On the Analysis of Two-Stage Stochastic Bandit

Yumou Liu[1,2], Haoming Li[2], Zhenzhe Zheng[2], Fan Wu[2], and Guihai Chen[2]
[1]The Chinese University of Hong Kong, Shenzhen, [2]Shanghai Jiao Tong University
yumouliu@link.cuhk.edu.cn,{wakkkka,zhengzhenzhe}@sjtu.edu.cn,{fwu,gchen}@cs.sjtu.edu.cn

ABSTRACT

Two-stage bandit-based algorithms have found widespread application in modern online platforms, offering a balance between cost and accuracy. The initial stage involves coarse filtering of a small candidate set of promising items from a large corpus, while the subsequent stage refines the selection and presents a single item to the user. In this work, to the best of our knowledge, we for the first time undertake a theoretical analysis of the two-stage stochastic multi-armed bandit problem. Specifically, we model the two-stage bandit problem as a two-stage online optimization, and conduct a theoretical analysis. We demonstrate that while the optimization objective of the first stage may seem intuitive, it is, in fact, non-trivial. We devise a proxy optimization objective, emphasize the importance of a carefully designed exploration strategy, and establish the theoretical analysis for the application of Upper Confidence Bound (UCB)-based algorithms in the first stage. Furthermore, we provide a regret analysis of the proposed two-stage bandit algorithm, demonstrating a gap-dependent upper bound of $O(\frac{1}{\bar{\Delta}} \log n\bar{\Delta}^2)$, where $\bar{\Delta}$ is the largest reward gap, and a gap-independent lower bound of $\Omega(\sqrt{n})$, where n represents the horizon.

CCS CONCEPTS

• **Networks** → **Network algorithms**; • **Theory of computation** → **Theory and algorithms for application domains**.

KEYWORDS

Multi-Armed Bandit, Two-Stage Systems, Machine Learning

ACM Reference Format:
Yumou Liu[1,2], Haoming Li[2], Zhenzhe Zheng[2], Fan Wu[2], and Guihai Chen[2]. 2024. On the Analysis of Two-Stage Stochastic Bandit. In *Proceedings of ACM Conference (MobiHoc '24).* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3641512.3686360

This work was completed during Yumou Liu's internship at Shanghai Jiao Tong University.
This work was supported in part by National Key R&D Program of China (No. 2023YFB4502400), in part by China NSF grant No. 62322206, 62132018, U2268204, 62025204, 62272307, 62372296. The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies or the government.
Zhenzhe Zheng is the corresponding author.

MOBIHOC '24, October 14–17, 2024, Athens, Greece

ACM ISBN 979-8-4007-0521-2/24/10.
https://doi.org/10.1145/3641512.3686360

1 INTRODUCTION

Online platforms have gained widespread adoption in industry, such as recommendation systems and online advertising [4, 11, 20]. The primary objective of these online platforms is to select a single or several items presented to the user, aiming to maximize Key Performance Indicators (KPIs) such as Click-Through Rate (CTR), Conversion Rate (CVR), etc. Bandit algorithms [18, 19] are widely deployed to the scenarios, where the online platforms do not know user's preference over items, and would like to explore and exploit this information to provide personalized services.

To tackle the challenge of recommending personalized items from an extensive collection of corpus within a constrained response time, the two-stage service paradigm [4, 11, 20] has gained widespread adoption by online platforms. In this two-stage architecture, the first stage serves to coarsely filter a candidate set of items from a large corpus, while the second stage refines the selection, further selecting one item from the candidate set to the user. To mitigate computation overhead, the models in the first stage are designed to be lightweight to make a trade-off between the computational complexity and the accuracy [2, 22, 29]. To guarantee the system performance, a more sophisticated model in the second stage delivers precise KPI predictions, and the item with the highest estimated KPI in the candidate set is recommended to the user. However, these works lack theoretical analysis of the performance guarantee in this new two-stage architecture. A line of recent works on two-stage systems has primarily focused on mobile computing applications, such as on-device recommendation systems and on-device machine learning applications [9, 32]. Aiming to address privacy concerns and reduce response latency, the first stage on the cloud only observes the partial features of the items, where the unobserved features are viewed as the user's privacy, and the second stage on the device observes all the features, representing constraints on computation overhead and privacy concerns [26].

In this work, we analyze the performance of the two-stage online platform in stochastic bandit learning. Specifically, we consider a two-stage stochastic bandit problem with k arms. At time t, the first-stage bandit algorithm filters a candidate set $\mathcal{S}_t$ containing h arms. Subsequently, the second-stage bandit algorithm further selects one arm, and observes the corresponding reward. This procedure repeats for n rounds, and the target of the bandit algorithm is to maximize the cumulative reward.

The first challenge of solving the two-stage stochastic bandit is the design of the first stage. Specifically, since the primary objective of the classical bandit algorithms focused on the result of the second stage, the first stage lacks consideration. In this work, we show that the optimization objective of the first stage is intuitive but non-trivial. We observe that the goal of the first stage is to include the best arm in the candidate set, formulated as an indicator function. However, this formulation cannot be directly optimized, because the online platform lacks the knowledge of which arm is the best.

Thus, we need to design a proxy optimization objective for the first stage. The difficulty of designing the proxy objective lies in the requirement of balancing exploration and exploitation. Specifically, we illustrate the necessity of designing a proper proxy objective by constructing counterexamples, where the regret scales with $O(n)$.

The second challenge arises with the regret analysis of the two-stage stochastic bandit algorithm. Specifically, the challenge lies in the analysis of the first stage, compared with the regret analysis of the classical single-stage bandit algorithms. The crux of the regret analysis is bounding the probability that the optimal arm is selected into the candidate set for the second stage. This event can be decomposed into $O(\binom{k}{h})$ sub-events, making it difficult to formulate and bound its probability.

We propose a new two-stage stochastic bandit algorithm to address the first challenge. Specifically, the optimization objective of the first stage is designed to maximize the probability of the event that the best arm is chosen to the second stage. Then, we propose UCB-SR (Algorithm 1) and UCB-LR (Algorithm 2) to tackle this problem. Specifically, UCB-SR is designed for a special case where the first stage has no access to the feature vectors, while UCB-LR is designed for the general case where the first stage observes some partial feature dimensions. Our algorithms select h arms with the largest UCBs out of the total k arms in the first stage, and play the arm with the largest UCB in the second stage. Then, we establish the regret analysis for the upper bound of our proposed algorithms and the lower bound of this problem. For the second challenge, we relax the probability of the combination of the sub-events by jointly considering all the events that a sub-optimal is played, so that we adopt the two-stage stochastic bandit analysis to the existing regret analysis framework [18]. We prove a sublinear bound of $O\left(k \log\left(\gamma n\right)\right) + R_2(n,h)$, where $\gamma = \binom{k-1}{\lceil \frac{k-1}{2} \rceil} \cdot (k-h)$, and $R_2(n,h)$ is the regret of a bandit algorithm with h arms in horizon n. Regarding the lower bound, we demonstrate that the problem can achieve $\Omega\left(\sqrt{\frac{n(k-1)}{h}}\right)$ regret.

To summarize, our major contributions in this work include:

- We for the first time touch the two-stage stochastic bandit problem, and theoretically analyze the optimization objective of the first stage, demonstrating the impossibility of directly optimizing the objective. We design a proxy optimization objective that makes it possible and efficient for the online platform to optimize.
- We propose two algorithms, UCB-SR and UCB-LR, to solve the two-stage stochastic bandit problem, and establish the regret analysis. We prove an upper bound of $O(\log n)$ for our algorithm and a lower bound of $\Omega(\sqrt{n})$ for the problem.
- We validate our proposed algorithms through experiments on both synthetic and real-world data, with results aligning well with our theoretical claims. The regret of UCB-LR is 83% less than the UCB algorithm with random selection in the first stage.

2 PRELIMINARIES

In this section, we introduce the modeling of the two-stage online platform (Sec. 2.1), and formulate the corresponding problem of two-stage stochastic bandit (Sec. 2.2).

2.1 System Modeling

We model the two-stage online platform in the bandit setting. Generally, an online platform would like to select one item with the largest estimated KPI (such as CTR [34] and CVR [23]) from a large item set to the user, and observes the realized KPI through the user's feedback. If the KPIs are not known to the online platform in advance, this problem can be modeled as a linear stochastic bandit problem [19], in which every item is represented as an arm, and selecting an item is viewed as pulling the corresponding arm. The online platform interacts with the user for several rounds, and learns the KPIs using the feedback. The goal of the online platform is to maximize the cumulative reward, e.g., maximize the number of clicks within a given horizon. Specifically, at the beginning of round t within a finite horizon n, the learner is given the arm set $\mathcal{A}_t \subset \mathbb{R}^d$ where d is the dimension of the arm's feature, and $|\mathcal{A}_t| = k$, from which it chooses an arm $a_t \in \mathcal{A}_t$, and receives the reward as

$$X_t = \langle \theta^*, a_t \rangle + \epsilon_t, \tag{1}$$

where ϵ_t is a random noise with zero mean, and $\theta^* \in \mathbb{R}^d$ is the linear coefficient which captures the relation between the item's feature a_t to its corresponding reward. The θ^* is fixed but unknown to the online platform. Without loss of generality, we denote μ_i as the expected reward of the arm i and $\hat{\mu}_i$ as the empirical mean. The regret is defined by

$$R(n,h) = \mathbb{E}\left[\sum_{t=1}^{n} \max_{a_t \in \mathcal{A}_t} \langle \theta^*, a_t \rangle - \sum_{t=1}^{n} X_t \right], \tag{2}$$

where the expectation is with respect to the selected arms $a_1, \ldots, a_n$ and the corresponding noise $\epsilon_1, \ldots, \epsilon_n$.

The motivation behind splitting the bandit problem in online platform into two stages stems from the computational costs associated with linear stochastic bandits. Computing the upper confidence bound, denoted as $\bar{\mu}_i$, involves evaluating the expression $\hat{\mu}_i + \alpha\sqrt{a_i^T (V_t^T V_t + I_d)^{-1} a_i}$, where a_i is the feature vector of arm i, $V_t := \sum_{\tau=1}^{t} a_\tau^T a_\tau \in \mathbb{R}^{m \times d}$, α is a hyper-parameter, and $I_d \in \mathbb{R}^{d \times d}$ denotes an identity matrix. Computing $\bar{\mu}_i$ is computationally intensive due to the matrix multiplications and inversions involved. Furthermore, in the online platform, we usually have a large corpus of items (around billion-scale [28]), Hence, a two-stage retrieval procedure is employed to alleviate the computational overhead. Specifically, the first stage with an acceptable computation overhead algorithm to coarse-grained filter a small candidate set from the whole item set. The second stage is to select one item out of the candidate set. The first stage alleviates the high computation overhead, while the second stage ensures a high selection accuracy.

Before formulating the two-stage bandit problem, we discuss the objectives of the two stages in detail. The second stage can be viewed as a vanilla one-stage bandit that selects one item within the candidate set from the first stage to get the largest accumulated reward. Thus, the objective of the second stage can be modeled as minimizing the cumulative regret, the same as the vanilla linear stochastic bandit. In contrast, the objective for the first stage is different. In round t, if the first stage fails to filter the best arm into the candidate set, the whole system will incur a constant regret, irrespective of the algorithm applied to the second stage. On the contrary, if the best arm is filtered into the candidate set, whether

the best arm can be finally chosen depends on the second stage. Thus, as long as the best arm is filtered into the candidate set, we can conclude that the first stage succeeds.

2.2 Problem Formulation

We formulate the problem of two-stage bandits in the online platform. In the first stage, the player selects h candidate arms out of all the k candidate arms as the arm set for the subsequent stage. In the second stage, the player evaluates the h candidate arms, and pulls one of them. The problem can be formulated as a two-stage online optimization problem within a horizon n, such that:

$$\max_{a_t \in \mathcal{S}_t} \sum_{t=1}^{n} X_t, \tag{3}$$

$$\text{s.t.} \quad \mathcal{S}_t = \arg\max_{S \subset \mathcal{A}_t, |\mathcal{S}|=h} \bar{r}_t(\mathcal{S}), \quad \forall\, t \in [n], \tag{4}$$

where $\bar{r}_t(\cdot)$ is the optimization objective of the first stage at time t. As discussed above, the goal of the first stage is to ensure that the optimal arm a_t^* is contained in the selected set $\mathcal{S}_t$, and thus:

$$\bar{r}(\mathcal{S}) = \mathbb{1}(a_t^* \in \mathcal{S}). \tag{5}$$

We note that the optimal solution of Equation (5) is not unique.

We explain the above problem formulation from the perspective of the two-stage bandits in detail. First, we consider the objective of the second stage, which is to maximize the cumulative reward (the sum of the observed reward x_t) of the pulled arms, subject to the constraint of the arm set selected by the first stage. Next, we consider the first stage, where Equation (4) shows that the objective of the first stage at time t is to maximize a set indicator function $\bar{r}_t(\cdot)$ under the cardinality constraints.

3 TWO-STAGE BANDITS

In this section, we discuss the design of two-stage bandit algorithms. Specifically, in Sec. 3.1, we consider the setting where no features are available for the first stage. In Sec. 3.2, we consider the case where the first stage can see several dimensions of the feature vector.

3.1 Two-Stage Bandit with Stochastic Retrieval

In this subsection, we examine a simplified scenario of the two-stage bandit. Assuming no features are available for the first stage, we can regard the first stage as a stochastic bandit problem under tabular setting [18] and name it as stochastic retrieval.

Before introducing the detailed method for the first stage, we emphasize the importance of exploration in the first stage. Firstly, we show that exploration in the first stage is necessary by providing a counterexample. Suppose that the first stage selects the items with the top-h estimated reward. When there are h different arms pulled, the first stage will no longer select the other arms into the candidate set, since only the selected h arms have means greater than 0. Thus, the lack of exploration in the first stage leads to a linear regret. Next, we show that a carefully designed exploration strategy is necessary by providing another counterexample. Suppose the first stage selects candidates by uniformly sampling the arms. The probability of the optimal arm being chosen into the candidate set is a constant in every round, denoted by P. Thus, the regret of this algorithm is at least $\Omega(\Delta_{\min} Pn)$, where $\Delta_{\min}$ is the smallest gap between the optimal and sub-optimal arm, indicating linear regret with respect to the horizon n.

Algorithm 1: Two-Stage UCB with Stochastic Retrieval (UCB-SR)

Input: $T_i = 0, \hat{\mu}_{i,0} = 0, \bar{\mu}_{i,0} = 0, \forall i \in [k]$;
Output: The selected arms;

```
for t ← 1, ... T do
    S_t ← arg top_h({μ̄_{1,t-1}, ..., μ̄_{n,t-1}});
    i ← arg max_{i∈S_t} LinUCB(A_i);
    Play arm i and observe X_{i,t};
    for j ∈ S_t do
        T_j ← T_j + 1;
        if j = i then
            μ̂_{i,t} ← (T_i × μ̂_{i,t-1} + X_{i,t})/(T_i + 1);
        μ̄_{j,t} ← μ̂_{j,t} + sqrt(2 log(T) / T_j);
return i;
```

Next, we delve into the design of the exploration method to maximize Equation (5) of the first stage. We revise the optimization objective of the first stage. It is challenging to directly optimize Equation (5) since the bandit player only observes the feedback and cannot know exactly whether the pulled arm is optimal or not. However, the player can be sure about whether an arm is optimal after several rounds with a high probability. Thus, it is possible to introduce a proxy optimization objective whose maximizer is also a maximizer of Equation (5) with a high probability. For simplicity, we assume that all noise ϵ_i is sampled from $\mathcal{N}(0,1)$. Let r_i be the stochastic reward of arm i, and E denote the event that the best arm is in the candidate set. Let E^c as the complementary event of E (formal definition will be given in Section 4.1). Maximizing $\mathbb{P}(E)$ or minimizing $\mathbb{P}(E^c)$ can be viewed as maximizing the probability of $\bar{r}(\mathcal{S}) = 1$ in Equation (5).

However, computing $\mathbb{P}(E^c)$ or $\mathbb{P}(E)$ requires much computational burden and it is necessary to be simplified. We note that $\mathbb{P}(E^c)$ does not equal $\prod_{i\in\mathcal{S}} \mathbb{P}(r_i \le \max_{j\in\mathcal{A}\setminus i} r_j)$ due to the lack of independence among the event $\{r_i \le \max_{j\in\mathcal{A}\setminus i} r_j\}$ for all the arms. To avoid the computational burden of enumerating all event combinations, we introduce a hyper-parameter v and consider events $H_i = \{r_i < v\}$ for all items, ensuring the independence of these events. We then focus on the probability $\prod_{i\in\mathcal{S}} \mathbb{P}(r_i < v) = \prod_{i\in\mathcal{S}} CDF_{\mathcal{N}(0,1)}\left(\frac{v-\mu_i}{\sigma_i}\right)$, which decreases with v. Setting v to be the second-largest value among all μ_i, to minimize $\prod_{i\in\mathcal{S}} \mathbb{P}(r_i < v)$ is equivalent to maximizing the RHS of Equation (5). Since the second largest of μ is unknown to the online platform, we need more assumptions for further analysis. Specifically, we assume that r_i for all the arms have the same variance. Under this assumption, the $\mathcal{S}^* = \arg\min_{\mathcal{S}} \prod_{i\in\mathcal{S}} \mathbb{P}(r_i < v)$ remains constant for any value of v, which will be proved in Appendix B.1. So in the following analysis, we assume v is fixed.

Then, we clarify the optimization objective of the first stage. The probability $\mathbb{P}(\mu_i < v)$ can be estimated by $CDF_{\mathcal{N}(0,1)}\left(\frac{v-\hat{\mu}_i}{\hat{\sigma}_i}\right)$. Then,

considering CDF is non-negative and non-decreasing, minimizing $\prod_{i\in\mathcal{S}} CDF_{\mathcal{N}(0,1)}\left(\frac{v-\mu_i}{\sigma_i}\right)$ is equivalent to minimizing

$$\bar{r}(\mathcal{S}) := \prod_{i\in\mathcal{S}} CDF_{\mathcal{N}(0,1)}\left(\frac{v-\hat{\mu}_i}{\hat{\sigma}_i}\right), \quad \forall i \in \mathcal{A}, \quad |S| = h. \tag{6}$$

However, real applications have complex structures of the environment, and require bandit algorithms to be flexible in tuning the trade-off between exploration and exploitation [6], but Equation (6) lacks flexibility and can hardly be tuned when using it as the first stage optimization objective. Specifically, compared with the vanilla UCB algorithms which can use hyper-parameters to scale the weight of the upper bound and tune the level of exploration, the level of exploration by optimizing Equation (6) cannot be manually tuned. Thus, we need to modify Equation (6) towards the UCB-style. Let $\frac{v-\hat{\mu}_i}{\hat{\sigma}_i} = \alpha$, $v = \hat{\mu}_i + \alpha\hat{\sigma}_i$. Since Equation (6) indicates a smaller α would like to be selected to the second stage, for a fixed v, an arm with larger $\hat{\mu}_i$ and $\hat{\sigma}_i$ is more likely to the selected. Thus, in practice, we set α to be a constant, and an arm with larger $\hat{\mu}_i + \alpha\hat{\sigma}_i$ is more likely to be selected, which is also known as the upper confidence bound $\bar{\mu}_i$ in bandit literature.

Based on the above discussions, we introduce another formulation that is easier to optimize to replace Equation (6), such that:

$$\bar{r}(\mathcal{S}) := \sum_{i\in\mathcal{S}} \hat{\mu}_i + \alpha\hat{\sigma}_i, \quad \forall i \in \mathcal{S}. \tag{7}$$

It is worth noticing that optimizing Equation (7) is equivalent to finding the top-h UCBs at time t. The optimality of Equation (7) will be discussed in the following section.

Finally, we present Algorithm 1 to address the two-stage bandit with stochastic retrieval, which can be viewed as the combination of a stochastic bandit for the first stage and a linear bandit for the second stage. Line 2 applies top-h selection on the estimated UCBs to select a set of arms as the arm set for the subsequent second stage. Line 3 indicates that we employ a classical bandit algorithm, for example, LinUCB, for the second stage. Lines 5-9 describe the update procedure for the first stage. After observing the reward of the pulled arm i from the second stage, the first stage updates the empirical mean of arm i and updates the UCBs of all the arms in $\mathcal{S}_t$. Updating the UCBs of the unpulled arms in $\mathcal{S}_t$ encourages exploration on the other arms outside $\mathcal{S}_t$.

3.2 Two-Stage Bandit with Linear Retrieval

In this subsection, we examine a general case in which the first stage has access to several dimensions of the arm feature vector. Since in Section 2, we assume that the expected reward is generated by a linear model, we apply a linear model in the first stage to retrieve arms. Thus, we name the first stage in this scenario linear retrieval.

We introduce Algorithm 2 tailored to address the linear retrieval scenario. In essence, Algorithm 2 is the fusion of two linear bandit algorithms. Specifically, in Line 2, we select the top-h arms based on the UCBs, which will be analyzed in Section 4.2. Line 3 denotes the application of a linear bandit algorithm for the second stage. Lines 5-9 outline the update procedure of the linear bandit for the first stage after observing the feedback from the second stage. It is noteworthy that, unlike Algorithm 1, Algorithm 2 updates the UCBs of all arms after a single round of play, and the size of the UCB is implicitly represented in Line 9.

Algorithm 2: Two-Stage UCB with Linear Retrieval (UCB-LR)

Input: $\hat{\theta}_{c,0} = 0, \hat{\theta}_{e,0} = 0, V_{c,0} = V_{e,0} = \lambda I$
Output: The selected arms;

1 **for** $t \leftarrow 1, \dots T$ **do**
2 $\quad \mathcal{S}_t \leftarrow$ $\arg\text{top\_h}\left(\left\{\hat{\theta}^T_{c,t-1}a_{c,1} + u_{t,i}, \dots, \hat{\theta}^T_{c,t-1}a_{c,k} + u_{t,k}\right\}\right)$;
3 $\quad i \leftarrow \arg\max_{i\in\mathcal{S}_t} \text{LinUCB}\left(a_{e,i}, \hat{\theta}_{i-1}\right)$;
4 $\quad$ Play arm i and observe $X_{i,t}$;
5 $\quad A_{c,t} \leftarrow a_{c,t}, X_t \leftarrow X_{i,t}$;
6 $\quad V_{c,t} = V_{c,t-1} + a_{c,i}a^T_{c,i}$;
7 $\quad \hat{\theta}_{c,t} \leftarrow V^{-1}_{c,t}\sum_{s=1}^{t} A_{c,s}X_s$;
8 $\quad$ **for** $j \in [k]$ **do**
9 $\quad\quad u_{t+1,j} \leftarrow ||a_{c,j}||_{V^{-1}_{c,t}}\sqrt{2\log(T)}$;
10 **return** i;

4 REGRET ANALYSIS

4.1 Upper Bound of Algorithm 1

In this subsection, we present the regret upper bound for Algorithm 1 of$O(\frac{1}{\bar{\Delta}}\log n\bar{\Delta}^2)$, where $\bar{\Delta}$ is the maximum reward gap. Prior to presenting the result, we establish the optimality of the proxy optimization objective for the first stage, as defined in Equation (7). This is achieved by demonstrating that the maximizer of Equation (7) is also a maximizer of Equation (5) with a high probability. Subsequently, leveraging this high probability, we decompose and relax the regret to derive an upper bound. In the following analysis, without additional explanation, all the mathematical notations are relevant only to the first stage. In this section, only the sketch of the theoretical analysis is provided and please refer to Section A.1 for detailed proofs.

Firstly, we establish the optimality of Equation (7) as a proxy optimization objective for the first stage. To ease the analysis, we define several events that are crucial to the regret.

Definition 1. *Let F_i be the 'good' event for the sub-optimal arm i defined by $F_i := \left\{\bar{\mu}_{i,u_i} < \mu_1\right\}$, where $\bar{\mu}_{i,u_i}$ is the UCB of the arm i after u_i times of play, and $u_i \in [n]$ is a constant to be chosen later.*

If F_i were true, it would indicate that arm i is not over-estimated after pulled for u_i times. Following the idea of this definition, we define the good event for all the sub-optimal arms.

Definition 2. *Let F be the 'good' event for all the sub-optimal arms defined by $F := \left\{\forall\mathcal{T} \subseteq \mathcal{A}\backslash\{1\}, |\mathcal{T}| \geq h, \exists i \in \mathcal{T}, \bar{\mu}_{i,u_i} < \mu_1\right\}$.*

If F were true, it would be indicated that there are at most $h-1$ arms over-estimated after pulled for u_i arms, respectively. Then, we consider the estimation condition of arm 1 and put everything together.

Definition 3. *Let E be the "good" event for the whole environment defined by $E := \left\{\mu_1 < \min_{t\in[n]} \bar{\mu}_{1,t}\right\} \wedge F$.*

If E were true, it would be indicated that the arm 1 is not underestimated and there are at most $h-1$ sub-optimal arms are overestimated, such that the arm 1 would be selected into the candidate set. In the following analysis, we aim to demonstrate two key points:

1. If E occurs, then the event that a sub-optimal arm is pulled will happen at most $T(n) \leq (k-1)\bar{u}$ times.
2. The complement event E^c occurs with low probability.

In the following analysis, we denote the times that arm i till round n is pulled by $T_i(n)$.

Lemma 1 (Bound of the Probability of E^c). *Under the assumption that the reward of each arm follows an 1-sub-Gaussian distribution, the probability of the event E^c is upper bounded by*

$$\mathbb{P}(E^c) \leq n\delta_1 + \gamma \exp\left(-\frac{\bar{u}hc^2\bar{\Delta}^2}{2}\right), \tag{8}$$

where $\gamma = \binom{k-1}{\lceil \frac{k-1}{2} \rceil} \cdot (k-h)$, $\bar{u} = \max_{i \in \mathcal{A}} u_i$ and $\bar{\Delta} = \max_{i \in \mathcal{A}} \Delta_i$.

Lemma 1 indicates that $\mathbb{P}(E^c)$ decreases with $\bar{u}$. After choosing a proper $\bar{u}$, this result will indicate that the bad event E^c happens with a low probability with respect to the horizon n.

Before proving the first claim, we show that if the first and second stage model were asked to select the best one arm, it is less possible for the second stage model to give a wrong answer. We will demonstrate this by showing the variance of in-sample error of the linear regression model.

Lemma 2 (Variance of In-Sample Error of Linear Regression). *Suppose y is a vector of 1-sub-Gaussian variables and A has full rank. If more labels of one feature vector are sampled in the training set, the variance of the prediction error of this vector is reduced.*

An intuitive interpretation of Lemma 2 can be drawn from the bias-variance trade-off perspective. Duplicating a feature vector introduces additional information to the training set, yet the models trained on both pre-duplicated and post-duplicated data remain unbiased. Thus, duplicating the feature vector intuitively aids in reducing the variance. Another interpretation of Lemma 2 is that compared with unstructured stochastic bandit, arms with features will reduce the variance of estimated mean. This lemma indicates that it is easier to train the second stage model since arm features are available, and it will help us bound $T(n)$ in the appendix.

Then, we prove the first claim.

Lemma 3. *If E is true, the event that a sub-optimal arm is pulled will happen at most $T(n) \leq 2(k-1)\bar{u}$ times.*

Next, we decompose the regret into two terms to bound them separately. This decomposition is rooted in Lemma 1. As demonstrated in the lemma, E^c occurs with a bounded low probability, and conversely, E occurs with high probability. By relaxing $\mathbb{P}(E) \to 1$, we establish that the corresponding regret is upper-bounded by the regret of the second stage. Therefore, the subsequent analysis focuses on cases where the first stage fails to retrieve the best arm.

Lemma 4 (Regret Decomposition). *Consider Algorithm 1 on a stochastic bandit instance with k arms and 1-sub-Gaussian rewards. For a horizon n, with probability at least $(1-\delta_1-\delta_2)$, the regret $R(n,k)$ satisfies:*

$$R(n,k) \leq \bar{\Delta} \cdot \left(2(k-1)\bar{u} + \mathbb{P}(E^c)n\right) + R_2(n,h),$$

where $R_L(n,h)$ is upper-bounded with probability $1-\delta_2$.

Then, by putting Lemma 1 and Lemma 4 together, we derive the regret of both Algorithm 1.

Theorem 1 (Regret Upper Bound of Algorithm 1). *Consider the two-stage bandit algorithm presented in Algorithm 1 applied to a k-armed 1-sub-Gaussian bandit problem. For a horizon n, with probability at least $(1-\frac{1}{n^2}-\delta_2)$, the regret is bounded by*

$$R(n) \leq \underbrace{\frac{16(k-1)}{h\bar{\Delta}}\left(1+\log\left(\frac{\gamma n h\bar{\Delta}^2}{16(k-1)}\right)\right)}_{\mathcal{T}_1} + \underbrace{(2k-1)\bar{\Delta}}_{\mathcal{T}_2} + \underbrace{R_2(n,h)}_{\mathcal{T}_3} \tag{9}$$

where $R_2(n,h)$ is the regret of a $h-$armed bandit algorithm with horizon n, and $\gamma = \binom{k-1}{\lceil \frac{k-1}{2} \rceil} \cdot (k-h)$.

Then, we discuss the implication of Theorem 1. Theorem 1 demonstrates that the regret increases with both k and $k-h$. This aligns with the intuition that if more items were available, it would be hard to find the best item, and if fewer items were retrieved, there would be a higher likelihood that the most preferred item remains unexplored, contributing to a larger regret.

Furthermore, we delve into the regret upper bound. $\mathcal{T}_2$ is irrelevant with h. $\mathcal{T}_3 \leq 8\sqrt{nh\log(n)} + 3h\bar{\Delta}$ according to [18]. By relaxing the log term in $\mathcal{T}_1$, there is $\mathcal{T}_1 + \mathcal{T}_3 \leq 8\sqrt{nh\log(n)} - (\gamma' n\bar{\Delta} - 3\bar{\Delta})h$, where $\gamma' = \binom{k-1}{\lceil \frac{k-1}{2} \rceil}$. Then, with reasonable k and n, the worst $h' = \frac{16n\log(n)}{(\gamma' n-3)^2\bar{\Delta}^2}$ is decreasing with k and n. This result indicates that when there is enough time and arms to explore, it is better to employ a larger h to avoid a high regret when the second stage can support more than h' arms.

4.2 Upper Bound of Algorithm 2

In this subsection, we modify the analysis in Section 4.1 to the linear retrieval case in Section 3.2. Before analysis, we assume that the arm vector is bounded.

Assumption 1. *The l^2-norm of the arm vector, i.e. the feature vector, is bounded.*

$$||a||_2 \leq L.$$

Firstly, we demonstrate that the retrieval stage with a linear regression model can be conceptualized as a stochastic unstructured bandit problem with arms having varying variances. We consider the prediction error of the offline linear regression model on a sample drawn from the training set. Suppose that $x = A\theta^* + \epsilon$, where ϵ is zero-mean Gaussian noise with unit variance, $\theta \in \mathbb{R}^d$, and $A \in \mathbb{R}^{n \times d}$ is the combination of arms in a linear bandit instance. Without loss of generality, we assume that A is a full rank matrix. Then we derive the concentration of the prediction error.

Lemma 5. *Under Assumption 1, the prediction error of a sample out of the training data set can be bounded by*

$$\mathbb{P}(e(a,x;t) > \delta) \leq e^{-\frac{\delta^2}{2||(U\Sigma^{\dagger T}W^T)a||_2^2}} \leq e^{-\frac{\lambda\delta^2}{2L^2}}, \tag{10}$$

where $A_t = U_t\Sigma_t W_t^T$ by SVD and A_t is the stack of previously pulled arm vector till t, and $\Sigma_t^\dagger$ is the pseudo-inverse of Σ_t, $e(a,x) =$

$\left(\hat{\theta} - \theta^*\right)^T a$, *and* t *is the number of training samples with* $t > d$, *and* λ *is the smallest eigenvalue of* $A^T A$ *with* $\lambda > 0$, *and* A *is the stacked matrix of all the arm vectors.*

Lemma 5 shows that we can transform the linear bandit problem into a stochastic bandit style with varying reward variances. Specifically, each arm a_i in the bandit instance follows the Gaussian distribution $\mathcal{N}\left(\theta^{*T} a_i, ||(U_t \Sigma_t^{\dagger T} W_t^T) a||_2^2\right)$, and the mean is fixed but the variance is varying. We will provide the relationship between the smallest eigenvalue of A_t and A in Section B.1. Thus, we can consider linear retrieval as a specific type of stochastic bandit, which will be analyzed in Section 4.2.

Theorem 2 (Regret Upper Bound of Algorithm 2). *Consider the two-stage bandit algorithm presented in Algorithm 2 applied to a* k*-armed* 1*-sub-Gaussian bandit problem. For a horizon* n*, with probability at least* $(1 - \frac{1}{n^2} - \delta_2)$*, the regret is bounded by*

$$R(n) \leq \frac{8(k-1)L^2}{h\lambda\bar{\Delta}} \left(1 + \log\left(\frac{\gamma n h \lambda \bar{\Delta}^2}{8(k-1)L^2}\right)\right) + (2k-1)\bar{\Delta} + R_2(n,h),$$

where $R_2(n,h)$ *is the regret of a* $h-$*armed bandit algorithm with horizon* n, $\gamma = \binom{k-1}{\lceil \frac{k-1}{2} \rceil} \cdot (k-h)$, *and* λ *is the smallest singular value of the matrix* $V = A^T A$.

4.3 Lower Bound

In this subsection, we establish the minimax lower bound for our two-stage bandit problem. The underlying principle for proving the lower bound involves constructing two bandit instances that share similarities but possess distinct optimal arms [18]. In such scenarios, distinguishing between instances from a finite-length sequence becomes challenging. This challenge arises from the nature of the problem settings rather than the characteristics of the algorithms employed. Through employing rigorous mathematical techniques, it becomes feasible to derive the lower bound for the sum of cumulative regrets in these two instances. Consequently, we obtain the lower bound for the regret of the two-stage bandit problem. In the subsequent analysis, we adhere to this fundamental approach to introduce and establish the lower bound.

Firstly, before deriving the regret lower bound, we introduce the divergence decomposition lemma [18].

Lemma 6 (Divergence Decomposition [18]). *Let* $\nu = (P_1, \ldots, P_k)$ *be the reward distributions associated with one* k*-armed bandit, and let* $\nu' = (P'_1, \ldots, P'_k)$ *be the reward distributions associated with another* k*-armed bandit. Fix some policy* π *and let* $\mathbb{P}_\nu = \mathbb{P}_{\nu\pi}$ *and* $\mathbb{P}_{\nu'} = \mathbb{P}_{\nu'\pi}$ *be the probability measures on the canonical bandit model induced by the* n*-round interconnection of* ν *and* π, *(respectively* ν' *and* π*). Then,*

$$D(\mathbb{P}_\nu, \mathbb{P}_{\nu'}) = \sum_{i=1}^{k} \mathbb{E}_\nu[T_i(n)] D(P_i, P'_i).$$

Lemma 6 suggests that the relative entropy between measures in the canonical bandit model can be decomposed as the sum of divergences between the reward probabilities of each arm.

Next, with Lemma 6, we show that the regret of the two-stage bandit problem is at least $\Omega\left(\sqrt{nk/h}\right)$.

Theorem 3 (Minimax Lower Bound). *Let* $r \in [0,1]$, *then for any policy:*

$$R(n) \geq \frac{1}{24}\sqrt{\frac{2n(k-1)}{eh}} + R_2(n,h),$$

where $R_2(n,h)$ *is upper-bounded of the second stage.*

The proof of Theorem 3 is given in Section B.2. Theorem 3 demonstrates that the regret increases with k and decreases with h. This aligns with the intuition that if fewer items are retrieved, there is a higher likelihood that the most preferred item remains unexplored, contributing to an increase in regret.

5 EVALUATION

5.1 Synthesis Data

5.1.1 Setup. We establish an environment with $k = 100$ arms, where each arm's context is a $d_e = 10$ dimensional vector, and rewards are generated from Gaussian distributions. Specifically, we randomly generate an arm matrix $A \in \mathbb{R}^{n \times d_e}$ with full rank and create a target model θ^*. The reward for arm i is sampled from a Gaussian distribution $\mathcal{N}(\theta^{*T} a_i, 0.1)$. For the first-stage features, we extract the first $d_c = 5$ dimensions of A, forming a matrix $A_c \in \mathbb{R}^{100 \times 5}$. The experiment is conducted with a horizon of $n = 1000$, and the results for each setting represent the average over 100 independent experiments.

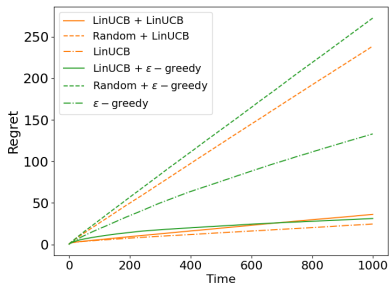
To compare the regret between one-stage and two-stage algorithms, we choose LinUCB and ϵ-Greedy as the second-stage algorithms. For the first stage, we select LinUCB, ϵ-Greedy, and uniformly random selection. We fix $h = 5$ for each two-stage algorithm. In this experiment, $\epsilon = 0.1$ for all ϵ-Greedy implementations, and $\lambda = 0.01$ for all LinUCB implementations.

To assess the impact of h on the cumulative regret of Algorithm 2, we vary h from $\{5, 10, 15, 20, 25\}$ while keeping other settings consistent with the previous experiment.

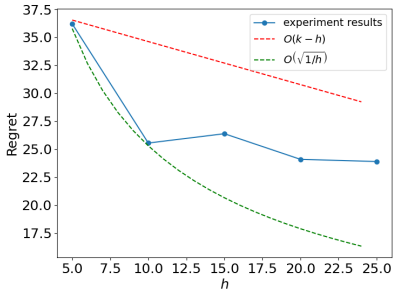
To illustrate the necessity of exploration strategy, we design an experiment based on two-stage ϵ-Greedy by tuning the ϵ of the first stage while fixing $\epsilon = 0.1$ for the second stage. We select ϵ from $\{0.01, 0.05, 0.1, 0.2, 0.4, 0.6, 0.8\}$.

5.1.2 Results and Discussion. Figure 1a depicts the regret of various one-stage and two-stage bandit algorithms on the synthetic data. The legend in Figure 1a follows the format "first stage algorithm + second stage algorithm." Solid lines represent two-stage algorithms with conventional bandit algorithms on each stage, dashed lines denote results with the first stage replaced by the uniformly random selection method, and dash-dot lines depict conventional one-stage bandit algorithms serving as the baseline. Comparing solid lines with corresponding dashed lines reveals that random selection in the first stage boosts performance, highlighting the importance of exploration strategy at the first stage. Furthermore, comparing solid lines with dash-dot lines indicates that, for LinUCB methods, the two-stage algorithm has a larger cumulative regret than the vanilla one-stage algorithm. Conversely, for ϵ-Greedy methods, employing LinUCB in the first stage results in a better performance for the two-stage algorithm compared to the one-stage algorithm, showcasing the superior performance of LinUCB over ϵ-Greedy.

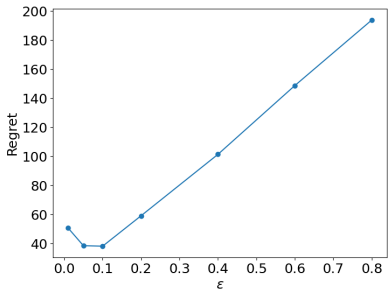
Figure 1b illustrates the regret of Algorithm 2 with varying h. Red and green dashed lines represent the theoretical upper and



(a) Regret comparison among two-stage bandit, random selection in first stage and single-stage bandit algorithms.

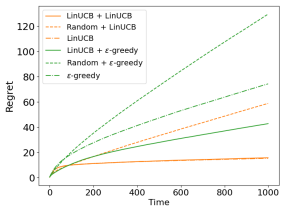


(b) Regret of two-stage bandit with various h, *i.e.* the size of candidate set.

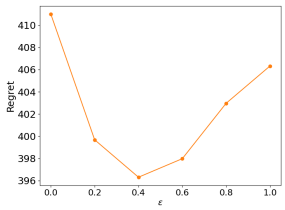


(c) Regret of two-stage bandit with ϵ-Greedy + LinUCB with various ϵ.

Figure 1: Evaluation results on synthesis data.



(a) Regret comparison among algorithms.



(b) Regret of two-stage bandit with ϵ_1-Greedy + ϵ_2-Greedy.

Figure 2: Evaluation results on MovieLens 1M dataset.

lower bounds, respectively. Generally, the regret decreases with increasing h, since a larger h makes it more likely for the best arm to be filtered into the candidate set by the first stage.

Figure 1c emphasizes the significance of exploration strategy at the first stage using ϵ-Greedy + LinUCB. When ϵ is too large, excessive exploration occurs, causing the first stage to uniformly select arms and making it less likely to filter the best arm into the candidate set, resulting in a large cumulative regret. Conversely, when ϵ is too small, the first stage model tends to be stuck, overfitting on noisy samples in the initial rounds and missing the best arm. The comparison between these two extreme cases underscores the importance of setting an appropriate ϵ for the first stage, highlighting the significance of exploration strategy in this context.

5.2 MovieLens 1M Dataset

5.2.1 Setup. We use the ratings of MovieLens 1M dataset to construct the environment for bandit. The ratings data can be reformulated as a big sparse matrix $R^{N_1 \times N_2}$ with missing values where $R_{ij} \in [0, 1]$ denotes the rating of user i to movie j. Then we apply PMF [25] to complete and factorize A into UM^T where $U \in \mathbb{R}^{N_1 \times d_e}$ and $A \in \mathbb{R}^{N_2 \times d_e}$. Each row of A is viewed as the feature vector of a movie and each row of U is viewed as the target model of the corresponding user.

We run the experiment on the first 10 users with the most number of ratings in the original dataset. For each user, we independently repeat the experiment with a horizon of $n = 1000$ for 100 times with random initialization and average the results. We set $d_e = 32$ for the data preparation. For the first-stage features, we extract the first $d_c = 16$ dimensions of A.

To compare the regret between one-stage and two-stage algorithms, we choose LinUCB and ϵ-Greedy as the second-stage algorithms. For the first stage, we select LinUCB, ϵ-Greedy, and uniformly random selection. We fix $h = 20$ for each two-stage algorithm. In this experiment, $\epsilon = 0.1$ for all ϵ-Greedy implementations, and $\lambda = 0.1$ for all LinUCB implementations.

To illustrate the necessity of exploration strategy, we design an experiment based on two-stage ϵ-Greedy by tuning the ϵ of the first stage while fixing $\epsilon = 0.1$ for the second stage. We select ϵ from $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$.

5.2.2 Results and Discussion. Figure 2a illustrates the regret of various one-stage and two-stage bandit algorithms on the synthetic data. The comparison between solid lines and corresponding dashed lines reveals that random selection in the first stage enhances performance, underscoring the importance of exploration strategy at the first stage. Furthermore, contrasting solid lines with dash-dot lines indicates that, for LinUCB methods, the two-stage algorithm has a larger cumulative regret than the vanilla one-stage algorithm. Conversely, for ϵ-Greedy methods, utilizing LinUCB in the first stage results in better performance for the two-stage algorithm compared to the one-stage algorithm, highlighting the superior capabilities of LinUCB over ϵ-Greedy.

Figure 2b accentuates the significance of exploration strategy at the first stage using two-stage ϵ-Greedy. When ϵ is too large, excessive exploration occurs, causing the first stage to uniformly select arms and reducing the likelihood of filtering the best arm into the candidate set, resulting in a substantial cumulative regret. Conversely, when ϵ is too small, the first stage model tends to become stuck, overfitting on noisy samples in the initial rounds and missing the best arm. The comparison between these two extreme cases underscores the importance of setting an appropriate ϵ for the first stage, emphasizing the significance of exploration strategy in this context.

6 RELATED WORK

6.1 Two-Stage Recommender System

Two-stage recommender systems have found widespread application in industries such as YouTube [4] and Pinterest [20]. In this setup, the first stage filters a candidate set with high precision, generally deemed relevant to the user, while the second stage ranks the items for optimal display. Various traditional methods are employed in the first stage, including collaborative filtering and matrix factorization [5, 25], as well as content-based filtering [27]. Lightweight designed deep neural networks are also utilized for candidate set generation [33].

An emerging trend is the deployment of recommender systems on both cloud and edge collaboratively, aiming to reduce cloud resource consumption, latency, and preserve privacy [32]. Alibaba introduced EdgeRec [9], a real-time edge recommender system for the reranking stage. Yao and Wang et al. [31] presented DCCL, a framework for large-scale on-device recommendation model personalization. Yang et al. [30] addressed the challenge of on-device models getting stuck when user interests undergo significant changes. Gong et al. [8] deployed a compact ranking model on devices to capture real-time feedback.

Several theoretical works have analyzed the performance of two-stage recommender systems under different settings. Hron and Krauth et al. [14] considered a different two-stage bandit model where there are multiple nominators (players) in the first stage observing partially overlapped action spaces. The study demonstrated the necessity of synchronizing exploration strategies between the ranker (second stage player) and the nominators. However, they did not provide a theoretical analysis of the regret of two-stage bandits, which is the main contribution of our work. Hron et al. [13] discovered that independent nominator training could lead to performance comparable to uniformly random recommendations and found that careful design of item pools, each assigned to a different nominator, alleviates these issues. Recent work [15] established the asymptotic characteristics of the two-stage recommender system, showing the convergence rate in an offline setting, compared with the online learning setting of our work.

6.2 Bandit in Recommendation

The application of linear contextual bandits to online recommendation was initially introduced by Yahoo [19] for news recommendation. In this context, a news article is considered as an arm, and a ridge regression model is trained using the LinUCB algorithm to estimate the CTR for each article. Subsequent approaches in industrial systems have incorporated various learning algorithms, including ϵ-greedy [24], Thompson sampling [3, 12], and others. In addition to linear models, diverse machine learning models have been explored, such as leveraging deep neural networks to capture KPIs and associated uncertainties [7], as well as employing deep Bayesian models [10].

Another line of research involves understanding user choice when multiple items are recommended, compared to single-item recommendation [19]. The cascade model [17] simplifies user choice, assuming evaluation of items from position 1 to k, clicking on the first satisfying item. Subsequent works extend this to allow multiple clicks with satisfaction probabilities [16] and captures CTR as the product of user preference and display position scores [35]. Industry practices also focus on designing algorithms for delayed feedback scenarios [1, 3].

Bandit algorithms find application in the two-stage recommendation framework as well. Apple [21] proposed a two-layer bandit framework for recommending items on top of search results. A Lower Confidence Bound (LCB) based method is employed in the first stage to prevent distracting users from search results.

7 CONCLUSION

In this paper, we delve into the theoretical analysis of the two-stage multi-armed bandit problem. We conduct a theoretical analysis of the optimization objective design for the first stage and propose a UCB-based two-stage bandit algorithm. Our algorithm is proven to achieve a gap-dependent regret upper bound of $O(\frac{1}{\Delta}\log n\tilde{\Delta}^2)$, while the gap-independent lower bound for this problem is established to be $\Omega(\sqrt{n})$.

A APPENDIX

Here we provide the missing proofs in the main text[1].

A.1 Related Proofs for Theorem 1

To start with, we prove the second claim first. To show that E^c happens with low probability, we employ the following lemmas to show that F_j^c and F^c happens with low probabilities first.

Lemma 7 (Bound of the Probability of F_j^c [18]). *Under the assumption that the reward of each arm follows an 1-sub-Gaussian distribution, the probability of the event F_j^c for sub-optimal arm j is upper bounded by*

$$\mathbb{P}(F_j^c) \le \exp\left(-\frac{u_j c^2 \Delta_j^2}{2}\right),$$

where $\Delta_j = \mu_1 - \mu_j$ is the gap between arm j and the optimal arm, $c \in (0,1)$ is a hyper-parameter to be chosen later and $\Delta_j - \sqrt{\frac{2\log(1/\delta_1)}{u_j}} \ge c\Delta_i$.

Lemma 8 (Bound of the Probability of $\{\mu_1 \ge \min_{t\in[n]} \bar{\mu}_{1,t}\}$[18]). *Under the assumption that the reward of each arm follows an 1-sub-Gaussian distribution, the probability of the event F_j^c for sub-optimal arm j is upper bounded by*

$$\mathbb{P}\left(\left\{\mu_1 \ge \min_{t\in[n]} \bar{\mu}_{1,t}\right\}\right) \le n\delta_1.$$

With Lemma 7 and Lemma 8, we can now derive the bound of the probability of E^c.

Proof of Lemma 1. First, we decompose the event F^c. Since the F^c indicates that there are at least h arms over-estimated, we enumerate all the possible cases such that there are h to $k-1$ arms over-estimated. It is worth noticing that all these cases are disjoint.

[1]More proof details are provided in https://drive.google.com/file/d/1t6Z7VXcZxm4jF2FPdDkIRBDbGHXYP5rT/view?usp=drive_link

Thus, we can decompose the probability of $\mathbb{P}(E^c)$ using a sum of probabilities, such that

$$\mathbb{P}(F^c) = \sum_{i=h}^{k-1} \sum_{j\in\mathcal{T}_i,\mathcal{T}_i\subseteq\mathcal{A},|\mathcal{T}_i|=i} \underbrace{\prod_{j\in\mathcal{T}_i} \mathbb{P}(F_j^c)}_{\mathsf{T}_1} \underbrace{\prod_{m\in\mathcal{A}\setminus\{1\}\setminus\mathcal{T}_i} (1-\mathbb{P}(F_m^c))}_{:=\mathsf{T}_2}$$

$$\leq \sum_{i=h}^{k-1} \binom{k-1}{i} \mathbb{P}\left(\bar{F}^c\right)^i$$

$$\leq \binom{k-1}{\lceil \frac{k-1}{2} \rceil} \cdot (k-h) \cdot \exp\left(-\frac{\bar{u}c^2\bar{\Delta}^2 h}{2}\right) = \gamma \cdot \exp\left(-\frac{\bar{u}c^2\bar{\Delta}^2 h}{2}\right),$$

where $\mathbb{P}\left(\bar{F}^c\right) = \max_{j\in\mathcal{A}\setminus\{1\}} \mathbb{P}(F_j^c)$. We obtain the second line by relaxing T_2 to 1 since $1-\mathbb{P}(F_m^c) < 1$, and take the maximum possible $\mathbb{P}(F_j^c)$ to relax T_1. The third line is obtained by using Lemma 7, and then taking the maximum over u_j, Δ_j and the number of combinations. It is worth noticing that $\bar{u}$ and $\bar{\Delta}$ may not be corresponded to the same arm.

By putting Lemma 8 and $\mathbb{P}(F^c)$ together, we obtain Equation (8). □

Proof of Lemma 2. Let X denote the training set with distinct feature vectors, and let $a \in X$ be a feature vector. The variance of the prediction error of a is $\sigma = \|U\Sigma^{\dagger T}W^T a\|^2$, where a is stacked into X. Suppose we sample one more label of a in an extended training set X_+. Then the variance of the prediction error becomes $\sigma_+ = \|U_+\Sigma_+^{\dagger T}W_+^T a\|^2$. Assume, for the sake of contradiction, that $\sigma < \sigma_+$. This implies $trace(\Sigma^{\dagger T}) \leq trace(\Sigma_+^{\dagger T})$, leading to $\|\Sigma\|_F > \|\Sigma_+\|_F$. Since X_+ has one more vector than X, $\|\Sigma\|_F \leq \|\Sigma\|$, which results in a contradiction. □

Proof of Lemma 3. If E is true, F and $\left\{\mu_1 < \min_{t\in[n]} \bar{\mu}_{1,t}\right\}$ are true. Then there are at least $k-1-h$ and at most $k-1$ sub-optimal arms, for example, arm i, such that F_i is true. Let $G_i = \left\{\mu_1 < \min_{t\in[n]} \bar{\mu}_{1,t}\right\} \wedge F_i$. Suppose that there is a single-bandit instance with the arms $\mathcal{A}' \subseteq \mathcal{A}$, $\{1,i\} \subseteq \mathcal{A}'$ and vanilla UCB algorithm, and it has been proven by [18] that $T_i'(n) \leq u_i$ if G_i is true where $T_i'(n)$ is the times that arm i is pulled in this single-stage instance. We notice that

$$T_i'(n) = \sum_{t=1}^{n} \mathbb{1}(a_t' = i|\mathcal{A}'),$$

$$T_i(n) = \sum_{t=1}^{n} \mathbb{1}(i\in\mathcal{S}_t)\mathbb{1}(a_t = i|\mathcal{S}_t)(\mathbb{1}(1\in\mathcal{S}_t) + \mathbb{1}(1\notin\mathcal{S}_t))$$

$$\leq T_i'(n) + \sum_{t=1}^{n} \mathbb{1}(i\in\mathcal{S}_t)\mathbb{1}(a_t = i|\mathcal{S}_t)\mathbb{1}(1\notin\mathcal{S}_t),$$

where the inequality holds because when $\mathbb{1}(i\in\mathcal{S}_t)\mathbb{1}(a_t = i|\mathcal{S}_t) = 1$, the second stage can be viewed as the instance $\mathcal{A}'$. Although Lemma 5 shows that the variance of the empirical mean of an arm has a potentially large upper bound for infinitely many arms, in the bandit instance with fixed finitely many arms, the variance is still $O(1/u_i)$ which can be deduced by Lemma 2. Thus the result in [18] that when G_i happens, the arm i is played for at most u_i times still holds. Thus,

$$T(n) = \sum_{i\in\mathcal{A}\setminus\{1\}} T_i(n)$$

$$\leq (k-1)\bar{u} + \sum_{i\in\mathcal{A}\setminus\{1\}} \sum_{t=1}^{n} \mathbb{1}(i\in\mathcal{S}_t)\mathbb{1}(a_t = i|\mathcal{S}_t)\mathbb{1}(1\notin\mathcal{S}_t)$$

$$= (k-1)\bar{u} + \sum_{t=1}^{n} \mathbb{1}(1\notin\mathcal{S}_t) \leq 2(k-1)\bar{u},$$

where the last inequality holds because when every time the optimal arm 1 is not in $\mathcal{S}_t$, there is a sub-optimal arm played. So the total times that $1 \notin \mathcal{S}_t$ should not be larger than the times that sub-optimal arms are played. □

Proof of Lemma 4. We denote $T(n)$ by the times that the optimal arm is not played in horizon n. Because $T(n) \leq n$, this will mean that

$$\mathbb{E}[T(n)] = \mathbb{E}[\mathbb{I}\{E\}T(n)] + \mathbb{E}[\mathbb{I}\{E^c\}T(n)] \leq 2(k-1)\bar{u} + \mathbb{P}(E^c)n. \tag{11}$$

Similarly, the regret can also be regarded as the composition of the regret when E occurs and when it does not.

$$R(n) \leq \sum_{t=1}^{n} \bar{\Delta}\cdot\mathbb{I}\{1\notin\mathcal{S}_t\} + r_{2,t}\mathbb{I}\{1\in\mathcal{S}_t\} \leq \bar{\Delta}\cdot\mathbb{E}[T(n)] + R_2(n,h). \tag{12}$$

Equation (12) indicates that the regret can be decomposed into two terms. The first term represents the upper bound of regret when the first stage fails to filter the best arm into the candidate set. The second term represents the complementary case. Since the event that the best arm is in the candidate set should happen with a high probability, we relax this probability to 1 to ease the analysis.

Substituting Equation (11) in Equation (12), then the result is obtained. □

Proof of Theorem 1. By substituting Equation (8) into Equation (4), we get

$$R(n,k) \leq \bar{\Delta}\cdot\left(2(k-1)\bar{u} + n\left(n\delta_1 + \gamma\exp\left(-\frac{\bar{u}hc^2\bar{\Delta}^2}{2}\right)\right)\right) + R_2(n,h).$$

Let $\bar{u} = \left\lceil \frac{2}{hc^2\Delta^2}\log\left(\frac{\gamma nhc^2\bar{\Delta}^2}{4(k-1)}\right)\right\rceil + 1$,

$$R(n,k) \leq \bar{\Delta}\cdot\left(2(k-1)\left\lceil\frac{2}{hc^2\Delta^2}\log\left(\frac{\gamma nhc^2\bar{\Delta}^2}{4(k-1)}\right)\right\rceil + k + \frac{4(k-1)}{hc^2\bar{\Delta}^2}\right) + R_2(n,h)$$

$$\leq \frac{4(k-1)}{hc^2\bar{\Delta}}\left(1+\log\left(\frac{\gamma nhc^2\bar{\Delta}^2}{4(k-1)}\right)\right) + (2k-1)\bar{\Delta} + R_2(n,h)$$

$$\underset{c=1/2}{\leq} \frac{16(k-1)}{h\bar{\Delta}}\left(1+\log\left(\frac{\gamma nh\bar{\Delta}^2}{16(k-1)}\right)\right) + (2k-1)\bar{\Delta} + R_2(n,h).$$

Then we discuss the probability that the upper bound holds. The right-hand side of Equation (9) can be viewed as $R_1 + R_2$. We denote the event that R_i holds as J_1 and the event that R_2 holds as J_2. The probability of each event is $1-\delta_1$ and $1-\delta_2$, respectively. Then the probability of $R_1 + R_2$ holds is $\mathbb{P}(J_2 \wedge J_1) = 1 - \mathbb{P}(J_1^c \vee J_2^c) \geq 1 - (\mathbb{P}(J_1^c) + \mathbb{P}(J_2^c)) \geq 1 - \delta_1 - \delta_2$. □

REFERENCES

[1] Bendada, W., Salha, G., and Bontempelli, T. Carousel personalization in music streaming apps with contextual bandits. In *Proceedings of the 14th ACM Conference on Recommender Systems* (2020), pp. 420–425.

[2] Borisyuk, F., Kenthapadi, K., Stein, D., and Zhao, B. Casmos: A framework for learning candidate selection models over structured queries and documents. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2016), pp. 441–450.

[3] Chapelle, O., and Li, L. An empirical evaluation of thompson sampling. *Advances in neural information processing systems 24* (2011).

[4] Covington, P., Adams, J., and Sargin, E. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems* (2016), pp. 191–198.

[5] Das, A. S., Datar, M., Garg, A., and Rajaram, S. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web* (2007), pp. 271–280.

[6] Ding, Q., Kang, Y., Liu, Y.-W., Lee, T. C. M., Hsieh, C.-J., and Sharpnack, J. Syndicated bandits: A framework for auto tuning hyper-parameters in contextual bandit algorithms. *Advances in Neural Information Processing Systems* (2022), 1170–1181.

[7] Eide, S., and Zhou, N. Deep neural network marketplace recommenders in online experiments. In *Proceedings of the 12th ACM Conference on Recommender Systems* (2018), pp. 387–391.

[8] Gong, X., Feng, Q., Zhang, Y., Qin, J., Ding, W., Li, B., Jiang, P., and Gai, K. Real-time short video recommendation on mobile devices. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management* (2022), pp. 3103–3112.

[9] Gong, Y., Jiang, Z., Feng, Y., Hu, B., Zhao, K., Liu, Q., and Ou, W. Edgerec: recommender system on edge in mobile taobao. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (2020), pp. 2477–2484.

[10] Guo, D., Ktena, S. I., Myana, P. K., Huszar, F., Shi, W., Tejani, A., Kneier, M., and Das, S. Deep bayesian bandits: Exploring in online personalized recommendations. In *Proceedings of the 14th ACM Conference on Recommender Systems* (2020), pp. 456–461.

[11] Higley, K., Oldridge, E., Ak, R., Rabhi, S., and de Souza Pereira Moreira, G. Building and deploying a multi-stage recommender system with merlin. In *Proceedings of the 16th ACM Conference on Recommender Systems* (2022), pp. 632–635.

[12] Hill, D. N., Nassif, H., Liu, Y., Iyer, A., and Vishwanathan, S. An efficient bandit algorithm for realtime multivariate optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2017), pp. 1813–1821.

[13] Hron, J., Krauth, K., Jordan, M., and Kilbertus, N. On component interactions in two-stage recommender systems. *Advances in neural information processing systems 34* (2021), 2744–2757.

[14] Hron, J., Krauth, K., Jordan, M. I., and Kilbertus, N. Exploration in two-stage recommender systems. *arXiv preprint arXiv:2009.08956* (2020).

[15] Jaiswal, A. K. Towards a theoretical understanding of two-stage recommender systems, 2024.

[16] Katariya, S., Kveton, B., Szepesvari, C., and Wen, Z. Dcm bandits: Learning to rank with multiple clicks. In *International Conference on Machine Learning* (2016), pp. 1215–1224.

[17] Kveton, B., Szepesvari, C., Wen, Z., and Ashkan, A. Cascading bandits: Learning to rank in the cascade model. In *International Conference on Machine Learning* (2015), pp. 767–776.

[18] Lattimore, T., and Szepesvári, C. *Bandit algorithms*. Cambridge University Press, 2020.

[19] Li, L., Chu, W., Langford, J., and Schapire, R. E. A contextual-bandit approach to personalized news article recommendation. In *WWW* (2010).

[20] Liu, D. C., Rogers, S., Shiau, R., Kislyuk, D., Ma, K. C., Zhong, Z., Liu, J., and Jing, Y. Related pins at pinterest: The evolution of a real-world recommender system. In *Proceedings of the 26th international conference on world wide web companion* (2017), pp. 583–592.

[21] Ma, S., Das, P., Nikolakaki, S. M., Chen, Q., and Topcu Altintas, H. Two-layer bandit optimization for recommendations. In *Proceedings of the 16th ACM Conference on Recommender Systems* (2022), pp. 509–511.

[22] Ma, X., Wang, P., Zhao, H., Liu, S., Zhao, C., Lin, W., Lee, K.-C., Xu, J., and Zheng, B. Towards a better tradeoff between effectiveness and efficiency in pre-ranking: A learnable feature selection based approach. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval* (2021), pp. 2036–2040.

[23] Ma, X., Zhao, L., Huang, G., Wang, Z., Hu, Z., Zhu, X., and Gai, K. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval* (2018), pp. 1137–1140.

[24] McInerney, J., Lacker, B., Hansen, S., Higley, K., Bouchard, H., Gruson, A., and Mehrotra, R. Explore, exploit, and explain: personalizing explainable recommendations with bandits. In *Proceedings of the 12th ACM Conference on Recommender Systems* (2018), pp. 31–39.

[25] Mnih, A., and Salakhutdinov, R. R. Probabilistic matrix factorization. *Advances in Neural Information Processing Systems* (2007), 1257–1264.

[26] Niu, C., Wu, F., Tang, S., Hua, L., Jia, R., Lv, C., Wu, Z., and Chen, G. Billion-scale federated learning on mobile clients: A submodel design with tunable privacy. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking* (2020), pp. 1–14.

[27] Pazzani, M. J., and Billsus, D. Content-based recommendation systems. In *The adaptive web: methods and strategies of web personalization*. Springer, 2007, pp. 325–341.

[28] Wang, J., Huang, P., Zhao, H., Zhang, Z., Zhao, B., and Lee, D. L. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2018), pp. 839–848.

[29] Wang, Z., Zhao, L., Jiang, B., Zhou, G., Zhu, X., and Gai, K. Cold: Towards the next generation of pre-ranking system. *arXiv preprint arXiv:2007.16122* (2020).

[30] Yao, J., Wang, F., Ding, X., Chen, S., Han, B., Zhou, J., and Yang, H. Device-cloud collaborative recommendation via meta controller. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2022), pp. 4353–4362.

[31] Yao, J., Wang, F., Jia, K., Han, B., Zhou, J., and Yang, H. Device-cloud collaborative learning for recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (2021), pp. 3865–3874.

[32] Yin, H., Qu, L., Chen, T., Yuan, W., Zheng, R., Long, J., Xia, X., Shi, Y., and Zhang, C. On-device recommender systems: A comprehensive survey. *arXiv preprint arXiv:2401.11441* (2024).

[33] Zhou, G., Fan, Y., Cui, R., Bian, W., Zhu, X., and Gai, K. Rocket launching: A universal and efficient framework for training well-performing light net. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2018), pp. 4580–4587.

[34] Zhou, G., Zhu, X., Song, C., Fan, Y., Zhu, H., Ma, X., Yan, Y., Jin, J., Li, H., and Gai, K. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2018), pp. 1059–1068.

[35] Zoghi, M., Tunys, T., Ghavamzadeh, M., Kveton, B., Szepesvari, C., and Wen, Z. Online learning to rank in stochastic click models. In *International Conference on Machine Learning* (2017), pp. 4199–4208.