}_{X\sim P}[g(X)] - \mathbb{E}_{\tilde{X}\sim\tilde{P}}[g(\tilde{X})]\right| \le \frac{C \cdot n}{s}. \tag{25}$$

Proof. Throughout the proof, we consider $X = (X_1, \ldots, X_n) \sim P$ and $\tilde{X} = (\tilde{X}_1, \ldots, \tilde{X}_n) \sim \tilde{P}$. Let $v_j = \frac{j}{s}$ ($j = 0, 1, \ldots, s$) and

$$p_{i,j} = \Pr[\tilde{X}_i = v_j] = \Pr[X_i \in I_j] \qquad i \in [n], j \in [s].$$

We prove (25) by induction on n.

(1) When $n = 1$, we have

$$\mathbb{E}[g(X_1)] = \sum_{j \in [s], p_{1,j} > 0} p_{1,j} \cdot \mathbb{E}\left[g(X_1) \middle| X_1 \in I_j\right]. \tag{26}$$

Since g is continuous, for each $j \in [s]$ such that $p_{1,j} > 0$, there exists $\xi_j \in [v_{j-1}, v_j]$ such that

$$\mathbb{E}\left[g(X_1) | X_1 \in I_j\right] = g(\xi_j)$$

From the Lipschitz continuity of g we have

$$|g(v_j) - g(\xi_j)| \le C|v_j - \xi_j| \le C|v_j - v_{j-1}| = \frac{C}{s}.$$

Hence

$$\begin{aligned}
\left|\mathbb{E}[g(X_1)] - \mathbb{E}[g(\tilde{X}_1)]\right| &= \left| \sum_{j \in [s], p_{1,j} > 0} p_{1,j} \cdot \mathbb{E}[g(X_1) | X_1 \in I_j] - \sum_{j \in [s], p_{1,j} > 0} p_{1,j} \cdot g(v_j) \right| \\
&= \left| \sum_{j \in [s], p_{1,j} > 0} p_{1,j} \cdot g(\xi_j) - \sum_{j \in [s], p_{1,j} > 0} p_{1,j} \cdot g(v_j) \right| \\
&\le \sum_{j \in [s], p_{1,j} > 0} p_{1,j} \cdot |g(\xi_j) - g(v_j)| \\
&\le \sum_{j \in [s], p_{1,j} > 0} p_{1,j} \cdot \frac{C}{s} \\
&= \frac{C}{s}.
\end{aligned}$$

This proves (25) for $n = 1$.

(ii) Suppose (25) is correct for $n = 1, 2, \ldots, k-1$. Now we prove it for $n = k$ ($k \ge 2$).
We define two functions on $[0, 1]^{k-1}$:

$$h(x_1, \ldots, x_{k-1}) = \mathbb{E}_{X_k}[g(x_1, \ldots, x_{k-1}, X_k)]$$

and

$$\tilde{h}(x_1, \ldots, x_{k-1}) = \mathbb{E}_{\tilde{X}_k}[g(x_1, \ldots, x_{k-1}, \tilde{X}_k)].$$

For any fixed $x_1, \ldots, x_{k-1} \in [0, 1]$, the function $g(x_1, \ldots, x_{k-1}, x)$ on $x \in [0, 1]$ is Lipschitz continuous. Therefore from the result for $n = 1$ we have

$$\left| h(x_1, \ldots, x_{k-1}) - \tilde{h}(x_1, \ldots, x_{k-1}) \right| \le \frac{C}{s} \qquad \forall x_1, \ldots, x_{k-1} \in [0, 1].$$

Then we have

$$
\begin{aligned}
&\left|\mathbb{E}[g(X)] - \mathbb{E}[g(\tilde{X})]\right| \\
=& \left|\mathbb{E}_{X_1,\dots,X_{k-1}}\left[\mathbb{E}[g(X)|X_1,\dots,X_{k-1}]\right] - \mathbb{E}[g(\tilde{X})]\right| \\
=& \left|\mathbb{E}_{X_1,\dots,X_{k-1}}\left[h(X_1,\dots,X_{k-1})\right] - \mathbb{E}[g(\tilde{X})]\right| \\
\le& \left|\mathbb{E}_{X_1,\dots,X_{k-1}}[h(X_1,\dots,X_{k-1})] - \mathbb{E}_{X_1,\dots,X_{k-1}}[\tilde{h}(X_1,\dots,X_{k-1})]\right| \\
&+ \left|\mathbb{E}_{X_1,\dots,X_{k-1}}[\tilde{h}(X_1,\dots,X_{k-1})] - \mathbb{E}[g(\tilde{X})]\right| \\
\le& \mathbb{E}_{X_1,\dots,X_{k-1}}\left[\left|h(X_1,\dots,X_{k-1}) - \tilde{h}(X_1,\dots,X_{k-1})\right|\right] \\
&+ \left|\mathbb{E}_{X_1,\dots,X_{k-1},\tilde{X}_k}[g(X_1,\dots,X_{k-1},\tilde{X}_k)] - \mathbb{E}[g(\tilde{X})]\right| \\
\le& \mathbb{E}_{X_1,\dots,X_{k-1}}\left[\frac{C}{s}\right] + \left|\mathbb{E}_{\tilde{X}_k}\left[\mathbb{E}[g(X_1,\dots,X_{k-1},\tilde{X}_k)|\tilde{X}_k] - \mathbb{E}[g(\tilde{X}_1,\dots,\tilde{X}_{k-1},\tilde{X}_k)|\tilde{X}_k]\right]\right| \\
\le& \frac{C}{s} + \mathbb{E}_{\tilde{X}_k}\left[\left|\mathbb{E}[g(X_1,\dots,X_{k-1},\tilde{X}_k)|\tilde{X}_k] - \mathbb{E}[g(\tilde{X}_1,\dots,\tilde{X}_{k-1},\tilde{X}_k)|\tilde{X}_k]\right|\right] \\
=& \frac{C}{s} + \sum_{j\in[s],p_{k,j}>0} p_{k,j}\cdot\left|\mathbb{E}[g(X_1,\dots,X_{k-1},v_j)] - \mathbb{E}[g(\tilde{X}_1,\dots,\tilde{X}_{k-1},v_j)]\right|.
\end{aligned}
\tag{27}
$$

For any $j \in [s]$, the function $g(x_1,\dots,x_{k-1},v_j)$ on $(x_1,\dots,x_{k-1}) \in [0,1]^{k-1}$ is Lipschitz continuous. Then from the induction hypothesis at $n = k-1$, we have

$$
\left|\mathbb{E}[g(X_1,\dots,X_{k-1},v_j)] - \mathbb{E}[g(\tilde{X}_1,\dots,\tilde{X}_{k-1},v_j)]\right| \le \frac{C(k-1)}{s} \qquad \forall j \in [s]. \tag{28}
$$

From (27) and (28) we have

$$
\begin{aligned}
\left|\mathbb{E}[g(X)] - \mathbb{E}[g(\tilde{X})]\right| &\le \frac{C}{s} + \sum_{j\in[s],p_{k,j}>0} p_{k,j}\cdot\frac{C(k-1)}{s} \\
&= \frac{C}{s} + \frac{C(k-1)}{s} \\
&= \frac{Ck}{s}.
\end{aligned}
$$

This concludes the proof for $n = k$. □

Now we prove Lemma 7.

Proof of Lemma 7. We have

$$
r_D(S) = \mathbb{E}_{X\sim D}[R(X,S)] = \mathbb{E}_{X\sim D}[R_S(X_S)] = \mathbb{E}_{X_S\sim D_S}[R_S(X_S)],
$$

where $X_S = (X_i)_{i\in S}$ and $D_S = (D_i)_{i\in S}$. Similarly, we have

$$
r_{\tilde{D}}(S) = \mathbb{E}_{\tilde{X}_S\sim\tilde{D}_S}[R_S(\tilde{X}_S)].
$$

According to Assumption 4, the function R_S defined on $[0,1]^S$ is Lipschitz continuous. Then from Lemma 8 we have

$$
|r_D(S) - r_{\tilde{D}}(S)| = \left|\mathbb{E}_{X_S\sim D_S}[R_S(X_S)] - \mathbb{E}_{\tilde{X}_S\sim\tilde{D}_S}[R_S(\tilde{X}_S)]\right| \le \frac{C\cdot|S|}{s} \le \frac{C\cdot K}{s}.
$$

This completes the proof. □

B.2 Proof of Theorem 2

Proof of Theorem 2. Let $S^* = \text{argmax}_{S \in \mathcal{F}}\{r_D(S)\}$ and $\tilde{S}^* = \text{argmax}_{S \in \mathcal{F}}\{r_{\tilde{D}}(S)\}$ be the optimal super arms in problems $([m], \mathcal{F}, D, R)$ and $([m], \mathcal{F}, \tilde{D}, R)$, respectively. Suppose Algorithm 2 selects super arm S_t in the t-th round $(1 \le t \le T)$. Then its α-approximation regret is bounded as

$$\begin{aligned}
&\mathsf{Reg}_{D,\alpha}^{\text{Alg. 2}}(T) \\
&= T \cdot \alpha \cdot r_D(S^*) - \sum_{t=1}^{T} \mathbb{E}\left[r_D(S_t)\right] \\
&= T \cdot \alpha \left(r_D(S^*) - r_{\tilde{D}}(\tilde{S}^*)\right) + \sum_{t=1}^{T} \mathbb{E}\left[r_{\tilde{D}}(S_t) - r_D(S_t)\right] + \left(T \cdot \alpha \cdot r_{\tilde{D}}(\tilde{S}^*) - \sum_{t=1}^{T} \mathbb{E}\left[r_{\tilde{D}}(S_t)\right]\right) \\
&\le T \cdot \alpha \left(r_D(S^*) - r_{\tilde{D}}(S^*)\right) + \sum_{t=1}^{T} \mathbb{E}\left[r_{\tilde{D}}(S_t) - r_D(S_t)\right] + \mathsf{Reg}_{\tilde{D},\alpha}^{\text{Alg. 1}}(T).
\end{aligned}$$

where the inequality is due to $r_{\tilde{D}}(\tilde{S}^*) \ge r_{\tilde{D}}(S^*)$.

Then from Lemma 7 and the distribution-independent bound in Theorem 1 we have

$$\begin{aligned}
\mathsf{Reg}_{D,\alpha}^{\text{Alg. 2}}(T) &\le T \cdot \alpha \cdot \frac{CK}{s} + T \cdot \frac{CK}{s} + 93M\sqrt{mKT \ln T} + \left(\frac{\pi^2}{3} + 1\right)\alpha Mm \\
&\le 2 \cdot \frac{CKT}{s} + 93M\sqrt{mKT \ln T} + \left(\frac{\pi^2}{3} + 1\right)\alpha Mm \qquad (29) \\
&\le 93M\sqrt{mKT \ln T} + 2CK\sqrt{T} + \left(\frac{\pi^2}{3} + 1\right)\alpha Mm.
\end{aligned}$$

Here in the last two inequalities we have used $\alpha \le 1$ and $s = \lceil\sqrt{T}\rceil \ge \sqrt{T}$. The proof is completed.

□

B.3 Proof of Theorem 3

Proof of Theorem 3. Let $n = \lceil \log_2 T \rceil$. Then we have $2^{n-1} < T \le 2^n$.

If $n \le q = \lceil \log_2 m \rceil$, then $T \le 2m$ and the regret in T rounds is at most $2m \cdot \alpha M$. The regret bound holds trivially.

Now we assume $n \ge q + 1$. Using Theorem 2, we have

$$\begin{aligned}
&\mathsf{Reg}_{D,\alpha}^{\text{Alg. 3}}(T) \\
&\le \mathsf{Reg}_{D,\alpha}^{\text{Alg. 3}}(2^n) \\
&= \mathsf{Reg}_{D,\alpha}^{\text{Alg. 2}}(2^q) + \sum_{k=q}^{n-1} \mathsf{Reg}_{D,\alpha}^{\text{Alg. 2}}(2^k) \\
&\le \mathsf{Reg}_{D,\alpha}^{\text{Alg. 2}}(2m) + \sum_{k=q}^{n-1} \mathsf{Reg}_{D,\alpha}^{\text{Alg. 2}}(2^k) \\
&\le 2m \cdot \alpha M + \sum_{k=q}^{n-1} \left(93M\sqrt{mK \cdot 2^k \ln 2^k} + 2CK\sqrt{2^k} + \left(\frac{\pi^2}{3} + 1\right)\alpha Mm\right) \\
&\le 2\alpha Mm + \left(93M\sqrt{mK \ln 2^{n-1}} + 2CK\right) \cdot \sum_{k=1}^{n-1} \sqrt{2^k} + (n-1) \cdot \left(\frac{\pi^2}{3} + 1\right)\alpha Mm \\
&\le \left(93M\sqrt{mK \ln 2^{n-1}} + 2CK\right) \cdot \frac{\sqrt{2^n}}{\sqrt{2} - 1} + \left(\frac{\pi^2}{3} + 3\right)(n-1) \cdot \alpha Mm
\end{aligned}$$

Algorithm 5 `Greedy-K-MAX`

$S \leftarrow \emptyset$
for $i = 1$ **to** K **do**
  $k \leftarrow \operatorname{argmax}_{j \in [m] \setminus S} r_D(S \cup \{j\})$
  $S \leftarrow S \cup \{k\}$
end for
Output: S

$$
\begin{aligned}
&\leq \left(93M\sqrt{mK \ln T} + 2CK\right) \cdot \frac{\sqrt{2T}}{\sqrt{2}-1} + \left(\frac{\pi^2}{3} + 3\right) \log_2 T \cdot \alpha M m \\
&\leq 318M\sqrt{mKT \ln T} + 7CK\sqrt{T} + 10\alpha M m \ln T.
\end{aligned}
$$
□

C The Offline K-MAX Problem

In this section, we consider the offline K-MAX problem. Recall that we have m independent random variables $\{X_i\}_{i\in[m]}$. X_i follows the discrete distribution D_i with support $\{v_{i,1}, \ldots, v_{i,s_i}\} \subset [0,1]$, and $D = D_1 \times \cdots \times D_m$ is the joint distribution of $X = (X_1, \ldots, X_m)$. Let $p_{i,j} = \Pr[X_i = v_{i,j}]$. Define $r_D(S) = \mathbb{E}_{X\sim D}[\max_{i\in S} X_i]$ and $\mathsf{OPT} = \max_{S:|S|=K} r_D(S)$. Our goal is to find (in polynomial time) a subset $S \subseteq [m]$ of cardinality K such that $r_D(S) \geq \alpha \cdot \mathsf{OPT}$ (for certain constant α).

First, we show that $r_D(S)$ can be calculated in polynomial time given any $S \subseteq [m]$. Let $S = \{i_1, i_2, \ldots, i_n\}$. Note that for $X \sim D$, $\max_{i\in S} X_i$ can only take values in the set $V(S) = \bigcup_{i\in S} \mathsf{supp}(D_i)$. For any $v \in V(S)$, we have

$$
\begin{aligned}
&\Pr_{X\sim D}\left[\max_{i\in S} X_i = v\right] \\
= &\Pr_{X\sim D}[X_{i_1} = v, X_{i_2} \leq v, \ldots, X_{i_n} \leq v] \\
&+ \Pr_{X\sim D}[X_{i_1} < v, X_{i_2} = v, X_{i_3} \leq v, \ldots, X_{i_n} \leq v] \\
&+ \cdots \\
&+ \Pr_{X\sim D}[X_{i_1} < v, \ldots, X_{i_{n-1}} < v, X_{i_n} = v].
\end{aligned}
\tag{30}
$$

Since $X_{i_1}, \ldots, X_{i_n}$ are mutually independent, each probability appearing in (30) can be calculated in polynomial time. Hence for any $v \in V(S)$, $\Pr_{X\sim D}[\max_{i\in S} X_i = v]$ can be calculated in polynomial time using (30). Then $r_D(S)$ can be calculated by

$$
r_D(S) = \sum_{v\in V(S)} v \cdot \Pr_{X\sim D}\left[\max_{i\in S} X_i = v\right]
$$

in polynomial time.

C.1 $(1-1/e)$-Approximation

We now show that a simple greedy algorithm (Algorithm 5) can find a $(1-1/e)$-approximate solution, by proving the submodularity of $r_D(S)$. In fact, this is implied by a slightly more general result [13, Lemma 3.2]. We provide a simple and direct proof for completeness.

Lemma 9. *Algorithm 5 can output a subset S such that $r_D(S) \geq (1-1/e) \cdot \mathsf{OPT}$.*

Proof. For any $x \in [0,1]^m$, let $f_x(S) = \max_{i\in S} x_i$ be a set function defined on $2^{[m]}$. (Define $f_x(\emptyset) = 0$.) We can verify that $f_x(S)$ is monotone and submodular:

- *Monotonicity.* For any $A \subseteq B \subseteq [m]$, we have $f_x(A) = \max_{i\in A} x_i \leq \max_{i\in B} x_i = f_x(B)$.

- *Submodularity.* For any $A \subseteq B \subseteq [m]$ and any $k \in [m] \setminus B$, there are three cases (note that $\max_{i \in A} x_i \le \max_{i \in B} x_i$):
  (i) If $x_k \le \max_{i \in A} x_i$, then $f_x(A \cup \{k\}) - f_x(A) = 0 = f_x(B \cup \{k\}) - f_x(B)$.
  (ii) If $\max_{i \in A} x_i < x_k \le \max_{i \in B} x_i$, then $f_x(A \cup \{k\}) - f_x(A) = x_k - \max_{i \in A} x_i > 0 = f_x(B \cup \{k\}) - f_x(B)$.
  (iii) If $x_k > \max_{i \in B} x_i$, then $f_x(A \cup \{k\}) - f_x(A) = x_k - \max_{i \in A} x_i \ge x_k - \max_{i \in B} x_i = f_x(B \cup \{k\}) - f_x(B)$.

  Therefore, we always have $f_x(A \cup \{k\}) - f_x(A) \ge f_x(B \cup \{i\}) - f_x(B)$. The function $f_x(S)$ is submodular.

For any $S \subseteq [m]$ we have

$$r_D(S) = \sum_{j_1=1}^{s_1} \sum_{j_2=1}^{s_2} \cdots \sum_{j_m=1}^{s_m} f_{(v_{1,j_1}, \ldots, v_{m,j_m})}(S) \prod_{i=1}^{m} p_{i,j_i}.$$

Since each set function $f_{(v_{1,j_1}, \ldots, v_{m,j_m})}(S)$ is monotone and submodular, $r_D(S)$ is a convex combination of monotone submodular functions on $2^{[m]}$. Therefore, $r_D(S)$ is also a monotone submodular function. According to the classical result on submodular maximization [25], the greedy algorithm can find a $(1 - 1/e)$-approximate solution to $\max_{S \subseteq [m], |S| \le K} \{r_D(S)\}$. □

C.2 PTAS

Now we provide a PTAS for the K-MAX problem. In other words, we give an algorithm which, given any fixed constant $0 < \varepsilon < 1/2$, can find a solution S of cardinality $|K|$ such that $r_D(S) \ge (1 - \varepsilon) \cdot \mathsf{OPT}$ in polynomial time.

We first provide an overview of our approach, and then spell out the details later.

1. (Discretization) We first transform each X_i to another discrete distribution $\tilde{X}_i$, such that all $\tilde{X}_i$'s are supported on a set of size $O(1/\varepsilon^2)$.
2. (Computing signatures) For each X_i, we can compute from $\tilde{X}_i$ a signature $\mathsf{Sig}(X_i)$ which is a vector of size $O(1/\varepsilon^2)$. For a set S, we define its signature $\mathsf{Sig}(S)$ to be $\sum_{i \in S} \mathsf{Sig}(X_i)$. We show that if two sets S_1 and S_2 have the same signature, their objective values are close (Lemma 12).
3. (Enumerating signatures) We enumerate all possible signatures (there are polynomial number of them when treating ε as a constant) and try to find the one which is the signature of a set of size K, and the objective value is maximized.

C.2.1 Discretization

We first describe the discretization step. We say that a random variable X follows the Bernoulli distribution $B(v, q)$ if X takes value v with probability q and value 0 with probability $1 - q$. For any discrete distribution, we can rewrite it as the maximum of a set of Bernoulli distributions.

Definition 1. *Let X be a discrete random variable with support $\{v_1, v_2, \ldots, v_s\}$ $(v_1 < v_2 < \cdots < v_s)$ and $\Pr[X = v_j] = p_j$. We define a set of independent Bernoulli random variables $\{Z_j\}_{j \in [s]}$ as*

$$Z_j \sim B\left(v_j \ , \frac{p_j}{\sum_{j' \le j} p_{j'}}\right).$$

We call $\{Z_j\}$ the Bernoulli decomposition of X_i.

Lemma 10. *For a discrete distribution X and its Bernoulli decomposition $\{Z_j\}$, $\max_j\{Z_j\}$ has the same distribution with X.*

Proof. We can easily see the following:

$$\Pr[\max_j\{Z_j\} = v_i] = \Pr[Z_i = v_i] \prod_{i' > i} \Pr[Z_{i'} = 0]$$

Algorithm 6 Discretization

We first run `Greedy-K-MAX` to obtain a solution S_G and let $\mathsf{W} = r_D(S_G)$.
for $i = 1$ **to** m **do**
  Compute the Bernoulli decomposition $\{Z_{i,j}\}_j$ of X_i.
  for all $Z_{i,j}$ **do**
    Create another Bernoulli variable $\tilde{Z}_{i,j}$ as follows:
    if $v_{i,j} > \mathsf{W}/\varepsilon$ **then**
      Let $\tilde{Z}_{i,j} \sim B\left(\frac{\mathsf{W}}{\varepsilon}, \mathbb{E}[Z_{i,j}]\frac{\varepsilon}{\mathsf{W}}\right)$ (Case 1)
    else
      Let $\tilde{Z}_{i,j} = \lfloor \frac{Z_{i,j}}{\varepsilon \mathsf{W}} \rfloor \varepsilon \mathsf{W}$ (Case 2)
    end if
  end for
  Let $\tilde{X}_i = \max_j\{\tilde{Z}_{ij}\}$
end for

$$= \frac{p_i}{\sum_{i' \le i} p_{i'}} \prod_{h>i} \left(1 - \frac{p_h}{\sum_{h' \le h} p_{h'}}\right)$$
$$= \frac{p_i}{\sum_{i' \le i} p_{i'}} \prod_{h>i} \frac{\sum_{h' \le h-1} p_{h'}}{\sum_{h' \le h} p_{h'}} = p_i.$$

Hence, $\Pr[\max_j\{Z_j\} = v_i] = \Pr[X = v_i]$ for all $i \in [s]$. $\square$

Now, we describe how to construct the discretization $\tilde{X}_i$ of X_i for all $i \in [m]$. The pseudocode can be found in Algorithm 6. We first run `Greedy-K-MAX` to obtain a solution S_G. Let $\mathsf{W} = r_D(S_G)$. By Lemma 9, we know that $\mathsf{W} \ge (1 - 1/e)\mathsf{OPT}$. Then we compute the Bernoulli decomposition $\{Z_{i,j}\}_j$ of X_i. For each $Z_{i,j}$, we create another Bernoulli variable $\tilde{Z}_{i,j}$ as follows: Recall that $v_{i,j}$ is the nonzero possible value of Z_{ij}. We distinguish two cases. Case 1: If $v_{i,j} > \mathsf{W}/\varepsilon$, then we let $\tilde{Z}_{i,j} \sim B\left(\frac{\mathsf{W}}{\varepsilon}, \mathbb{E}[Z_{i,j}]\frac{\varepsilon}{\mathsf{W}}\right)$. It is easy to see that $\mathbb{E}[\tilde{Z}_{ij}] = \mathbb{E}[Z_{ij}]$. Case 2: If $v_{i,j} \le \mathsf{W}/\varepsilon$, then we let $\tilde{Z}_{i,j} = \lfloor \frac{Z_{i,j}}{\varepsilon \mathsf{W}} \rfloor \varepsilon \mathsf{W}$. We note that more than one $\tilde{Z}_{ij}$'s may have the same support, and all $\tilde{Z}_{ij}$'s are supported on $\mathsf{DS} = \{0, \varepsilon\mathsf{W}, 2\varepsilon\mathsf{W}, \ldots, \mathsf{W}/\varepsilon\}$. Finally, we let $\tilde{X}_i = \max_j\{\tilde{Z}_{ij}\}$, which is the discretization of X_i. Since $\tilde{X}_i$ is the maximum of a set of Bernoulli distributions, it is also a discrete distribution supported on DS. We can easily compute $\Pr[\tilde{X}_i = v]$ for any $v \in \mathsf{DS}$.

Now, we show that the discretization only incurs a small loss in the objective value. The key is to show that we do not lose much in the transformation from $Z_{i,j}$'s to $\tilde{Z}_{i,j}$'s. We prove a slightly more general lemma as follows.

Lemma 11. *Consider any set of Bernoulli variables $\{Z_i \sim B(a_i, p_i)\}_{1 \le i \le n}$. Assume that $\mathbb{E}[\max_{i\in[n]} Z_i] < c\mathsf{W}$, where c is a constant such that $c\varepsilon < 1/2$. For each Z_i, we create a Bernoulli variable $\tilde{Z}_i$ in the same way as Algorithm 6. Then the following holds:*

$$\mathbb{E}[\max Z_i] \ge \mathbb{E}[\max \tilde{Z}_i] \ge \mathbb{E}[\max Z_i] - (2c+1)\varepsilon\mathsf{W}.$$

Proof. Assume a_1 is the largest among all a_i's.

If $a_1 < \mathsf{W}/\varepsilon$, all $\tilde{Z}_i$ are created in Case 2. In this case, it is obvious to have that

$$\mathbb{E}[\max Z_i] \ge \mathbb{E}[\max \tilde{Z}_i] \ge \mathbb{E}[\max Z_i] - \varepsilon\mathsf{W}.$$

If $a_1 \ge \mathsf{W}/\varepsilon$, the proof is slightly more complicated. Let $L = \{i \mid a_i \ge \mathsf{W}/\varepsilon\}$. We prove by induction on n (i.e., the number of the variables) the following more general claim:

$$\mathbb{E}[\max Z_i] \ge \mathbb{E}[\max \tilde{Z}_i] \ge \mathbb{E}[\max Z_i] - \varepsilon\mathsf{W} - c\sum_{i\in L} \varepsilon a_i p_i. \tag{31}$$

Consider the base case $n = 1$. The lemma holds immediately in Case 1 as $\mathbb{E}[Z_1] = \mathbb{E}[\tilde{Z}_1]$.

Assuming the lemma is true for $n = k$, we show it also holds for $n = k+1$. Recall we have $\tilde{Z}_1 \sim B(\frac{\mathsf{W}}{\varepsilon}, \varepsilon\mathbb{E}[Z_1]/\mathsf{W})$. Thus

$$\begin{aligned}\mathbb{E}[\max_{i\geq 1} Z_i] - \mathbb{E}[\max_{i\geq 1} \tilde{Z}_i] =& a_1p_1 + (1-p_1)\mathbb{E}[\max_{i\geq 2} Z_i] - a_1p_1 - (1-\varepsilon\mathbb{E}[Z_1]/\mathsf{W})\mathbb{E}[\max_{i\geq 2} \tilde{Z}_i] \\ \geq& (1-p_1)\mathbb{E}[\max_{i\geq 2} \tilde{Z}_i] - (1-\varepsilon\mathbb{E}[Z_1]/\mathsf{W})\mathbb{E}[\max_{i\geq 2} \tilde{Z}_i] \\ =& (\varepsilon a_1p_1/\mathsf{W} - p_1)\mathbb{E}[\max_{i\geq 2} \tilde{Z}_i] \geq 0,\end{aligned}$$

where the first inequality follows from the induction hypothesis and the last from $a_1 \geq \mathsf{W}/\varepsilon$. The other direction can be seen as follows:

$$\begin{aligned}\mathbb{E}[\max_{i\geq 1} \tilde{Z}_i] - \mathbb{E}[\max_{i\geq 1} Z_i] =& a_1p_1 + (1-\varepsilon\mathbb{E}[Z_1]/\mathsf{W})\mathbb{E}[\max_{i\geq 2} \tilde{Z}_i] - (a_1p_1 + (1-p_1)\mathbb{E}[\max_{i\geq 2} Z_i]) \\ \geq& (1-\varepsilon\mathbb{E}[Z_1]/\mathsf{W})\mathbb{E}[\max_{i\geq 2} Z_i] - (1-p_1)\mathbb{E}[\max_{i\geq 2} Z_i] - \varepsilon\mathsf{W} - c\sum_{i\in L\setminus\{1\}} \varepsilon a_ip_i \\ \geq& (-\varepsilon\mathbb{E}[Z_1]/\mathsf{W})\mathbb{E}[\max_{i\geq 2} Z_i] - \varepsilon\mathsf{W} - c\sum_{i\in L\setminus\{1\}} \varepsilon a_ip_i \\ \geq& -\varepsilon\mathsf{W} - c\sum_{i\in L} \varepsilon a_ip_i,\end{aligned}$$

where the last inequality holds since $\mathbb{E}[\max_{i\geq 2} Z_i] \leq c\mathsf{W}$. This finishes the proof of (31).

Now, we show that $\sum_{i\in L} a_ip_i \leq 2\mathsf{W}$. This can be seen as follows. First, we can see from Markov inequality that

$$\Pr[\max Z_i > \mathsf{W}/\varepsilon] \leq c\varepsilon.$$

Equivalently, we have $\prod_{i\in L}(1-p_i) \geq 1 - c\varepsilon$. Then, we can see that

$$\mathsf{W} \geq \sum_{i\in L} a_i \prod_{j<i}(1-p_j)p_i \geq (1-c\varepsilon)\sum_{i\in L} a_ip_i \geq \frac{1}{2}\sum_{i\in L} a_ip_i.$$

Plugging this into (31), we prove the lemma. □

Corollary 1. *For any set $S \subseteq [m]$, suppose $\mathbb{E}[\max_{i\in S} X_i] < c\mathsf{W}$, where c is a constant such that $c\varepsilon < 1/2$. Then the following holds:*

$$\mathbb{E}[\max_{i\in S} X_i] \geq \mathbb{E}[\max_{i\in S} \tilde{X}_i] \geq \mathbb{E}[\max_{i\in S} X_i] - (2c+1)\varepsilon\mathsf{W}.$$

C.2.2 Signatures

For each X_i, we have created its discretization $\tilde{X}_i = \max_j\{\tilde{Z}_{ij}\}$. Since $\tilde{X}_i$ is a discrete distribution, we can define its Bernoulli decomposition $\{Y_{ij}\}_{j\in[h]}$ where $h = |\mathsf{DS}|$. Suppose $Y_{ij} \sim B(j\varepsilon\mathsf{W}, q_{ij})$. Now, we define the signature of X_i to be the vector $\mathsf{Sig}(X_i) = (\mathsf{Sig}(X_i)_1, \ldots, \mathsf{Sig}(X_i)_h)$ where

$$\mathsf{Sig}(X_i)_j = \min\left(\left\lfloor \frac{-\ln(1-q_{ij})}{\varepsilon^4/m} \right\rfloor, \left\lfloor \frac{\ln(1/\varepsilon^4)}{\varepsilon^4/m} \right\rfloor\right) \cdot \frac{\varepsilon^4}{m} \qquad j \in [h].$$

For any set S, define its signature to be

$$\mathsf{Sig}(S) = \sum_{i\in S} \mathsf{Sig}(X_i).$$

Define the set SG of *signature vectors* to be all nonnegative h-dimensional vectors, where each coordinate is an integer multiple of ε^4/m and at most $m\ln(1/\varepsilon^4)$. Clearly, the size of SG is $O\left(\left(m\varepsilon^{-4}\log(h/\varepsilon^2)\right)^{h-1}\right) = \tilde{O}(m^{O(1/\varepsilon^2)})$, which is polynomial for any fixed constant $\varepsilon > 0$ (recall $h = |\mathsf{DS}| = O(1/\varepsilon^2)$).

Now, we prove the following crucial lemma.

Lemma 12. *Consider two sets S_1 and S_2. If $\mathsf{Sig}(S_1) = \mathsf{Sig}(S_2)$, the following holds:*

$$\left|\mathbb{E}[\max_{i\in S_1}\tilde{X}_i] - \mathbb{E}[\max_{i\in S_2}\tilde{X}_i]\right| \le O(\varepsilon)\mathsf{W}.$$

Proof. Suppose $\{Y_{ij}\}_{j\in[h]}$ is the Bernoulli decomposition of $\tilde{X}_i$. For any set S, we define $Y_k(S) = \max_{i\in S} Y_{ik}$ (it is the max of a set of Bernoulli distributions). It is not hard to see that $Y_k(S)$ has a Bernoulli distribution $B(k\varepsilon\mathsf{W}, p_k(S))$ with $p_k(S) = 1 - \prod_{i\in S}(1-q_{ik})$. As $\mathsf{Sig}(S_1) = \mathsf{Sig}(S_2)$, we have that

$$\begin{aligned}
|p_k(S_1) - p_k(S_2)| &= |\prod_{i\in S_1}(1-q_{ik}) - \prod_{i\in S_2}(1-q_{ik})| \\
&= \left|\exp\left(\sum_{i\in S_1}\ln(1-q_{ik})\right) - \exp\left(\sum_{i\in S_2}\ln(1-q_{ik})\right)\right| \\
&\le 2\varepsilon^4 \qquad \forall k\in[h].
\end{aligned}$$

Noticing $\max_{i\in S}\tilde{X}_i = \max_k Y_k(S)$, we have that

$$\begin{aligned}
\left|\mathbb{E}[\max_{i\in S_1}\tilde{X}_i] - \mathbb{E}[\max_{i\in S_2}\tilde{X}_i]\right| = &\left|\mathbb{E}[\max_k Y_k(S_1)] - \mathbb{E}[\max_k Y_k(S_2)]\right| \\
\le &\frac{\mathsf{W}}{\varepsilon}\left(\sum_k |p_k(S_1) - p_k(S_2)|\right) \\
\le &4h\varepsilon^3\mathsf{W} = O(\varepsilon)\mathsf{W}
\end{aligned}$$

where the first inequality follows from Lemma 1. ☐

For any signature vector sg, we associate to it a set of random variables $\{B_k \sim B(k\varepsilon\mathsf{W}, 1 - e^{-\mathsf{sg}_k})\}_{k=1}^{h}$.[8] Define the value of sg to be $\mathsf{Val}(\mathsf{sg}) = \mathbb{E}[\max_{k\in[h]} B_k]$.

Corollary 2. *For any feasible set S with $\mathsf{Sig}(S) = \mathsf{sg}$, $|\mathbb{E}[\max_{i\in S}\tilde{X}_i] - \mathsf{Val}(\mathsf{sg})| \le O(\varepsilon)\mathsf{W}$. Moreover, combining with Corollary 1, we have that $|\mathbb{E}[\max_{i\in S} X_i] - \mathsf{Val}(\mathsf{sg})| \le O(\varepsilon)\mathsf{W}$.*

C.2.3 Enumerating Signatures

Our algorithm enumerates all signature vectors sg in SG. For each sg, we check if we can find a set S of size K such that $\mathsf{Sig}(S) = \mathsf{sg}$. This can be done by a standard dynamic program in $\tilde{O}(m^{O(1/\varepsilon^2)})$ time as follows: We use Boolean variable $R[i][j][\mathsf{sg}']$ to represent whether signature vector $\mathsf{sg}' \in \mathsf{SG}$ can be dominated by i variables in set $\{X_1, \ldots, X_j\}$. The dynamic programming recursion is

$$R[i][j][\mathsf{sg}'] = R[i][j-1][\mathsf{sg}'] \wedge R[i-1][j-1][\mathsf{sg}' - \mathsf{Sig}(X_j)].$$

If the answer is yes (i.e., we can find such S), we say sg is a feasible signature vector and S is a candidate set. Finally, we pick the candidate set with maximum $r_D(S)$ and output the set. The pseudocode can be found in Algorithm 7.

Now, we are ready to prove Theorem 4 by showing Algorithm 7 is a PTAS for the K-MAX problem.

Proof of Theorem 4. Suppose S^* is the optimal solution and sg^* is the signature of S^*. By Corollary 2, we have that $|\mathsf{OPT} - \mathsf{Val}(\mathsf{sg}^*)| \le O(\varepsilon)\mathsf{W}$.

When Algorithm 7 is enumerating sg^*, it can find a set S such that $\mathsf{Sig}(S) = \mathsf{sg}^*$ (there exists at least one such set since S^* is one). Therefore, we can see that

$$|\mathbb{E}[\max_{i\in S} X_i] - \mathbb{E}[\max_{i\in S^*} X_i]| \le |\mathsf{Val}(\mathsf{sg}^*) - \max_{i\in S} X_i| + |\mathsf{Val}(\mathsf{sg}^*) - \mathbb{E}[\max_{i\in S^*} X_i]| \le O(\varepsilon)\mathsf{W}.$$

Let U be the output of Algorithm 7. Since $\mathsf{W} \ge (1-1/e)\mathsf{OPT}$, we have $r_D(U) \ge r_D(S) = \mathbb{E}[\max_{i\in S} X_i] \ge (1-O(\varepsilon))\mathsf{OPT}$.

The running time of the algorithm is polynomial for a fixed constant $\varepsilon > 0$, since the number of signature vectors is polynomial and the dynamic program in each iteration also runs in polynomial time. Hence, we have a PTAS for the K-MAX problem. ☐

[8] It is not hard to see the signature of $\max_{k\in[h]} B_k$ is exactly sg.

Algorithm 7 PTAS-K-MAX

1: $U \leftarrow \emptyset$
2: **for all** signature vector $\mathsf{sg} \in \mathsf{SG}$ **do**
3: Find a set S such that $|S| = K$ and $\mathsf{Sig}(S) = \mathsf{sg}$
4: **if** $r_D(S) > r_D(U)$ **then**
5: $U \leftarrow S$
6: **end if**
7: **end for**
Output: U

Algorithm 8 Online Submodular Maximization [26]

Let $\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_K$ be K instances of Exp3
for $t = 1, 2, \ldots$ **do**
// Action in the t-th round
for $i = 1$ **to** K **do**
Use $\mathcal{A}_i$ to select an arm $a_{t,i} \in [m]$
end for
Play the super arm $S_t \leftarrow \bigcup_{i=1}^{K}\{a_{t,i}\}$
for $i = 1$ **to** K **do**
Feed back $f_t(\bigcup_{j=1}^{i}\{a_{t,j}\}) - f_t(\bigcup_{j=1}^{i-1}\{a_{t,j}\})$ as the payoff $\mathcal{A}_i$ receives for choosing $a_{t,i}$
end for
end for

Remark. In fact, Theorem 4 can be generalized in the following way: instead of the cardinality constraint $|S| \le K$, we can have more general combinatorial constraint on the feasible set S. As long as we can execute line 3 in Algorithm 7 in polynomial time, the analysis wound be the same. Using the same trick as in [20], we can extend the dynamic program to a more general class of combinatorial constraints where there is a pseudo-polynomial time for the exact version[9] of the deterministic version of the corresponding problem. The class of constraints includes s-t simple paths, knapsacks, spanning trees, matchings, etc.

D Empirical Comparison between the SDCB Algorithm and Online Submodular Maximization on the K-MAX Problem

We perform experiments to compare the SDCB algorithm with the online submodular maximization algorithm in [26], on the K-MAX problem.

Online Submodular Maximization. First we briefly describe the online submodular maximization problem considered in [26] and the algorithm therein. At the beginning, an oblivious adversary sets a sequence of submodular functions $f_1, f_2, \ldots, f_T$ on $2^{[m]}$, where f_t will be used to determine the reward in the t-th round. In the t-th round, if the player selects a feasible super arm S_t, the reward will be $f_t(S_t)$. This model covers the K-MAX problem as an instance: suppose $X^{(t)} = (X_1^{(t)}, \ldots, X_m^{(t)}) \sim D$ is the outcome vector sampled in the t-th round, then the function $f_t(S) = \max_{i \in S} X_i^{(t)}$ is submodular and will determine the reward in the t-th round. We summarize the algorithm in Algorithm 8. It uses K copies of the Exp3 algorithm (see [3] for an introduction). For the K-MAX problem, Algorithm 8 achieves an $O(K\sqrt{mT\log m})$ upper bound on the $(1-1/e)$-approximation regret.

Setup. We set $m = 9$ and $K = 3$, i.e., there are 9 arms in total and it is allowed to select at most 3 arms in each round. We compare the performance of SDCB/Lazy-SDCB and the online

[9] In the exact version of a problem, we ask for a feasible set S such that total weight of S is exactly a given target value B. For example, in the exact spanning tree problem where each edge has an integer weight, we would like to find a spanning tree of weight exactly B.

submodular maximization algorithm on four different distributions. Here we use the greedy algorithm `Greedy-K-MAX` (Algorithm 5) as the offline oracle.

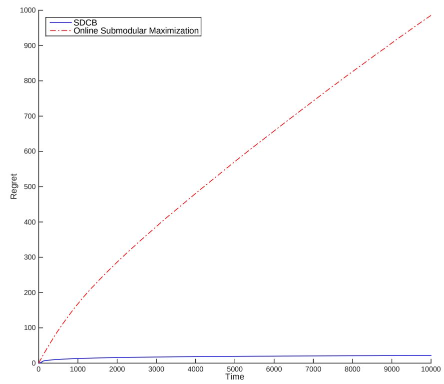
Let $X_i \sim D_i$ ($i = 1, \ldots, 9$). We consider the following distributions. For all of them, the optimal super arm is $S^* = \{1, 2, 3\}$.

- Distribution 1: All D_i's have the same support $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$.
  For $i \in \{1, 2, 3\}$, $\Pr[X_i = 0] = \Pr[X_i = 0.2] = \Pr[X_i = 0.4] = \Pr[X_i = 0.6] = \Pr[X_i = 0.8] = 0.1$ and $\Pr[X_i = 1] = 0.5$.
  For $i \in \{4, 5, 6, \ldots, 9\}$, $\Pr[X_i = 0] = 0.5$ and $\Pr[X_i = 0.2] = \Pr[X_i = 0.4] = \Pr[X_i = 0.6] = \Pr[X_i = 0.8] = \Pr[X_i = 1] = 0.1$.
- Distribution 2: All D_i's have the same support $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$.
  For $i \in \{1, 2, 3\}$, $\Pr[X_i = 0] = \Pr[X_i = 0.2] = \Pr[X_i = 0.4] = \Pr[X_i = 0.6] = \Pr[X_i = 0.8] = 0.1$ and $\Pr[X_i = 1] = 0.5$.
  For $i \in \{4, 5, 6, \ldots, 9\}$, $\Pr[X_i = 0] = \Pr[X_i = 0.2] = \Pr[X_i = 0.4] = \Pr[X_i = 0.6] = \Pr[X_i = 0.8] = 0.12$ and $\Pr[X_i = 1] = 0.4$.
- Distribution 3: All D_i's have the same support $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$.
  For $i \in \{1, 2, 3\}$, $\Pr[X_i = 0] = \Pr[X_i = 0.2] = \Pr[X_i = 0.4] = \Pr[X_i = 0.6] = \Pr[X_i = 0.8] = 0.1$ and $\Pr[X_i = 1] = 0.5$.
  For $i \in \{4, 5, 6\}$, $\Pr[X_i = 0] = \Pr[X_i = 0.2] = \Pr[X_i = 0.4] = \Pr[X_i = 0.6] = \Pr[X_i = 0.8] = 0.12$ and $\Pr[X_i = 1] = 0.4$.
  For $i \in \{7, 8, 9\}$, $\Pr[X_i = 0] = \Pr[X_i = 0.2] = \Pr[X_i = 0.4] = \Pr[X_i = 0.6] = \Pr[X_i = 0.8] = 0.16$ and $\Pr[X_i = 1] = 0.2$.
- Distribution 4: All D_i's are continuous distributions on $[0, 1]$.
  For $i \in \{1, 2, 3\}$, D_i is the uniform distribution on $[0, 1]$.
  For $i \in \{4, 5, 6, \ldots, 9\}$, the probability density function (PDF) of X_i is

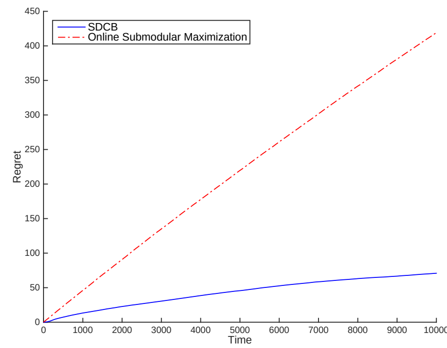
$$f(x) = \begin{cases} 1.2 & x \in [0, 0.5], \\ 0.8 & x \in (0.5, 1]. \end{cases}$$

These distributions represent several different scenarios. Distribution 1 is relatively "easy" because the suboptimal arms 4-9's distribution is far away from arms 1-3's distribution, whereas distribution 2 is "hard" since the distribution of arms 4-9 is close to the distribution of arms 1-3. In distribution 3, the distribution of arms 4-6 is close to the distribution of arms 1-3's, while arms 7-9's distribution is further away. Distribution 4 is an example of a group of continuous distributions for which `Lazy-SDCB` is more efficient than `SDCB`.

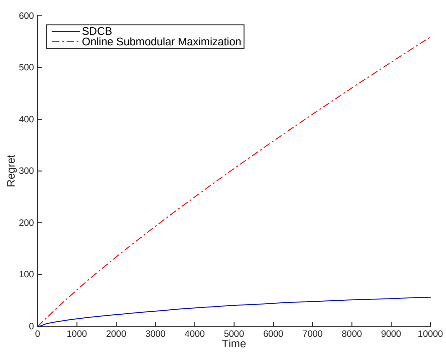
We use `SDCB` for distributions 1-3, and `Lazy-SDCB` (with known time horizon) for distribution 4. Figure 1 shows the regrets of both `SDCB` and the online submodular maximization algorithm. We plot the 1-approximation regrets instead of the $(1 - 1/e)$-approximation regrets, since the greedy oracle usually performs much better than its $(1 - 1/e)$-approximation guarantee. We can see from Figure 1 that our algorithms achieve much lower regrets in all examples.



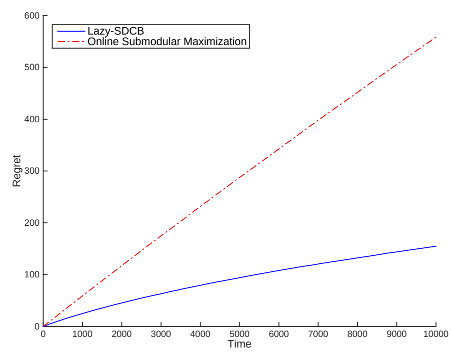
(a) Distribution 1



(b) Distribution 2



(c) Distribution 3



(d) Distribution 4

Figure 1: Regrets of SDCB/Lazy-SDCB and Algorithm 8 on the K-MAX problem, for distributions 1-4. The regrets are averaged over 20 independent runs.