# Combinatorial Categorized Bandits with Expert Rankings

**Sayak Ray Chowdhury**[*1] **Gaurav Sinha**[*1] **Nagarajan Natarajan**[1] **Amit Sharma**[1]

[1]Microsoft Research, Bengaluru, India

## Abstract

Many real-world systems such as e-commerce websites and content-serving platforms employ two-stage recommendation — in the first stage, multiple nominators (experts) provide ranked lists of items (one nominator per category, e.g., sports and political news articles), and in the second stage, an aggregator filters across the lists and outputs a single (short) list of $K$ items to the users. The aggregation stage can be posed as a combinatorial multi-armed bandit problem, with the additional structure that the arms are grouped into categories (disjoint sets of items) and the ranking of arms within each category is known. We propose algorithms for selecting top $K$ items in this setting under two learning objectives, namely minimizing regret over rounds and identifying the top $K$ items within a fixed number of rounds. For each of the objectives, we provide sharp regret/error analysis using carefully defined notion of "gap" that exploits our problem structure. The resulting regret/error bounds strictly improve over prior work in combinatorial bandits literature. We also provide supporting evidence from simulations on synthetic and semi-synthetic problems.

## 1 INTRODUCTION

Multi-Armed Bandits (MAB) is a popular approach to model sequential decision making problems [Bouneffouf et al., 2020]; and has been applied to real-world situations such as recommendation systems [Glowacka, 2019] and online advertising [Lu et al., 2010, Avadhanula et al., 2021]. In many of these applications, however, the decision maker (called agent) needs to identify a *combination* of arms which when pulled together could yield high rewards. For instance, recommender systems often recommend a subset of relevant items to its users. The decision making problem is much more challenging in this setting, as the search space is combinatorially large. This is typically formulated as a "combinatorial multi-armed bandit problem" [Kveton et al., 2015] (`Comb-MAB`), when the goal is to optimize cumulative rewards over rounds, or as a "K-best arm identification problem" [Bubeck et al., 2013] (`K-BAI`), when the goal is to find the best $K$ arms within a fixed number of rounds.

As a motivating example, consider the "whole page optimization" problem arising in recommendation systems for e-commerce, news articles, etc. Here, the web page has a real estate for, say, at most $K$ items, which are typically a combination of products or news articles from different categories. Selecting the "best" $K$ items can be posed as a standard combinatorial multi-armed bandits problem, with the goal of optimizing for click-based rewards. However, in practice, there is more structure to the problem. In particular, the recommender systems employed in large-scale commercial settings typically comprise two stages [Hron et al., 2021, Ma et al., 2020], wherein there are multiple *nominators* in the first stage, each producing a ranked list of items (e.g., a nominator for ranking news articles from the sports category, and another for ranking from political category, and so on), and an *aggregator* that selects the top $K$ items from the ranked lists to populate the web page.

In such two-stage settings where nominators provide reliable rankings, the core online learning problem then is to perform optimal filtering, i.e., subset selection in the second stage (across different nominators) in a sample-efficient manner. In such scenarios, directly applying the known algorithms for `Comb-MAB` [Kveton et al., 2015] and `K-BAI` [Bubeck et al., 2013] for selecting subset of arms can be sub-optimal. Designing and analysing algorithms for the second stage given the structure induced by the first stage is the core technical problem we address in this paper.

Specifically, we study learning algorithms for the setting where arms (i.e., items) are grouped into categories (or dis-

[*]Equal contribution

*Accepted for the 39th Conference on Uncertainty in Artificial Intelligence* (UAI 2023).

joint clusters) and ranking of arms within each category is known to the learning algorithm. Note that only rankings of arms within each category is assumed to be known and not the actual reward distributions. For instance, in the above web page population scenario, the second-stage recommender system would have access to the correct ranking of all the news articles in sports (first category), politics (second category), and so on.

### 1.1 OUR CONTRIBUTIONS

We design new algorithms for the `Comb-MAB` and `K-BAI` problems under the above mentioned structural assumption, i.e., the arms are grouped into different disjoint categories and the true ordering of arms (with respect to their rewards) within each category is known. We summarize our main contributions below.

**Regret Minimization:** For `Comb-MAB`, our objective is to minimize the expected cumulative regret, over $T$ rounds, of selecting $K$ candidate arms to play at each round. We propose *Ordered Combinatorial UCB*, based on the widely-used upper confidence bound (UCB) algorithm (Section 3). We adapt the strategy of Kveton et al. [2015] to incorporate the knowledge of the expert rankings in two key respects: (a) designing a computationally efficient and provably correct sub-routine for selecting $K$ items at each round; (b) providing a regret analysis for our algorithm that strictly improves over Kveton et al. [2015], via defining an appropriate notion of sub-optimality gap for our setting (Theorem 3.2).

**K-Best Arm Identification:** For `K-BAI`, we seek to discover the $K$ best arms at the end of $T$ rounds with high probability. We propose *Ordered SAR* that adapts the Successive-Accept-Reject (SAR) strategy of Bubeck et al. [2013] (Section 4) to our setting with the "prefix structure", i.e., the optimal ranking of items across different experts must necessarily incorporate prefixes of ranked lists from experts. We give a sharper analysis of the error bound for our algorithm that strictly improves over the bound of Bubeck et al. [2013], via a novel definition of "instance-specific complexity", for our structured setting (Theorem 4.1).

We show findings from simulations, that support our theoretical results, in Section 5. Our work leads to interesting follow-up research questions in two-stage recommender systems that we highlight in Section 6. Before we proceed with formally setting up the problem and metrics in Section 2, we review closely related work next.

### 1.2 RELATED WORK

Combinatorial bandits is a generalization of the well-studied multi-armed bandit problem [Auer et al., 2002, Bubeck et al., 2012]. While the problem has been studied in the adversarial setting Cesa-Bianchi and Lugosi [2012], Kale et al. [2010], in this work we focus on stochastic combinatorial bandits.

The cumulative regret minimization problem in stochastic combinatorial bandits under additive rewards was first studied by Gai et al. [2012]. The theoretical guarantee of the proposed algorithm in the above work was subsequently analyzed by Kveton et al. [2015]. Wang and Chen [2017] further generalize these results to the combinatorial bandit setting with general reward structure. In our work, departing from the above line of research, we consider a structured arm set, i.e., where the arms are grouped into different disjoint categories.

The K-Best arm identification problem was first studied under the fixed budget setting by Bubeck et al. [2013], and under the fixed confidence setting by Kalyanakrishnan and Stone [2010], Kalyanakrishnan et al. [2012]. Follow-up works either a) generalized the results of the above papers to the general combinatorial bandit setting [Chen et al., 2014, Gabillon et al., 2012]; or b) considered specific combinatorial structures like matroids [Chen et al., 2017]; or c) improved the algorithms [Chen et al., 2016, Jiang et al., 2017]. Chaudhuri and Tewari [2017] considers the learning-to-rank problem with feedback from K-best arms. The goal is to rank the K-best arms in an online fashion, whereas we focus on identifying the K-best arms.

A well-studied special case of K-Best-Arm is the Best-Arm identification problem [Even-Dar et al., 2006, Audibert et al., 2010], in which we are required to identify the single arm with the largest mean reward. Nearly tight sample complexity bounds as well as error probabilities for Best Arm identification problem were obtained by Jamieson et al. [2014], Kaufmann and Kalyanakrishnan [2013]. However, completely understanding the exact complexity of Best-Arm identification continues to attract significant attention. The same is the case for K-Best arm identification problem, which is the focus of this work.

## 2 PROBLEM SETUP

We are given $N$ ranked lists of items corresponding to different categories, from $N$ experts, as introduced in Section 1. We refer to items as actions or arms interchangeably, as in the bandits literature. Each list $i$ consists of $M$ items $a_{i,1}, \ldots, a_{i,M}$. In typical scenarios discussed in Section 1, $M \gg N$. We assume that lists are disjoint, i.e., each list has a unique set of items.[*] Each item $a_{i,j}, i \in [N], j \in [M]$ is associated with a reward distribution (with a well-defined density) $\nu_{i,j}$, supported on $[0, 1]$, with mean $\mu^*_{i,j} := \mathbb{E}_{r\sim\nu_{i,j}}[r]$. We adopt the setting of combinatorial multi-armed bandits, but with the following structure. We assume that each list

[*]When an item appears in multiple lists, we can keep any one copy of it and throw away the rest. Since our objective is to find the top $K$ (distinct) items in the union of all lists, the best arm (i.e., top $K$ items) does not change by doing so.

is sorted with respect to $\mu^*$, i.e, $\mu^*_{i,1} \geqslant \mu^*_{i,2} \geqslant \ldots \geqslant \mu^*_{i,M}$ for all $i \in [N]$. The $N$-by-$M$ matrix of mean rewards $\mu^* := \left[\mu^*_{i,j}\right]_{i,j}$ is unknown to the learning agent, but she has the *side information* that each list is sorted.

The learning agent has a budget of $T$ rounds (or pulls), and at each round, the agent has to return a set of $K$ items from the $N$ sorted lists of $M$ items each. In practice, $K \ll M$ (see Remark 2.1).

We assume semi-bandit feedback model, i.e., the agent can observe rewards for each selected item, consistent with the literature [Kveton et al., 2015, Chen et al., 2014]. To guide the agent make the combinatorial selection at each round, we consider two widely-used learning objectives:

**Regret minimization.** The first objective we consider is the *optimization goal*, where the agent aims to maximize her expected cumulative reward over time by repeatedly interacting with the unknown environment. The learning protocol is as follows: at each time $t$, (i) the agent chooses a set of $K$ items from the $N$ sorted lists of $M$ items each based on the rewards received before time $t$; equivalently, with the knowledge of the lists being sorted w.r.t. $\mu^*$, she chooses top $z_{t,i}$ items from each list $i$ such that $\sum_{i=1}^{N} z_{t,i} = K$, where each $z_{t,i}$ can take values in $\{0, 1, \ldots, K\}$, and (ii) observes rewards of all the $K$ chosen items $r_t(a_{i,j}), i \leqslant N, j \leqslant z_{t,i}$.

Let $\mathcal{Z}$ denote the set of all possible "list prefixes" or "allocations" the agent can make at any given round, i.e.,

$$\mathcal{Z} = \left\{(z_1, \ldots, z_N) : z_i \in \{0, 1, \ldots, K\}, \sum_{i=1}^{N} z_i = K\right\}. \quad (1)$$

Let $f^{\mu*}(z)$ denote the total expected reward or *utility* of an allocation $z \in \mathcal{Z}$. If the agent knew $\mu^*$ a priori, she could choose the optimal allocation $z^* \in \operatorname{argmax}_{z \in \mathcal{Z}} f^{\mu^*}(z)$ at each round $t$. In this setting, we evaluate the performance of the agent's strategy using *expected cumulative regret* due to not knowing $\mu^*$, that is

$$R_T = \sum_{t=1}^{T} \left[ f^{\mu^*}(z^*) - f^{\mu^*}(z_t) \right]. \quad (2)$$

For simplicity of presentation, we assume that utility function is additive, i.e., $f^{\mu}(z) = \sum_{i=1}^{N} \sum_{j=1}^{z_i} \mu_{i,j}$ for any set of parameters $[\mu]_{i,j}$. However, our results would hold for any monotone utility function, i.e., for any $f$ satisfying $f^{\mu}(z) \leqslant f^{\mu'}(z)$ if $\mu_{i,j} \leqslant \mu'_{i,j}$ for all $i \leqslant N, j \leqslant M$.

**$K$-Best arm identification.** We also study a related objective of the *search goal*, i.e., the agent has a budget of $T$ rounds (or pulls), and is tasked to find top-$K$ items from the $N$ sorted lists of $M$ items each, where $1 < K \leqslant M$. Equivalently, with the knowledge of the lists being sorted w.r.t. $\mu^*$, she needs to find the optimal allocation $z^* \in \mathcal{Z}$, that corresponds to the set of $K$ items with the highest mean rewards across all lists.

The sequential evaluation protocol proceeds as follows: at each round $t = 1, \ldots, T$, the agent chooses an item $a_{i,j}$ and observes a reward $r_t(a_{i,j})$, drawn from $\nu_{i,j}$ independent of the past given $a_{i,j}$. At the end of $T$ rounds, she returns an allocation $z^{\text{out}} \in \mathcal{Z}$. We evaluate the performance of the agent's strategy by the *probability of error* (or misidentification), that is

$$\delta_T = \mathbb{P}\left[z^{\text{out}} \neq z^*\right]. \quad (3)$$

**Remark 2.1.** Note that, without loss of generality, we can trim each of the lists from size $M$ down to size $K$, given (a) our problem structure, i.e., we want the $K$ items returned to be a prefix of the lists, and (b) the monotonicity of the utility function. This reduces the search space, and so, in effect, $M = K$.

## 3 COMB-MAB: REGRET MINIMIZATION

In this section, we present the learning algorithm for the first objective set up in Section 2, i.e., *optimization* goal, its regret bounds, and show how our guarantees improve over the state-of-the-art results in the literature by exploiting the problem structure.

### 3.1 ALGORITHM

Motivated by the simplicity of the widely-used Upper Confidence Bound (UCB) strategy, Kveton et al. [2015] designed and analyzed an algorithm for stochastic combinatorial semi-bandits for regret minimization. We adapt this algorithm to our setting, where the agent knows the true ordering of items in each list. We call this algorithm *Ordered Combinatorial UCB*.

Informally, our algorithm consists of three steps at each time $t$. First, we compute the UCBs on the expected reward $\mu^*_e$ of each item $e \in \{a_{i,j}\}_{i \leqslant N, j \leqslant M}$ as

$$U_t(e) = \widehat{\mu}_{T_{t-1}(e)}(e) + \beta_{t-1, T_{t-1}(e)},$$

where $T_t(e)$ denotes the number of times item $e$ is observed in $t$ rounds, $\widehat{\mu}_s(e)$ denotes the empirical mean of $s$ samples from $\nu_e$ and $\beta_{t,s}$ denotes the radius of a confidence interval around $\widehat{\mu}_s(e)$. Choosing $\beta_{t,s} = \sqrt{\frac{3 \log t}{2s}}$, it holds that $\mu^*_e$ lies in the said confidence interval with high probability. Next, we choose an allocation $z_t \in \mathcal{Z}$ by solving a *combinatorial optimization* problem using UCB estimates:

$$z_t \in \operatorname*{argmax}_{z \in \mathcal{Z}} f^{U_t}(z) = \operatorname*{argmax}_{z \in \mathcal{Z}} \sum_{i=1}^{N} \sum_{j=1}^{z_i} U_t(a_{i,j}). \quad (4)$$

Now, we play the set of $K$ items $a_{i,j}, i \leqslant N, j \leqslant z_{t,i}$, given by the allocation $z_t$ and observe rewards of all the items. Finally, we update the estimates $T_t(a_{i,j})$ and $\widehat{\mu}_{T_t(a_{i,j})}(a_{i,j})$ of these items. See Algorithm 1 for complete pseudocode.

**DP-based optimization solution.** Now, we provide a subroutine to find the allocation $z_t$ as given in (4). Our proposed solution is based on dynamic programming (DP), and is computationally efficient. Given a set of parameters $\theta_{i,j}, i \leqslant N, j \leqslant M$, the objective is to compute $\operatorname{argmax}_{z\in\mathcal{Z}} \sum_{i=1}^{N}\sum_{j=1}^{z_i} \theta_{i,j}$. In other words, we need to find optimal selection of $K$ items w.r.t. the parameter $\theta$ using prefixes from lists $1,\dots,N$.

To this end, let $V^{\theta}_{i,j}$ denote the *value* of the selection of $j$ items using prefixes from lists $1,\dots,i$. Also, for each list $i$, let $s^{\theta}_{i,j} = \sum_{k=1}^{j}\theta_{i,k}$ denote the sum of $\theta$'s of first $j$ items. By definition, $V^{\theta}_{1,j} = s^{\theta}_{1,j}$ for all $j$ and $V^{\theta}_{i,0} = s^{\theta}_{i,0} = 0$ for all $i$. Now, for each $2 \leqslant i \leqslant N$ and $1 \leqslant j \leqslant K$, we compute the value $V^{\theta}_{i,j}$ using the following recurrence:

$$V^{\theta}_{i,j} = \max \begin{cases} V^{\theta}_{i-1,j} + s^{\theta}_{i,0} & \text{(no item from list } i\text{)} \\ V^{\theta}_{i-1,j-1} + s^{\theta}_{i,1} & \text{(first item from list } i\text{)} \\ V^{\theta}_{i-1,j-2} + s^{\theta}_{i,2} & \text{(first 2 items from list } i\text{)} \\ \vdots & \\ V^{\theta}_{i-1,0} + s^{\theta}_{i,j} & \text{(all } j \text{ items from list } i\text{)} \end{cases} \quad (5)$$

We return as $z_t$ the selection of $K$ items from lists $1,\dots,N$ that attain the value $V^{U_t}_{N,K}$, where $U_t(a_{i,j})$ denotes UCB estimates at round $t$. The following lemma shows the optimality of this solution. It can be proved using simple induction argument.

**Lemma 3.1** (Optimality of DP)**.** *Let $\mathcal{Z}$ be given by* (1)*. Then, for any set of parameters $\theta_{i,j} > 0$, $i \in [N]$, $j \in [M]$, we have $V^{\theta}_{N,K} = \max_{z\in\mathcal{Z}} \sum_{i=1}^{N}\sum_{j=1}^{z_i}\theta_{i,j}$.*

The time complexity of finding the allocation $z_t$ is $O(NK^2)$ implying our algorithm is also computationally efficient, especially since, in general, $K$ is small.

### 3.2 CUMULATIVE REGRET

We introduce the following notion of gap based on the utility function $f$ and optimal allocation $z^*$, which is essential to characterize the performance of our regret minimization algorithm. First, we define the *gap* of an allocation $z \in \mathcal{Z}$ as $\Delta_z = f^{\mu^*}(z^*) - f^{\mu^*}(z)$. Now, we define the minimum gap of any sub-optimal allocation $z = (z_1,\dots,z_N)$ that selects top $j$ items, $j \leqslant M$, from $i$-th list, $i \leqslant N$, as

$$\Delta_{i,j} := \min_{z\neq z^\star : z_i = j} \Delta_z = f^{\mu^*}(z^*) - \max_{z\neq z^\star : z_i = j} f^{\mu^*}(z). \quad (6)$$

With this definition of gap, we bound the cumulative regret of Algorithm 1 as follows.

**Theorem 3.2** (Cumulative regret)**.** *After $T$ rounds, the Ordered UCB algorithm enjoys the regret bound*

$$R_T \leqslant \sum_{(i,j): z^*_i + 1 \leqslant j \leqslant M} \frac{CK\log T}{\Delta_{i,j}} + \left(\frac{\pi^2}{3} + 1\right) KMN,$$

**Algorithm 1:** Ordered Combinatorial UCB

**Input:** $N$ lists of items $(a_{i,1},\dots,a_{i,M}), i \leqslant N$, and $K$ (#items to retrieve).

**Initialize:** Play each arm $a_{i,j}, i \leqslant N, j \leqslant M$ once and observe reward $r(a_{i,j}) \sim \nu_{i,j}$.

Set $T_{MN}(a_{i,j}) = 1$, $\widehat{\mu}_{MN}(e) = r(a_{i,j})$.

**for** *round* $t = MN+1, 2, \dots$ **do**

  For each $i \leqslant N, j \leqslant M$, compute UCBs: $U_t(a_{i,j}) = \widehat{\mu}_{T_{t-1}(a_{i,j})}(a_{i,j}) + \sqrt{\frac{3\log(t-1)}{2T_{t-1}(a_{i,j})}}$.

  Choose allocation $z_t \in \mathcal{Z}$ using (4) and (5).

  For each $i \leqslant N$ and each $j \leqslant z_{t,i}$, play arm $a_{i,j}$ and observe its reward $r_t(a_{i,j})$.

  Update number of plays for $a_{i,j}$'s played in the above step: $T_t(a_{i,j}) = T_{t-1}(a_{i,j}) + 1$.

  Update their mean estimates: $\widehat{\mu}_{T_t(a_{i,j})}(a_{i,j}) = \frac{\widehat{\mu}_{T_{t-1}(a_{i,j})}(a_{i,j})T_{t-1}(a_{i,j}) + r_t(a_{i,j})}{T_t(a_{i,j})}$.

*where $\Delta_{i,j}$ is given by* (6) *and $C > 0$ is a universal constant.*

It is worth noting that the summation in the above expression is over the sub-optimal items, i.e., the items not appearing in the optimal allocation $z^*$. At the same time, the sum is over the items that can appear in a sub-optimal allocation, i.e., those within $K$ positions from the top of each list. This is because any item below the $K$-th position of any would never be played even by a sub-optimal allocation due to the ordering structure. Also, note that $M = K$ for the Ordered UCB algorithm due to Remark 2.1.

**Comparison with prior work.** One can directly employ the *CombUCB1* algorithm of Kveton et al. [2015] to solve the above regret minimization problem. To do so, one needs to instantiate the feasible set $\Theta$, which CombUCB1 takes as an input, with the allocation set $\mathcal{Z}$ as given in (1). This is due to the fact that any allocation $z \in \mathcal{Z}$ induces a subset $\Theta \in 2^{MN}$ of size $K$, where $z_i \geqslant j$ implies $a_{i,j} \in \Theta$. This simple tweak of CombUCB1 would achieve a regret similar to Theorem 3.2 with the gaps $\Delta_{i,j}$ being replaced by

$$\widetilde{\Delta}_{i,j} = \min_{z\neq z^*: z_i \geqslant j} \Delta_z \,.$$

Note that $\Delta_{i,j} \geqslant \widetilde{\Delta}_{i,j}$, since the minimum in (6) is over a smaller set of allocations in $\mathcal{Z}$. Hence, Algorithm 1 enjoys a smaller regret bound as compared to CombUCB1. This is because our regret analysis is carefully fine-tuned to the prefix structure present in the problem, whereas Kveton et al. [2015] present a general analysis for combinatorial action sets oblivious to the ordering in each list.

### 3.2.1 Proof Sketch

In this section, we provide the main ideas to prove Theorem 3.2, and contrast it to the analysis of Kveton et al. [2015] when needed. First, we define the event

$$\mathcal{E}_t = \left\{ \Delta_{z_t} \leqslant \sum_{(i,j):z_i^* < j \leqslant z_{t,i}} 2\sqrt{\frac{1.5\log T}{T_{t-1}(a_{i,j})}}, \Delta_{z_t} > 0 \right\},$$

where $\Delta_z$ denotes the gap of an allocation $z$. Now, defining $\widehat{R}_T = \sum_{t=MN+1}^{T} \Delta_{z_t} \mathbb{1}\{\mathcal{E}_t\}$, we see from Kveton et al. [2015, Lemma 1] that

$$R_T \leqslant \mathbb{E}\left[\widehat{R}_T\right] + (1+\pi^2/3)KMN\ .$$

Now, let us consider two sequences of constants $(\alpha_l)_{l\geqslant 1}$ and $(\beta_l)_{l\geqslant 0}$ as in Kveton et al. [2015] and define $m_{l,t} = \frac{\alpha_l K^2 \log T}{\Delta_{z_t}^2}$. Furthermore, let $\tilde{A}_t$ denote the subset of items included in the allocation $z_t$ but not in $z^*$. Then, we define a series of mutually exclusive events $(G_{l,t})_{l\geqslant 1}$, where $G_{l,t}$ denotes the event that at least $\beta_l K$ items in $\tilde{A}_t$ were observed at most $m_{l,t}$ times and for all $j < i$, less than $\beta_1 K$ items in $\tilde{A}_t$ were observed at most $m_{l-1,t}$ times. Then, under $\mathcal{F}_t$, it holds that the event $\bigcup_{l\geqslant 1} G_{l,t}$ happens, and hence

$$\widehat{R}_T = \sum_{l=1}^{\infty} \sum_{t=MN+1}^{T} \Delta_{z_t} \mathbb{1}\{G_{l,t}, \Delta_{z_t} > 0\}.$$

Now, let $G_{a_{i,j},l,t} = G_{l,t} \bigcap F_{a_{i,j},l,t}$ be the event that item $a_{i,j}$ is not observed sufficiently often under $G_{i,t}$, where $F_{a_{i,j},l,t} = \{z_i^* < j \leqslant z_{t,i}, T_{t-1}(a_{i,j}) \leqslant m_{l,t}\}$. Then Kveton et al. [2015] bound $\widehat{R}_T$ as

$$\widehat{R}_T \leqslant \sum_l \sum_t \sum_{(i,j):z_i^* < j} \mathbb{1}\{j \leqslant z_{t,i}, T_{t-1}(a_{i,j}) \leqslant m_{l,t}\} \frac{\Delta_{z_t}}{\beta_l K}.$$

Our main analytical novelty is to identify some "double counting" present in the above regret expression under the ordered structure. To do so, we define for $k \geqslant 0$, the events

$$F^k_{a_{i,j},l,t} = \{z_i^* < j + k = z_{t,i}, T_{t-1}(a_{i,j+k}) \leqslant m_{l,t}\}\ .$$

Note that, because of the ordered structure, if $a_{i,j}$ has only been observed a certain number of times, then $a_{i,j_k}$ would be observed less than or equal number of times i.e., $T_{t-1}(a_{i,j+k}) \leqslant T_{t-1}(a_{i,j})$, which implies that the event

$$\{z_i^* < j + k = z_{t,i}, T_{t-1}(a_{i,j}) \leqslant m_{l,t}\} \subseteq F^k_{a_{i,j},l,t}\ .$$

This yields $F_{a_{i,j},l,t} \subseteq \bigcup_{k=0}^{M-j} F^k_{a_{i,j},l,t} =: H_{a_{i,j},l,t}$, which gives $\bigcup_{j=1}^{M}\{G_{l,t} \bigcap F_{a_{i,j},l,t}\} \subseteq \bigcup_{j=1}^{M}\{G_{l,t} \bigcap H_{a_{i,j},l,t}\}$. Now observe that $H_{a_{i,1},l,t} \supseteq H_{a_{i,2},l,t} \supseteq \ldots H_{a_{i,M},l,t}$, implying that the RHS of the above is a union over decreasing sets and hence $\bigcup_{j=1}^{M}\{G_{l,t} \bigcap F_{a_{i,j},l,t}\} \subseteq \{G_{l,t} \bigcap H_{a_{i,i},l,t}\}$. This further implies that

$$\bigcup_{j=1}^{M} G_{a_{i,j},l,t} \subseteq \bigcup_{j=1}^{M} \Big\{ G_{l,t} \bigcap \{z_i^* < j = z_{t,i}, T_{t-1}(a_{i,j}) \leqslant m_{l,t}\} \Big\}.$$

Therefore, we can bound $\widehat{R}_T$ as

$$\widehat{R}_T \leqslant \sum_l \sum_t \sum_{(i,j):z_i^* < j} \mathbb{1}\{j = z_{t,i}, T_{t-1}(a_{i,j}) \leqslant m_{l,t}\} \frac{\Delta_{z_t}}{\beta_l K},$$

which corrects for the "double counting" mentioned above. To bound the regret, we now need to look at sub-optimal allocations that end at item $a_{i,j}$ for list $i$, which can be accounted with our definition of *minimum gap* $\Delta_{i,j}$ (see (6)). The rest of the proof follows similar arguments as in Kveton et al. [2015]. See Appendix **A.1** for details.

# 4 K-BAI: MINIMIZE ERROR PROBABILITY

In this section, we present the learning algorithm for the second objective set up in Section 2, i.e., the *search* goal. We prove a bound on the probability of error of our algorithm and show that our guarantee improves the state-of-the-art results in the literature by exploiting the problem structure.

## 4.1 ALGORITHM

We propose an algorithm for finding top-$K$ items from $N$ lists obeying the ordered structure. We adapt the Successive Accepts and Rejects (SAR) strategy of Bubeck et al. [2013] to our setting, originally designed for top-$K$ identification in stochastic combinatorial semi-bandits. We call this algorithm *Ordered SAR*.

Informally, our algorithm proceeds as follows. We divide the total budget of $T$ rounds into $MN-1$ phases. At the end of each phase, we either *accept* an item from the *top* of a list or *reject* an item from the *bottom* of a list. In any case, that item is "deactivated". The items that are still active are sampled for an equal number of rounds in the next phase. Now, we describe the procedure for choosing an item to accept or reject. Let $\Phi_k$ denote the set of active items at the start of phase $k$. We pull each item $e \in \Phi_k$ for $T_k - T_{k-1}$ rounds and update their empirical means with observed rewards, where

$$T_k = \left\lceil \frac{1}{\overline{\log}(MN)} \frac{T - MN}{MN + 1 - k} \right\rceil,\ \overline{\log}(n) = \frac{1}{2} + \sum_{i=2}^{n} \frac{1}{i}\ , \quad (7)$$

with $T_0 := 0$. Similar to Bubeck et al. [2013], the key to decide whether to accept or reject an item is to consider estimates of the gaps $\Delta_e$. To this end, let $m_k > 0$ denote the number of items left to find at the start of phase $k$. First,

**Algorithm 2:** Ordered Successive-Accept-Reject

**Input:** $N$ lists of items $(a_{i,1}, \ldots, a_{i,M}), i \leqslant N$, phase lengths $(T_k)_{0 \leqslant k < MN}$, and $K$ (#items to retrieve).

**Initialize:** $\Phi_1 = \{a_{i,j}\}_{i \leqslant N, j \leqslant M}$, $m_1 = K$, $z_i^{\text{out}} = 0$, $\text{top}_i = 1, \text{bot}_i = M \ \forall i \leqslant N$

**for** *each phase* $k = 1, 2, \ldots, MN-1$ **do**

  Pull each arm $e \in \Phi_k$ for $T_k - T_{k-1}$ rounds and update its empirical mean $\widehat{\mu}_{k,e}$.

  Compute empirical gap $\widehat{\Delta}_{k,e}$ for each arm $e \in \Phi_k$ using (8).

  Let $e_k \in \operatorname{argmax}_{e \in \Phi_k} \widehat{\Delta}_{k,e}$ (ties broken arbitrarily) and $i_k$ be such that $e_k = a_{i_k,j}$ for some $j$.

  **if** $\widehat{\mu}_{k,e_k} > \widehat{\mu}_{k,[m_k]}$ **then**

    Set $j_k = \text{top}_{i_k}$, $\text{top}_{i_k} = \text{top}_{i_k} + 1$,

    $m_{k+1} = m_k - 1$, $z_{i_k}^{\text{out}} = z_{i_k}^{\text{out}} + 1$.

  **else**

    Set $j_k = \text{bot}_{i_k}$, $\text{bot}_{i_k} = \text{bot}_{i_k} - 1$.

  Set $\Phi_{k+1} = \Phi_k \backslash \{a_{i_k,j_k}\}$.

**Output:** Allocation $z^{\text{out}}$ of accepted arms.

we compute the "empirical gap" of each item $e \in \Phi_k$ :

$$\widehat{\Delta}_{k,e} = \begin{cases} \widehat{\mu}_{k,e} - \widehat{\mu}_{k,[m_k+1]}, & \text{if } \widehat{\mu}_{k,e} \geqslant \widehat{\mu}_{k,[m_k]} \\ \widehat{\mu}_{k,[m_k]} - \widehat{\mu}_{k,e}, & \text{if } \widehat{\mu}_{k,e} \leqslant \widehat{\mu}_{k,[m_k+1]} \end{cases}, \quad (8)$$

where $\widehat{\mu}_{k,[l]}$ denotes the $l$-th largest empirical mean among all items in $\Phi_k$. Then, we find the item $e_k$ which has the largest empirical gap among all active items $\Phi_k$. Now, let $e_k$ be an item from list $i_k$. If $e_k$ is the current empirical best item, we accept the current topmost active item from list $i_k$. Else, we reject the current bottom most active item from it. In any case, we deactivate the accepted or rejected item, and update the top or bottom of the list accordingly. See Algorithm 2 for pseudo-code.

### 4.2 PROBABILITY OF ERROR

We introduce the following complexity measure to characterize the performance of our top-$K$ identification algorithm. Recall that $(z_1^*, \ldots, z_N^*) \in \mathcal{Z}$ is the optimal allocation corresponding to the set of $K$ arms with highest mean rewards. Define the set of "boundary" arms

$$\Phi = \bigcup_{i=1}^{N} \left\{ a_{i,z_i^*}, a_{i,z_i^*+1} \right\}, \quad (9)$$

where $a_{i,0} := \varnothing$ for all $i \in [N]$. That is, $\Phi$ contains only $z_i$-th and $z_i + 1$-st arms from the top of each list $i$. Note that the cardinality of this boundary set is at most twice the number of lists , i.e., $|\Phi| \leqslant 2N$.

Let $\mu_{[l]}^*, 1 \leqslant l \leqslant MN$, denote the $l$-th largest mean reward among all arms, i.e., $\mu_{[1]}^* \geqslant \ldots \geqslant \mu_{[MN]}^*$. Now, similar to Bubeck et al. [2013], we define the gap of each arm $e \in \{a_{i,j}\}_{i \leqslant N, j \leqslant M}$:

$$\Delta_e = \begin{cases} \mu_e^* - \mu_{[K+1]}^*, & \text{if } \mu_e^* \geqslant \mu_{[K]}^* \\ \mu_{[K]}^* - \mu_e^*, & \text{if } \mu_e^* \leqslant \mu_{[K+1]}^* \end{cases}. \quad (10)$$

Let $\Delta_{[l]}$ be the $l$-th smallest such gap, i.e., $\Delta_{[1]} \leqslant \ldots \leqslant \Delta_{[MN]}$. Let $k_1 \leqslant \ldots \leqslant k_{|\Phi|}$ be the phases in which Algorithm 2 accepts or rejects an arm from the boundary set $\Phi$. We define the complexity measure

$$H_\Phi = \max_{1 \leqslant j \leqslant |\Phi|} \frac{(MN + 1 - k_j)}{\Delta_{[MN+1-k_j]}^2}. \quad (11)$$

With these definitions in place, we bound the probability of error of Algorithm 2 as follows.

**Theorem 4.1** (Probability of error)**.** *Given a time budget $T > MN$, running the ordered SAR algorithm with choice of $T_k$'s given in* (7)*, achieves the probability of error*

$$\delta_T \leqslant 2MN|\Phi| \exp\left( -\frac{T - MN}{8\overline{\log}(MN) H_\Phi} \right),$$

*where $H_\Phi$ is given by* (11)*.*

It is worth noting that gaps of only $|\Phi|$ many arms influence the final error in the selected arms, which is a consequence of our algorithm exploiting the prefix structure. Furthermore, if reward gaps are large for these $|\Phi|$ many arms, then $H_\Phi$ is small and hence, the probability of error is also small, i.e., it is easy to distinguish the top-$K$ arms from the rest.

Furthermore, in our setting, the dependence on $M$ in the bound above is extraneous as stated in the remark below.

**Remark 4.2.** Given our problem structure and assumption on the utility function, the top-$K$ items must necessarily incorporate prefixes of the lists. So, when $M > K$, the lists can be trimmed to size $K$, before presenting to the algorithm, as mentioned in Remark 2.1. Thus, we can replace $M$ with $K$ in the bound of Theorem 4.1.

**Comparison with prior work.** Observe that one can directly apply the SAR algorithm of Bubeck et al. [2013] to find the optimal allocation $z^*$. This algorithm gives a guarantee that the probability of error is

$$\leqslant 2M^2N^2 \exp\left( -\frac{T - MN}{8\overline{\log}(MN) H} \right)$$

where the complexity measure $H$ is defined as:

$$H = \max_{1 \leqslant l \leqslant MN} l \, \Delta_{[l]}^{-2}.$$

Note that $H_\Phi \leqslant H$ since the maximum in (11) is over a much smaller set of arms $\Phi$ of size $\leqslant 2N$ compared to the maximum over all the $MN$ arms in $H$. Hence, Algorithm 2 achieves a smaller probability of error compared to the above work. This is because we adapt our strategy to the ordering of the lists, whereas the SAR algorithm does not. Comparing the terms outside the negative exponential* in both these error guarantees, we can see that our guarantee depends linearly on $M$, whereas the guarantee for the SAR algorithm in Bubeck et al. [2013] has a quadratic dependence. Our experiments on $K$-best arm identification in Section 5 provide good support to these theoretical findings.

#### 4.2.1 Proof Sketch

In this section we provide a high level sketch of our proof for Theorem 4.1. Complete details are provided in Appendix **A.2**. At a high level, our proof uses ideas from the proof of Theorem 1 in Bubeck et al. [2013]. However, there are some crucial differences that take advantage of the known ordered structure between items and therefore leads to better guarantees. Let $k_1 < \ldots < k_{|\Phi|}$ be the phases where an item from $\Phi$ (i.e. the boundary set) was accepted or rejected by Algorithm 2. Since we always accept an item that is the top item of some list and reject an item that is the bottom item of some list, the first error can only occur at a boundary item i.e. there can be no errors before phase $k_1$. During phase $k_1$, there will be $MN + 1 - k_1$ active items, let's call them $a_1, \ldots, a_{MN+1-k_1}$ such that $\mu_{a_1} \geqslant \ldots \geqslant \mu_{a_{MN+1-k_1}}$.

Now let's say an error occurs at phase $k_1$ and an item $a_l \in \Phi$ was accepted when it should have actually been rejected. We follow the proof idea in Bubeck et al. [2013] and prove that this cannot hold by showing that it leads to a contradiction. In particular we show $\Delta_{[MN+1-k_1]} > \max\{\mu_{a_1} - \mu_K, \mu_K - \mu_{a_{MN+1-k_1}}\}$, where $\mu_K$ is the $K^{th}$ largest mean rewards. It's a contradiction because at stage $k_1$ only $k_1 - 1$ items would have been accepted or rejected, implying $\Delta_{[MN+1-k_1]} \leqslant \max\{\mu_{a_1} - \mu_K, \mu_K - \mu_{a_{MN+1-k_1}}\}$. We create a high probability event where this can be shown:

$$\eta_1 = \left\{ \forall \text{ items } a : \left| \frac{1}{n_{k_1}} \sum_{s=1}^{n_{k_1}} X_{a,s} - \mu_a \right| < \frac{1}{4} \Delta_{[MN+1-k_1]} \right\},$$

where $X_{a,s}$ is the reward received on the $s^{th}$ pull of item $a$. Proof of why this holds under $\eta_1$ is technical and is presented with all details in Appendix **A.2**. The general idea is similar to the one presented in Bubeck et al. [2013]. However, the proof needs to be crucially modified at many places to make it work. Proof in Bubeck et al. [2013] directly uses the item $a_l$ that was accepted (by mistake) in its technical calculations to show the above inequality. In their SAR algorithm, they accept $a_l$ when it has the largest empirical gap and the largest mean empirical reward among all active items. This fact is crucial in showing the inequality mentioned above. However, our algorithm accepts $a_l$ without actually considering its own empirical mean reward and therefore we cannot directly use the proof in Bubeck et al. [2013]. To get around this problem, we note that $a_l$ is accepted by Algorithm 2, only when there is some item $a_p$ in the same list as $a_l$ with true mean reward $\mu_{a_p} \leqslant \mu_{a_l}$, largest empirical gap and largest empirical mean reward compared to all other active items. Since $a_l$ should have been rejected it is not in top $K$ items and therefore $a_p$ is also not in the top $K$ items. For the rest of the proof we follow the steps in Bubeck et al. [2013] but work with $a_p$ instead of $a_l$ and all steps go through. The technical calculations only require that within all the active items, $a_p$ has the largest empirical gap and the largest empirical mean reward and that it is not a part of top $K$ items.

Now we can extend this argument to phase $k_2$ since if there was no error at phase $k_1$, the next error has to happen at the next item from the boundary set i.e. at phase $k_2$. To prevent this we assume the event:

$$\eta_2 = \left\{ \forall \text{ items } a : \left| \frac{1}{n_{k_2}} \sum_{s=1}^{n_{k_2}} X_{a,s} - \mu_a \right| < \frac{1}{4} \Delta_{[MN+1-k_2]} \right\}.$$

Continuing in this fashion, we define the intersection of all these events and our proof holds when this event is assumed. Since we only had to repeat this argument for $|\Phi|$ many phases compared to the SAR algorithm which repeats it for all $MN - 1$ phases, we are able to guarantee a better bound.

## 5 SIMULATIONS

The objective of this section is to verify if the theoretical guarantees on improvements over existing methods in terms of upper bounds hold empirically. We show results on synthetic and semi-synthetic problem instances.

### 5.1 REGRET MINIMIZATION

In this section, we empirically evaluate the regret performance of Algorithm 1 on bandit instances generated from synthetic and real-world data. We compare against the combinatorial bandit algorithm of Kveton et al. [2015] which is oblivious to the ordering of the lists. Specifically, we instantiate the algorithm of Kveton et al. [2015] with the feasible set $\Theta = 2^{MN}$ to serve as a baseline (referred to as CombUCB). We plot cumulative regret as a function of number of rounds, and we average results over 20 trial runs with different seeds.

**Synthetic bandit instances.** First, we generate combinatorial bandit instances with $N = 5$ ordered lists, with each list consisting of $M = 10$ items (arms). The arm means are sampled uniformly in $[0.25, 0.75]$. We consider real-valued

*these arise due to application of union bounds

rewards sampled from Gaussian (Figure 1), Bernoulli (Figure 2) distributions with aforementioned means, projected to $[0, 1]$. We show the results for $K = 5$ in Figures 1 and 2; the growth of cumulative regret for the two algorithms aligns with our theoretical findings in Section 3.

**Semi-synthetic bandit instance.** Next, we generate bandit instances from Microsoft Learning to Rank dataset MSLR-WEB10K [Qin and Liu, 2013]. The dataset consists of 1,200,192 rows and 138 columns, where each row corresponds to a query-url pair. The first column is relevance label $\{0, 1, 2, 3, 4\}$ of the pair, which we take as rewards. The second column denotes the query id, and the rest 136 columns denote contexts of a query-url pair. We cluster the data by running K-means algorithm with 50 clusters. We treat each cluster as a bandit arm with mean reward as the empirical mean of the individual ratings in the cluster. This way, we obtain a bandit instance with 50 total arms. We then divide them into $N = 5$ lists of $M = 10$ arms in each. The results are shown in Figure 3 for $K = 5$.

In all the simulations above, we observe that the cumulative regret of our algorithm (*Ordered CombUCB*) is much lower than the baseline (Vanilla *CombUCB*), consistent with our theoretical result.

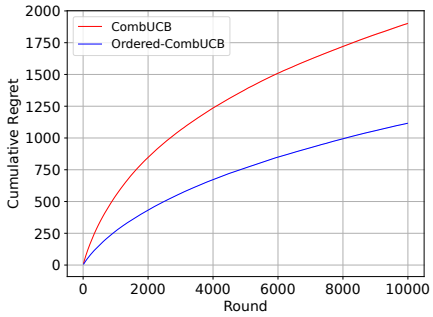


Figure 1: Comparison of cumulative regret for CombUCB and Ordered CombUCB on synthetic Gaussian bandit instance.

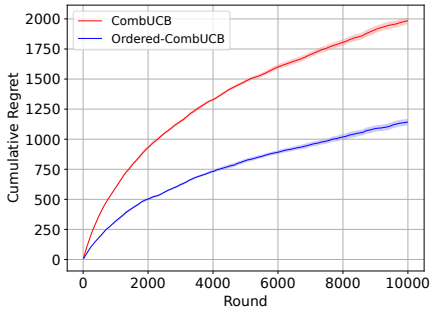


Figure 2: Comparison of cumulative regret for CombUCB and Ordered CombUCB on synthetic Bernoulli bandit instance.

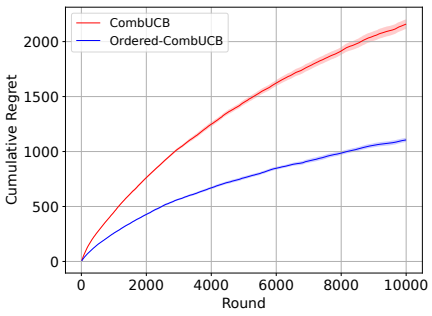


Figure 3: Comparison of cumulative regret for CombUCB and Ordered CombUCB on semi-synthetic bandit instance.

### 5.2 $K$-BEST ARM IDENTIFICATION

In this section, we first aim to find top $K = 5$ arms of the (first) synthetic bandit instance used in the above experiments, i.e., when means of each of the 50 arms are sampled uniformly in $[0.25, 0.75]$. We use the algorithm of Bubeck et al. [2013] as baseline (referred to as SAR) against our proposed Algorithm 2. We observe that (plot not shown), within 5000 rounds, both the algorithms are able to find top 5 arms. This, we believe, is due to the fact that the *problem instance is easy* (i.e., top-5 arms are easy to find when the mean rewards are fairly spread out).

To demonstrate the advantage of our algorithm, we generate a *hard instance* by sampling arm means uniformly in $[0.45, 0.55]$. The rewards are sampled from Gaussian (Figure 4) and Bernoulli (Figure 5) distributions with aforementioned means and projected to $[0, 1]$. We run both the algorithms for rounds $T \in [1000, \ldots, 10000]$ for 100 independent trials and compute the fraction of trials for which they fail to output the optimal allocation. In Figures 4 and 5, we compare the probability of error of Ordered SAR (Algorithm 2) with the SAR algorithm of Bubeck et al. [2013] as a function of the budget, i.e., number of rounds. We find that the failure probability of Ordered SAR is consistently lower than that of SAR, which validates our theory.

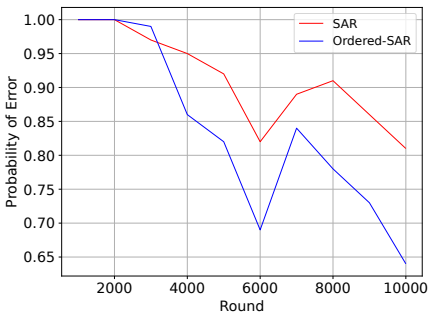


Figure 4: Comparison of probability of error for SAR and Ordered SAR on synthetic Gaussian bandit instance.

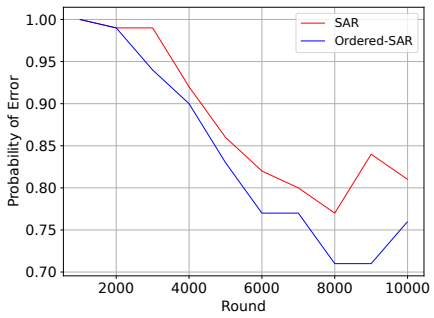


Figure 5: Comparison of probability of error for SAR and Ordered SAR on synthetic Bernoulli bandit instance.

## 6 CONCLUSIONS AND FUTURE WORK

We identify and formulate an important problem arising in two-stage recommendation systems that employ different experts for different categories of items. We propose solutions, adapting existing algorithms for combinatorial multi-arm bandits, and provide regret/error bounds that strictly improve over state-of-the-art for our setting. Our work opens up interesting follow-up research questions: i) can we incorporate user context while selecting top $K$ items, when available? ii) can we design an algorithm to find the optimal allocation with a *fixed confidence*, say $\delta$, and find the sample complexity of this strategy as a function of $\delta$? We conjecture that a variant of the *Combinatorial Lower Upper Confidence Bound* algorithm of Kalyanakrishnan et al. [2012] adapted to the ordering of lists would work in this setting. Another interesting direction is to lift these results to the setting distributed bandits[Korda et al., 2016, Mahadik et al., 2020].

### References


Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In *COLT*, pages 41–53, 2010.

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

Vashist Avadhanula, Riccardo Colini Baldeschi, Stefano Leonardi, Karthik Abinav Sankararaman, and Okke Schrijvers. Stochastic bandits for multi-platform budget optimization in online advertising. WWW '21, page 2805–2817, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383127. doi: 10.1145/3442381.3450074. URL https://doi.org/10.1145/3442381.3450074.

Djallel Bouneffouf, Irina Rish, and Charu Aggarwal. Survey on applications of multi-armed and contextual bandits. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2020. doi: 10.1109/CEC48606.2020.9185782.

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

Séebastian Bubeck, Tengyao Wang, and Nitin Viswanathan. Multiple identifications in multi-armed bandits. In *International Conference on Machine Learning*, pages 258–265. PMLR, 2013.

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

Sougata Chaudhuri and Ambuj Tewari. Online learning to rank with top-k feedback. *The Journal of Machine Learning Research*, 18(1):3599–3648, 2017.

Lijie Chen, Anupam Gupta, and Jian Li. Pure exploration of multi-armed bandit under matroid constraints. In *Conference on Learning Theory*, pages 647–669. PMLR, 2016.

Lijie Chen, Jian Li, and Mingda Qiao. Nearly instance optimal sample complexity bounds for top-k arm selection. In *Artificial Intelligence and Statistics*, pages 101–110. PMLR, 2017.

Shouyuan Chen, Tian Lin, Irwin King, Michael R Lyu, and Wei Chen. Combinatorial pure exploration of multi-armed bandits. *Advances in neural information processing systems*, 27, 2014.

Eyal Even-Dar, Shie Mannor, Yishay Mansour, and Sridhar Mahadevan. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(6), 2006.

Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. *Advances in Neural Information Processing Systems*, 25, 2012.

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.

Dorota Glowacka. Bandit algorithms in recommender systems. RecSys '19, page 574–575, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362436. doi: 10.1145/3298689.3346956. URL https://doi.org/10.1145/3298689.3346956.

Jiri Hron, Karl Krauth, Michael Jordan, and Niki Kilbertus. On component interactions in two-stage recommender systems. *Advances in neural information processing systems*, 34:2744–2757, 2021.

Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil'ucb: An optimal exploration algorithm for multi-armed bandits. In *Conference on Learning Theory*, pages 423–439. PMLR, 2014.

Haotian Jiang, Jian Li, and Mingda Qiao. Practical algorithms for best-k identification in multi-armed bandits. *arXiv preprint arXiv:1705.06894*, 2017.

Satyen Kale, Lev Reyzin, and Robert E Schapire. Non-stochastic bandit slate problems. *Advances in Neural Information Processing Systems*, 23, 2010.

Shivaram Kalyanakrishnan and Peter Stone. Efficient selection of multiple bandit arms: Theory and practice. In *ICML*, volume 10, pages 511–518, 2010.

Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. Pac subset selection in stochastic multi-armed bandits. In *ICML*, volume 12, pages 655–662, 2012.

Emilie Kaufmann and Shivaram Kalyanakrishnan. Information complexity in bandit subset selection. In *Conference on Learning Theory*, pages 228–251. PMLR, 2013.

Nathan Korda, Balazs Szorenyi, and Shuai Li. Distributed clustering of linear bandits in peer to peer networks. In *International conference on machine learning*, pages 1301–1309. PMLR, 2016.

Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*, pages 535–543. PMLR, 2015.

Tyler Lu, Dávid Pál, and Martin Pál. Showing relevant ads via lipschitz context multi-armed bandits. In *Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010. URL `http://jmlr.csail.mit.edu/proceedings/papers/v9/lu10a/lu10a.pdf`.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Ji Yang, Minmin Chen, Jiaxi Tang, Lichan Hong, and Ed H. Chi. Off-policy learning in two-stage recommender systems. In *Proceedings of The Web Conference 2020*, WWW '20, page 463–473, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450370233. doi: 10.1145/3366423.3380130. URL `https://doi.org/10.1145/3366423.3380130`.

Kanak Mahadik, Qingyun Wu, Shuai Li, and Amit Sabne. Fast distributed bandits for online recommendation systems. In *Proceedings of the 34th ACM international conference on supercomputing*, pages 1–13, 2020.

Tao Qin and Tie-Yan Liu. Introducing LETOR 4.0 datasets. *CoRR*, abs/1306.2597, 2013. URL `http://arxiv.org/abs/1306.2597`.

Qinshi Wang and Wei Chen. Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications. *Advances in Neural Information Processing Systems*, 30, 2017.