Stochastic Low-rank Tensor Bandits for Multi-dimensional Online Decision Making

Jie Zhou[*], Botao Hao[†], Zheng Wen[‡], Jingfei Zhang[§] and Will Wei Sun[¶]

November 9, 2023

Abstract

Multi-dimensional online decision making plays a crucial role in many real applications such as online recommendation and digital marketing. In these problems, a decision at each time is a combination of choices from different types of entities. To solve it, we introduce stochastic low-rank tensor bandits, a class of bandits whose mean rewards can be represented as a low-rank tensor. We consider two settings, tensor bandits without context and tensor bandits with context. In the first setting, the platform aims to find the optimal decision with the highest expected reward, a.k.a, the largest entry of true reward tensor. In the second setting, some modes of the tensor are contexts and the rest modes are decisions, and the goal is to find the optimal decision given the contextual information. We propose two learning algorithms `tensor elimination` and `tensor epoch-greedy` for tensor bandits without context, and derive finite-time regret bounds for them. Comparing with existing competitive methods, `tensor elimination` has the best overall regret bound and `tensor epoch-greedy` has a sharper dependency on dimensions of the reward tensor. Furthermore, we develop a practically effective Bayesian algorithm called `tensor ensemble sampling` for tensor bandits with context. Extensive simulations and real analysis in online advertising data back up our theoretical findings and show that our algorithms outperform various state-of-the-art approaches that ignore the tensor low-rank structure.

Key Words: Bandit algorithms; Finite-time regret bounds; Online decision making; Tensor completion.

[*]Amazon, Email: jiezhoua@amazon.com

[†]Deepmind, Email: bhao@google.com

[‡]Deepmind, Email: zhengwen@google.com

[§]Goizueta Business School, Emory University. Email: jingfei.zhang@emory.edu

[¶]Daniels School of Business, Purdue University. Email: sun244@purdue.edu. Corresponding author.

1 Introduction

The tensor, which is also called multidimensional array, is well recognized as a powerful tool to represent complex and unstructured data. Tensor data are prevalent in a wide range of applications such as recommender systems, computer vision, bioinformatics, operations research, and etc (Frolov and Oseledets, 2017; Bi et al., 2018; Song et al., 2019; Bi et al., 2021, 2022). The growing availability of tensor data provides a unique opportunity for decision-makers to efficiently develop multi-dimensional decisions for individuals. In this paper, we introduce tensor bandits problem where a decision, also called an arm, is a combination of choices from different entity types, and the expected rewards formulate a tensor. The problem is motivated by numerous applications in which the agent (the platform) must recommend multiple different entity types as one arm. For example, in an advertising campaign a marketer wants to promote a new product with various promotion offers. The goal is to choose an optimal triple user segment×offer×channel for this new product to boost the effectiveness of the advertising campaign. At each time, after making an action, i.e., pulling the arm (user i, offer j, channel k), the leaner receives a reward, e.g., clicking status or revenue, indicating the user segment i's feedback on promotion offer j on marketing channel k. The rewards of all these three-dimensional arms formulate an order-three tensor, see Figure 1 for an illustration. Similarly, a clothing website may want to recommend the triple top×bottom×shoes to a user that fits the best together. Each arm is the triple of three entities. In these applications, the agent needs to pull an arm by considering multiple entities together and learn to decide which arm provides the highest reward.

Traditional tensor methods focus on static systems where agents do not interact with the environment, and typically suffer the cold-start issue in the absence of information from new customers, new products or new contexts (Song et al., 2019). However, in many real applications, agents receive feedback from the environment interactively and new subjects enter the system sequentially. See Figure 1 for an illustration of such interactive sequential

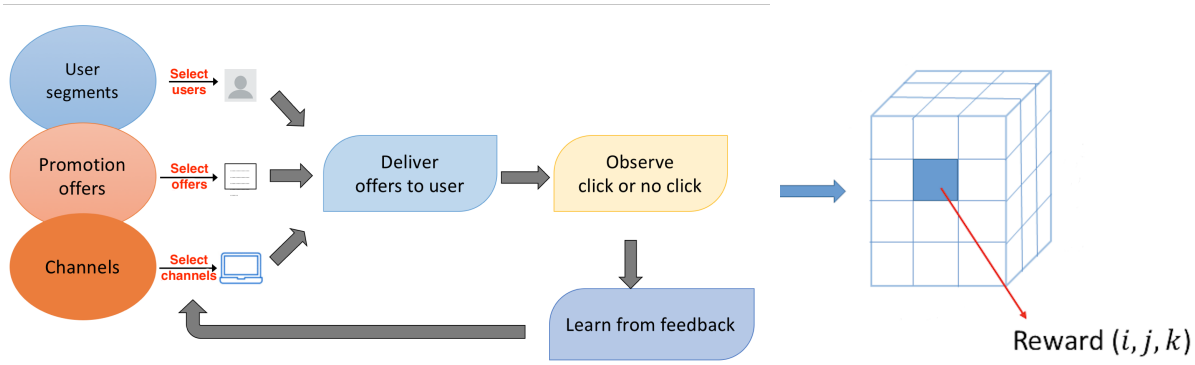


Figure 1: An example of interactive multi-dimensional online decision making. The rewards from all sequential multi-dimensional decisions formulate a tensor.

decision making. In each round, the agent recommends a promotion offer to a chosen user segment in a channel, and then the agent receives a feedback from this user segment. Based on this instant feedback, the agent needs to update the model to improve the user targeting accuracy in the future.

Bandit problems are basic instances of interactive sequential decision making and now play an important role in vast applications such as revenue management, online advertising, and recommender system (Li et al., 2010; Bubeck and Cesa-Bianchi, 2012; Lattimore and Szepesvári, 2020). In bandit problems, at each time step the agent chooses an arm/action from a list of choices based on the action-reward pairs observed so far, and receives a random reward that is conditionally independently drawn from the unknown reward distribution given the chosen action. The objective is to learn the optimal arm that maximizes the sum of the expected rewards. The heart of bandit problems is to address the fundamental trade-off between exploration and exploitation in sequential experiments. At each time step, after receiving the feedback from users, the agent faces a decision dilemma. The agent can either exploit the current estimates to optimize decisions or explore new arms to improve the estimates and achieve higher payoffs in the future. Our considered tensor bandits problem can be viewed as a higher-order extension of the standard bandit problem, which generalizes a scalar arm to a multi-dimensional arm and correspondingly generalizes a vector reward to

a tensor reward case.

In this article, we introduce stochastic low-rank tensor bandits for multi-dimensional online decision-making problems. These are a class of bandits whose mean rewards can be represented as a low-rank tensor and arms are selected from different entity types. The assumption of low-rankness is well-adopted in the literature on tensors. It effectively reduces model complexity and finds widespread applications in practical scenarios such as online recommendation systems and digital marketing (Sun et al., 2017; Bi et al., 2021; Idé et al., 2022). More practical justifications for the use of low-rank tensors can be found in the survey paper Song et al. (2019). To balance the exploration-exploitation trade-off, we propose two algorithms for tensor bandits, `tensor epoch-greedy` and `tensor elimination`. The `tensor epoch-greedy` proceeds in epochs, with each epoch consisting of an exploration phase and an exploitation phase. In the exploration phase, arms are randomly selected and in the exploitation phase, arms that expect the highest reward are pulled. The number of steps in each exploitation phase increases with number of epochs, guided by the fact that, as the number of epochs increases, the estimation accuracy of the true reward improves and more exploitation steps are desirable. For `tensor elimination`, we incorporate the low-rank structure of reward tensor to transform the tensor bandit into linear bandit problem with low-dimension and then employ the upper confidence band (UCB) (Lai and Robbins, 1985) to enable the uncertainty quantification. The UCB has been very successful in bandit problems, leading to an extensive literature on UCB algorithms for standard multi-armed bandits (Lattimore and Szepesvári, 2020). However, employing the successful UCB strategy in low-rank tensor bandits encounters a critical challenge, as the tensor decomposition is a non-convex problem. When the data is not uniformly randomly collected but adaptively collected, the concentration results for the low-rank tensor components remain elusive thus far. Our `tensor elimination` approach considers a tensor spectral-based rotation strategy that preserves the tensor low-rank information and meanwhile enables uncertainty quantification.

Algorithm	Regret bound
`tensor epoch-greedy`	$\widetilde{\mathcal{O}}(p^{d/2} + p^{(d+1)/3} n^{2/3})$
`tensor elimination`	$\widetilde{\mathcal{O}}(p^{d/2} + p^{(d-1)/2} n^{1/2})$
`vectorized UCB`	$\widetilde{\mathcal{O}}(p^{d} + p^{d/2} n^{1/2})$
`matricized ESTR`	$\widetilde{\mathcal{O}}(p^{d-1} + p^{3(d-1)/2} n^{1/2})$

Table 1: Regret bounds of our proposed `tensor epoch-greedy` and `tensor elimination`, as well as the competitors `vectorized UCB` and `matricized ESTR`. Here n denotes the time horizon, $p = \max\{p_1, \ldots, p_d\}$ denotes the maximum tensor dimension and d denotes the order of the reward tensor. We consider $d \geq 3$, the maximum tensor rank $r = \mathcal{O}(1)$, and use $\widetilde{\mathcal{O}}$ to denote $\mathcal{O}$ ignoring logarithmic factors.

In addition to these methodological contributions, in theory we further derive the finite-time regret bounds of our proposed algorithms and show the improvement over existing methods. Low-rank tensor structure has imposed fundamental challenges, as the proof strategies for existing bandit algorithms are not directly applicable to our tensor bandits problem. So the regret analysis of tensor bandits demands new technical tools. In theory, we show that two existing competitors: (1) `vectorized UCB` which vectorizes the reward tensor into a vector and then applies UCB (Auer, 2002); and (2) `matricized ESTR` which unfolds the reward tensor into a matrix and then applies matrix bandit ESTR (Jun et al., 2019), both lead to sub-optimal regret bounds. Table 1 illustrates the comparison of our regret bounds and the regret bounds of these two competitors. Importantly, we prove that `tensor epoch-greedy` has better dependency on tensor dimensions and worse dependency on time horizon compared with the other methods. Therefore, it has superiority over other methods in two scenarios: (1) when the time horizon is short, e.g., the market campaign has a small time budget; or (2) when the dimensions are high. In contrast, the regret bound of `tensor elimination` is always better than the two existing competitors due to its sharper dependency on the dimensions, and also has advantages over `tensor epoch-greedy` when time horizon is long since it has better dependency on time horizon. These theoretical guarantees and insights are important as they help us better understand the algorithms and when one might be preferred over the other.

Finally, we consider an interesting extension of tensor bandits when the contextual information is available. In the aforementioned tensor bandits setting, the goal is to find the optimal arm corresponding to the largest entry of the reward tensor. This setting is called tensor bandits without context. When some modes of the reward tensor are contextual information, we encounter contextual tensor bandits. Take the online advertising data considered in Section 6 as an example. Users use the online platform on some day of the week, and the platform can only decide which advertisement to show to this given user at the given time. In this example, the user mode and the day-of-week mode of the reward tensor are both contextual information and both are not decided by the platform. This is the key difference to the user targeting example shown in Figure 1. Because of this, many of the aforementioned methods are no longer applicable. In this paper we further develop `tensor ensemble sampling` for contextual tensor bandits that utilizes Thompson sampling (Russo et al., 2018) and ensemble sampling (Lu and Van Roy, 2017). Thompson sampling is a powerful Bayesian algorithm that can be used to address a wide range of online decision problems. The algorithm, in its basic form, first initializes a prior distribution over model parameters, and then samples from its posterior distribution calculated using past observations. Finally, an action is made to maximize the reward given the sampled parameters. The posterior distribution can be derived in closed-form in a few special cases such as the Bernoulli bandit (Russo et al., 2018). With more complex models such as our low-rank tensor bandit problem, the exact calculation of the posterior distribution becomes intractable. In this case, we consider an ensemble sampling approach (Lu and Van Roy, 2017) that aims to approximate Thompson sampling while maintaining computational tractability. In an online advertising application, our `tensor ensemble sampling` is empirically successful and reduces the cumulative regret by 75% compared to the benchmark methods.

There are several lines of research that are related to but also clearly distinctive of the problem we address. The first line is tensor completion (Yuan and Zhang, 2016; Song et al.,

2019; Zhang et al., 2019; Cai et al., 2021; Xia et al., 2021; Han et al., 2022). While we employ similar low-dimensional structures as tensor completion, the two problems have fundamental difference. First, a key assumption in existing tensor completion is to assume the observed entries are collected uniformly and randomly (the only exception is Zhang et al. (2019) which assumes a special cross structure of the missing mechanism). This is largely different from our interactive online decision problem where the observed entries are collected adaptively based on some bandit policy. The difference is analogous to that between linear regression and linear bandit (Lattimore and Szepesvári, 2020). Second, the goal of existing tensor completion is to predict all missing entries while the goal of tensor bandits is to find the *largest* entry in the reward tensor so that the cumulative regret is minimized. Third, these tensor completion algorithms are developed for off-line settings where data are collected all at once. They are not applicable to our online decision problem where data enter the system sequentially. On the other hand, existing online tensor completion (Yu et al., 2015; Ahn et al., 2021) for streaming data could not handle our interactive decision problem due to their uniform and random missing mechanism and non-interaction nature.

The second line of related work is low-rank matrix bandit. There are some works considering special rank-1 matrix bandits (Katariya et al., 2017b,a; Trinh et al., 2020). To find the largest entry of a non-negative rank-1 matrix, one just needs to identify the largest values of the left-singular and right-singular vectors. However, this is no longer applicable for higher-rank matrices. For general low-rank matrix bandits, Kveton et al. (2017) handled low-rank matrix bandits but imposed strong "hott topics" assumptions on the mean reward matrix. They assumed all rows of decomposed factor matrix can be written as a convex combination of a subset of rows. Sen et al. (2017) considered low-rank matrix bandits with one dimension choosing by the nature and the other dimension choosing by the agent. They derived a logarithmic regret under a constant gap assumption. However the gap may not be specified in advance. Lu et al. (2018) utilized ensemble sampling for low-rank matrix

bandits but did not provide any regret guarantee due to the theoretical challenges in handling sampling-based exploration. Jun et al. (2019) proposed a bilinear bandit that can be viewed as a contextual low-rank matrix bandit. However, their regret bound becomes sub-optimal in the context-free setting due to the use of LinUCB (Abbasi-Yadkori et al., 2011) for linear bandits with finitely many arms. In addition, our theory shows that unfolding reward tensor into matrix and then applying algorithm proposed by Jun et al. (2019) leads to a suboptimal regret bound. Lu et al. (2021) further generalized Jun et al. (2019) to a low-rank generalized linear bandit. To the best of our knowledge, there is no existing work that systematically studies tensor bandits problem. Low-rank tensor structure has imposed fundamental challenges. It is well known that many efficient tools for matrix data, such as nuclear norm minimization or singular value decomposition, cannot be simply extended to tensor framework (Richard and Montanari, 2014; Yuan and Zhang, 2016; Friedland and Lim, 2017; Zhang and Xia, 2018). Hence existing algorithms and proof strategies for linear bandits or matrix bandits are not directly applicable to our tensor bandits problem. Our proposed algorithms and their regret analysis demand new technical tools.

The rest of the paper is organized as follows. Section 2 reviews some notation and tensor algebra. Section 3 presents our model, two main algorithms and their theoretical analysis for the tensor bandits. Section 4 considers the extension to the contextual tensor bandits. Section 5 contains a series of simulation studies. Section 6 applies our algorithms to an online advertising application to illustrate their practical advantages. All proofs, an analysis of approximate Thompson sampling, and additional implementation details in the experiments are included in the supplemental material.

2 Notation and Tensor Algebra

A *tensor* is a multidimensional array and the *order* of a tensor is the number of dimensions it has, also referred to as the *mode*. We denote vectors using lower-case bold letters (e.g., $\mathbf{x}$),

matrices using upper-case bold letters (e.g., $\mathbf{X}$), and high-order tensors using upper-case bold script letters (e.g., $\mathcal{X}$). We denote the cardinality of a set by $|\cdot|$ and write $[k] = \{1, 2, \ldots, k\}$ for an integer $k \geq 1$. For a positive scalar x, let $\lceil x \rceil = \min\{z \in \mathbb{N}^+ : z \geq x\}$. We use $\mathbf{e}_j \in \mathbb{R}^p$ to denote a basis vector that takes 1 as its j-th entry and 0 otherwise. For a vector $\mathbf{a} \in \mathbb{R}^d$ and $s_1 \leq s_2 \in [d]$, let $\mathbf{a}_{s_1:s_2}$ be the sub-vector $(\mathbf{a}_{s_1}, \mathbf{a}_{s_1+1}, \ldots, \mathbf{a}_{s_2})$. For an order-$d$ tensor $\mathcal{X} \in \mathbb{R}^{p_1 \times \cdots \times p_d}$, define its mode-$j$ fibers as the p_j-dimensional vectors $\mathcal{X}_{i_1,\ldots,i_{j-1},\cdot,i_{j+1},\ldots,i_d}$, and its mode-$j$ matricization as $\mathcal{M}_j(\mathcal{X}) \in \mathbb{R}^{p_j \times (p_1 \cdots p_{j-1} p_{j+1} \cdots p_d)}$, where the column vectors of $\mathcal{M}_j(\mathcal{X})$ are the mode-j fibers of $\mathcal{X}$. For instance, for an order-3 tensor $\mathcal{X} \in \mathbb{R}^{p_1 \times p_2 \times p_3}$, its mode-1 matricization $\mathcal{M}_1(\mathcal{X}) \in \mathbb{R}^{p_1 \times (p_2 p_3)}$ is defined as, for $i \in [p_1], j \in [p_2], k \in [p_3]$,

$$[\mathcal{M}_1(\mathcal{X})]_{i,(j-1)p_3+k} = \mathcal{X}_{i,j,k}. \tag{1}$$

For a tensor $\mathcal{X} \in \mathbb{R}^{p_1 \times p_2 \times \cdots \times p_d}$ and a matrix $\mathbf{Y} \in \mathbb{R}^{r_1 \times p_1}$, we define the marginal multiplication $\mathcal{X} \times_1 \mathbf{Y} \in \mathbb{R}^{r_1 \times p_2 \times \cdots \times p_d}$ as

$$\mathcal{X} \times_1 \mathbf{Y} = \Big(\sum_{i_1'=1}^{p_1} \mathcal{X}_{i_1', i_2, \ldots, i_d} \mathbf{Y}_{i_1, i_1'} \Big)_{i_1 \in [r_1], i_2 \in [p_2] \ldots, i_d \in [p_d]}. \tag{2}$$

Marginal multiplications along other modes, i.e., $\times_2, \ldots, \times_d$, can be defined similarly. For $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{p_1 \times \cdots \times p_d}$, define the tensor inner product as $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1 \in [p_1], \ldots, i_d \in [p_d]} \mathcal{X}_{i_1,\ldots,i_d} \mathcal{Y}_{i_1,\ldots,i_d}$. The tensor Frobenius norm is defined as $\|\mathcal{X}\|_{\mathrm{F}} = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}$, and the element-wise tensor max norm is defined as $\|\mathcal{X}\|_\infty = \max_{i_1,\ldots,i_d} |\mathcal{X}_{i_1,\ldots,i_d}|$.

Consider again an order-d tensor $\mathcal{X} \in \mathbb{R}^{p_1 \times \cdots \times p_d}$. Letting r_j be the rank of matrix $\mathcal{M}_j(\mathcal{X})$, $j \in [d]$, the tensor Tucker rank of $\mathcal{X}$ is the d-tuple $(r_1, \ldots, r_d)$. Let $\mathbf{U}_1 \in \mathbb{R}^{p_1 \times r_1}, \ldots, \mathbf{U}_d \in \mathbb{R}^{p_d \times r_d}$ be the matrices whose columns are the left singular vectors of $\mathcal{M}_1(\mathcal{X}), \ldots, \mathcal{M}_d(\mathcal{X})$, respectively. Then, there exists a core tensor $\mathcal{S} \in \mathbb{R}^{r_1 \times \cdots \times r_d}$ such that

$$\mathcal{X} = \mathcal{S} \times_1 \mathbf{U}_1 \times_2 \cdots \times_d \mathbf{U}_d,$$

or equivalently, $\mathcal{X}_{i_1,\ldots,i_d} = \sum_{i_1' \in [r_1], \ldots, i_d' \in [r_d]} S_{i_1', \ldots, i_d'} [\mathbf{U}_1]_{i_1, i_1'} \cdots [\mathbf{U}_d]_{i_d, i_d'}$. The above decomposition is often referred to as the tensor Tucker decomposition (Kolda and Bader, 2009).

3 Tensor Bandits

In this section, we first introduce tensor bandits, followed by two new algorithms – `tensor elimination` and `tensor epoch-greedy`. We then establish the finite-time regret bounds of these two algorithms, which reveal their different performances under different rate conditions and provide a useful guidance for their implementations in practice.

In tensor bandits problem, the agent interacts with an environment for n time steps, and at each step, the agent faces a d-dimensional decision, indexed by $[p_1] \times \cdots \times [p_d]$. A standard multi-armed bandit can be regarded as a special case of tensor bandits with $d = 1$. At step $t \in [n]$ and given past interactions, the agent pulls an arm I_t, which denotes a d-tuple $(i_{1,t}, \ldots, i_{d,t}) \in [p_1] \times \ldots \times [p_d]$. Correspondingly, the agent observes a reward $y_t \in \mathbb{R}$, drawn from a probability distribution associated with the arm I_t. Specifically, denoting the true reward tensor as $\mathcal{X} \in \mathbb{R}^{p_1 \times \ldots \times p_d}$, the agent at time t receives a noisy reward

$$y_t = \langle \mathcal{X}, \mathcal{A}_t \rangle + \epsilon_t, \text{ with } \mathcal{A}_t = \mathbf{e}_{i_{1,t}} \circ \cdots \circ \mathbf{e}_{i_{d,t}}, \tag{3}$$

where "$\circ$" denotes the vector outer product, $\mathbf{e}_{i_{j,t}} \in \mathbb{R}^{p_j}$ is a basis vector, $j \in [d]$, and $\mathcal{A}_t$ is a tensor indicating the location of the arm I_t. For example, if the agent pulls $I_t = (i_{1,t}, \ldots, i_{d,t})$, then the $(i_{1,t}, \ldots, i_{d,t})$-th entry of $\mathcal{A}_t$ is 1 while all other entries are 0. In (3), ϵ_t is a random noise term, assumed to be sub-Gaussian in Assumption 1.

The goal of our work, aligned with the central task in bandit problems, is to strike the right balance between exploration and exploitation, and to minimize the cumulative regret. Let the arm with the maximum true reward be

$$(i_1^*, \ldots, i_d^*) = \underset{i_1 \in [p_1], \ldots, i_d \in [p_d]}{\operatorname{argmax}} \langle \mathcal{X}, \mathbf{e}_{i_1} \circ \cdots \circ \mathbf{e}_{i_d} \rangle$$

and correspondingly, denote $\mathcal{A}^* = \mathbf{e}_{i_1^*} \circ \cdots \circ \mathbf{e}_{i_d^*}$. Our objective is to minimize the cumulative regret (Audibert et al., 2009), defined as

$$\mathrm{R}_n = \sum_{t=1}^{n} \langle \mathcal{X}, \mathcal{A}^* \rangle - \sum_{t=1}^{n} \langle \mathcal{X}, \mathcal{A}_t \rangle. \tag{4}$$

Naturally, at each step $t \in [n]$, the agent faces an exploitation-exploration dilemma, in that the agent can either choose the arm that expects the highest reward based on historical data (exploitation), so as to reduce immediate regret, or choose some under-explored arms to gather information about their associated reward (exploration), so as to reduce future regret.

At first glance, the tensor bandit problem posed in (3)-(4) can be re-formulated, via vectorization, as a standard multi-armed bandit problem of dimension $p_1 \times \ldots \times p_d$. However, applying the existing algorithms for standard multi-armed bandits to vectorized tensor bandits may be inappropriate due to several reasons. First, the majority of existing solutions for multi-armed bandits require a proper initialization phase where each arm is pulled at least once, in order to give a well-defined solution (Auer et al., 2002). For tensor bandits, such an initialization step can be computationally expensive or even infeasible, especially when $p_1 \times \ldots \times p_d$ is large. Second, the vectorization approach may result in a severe loss of information, as the intrinsic structures (e.g., low-rank) of tensors are largely ignored after vectorization. Indeed, as commonly considered in recommendation systems and other applications (Kolda and Bader, 2009; Allen, 2012; Jain and Oh, 2014; Bi et al., 2018; Song et al., 2019; Xia et al., 2021; Bi et al., 2021), tensor objects usually have a low-rank structure and can be represented in a lower-dimensional space.

In this work, we propose to retain the tensor form of $\mathcal{X}$ and assume that it admits the following low-rank decomposition,

$$\mathcal{X} = \mathcal{S} \times_1 \mathbf{U}_1 \times_2 \cdots \times_d \mathbf{U}_d, \tag{5}$$

where $\mathcal{S} \in \mathbb{R}^{r_1 \times \cdots \times r_d}$ is a core tensor, and $\mathbf{U}_1 \in \mathbb{R}^{p_1 \times r_1}, \ldots, \mathbf{U}_d \in \mathbb{R}^{p_d \times r_d}$ are matrices with orthonormal columns; see more details on this decomposition in Section 2. We consider a low-rank model where the rank r_i is much smaller than p_i. The low-rank assumption in (5) exploits the structures in tensors and efficiently reduces the number of free parameters in $\mathcal{X}$. Consider a special case where r_i is fixed and $p_1 = \ldots = p_d = p$ for simplicity. This low-rank modeling allows us to consider an efficient initialization phase with $\mathcal{O}(p^{d/2})$ steps (see Lemma

S3), which is much reduced comparing to the p^d steps required in the simple vectorization strategy. As demonstrate in Table 1, comparing to the vectorized solutions that ignore the low-rank structure, our proposed low-rank tensor bandit algorithms have much improved finite-time regret bounds.

Before discussing the main algorithms, we first describe our initialization procedure. Thanks to the tensor low-rank structure, our initialization phase need not to pull every arm at least once, which is required in the majority of multi-armed bandit algorithms. Define an initial set of s_1 steps

$$\mathcal{E}_1 = \{t \mid t \in [s_1]\}, \tag{6}$$

where s_1 is an integer to be specified later in Assumption 3. In the initialization phase, arms are pulled with a uniform probability, equivalent to assuming $\mathbb{P}(i_{jt} = k) = 1/p_j$, $k \in [p_j]$, in (3). If some prior knowledge about the true reward tensor is available, a non-uniform sampling can also be considered in the initialization phase.

3.1 Tensor Elimination

The upper confidence band (UCB) strategies (Lai and Robbins, 1985) have been very successful in bandit problems, leading to an extensive literature on UCB algorithms for standard multi-armed bandits (Lattimore and Szepesvári, 2020). These UCB algorithms balance between exploration and exploitation based on a confidence bound that the algorithm assigns to each arm. Specifically, in each round of steps, the UCB algorithm constructs an upper confidence bound for the reward associated with each arm, and the arms with the highest upper bounds are pulled, as they may be associated with high rewards and/or large uncertainties (i.e., under-explored). Many work have analyzed the regret bounds of UCB algorithms and investigated their optimality (Auer et al., 2002; Garivier and Cappé, 2011).

Employing the successful UCB strategy in low-rank tensor bandits encounters a critical challenge, as the tensor decomposition in (5) is a non-convex problem, the data is adaptively

collected and the concentration results for $\widehat{\mathcal{S}}, \widehat{\mathbf{U}}_1, \ldots, \widehat{\mathbf{U}}_d$, to our knowledge, remain elusive thus far. Without such concentration results, constructing the confidence bounds becomes a very difficult problem. One straightforward strategy is to first vectorize the tensor bandits and then treat the problem as a standard multi-armed bandit problem. However, as discussed before, this strategy incurs a severe loss of structural information and is demanding, in terms of sample complexity, in its initialization phase. In our proposed approach, we consider a tensor spectral-based rotation strategy that preserves the low-rank information and at the same time, enables uncertainty quantification. We also consider an elimination step that eliminate less promising arms based on the calculated confidence bounds, which further improves the finite-time regret bound (see Theorem 1). Taken together, the proposed `tensor elimination` algorithm avoids directly characterizing the uncertainty of tensor decomposition estimators, effectively utilizes the low-rank information and achieves a desirable sub-linear finite-time regret bound. Next, we discuss the `tensor elimination` algorithm in details.

The `tensor elimination` shown in Algorithm 1 starts with an initialization phase of length s_1 and then proceeds to an exploration phase of length n_1, where arms in both phases are selected randomly. In this algorithm, the initialization phase and exploration phase are same. We choose to separate them so that the format is consistent with the `tensor epoch-greedy` algorithm introduced in next subsection. Here, s_1 is set to be the minimal sample size for tensor completion and n_1 is chosen to minimize cumulative regret, both of which will be specified later in Section 3.2. Based on the random samples collected from the initialization and exploration phases, we calculate estimates $\widehat{\mathbf{U}}_1, \ldots, \widehat{\mathbf{U}}_d$ of the matrices $\mathbf{U}_1, \ldots, \mathbf{U}_d$ in (5) using a low-rank tensor completion method (see Appendix S.3.2). Next, we consider a rotation technique that preserves the tensor low-rank structure, and enables vectorization and uncertainty quantification (see Lemma 1). Specifically, given $\widehat{\mathbf{U}}_j$, $j \in [d]$, define $\widehat{\mathbf{U}}_{j\perp}$ whose columns are the orthogonal basis of the subspace complement to the column

Algorithm 1 `Tensor elimination`

Input: number of total steps n, number of exploration steps n_1, regularization parameters λ_1, λ_2, length of confidence intervals ξ, ranks $r_1, \cdots, r_d$.
 # initialization and exploration phases
Initialize: $\mathcal{D} = \emptyset$.
for $t = 1, \cdots, s_1 + n_1$ **do**
  Randomly pull an arm $\mathcal{A}_t$ and receive its associated reward y_t. Let $\mathcal{D} = \mathcal{D} \cup \{(y_t, \mathcal{A}_t)\}$.
end for
Calculate $\widehat{\mathbf{U}}_1, \ldots, \widehat{\mathbf{U}}_d$ using $\mathcal{D}$, and then find $\widehat{\mathbf{U}}_{1\perp}, \ldots, \widehat{\mathbf{U}}_{d\perp}$.
 # reduction phase
Construct an action set $\mathbb{A}_1$ as in (9) and denote $q = \Pi_{j=1}^d p_j - \Pi_{j=1}^d (p_j - r_j)$.
for $k = 1$ to $\log_2(n)$ **do**
  Set $V_{t_k} = \mathrm{diag}(\underbrace{\lambda_1, \ldots, \lambda_1}_{q}, \lambda_2, \ldots, \lambda_2)$ and $\mathcal{D} = \emptyset$.
  for $t = t_k$ to $\min(t_{k+1} - 1, n - n_1 - s_1)$ **do**
    Pull the arm $A_t = \mathrm{argmax}_{a \in \mathbb{A}_k} \|\mathbf{a}\|_{V_t^{-1}}$.
    Receive its associated reward y_t and update $V_{t+1} = V_t + A_t A_t^\top$. Let $\mathcal{D} = \mathcal{D} \cup \{(y_t, A_t)\}$.
  end for
  Eliminate arms based on confidence intervals:

$$\mathbb{A}_{k+1} = \Big\{ \mathbf{a} \in \mathbb{A}_k : \langle \widehat{\boldsymbol{\beta}}_k, \mathbf{a} \rangle + \|\mathbf{a}\|_{V_t^{-1}} \xi \geq \max_{\mathbf{a} \in \mathbb{A}_k} \Big[\langle \widehat{\boldsymbol{\beta}}_k, \mathbf{a} \rangle - \|\mathbf{a}\|_{V_t^{-1}} \xi \Big] \Big\}, \text{ where} \tag{7}$$

$$\widehat{\boldsymbol{\beta}}_k = \underset{\boldsymbol{\beta}}{\mathrm{argmin}} \Big\{ \frac{1}{2} \sum_{(y_t, A_t) \in \mathcal{D}} \big(y_t - \langle A_t, \boldsymbol{\beta} \rangle \big)^2 + \frac{1}{2} \lambda_1 \|\boldsymbol{\beta}_{1:q}\|_2 + \frac{1}{2} \lambda_2 \|\boldsymbol{\beta}_{(q+1):\Pi_{j=1}^d p_j}\|_2 \Big\}. \tag{8}$$

end for

subspace of $\widehat{\mathbf{U}}_j$. Consider a rotation to the true reward tensor $\mathcal{X}$ calculated as

$$\mathcal{Y} = \mathcal{X} \times_1 [\widehat{\mathbf{U}}_1; \widehat{\mathbf{U}}_{1\perp}] \times_2 \cdots \times_d [\widehat{\mathbf{U}}_d; \widehat{\mathbf{U}}_{d\perp}] \in \mathbb{R}^{p_1 \times \cdots \times p_d},$$

where $\times_1, \ldots, \times_d$ are as defined in (2) and $[\widehat{\mathbf{U}}_j; \widehat{\mathbf{U}}_{j\perp}]$ is the concatenation (by columns) of $\widehat{\mathbf{U}}_j$ and $\widehat{\mathbf{U}}_{j\perp}$. Correspondingly, the reward defined in (3) can be re-written (see proof in Appendix S.3.1) as

$$y_t = \Big\langle \mathcal{Y}, [\widehat{\mathbf{U}}_1; \widehat{\mathbf{U}}_{1\perp}]^\top \mathbf{e}_{i_{1,t}} \circ \cdots \circ [\widehat{\mathbf{U}}_d; \widehat{\mathbf{U}}_{d\perp}]^\top \mathbf{e}_{i_{d,t}} \Big\rangle + \epsilon_t.$$

It is seen that replacing the reward tensor $\mathcal{X}$ with $\mathcal{Y}$ and the arm $\mathbf{e}_{i_1} \circ \cdots \circ \mathbf{e}_{i_d}$ with $[\widehat{\mathbf{U}}_1; \widehat{\mathbf{U}}_{1\perp}]^\top \mathbf{e}_{i_{1,t}} \circ \cdots \circ [\widehat{\mathbf{U}}_d; \widehat{\mathbf{U}}_{d\perp}]^\top \mathbf{e}_{i_{d,t}}$ does not change the tensor bandit problem. Define $\boldsymbol{\beta} = \mathrm{vec}(\mathcal{Y}) \in \mathbb{R}^{\Pi_{j=1}^d p_j}$, which vectorizes the reward tensor $\mathcal{Y}$ such that the first $\Pi_{j=1}^d r_j$ entries

of $\text{vec}(\mathcal{Y})$ are $\mathcal{Y}_{i_1,\ldots,i_d}$ for $i_j \in \{1,\ldots,r_j\}$, $j \in [d]$, and denote the corresponding vectorized arm set as

$$\mathbb{A} := \Big\{ \text{vec}\Big([\widehat{\mathbf{U}}_1; \widehat{\mathbf{U}}_{1\perp}]^\top \mathbf{e}_{i_1} \circ \cdots \circ [\widehat{\mathbf{U}}_d; \widehat{\mathbf{U}}_{d\perp}]^\top \mathbf{e}_{i_d} \Big), i_1 \in [p_1], \ldots, i_d \in [p_d] \Big\}. \tag{9}$$

Correspondingly, the tensor bandits in (3) with the true reward tensor $\mathcal{X}$ and arm set $\{\mathbf{e}_{i_1} \circ \cdots \circ \mathbf{e}_{i_d}, i_1 \in [p_1], \cdots, i_d \in [p_d]\}$ can be re-formulated as a multi-armed bandits with the reward vector $\boldsymbol{\beta}$ and arm set $\mathbb{A}$.

It is easy to see that in $\text{vec}(\mathcal{X} \times_1 [\mathbf{U}_1; \mathbf{U}_{1\perp}] \times_2 \cdots \times_d [\mathbf{U}_d; \mathbf{U}_{d\perp}])$, the first $\Pi_{j=1}^d r_j$ entries are nonzero and the last $\Pi_{j=1}^d (p_j - r_j)$ entries are zero. Such a sparsity pattern cannot be achieved if $\mathcal{X}$ is vectorized directly without the rotation. From this perspective, the rotation strategy preserves the structural information in the vectorized tensor. Specifically, when estimating the reward vector $\boldsymbol{\beta}$ in (8), we apply different regularizations to the first $\Pi_{j=1}^d r_j$ entries and the remaining $\Pi_{j=1}^d (p_j - r_j)$ entries, respectively.

The algorithm then proceeds to the elimination phase, where less promising arms are identified and eliminated. This phase aims to further improve the regret bound. Given a vector $\mathbf{a}$, we define its $\mathbf{A}$-norm as $\|\mathbf{a}\|_{\mathbf{A}} = \sqrt{\mathbf{a}^\top \mathbf{A} \mathbf{a}}$, where $\mathbf{A}$ is a positive definite matrix. During phase k with the arm set $\mathbb{A}_k$, the confidence ellipsoid of the mean reward of each arm $\mathbf{a} \in \mathbb{A}_k$ is constructed using $\widehat{\boldsymbol{\beta}}_k$. It is shown in Lemma 1 that the confidence width of the reward of arm $\mathbf{a}$ is $\|\mathbf{a}\|_{V^{-1}} \xi$, where V is the covariance matrix and ξ is a fixed constant term that does not depend on $\mathbf{a}$. At each time step t, the algorithm (line 13) then pulls the arm with the largest confidence interval width. The intuition of the arm selection in this step is that arms with the highest confidence widths are likely under-explored. At the end of phase k (line 16), we implement an elimination procedure that trims less promising arms. Specifically, we first update the estimate $\widehat{\boldsymbol{\beta}}_k$ in (8) based on the pulled arms and their associated rewards during phase k. Based on the estimated reward $\widehat{\boldsymbol{\beta}}_k$, we then construct confidence interval (7) for the mean reward of each arm and eliminate the arms whose upper confidence bound is lower than the maximum of lower confidence bounds of all arms in $\mathbb{A}_k$.

3.2 Regret Analysis of Tensor Elimination

In this section, we carry out the regret analysis of the `tensor elimination`. To ease notation, we assume the tensor rank $r_1 = \ldots = r_d = r$ and the tensor dimension $p_1 = \ldots = p_d = p$. The results for general ranks and dimensions can be established similarly using a more involved notation system. We first state some assumptions.

Assumption 1 (Sub-Gaussian noise)**.** *The noise term ϵ_t is assumed to follow a 1-sub-Gaussian distribution such that, for any $\lambda \in \mathbb{R}$,*

$$\mathbb{E}[\exp(\lambda \epsilon_t)] \leq \exp(\lambda^2/2).$$

Assumption 2. *Assume true reward tensor $\mathcal{X}$ admits the low-rank decomposition in (5) with $r = \mathcal{O}(p^{1/(d-1)})$. In addition, we assume $\|\mathcal{X}\|_\infty \leq 1$ and $p^{d/2}\|\mathcal{X}\|_\infty / \|\mathcal{X}\|_F = \mathcal{O}(1)$.*

The assumption $\|\mathcal{X}\|_\infty \leq 1$ assumes that the reward is bounded, and it is common in the multi-armed bandit literature (see, for example, Langford and Zhang, 2007). It implies that the immediate regret in each exploration step is $\mathcal{O}(1)$. Similar boundedness conditions on tensor entries can also be found in the tensor completion literature (see, for example, Cai et al., 2021; Xia et al., 2021). The assumption on the rank r refers to a low-rank model assumption and is to simplify the final sample size requirement. Moreover, $p^{d/2}\|\mathcal{X}\|_\infty / \|\mathcal{X}\|_F$ measures the spikiness of the true tensor and its boundedness ensures that low-rank tensor completion based on randomly observed samples can be reliable. This condition is a typical incoherence assumption that is used in Xia et al. (2021); Cai et al. (2021) and is also common in other low-rank models (Negahban and Wainwright, 2012).

Assumption 3. *Assume the number of steps in the initialization phase s_1 is*

$$s_1 = C_0 r^{(d-2)/2} p^{d/2}, \tag{10}$$

where C_0 is a positive constant as defined in Lemma S3.

This assumption requires the minimal sample complexity for provably recovering a low-rank tensor from noisy observations when the entries are observed randomly (see Lemma S3 and Xia et al. (2021)). Such random initialization phase is standard and important in all bandit algorithms (Lattimore and Szepesvári, 2020). As discussed before, the simple vectorization strategy would require $s_1 = \mathcal{O}(p^d)$, which is significantly larger.

The next lemma provides the confidence interval for the reward of a fixed arm $\mathbf{a}$.

Lemma 1. *For any fixed vector $\mathbf{a} \in \mathbb{R}^{p^d}$ and $\delta > 0$, we have that, if*

$$\xi = 2\sqrt{14\log(2/\delta)} + \sqrt{\lambda_1}\|\boldsymbol{\beta}_{1:q}\|_2 + \sqrt{\lambda_2}\|\boldsymbol{\beta}_{(q+1):p^d}\|_2, \tag{11}$$

with $\boldsymbol{\beta} = vec(\mathcal{Y})$, $\lambda_1 > 0$ and $\lambda_2 = n/(q\log(1+n/\lambda_1))$, then at the beginning of phase k

$$\mathbb{P}(|\mathbf{a}^\top(\widehat{\boldsymbol{\beta}}_k - \boldsymbol{\beta})| \le \xi\|\mathbf{a}\|_{V_t^{-1}}) \ge 1-\delta,$$

where $V_t = \sum_{s=1}^t A_s A_s^\top + \mathrm{diag}(\underbrace{\lambda_1, \ldots, \lambda_1}_{q}, \lambda_2, \ldots, \lambda_2)$.

Next, we show the finite-time regret bound for `tensor elimination`. Recall $q = \Pi_{j=1}^d p_j - \Pi_{j=1}^d (p_j - r_j)$.

Theorem 1. *Suppose Assumptions 1-3 hold. Let $t_k = 2^{k-1}$, $0 < \lambda_1 \le 1/p^d$, $\lambda_2 = n/(q\log(1+n/\lambda_1))$, and*

$$n_1 = \left\lceil n^{\frac{2}{d+2}} \frac{r^d}{\Pi_{j=1}^d \sigma_j} p^{\frac{d^2+d}{2}} \log^{d/2}(p) \right\rceil, \tag{12}$$

where σ_j is the smallest non-zero singular value of $\mathcal{M}_j(\mathcal{X})$, $j \in [d]$. The cumulative regret of Algorithm 1 satisfies

$$R_n \le C\Big(r^{\frac{d}{2}} p^{\frac{d}{2}} + \Big(\frac{r^d}{\Pi_{i=1}^d \sigma_i}\log^{d/2}(p)\Big)^{\frac{2}{d+2}} p^{\frac{d^2+d}{d+2}} n^{\frac{2}{d+2}} + \sqrt{(d\log(p)+\log(n))^2 p^{d-1} n}\Big),$$

with probability at least $1 - dp^{-10} - 1/n$, where $C > 0$ is some constant.

The detailed proof of Theorem 1 is deferred to Appendix S.1.1. It should be noted that this paper focuses on the high-dimensional setting, where p approaches infinity, to ensure

the probability approaches 1. A similar prerequisite of $p \to \infty$ is also essential in both the bilinear matrix bandits (Jun et al., 2019) and low-rank tensor model (Xia et al., 2021) to ensure that the probability approaches 1.probability approaches 1. Ignoring any logarithmic and constant factor, the above regret bound can be simplified to

$$\mathrm{R}_n = \widetilde{\mathcal{O}}(r^{\frac{d-2}{2}} p^{\frac{d}{2}} + r^{\frac{2d}{d+2}} p^{\frac{d^2+d}{d+2}} n^{\frac{2}{d+2}} + p^{\frac{d-1}{2}} n^{\frac{1}{2}}). \tag{13}$$

The upper bound on the cumulative regret is the sum of three terms, with the first two terms characterizing regret from the s_1 initialization steps and n_1 exploration steps, respectively, and the third term quantifying the regret in the $n - s_1 - n_1$ elimination steps. As the regret from the exploration phase increases with n_1 and the regret from the elimination phase decreases with n_1, the value for n_1 in (12) is chosen to minimize the sum of these two regrets. Note that after the rotation, the order of $\|\boldsymbol{\beta}_{1:q}\|_2$ is of $\widetilde{\mathcal{O}}(p^{d/2})$ which guides the choice of λ_1. One component of the upper bound of the cumulative regret is $\log\left(\frac{\det(V_k)}{\det(\Lambda)}\right)$ with $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_1, \lambda_2, \ldots, \lambda_2)$ and λ_2 is chosen to minimize the upper bound of the log term so as to minimize the upper bound of cumulative regret. Theorem 1 is derived assuming r is considerably smaller than p. In the case of a full-rank tensor with $r = p$, there is no benefit of considering a low-rank model and one could unfold the tensor into a long vector and employ an existing bandit algorithm such as LinUCB, which has been shown to be optimal in linear bandits.

Remark 1. *It is worth to compare the regret bound in Eq. (13) with other strategies. As summaized in Table 1, when $d = 3$ and $r = \mathcal{O}(1)$, `vectorized UCB` suffers $\widetilde{\mathcal{O}}(p^3 + p^{3/2}n^{1/2})$. If we unfold the tensor into a matrix and implement ESTR (Jun et al., 2019), it suffers $\widetilde{\mathcal{O}}(p^2 + p^3 n^{1/2})$. Both of these competitive methods obtain significantly sub-optimal regret bounds. By utilizing the low-rank tensor information, our bound greatly improves the dependency on the dimension p. Moreover, our advantage is even larger when the tensor order d is larger.*

Remark 2. *One may wonder whether we can extend the matrix bandit ESTR (Jun et al.,*

2019) to the tensor case. In this case, standard LinUCB (Abbasi-Yadkori et al., 2011) algorithm could be queried to handle the reshaped linear bandits as did in the matrix bandits (Jun et al., 2019). However, it is known that the algorithm of LinUCB is suboptimal for linear bandits with finitely many arms and the sub-optimality will be amplified as the order of tensor grows. Hence, using LinUCB in the reduction phase results in $\mathcal{O}(p^2 n^{1/2})$ for the leading term that is even worse than `vectorized UCB`.

One of the key challenges in our theoretical analysis is to quantify the cumulative regret in the elimination phase. Existing techniques are not applicable as we utilize a different eliminator with a modified regularization strategy. Furthermore, to bound the cumulative regret in the elimination phase, we need to bound the norm $\|\boldsymbol{\beta}_{(q+1):p^d}\|_2$ which is the last $p^d - q$ entries of $vec(\mathcal{Y})$. Recall that the reward tensor $\mathcal{Y}$ is a rotation of true reward tensor $\mathcal{X}$. We need to derive the upper bound of the norm of rotated reward vector by exploiting the knowledge of estimation error of $\mathcal{X}$. We use the elliptical potential lemma to bound the cumulative regret in elimination phase. Furthermore, all parameters such as the penalization parameter λ_2 and exploration phase length n_1 are carefully selected to obtain the best bound.

3.3 Tensor Epoch-greedy and Regret Analysis

Next, we propose an epoch-greedy type algorithm for low-rank tensor bandits, and compare its performance with `tensor elimination`. The epoch-greedy algorithm (Langford and Zhang, 2007) proceeds in epochs, with each epoch consisting of an exploration phase and an exploitation phase. One advantage of this epoch-greedy algorithm is that we do not need to know the total time horizon n in advance. In the exploration phase, arms are randomly selected and in the exploitation phase, arms that expect the highest reward are pulled. The number of steps in each exploitation phase increases with number of epochs, guided by the fact that, as the number of epochs increases, the estimation accuracy of the true reward improves and more exploitation steps are desirable. The epoch-greedy algorithm is

Algorithm 2 `Tensor epoch-greedy`

Input: initial set $\mathcal{E}_1$, exploration set $\mathcal{E}_2$.
Initialize $\mathcal{D} = \emptyset$.
for $t = 1, 2, \ldots, n$ **do**
  # initialization and exploration phases
  if $t \in \mathcal{E}_1 \cup \mathcal{E}_2$ **then**
    Randomly pull an arm $\mathcal{A}_t$ and receive its associated reward $y_t = \langle \mathcal{X}, \mathcal{A}_t \rangle + \epsilon_t$.
    Let $\mathcal{D} = \mathcal{D} \cup \{(y_t, \mathcal{A}_t)\}$.
  end if
  # exploitation phase
  if $t \notin \mathcal{E}_1 \cup \mathcal{E}_2$ **then**
    Based on $\mathcal{D}$, calculate a low-rank tensor estimate $\widehat{\mathcal{X}}_t$.
    Pull the arm $(i_{1,t}, \ldots, i_{d,t}) = \operatorname{argmax}_{i_1,\ldots,i_d} \langle \widehat{\mathcal{X}}_t, \mathbf{e}_{i_1} \circ \ldots \circ \mathbf{e}_{i_d} \rangle$.
    Receive the associated reward $y_t = \langle \mathcal{X}, \mathbf{e}_{i_{1,t}} \circ \ldots \circ \mathbf{e}_{i_{d,t}} \rangle + \epsilon_t$.
  end if
end for

straightforward to implement, and we find that compared to `tensor elimination`, `tensor epoch-greedy` algorithm has a better dependence on dimension p and a worse dependence on time horizon n.

The detailed steps of `tensor epoch-greedy` are given in Algorithm 2. In the initialization phase, i.e., $t \in \mathcal{E}_1$, arms are randomly pulled to collect samples for tensor completion. Recall the initialization phase has s_1 steps. Let the index set of steps in the exploration phases be

$$\mathcal{E}_2 = \Big\{ s_1 + l + 1 + \sum_{k=0}^{l} s_{2k} \mid l = 0, 1, \ldots \Big\}, \tag{14}$$

where s_{2k} denotes the number of exploitation steps in the kth epoch and it increases with k. In the exploration phase, i.e., $t \in \mathcal{E}_2$, an arm $\mathcal{A}_t$ is pulled (or sampled) randomly. These random samples collected in the exploration phases are important for unbiased estimation, as they do not depend on historical data, and their accumulation can improve estimation accuracy of the reward tensor. Meanwhile, as the exploration phase does not focus on the best arm, each step $t \in \mathcal{E}_2$ is expected to result in a large immediate regret, though it can potentially reduce regret from future exploitation steps. In the exploitation phase, i.e., $t \notin \mathcal{E}_1 \cup \mathcal{E}_2$, we construct a low-rank estimate $\widehat{\mathcal{X}}_t$ of the reward tensor using the random samples collected thus far in

$\mathcal{D}$. Then, the arm $(i_{1,t}, \ldots, i_{d,t})$ with the highest estimated reward in $\widehat{\mathcal{X}}_t$ is selected, i.e.,

$$(i_{1,t}, \ldots, i_{d,t}) = \underset{i_1,\ldots,i_d}{\operatorname{argmax}} \left\langle \widehat{\mathcal{X}}_t, \mathbf{e}_{i_1} \circ \ldots \circ \mathbf{e}_{i_d} \right\rangle.$$

Samples in the exploitation phase will not be used to estimate the reward tensor as they are biased and thus exploitation steps cannot improve estimation accuracy of the reward tensor.

We next derive the regret bound of proposed `tensor epoch-greedy`.

Theorem 2. *Suppose Assumptions 1-3 hold. Let*

$$s_{2k} = \left\lceil C_2 p^{-\frac{d+1}{2}} r^{-\frac{1}{2}} (\log p)^{-\frac{1}{2}} (k + s_1)^{\frac{1}{2}} \right\rceil, \tag{15}$$

for some small constant $C_2 > 0$. When $n \geq C_0 r^{\frac{d-2}{2}} p^{\frac{d}{2}}$, the cumulative regret of Algorithm 2 satisfy, with probability at least $1 - p^{-10}$,

$$R_n \leq C_0 r^{\frac{d-2}{2}} p^{\frac{d}{2}} + 8 n^{\frac{2}{3}} p^{\frac{d+1}{3}} (r \log p)^{\frac{1}{3}}. \tag{16}$$

The regret bound has two terms with the first term characterizing the regret accumulated during the initialization phase and the second term characterizing the regret accumulated over the exploration and exploitation phases. The first term depends on the tensor rank r and dimension p, but not n. It clearly highlights the benefit of exploiting a tensor low-rank structure since unfolding the tensor into a vector or a matrix requires much longer initialization phase. The second term in the regret bound is related to time horizon n and it increases with n at a rate of $n^{\frac{2}{3}}$.

It is worth to compare the leading term of regret bounds for high-order tensor bandits of `tensor elimination` in Eq. (13) and `tensor epoch-greedy` in Eq. (16). As summaized in Table 1, when $d \geq 3$ and $r = \mathcal{O}(1)$, `tensor elimination` suffers $\widetilde{\mathcal{O}}(p^{(d-1)/2}\sqrt{n})$ regret while `tensor epoch-greedy` suffers $\widetilde{\mathcal{O}}(p^{(d+1)/3} n^{2/3})$ regret. Although the latter one has a sub-optimal dependency on the horizon due to the ε-greedy paradigm, it enjoys a better regret than the prior one in the high-dimensional regime ($n \leq p^{d-5}$).

In the theoretical analysis, a key step is to determine the switch time between the two phases, i.e., s_{2k}. We set the length of exploitation phase to be the inverse of tensor estimation error. Intuitively, when the tensor estimation error is large, more exploration can increase the sample size and improve the estimation. When the tensor estimation error is small, there is no need to perform more randomly exploration. Instead, we exploit more to reduce instant regrets. After obtaining the regret in epoch, we need to derive the upper bound of number of epochs. Similar to the optimal tuning procedure in explore-then-commit regret analysis, we tune the parameter to determine the final bound of total number of exploration steps.

4 Contextual Tensor Bandits

In this section, we consider an extension of tensor bandits to contextual tensor bandits where some modes of the reward tensor are contextual information. Take the online advertising data considered in Section 6 as an example. Users use the online platform on some day of the week, and the platform can only decide which advertisement to show to this given user at the given time. In this example, the user mode and the day-of-week mode of the reward tensor are both contextual information and both are not decided by the platform.

The above example can be formalized as contextual tensor bandits. Specifically, at time t, the agent observes a d_0-dimensional context $(i_{1,t}, \cdots, i_{d_0,t}) \in [p_1] \times \cdots \times [p_{d_0}]$ and given the observed context, pulls an $(d-d_0)$-dimensional arm $(i_{d_0+1,t}, \cdots, i_{d,t}) \in [p_{d_0+1}] \times \cdots \times [p_d]$. Let $I_t = (i_{1,t}, \cdots, i_{d,t})$ collect the *context*$\times$*arm* information at time step t. Correspondingly, the agent observes a noisy reward y_t drawn from a probability distribution associated with I_t. The objective is to maximize the cumulative reward over the time horizon. This contextual tensor bandit problem is different from the tensor bandit problems considered in Section 3, as the agent does not have the ability to choose the context. Therefore, the `tensor elimination` algorithm can not be applied to contextual tensor bandits. To tackle this problem, we introduce a heuristic solution to contextual tensor bandits that utilizes Thompson sampling

(Russo et al., 2018) and ensemble sampling (Lu and Van Roy, 2017).

Thompson sampling is a powerful Bayesian algorithm that can be used to address a wide range of online decision problems. The algorithm, in its basic form, first initializes a prior distribution over model parameters, and then samples from its posterior distribution calculated using past observations. Finally, an action is made to maximize the reward given the sampled parameters. The posterior distribution can be derived in closed-form in a few special cases such as the Bernoulli bandit (Russo et al., 2018). With more complex models such as our low-rank tensor bandit problem, the exact calculation of the posterior distribution may become intractable. In this case, we consider an ensemble sampling approach that aims to approximate Thompson sampling while maintaining computational tractability. Specifically, ensemble sampling aims to maintain, incrementally update, and sample from a finite ensemble of models; and this ensemble of models approximates the posterior distribution (Lu and Van Roy, 2017).

Consider the true reward tensor $\mathcal{X} \in \mathbb{R}^{p_1 \times \cdots \times p_d}$ that admits the decomposition in (5), where the first d_0 dimensions of $\mathcal{X}$ correspond to the context and the last $d - d_0$ dimensions correspond to the decision (or arm). At time t and given the arm $\mathcal{A}_t = \mathbf{e}_{i_{1,t}} \circ \cdots \circ \mathbf{e}_{i_{d,t}}$, the reward y_t is assumed to follow $y_t = \langle \mathcal{X}, \mathcal{A}_t \rangle + \epsilon_t$. To ease the calculation of the posterior distribution, in contextual tensor bandits we consider $\epsilon_t \sim N(0, \sigma^2)$. For the prior distribution over model parameters, we assume the rows of $\mathbf{U}_k$ are drawn independently from

$$[\mathbf{U}_k]_{i,\cdot} \sim N(\boldsymbol{\mu}_{k,i}, \sigma_k^2 \mathbf{I}), \quad i \in [p_k], \ k \in [d].$$

Let $\mathcal{H}_{t-1} = \{(\mathcal{A}_s, y_s)\}_{s=1}^{t-1}$ denote the history of action×reward up to time t. Given the prior distribution, the posterior density function can be calculated as

$$f(\mathcal{X}|y_1, \cdots, y_{t-1}) \propto f(y_1 \cdots, y_{t-1}|\mathcal{X}) \Pi_{k,i} f([\mathbf{U}_k]_{i,\cdot}).$$

We maximize $f(\mathcal{X}|y_1, \cdots, y_{t-1})$ to obtain the maximum the posteriori (MAP) estimate as

$$(\widehat{\mathcal{S}}^{(t)}, \widehat{\mathbf{U}}_1^{(t)}, \cdots, \widehat{\mathbf{U}}_d^{(t)}) = \underset{\mathcal{S}, \mathbf{U}_1, \cdots, \mathbf{U}_d}{\operatorname{argmin}} \Big(\frac{1}{\sigma^2} \sum_{s=1}^{t-1} (y_s - \langle \mathcal{X}, \mathcal{A}_s \rangle)^2 + \sum_{k=1}^{d} \frac{1}{\sigma_k^2} \sum_{i=1}^{p_k} \Big\| [\mathbf{U}_k]_{i,\cdot} - \boldsymbol{\mu}_{k,i} \Big\|_2^2 \Big). \quad (17)$$

Algorithm 3 `Tensor ensemble sampling`

Input: rank $r_1, \ldots, r_d$, σ^2, $\{\boldsymbol{\mu}_{ki}\}_{i\in[p_k],k\in[d]}$, $\{\sigma_k^2\}_{k\in[d]}$, number of models M, variance of perturbed noise $\widetilde{\sigma}^2$.

 # initialize M models from prior distributions

Initialize sample $[\widehat{\mathbf{U}}_{km}]_{i,\cdot}^{(0)} \sim N(\boldsymbol{\mu}_{ki}, \sigma_k^2\mathbf{I})$ for $m \in [M]$, $i \in [p_k]$, $k \in [d]$. Normalize each column of matrix $\widehat{\mathbf{U}}_{km}^{(0)}$. Initialize the core tensor $\mathcal{S}_m^{(0)} = \mathbf{1} \circ \cdots \circ \mathbf{1} \in \mathbb{R}^{r_1\times 1\cdots\times_d r_d}$.

for t $= 1, 2\cdots$ **do**

 # exploitation phase

Sample $\widetilde{m} \sim \text{Unif}\{1, \cdots, M\}$

Observe context $\boldsymbol{x}_t = (i_{1t}, \cdots, i_{d_0 t})$

Update $(\widehat{\mathcal{S}}_{\widetilde{m}}^{(t)}, \widehat{\mathbf{U}}_{1\widetilde{m}}^{(t)}, \cdots, \widehat{\mathbf{U}}_{d\widetilde{m}}^{(t)})$ by solving (18).

Choose $\mathbf{a}_t = (i_{d_0+1,t}, \cdots, i_{dt}) = \text{argmax}_{\mathbf{a}=(i_{d_0+1},\cdots,i_d)\in[p_{d_0+1}]\times\cdots\times[p_d]}\, \widehat{R}_{\widetilde{m}}(\boldsymbol{x}_t, \mathbf{a})$, where

$$\widehat{R}_{\widetilde{m}}(\boldsymbol{x}_t, \mathbf{a}) = \widehat{\mathcal{S}}_{\widetilde{m}}^{(t)} \times_1 [\widehat{\mathbf{U}}_{1\widetilde{m}}]_{i_{1t},\cdot}^{(t)} \times \cdots \times_d [\widehat{\mathbf{U}}_{d\widetilde{m}}^{(t)}]_{i_{dt},\cdot}.$$

Receive reward y_t.

 # perturbation phase

Sample perturbation noise $\omega_{tm} \sim N(0, \widetilde{\sigma}^2)$ for $m \in [M]$.

Obtain perturbed rewards $\widetilde{y}_{tm} = y_t + \omega_{tm}$ for $m \in [M]$.

end for

The objective function in (17) can be equivalently written as

$$\frac{1}{\sigma^2}\sum_{s=1}^{t-1}(y_s - \mathcal{S} \times_1 [\mathbf{U}_1]_{i_{1s},\cdot} \times \cdots \times_d [\mathbf{U}_d]_{i_{ds},\cdot})^2 + \sum_{k=1}^{d}\frac{1}{\sigma_k^2}\sum_{i=1}^{p_k}\Big\|[\mathbf{U}_k]_{i,\cdot} - \boldsymbol{\mu}_{ki}\Big\|_2^2,$$

which is a non-convex optimization problem. In our proposed algorithm, we alternatively optimize $\mathbf{U}_k$, $k \in [d]$ and $\mathcal{S}$. Given all $\mathbf{U}_l$ such that $l \neq k$ and $\mathcal{S}$, we estimate the i-th row of $\mathbf{U}_k$ as

$$[\mathbf{U}_k]_{i,\cdot}^{(t)} = \left[\frac{1}{\sigma^2}\sum_{s=1}^{t-1}\mathbf{1}_{(i_{ks}=i)}\mathbf{v}^{(t-1)}(\mathbf{v}^{(t-1)})^\top + \frac{1}{\sigma_1^2}\mathbf{I}\right]^{-1}\left\{\frac{1}{\sigma^2}\sum_{s=1}^{t-1}\mathbf{1}_{(i_{1,s}=i)}y_s\mathbf{v}^{(t-1)} + \frac{1}{\sigma^2}\boldsymbol{\mu}_{k,i}\right\},$$

where $\mathbf{v}^{(t-1)} = \left\{\mathcal{S}^{(t-1)} \times_1 [\mathbf{U}_1]_{i_{1s},\cdot}^{(t-1)} \times \cdots \times_{k-1} [\mathbf{U}_{k-1}]_{i_{k-1,s},\cdot}^{(t-1)} \times_{k+1} [\mathbf{U}_{k+1}]_{i_{k+1,s},\cdot}^{(t-1)} \times \cdots \times_d [\mathbf{U}_d]_{i_{ds},\cdot}^{(t-1)}\right\}$. After updating all rows of $\mathbf{U}_k$ for $k \in [d]$, we then estimate $\mathcal{S}$ by solving (17).

`Tensor ensemble sampling` in Algorithm 3 consists of initialization, exploitation and perturbation phases. In the initialization phase, we sample M models from the prior distributions. The mean $\boldsymbol{\mu}_{ki}$ and variance σ_k^2 in the prior distributions could be determined

from prior knowledge or specified so that the range of models spans plausible outcomes. Then, at each time step t, a model $\widetilde{m}$ is uniformly sampled from the ensemble of M models. After observing a context $\boldsymbol{x}_t = (i_{1t}, \cdots, i_{d_0 t})$, the agent exploits the history data of model $\widetilde{m}$ to estimate the low-rank component of the reward tensor via

$$\min_{\mathcal{S}, \mathbf{U}_1, \cdots, \mathbf{U}_d} \frac{1}{\sigma^2} \sum_{s=1}^{t-1} \left\{ \widetilde{y}_{sm} - \langle \mathcal{X}, \mathcal{A}_s \rangle \right\}^2 + \sum_{k=1}^{d} \frac{1}{\sigma_k^2} \sum_{i=1}^{p_k} \left\| [\mathbf{U}_k]_{i,\cdot} - [\widehat{\mathbf{U}}_{k,m}]_{i,\cdot}^{(0)} \right\|_2^2. \tag{18}$$

Compared to (17), the objective in (18) uses perturbed rewards and perturbed priors, which helps to diversify the models and capture model uncertainty. The goal is for the ensemble to approximate the posterior distribution and the variance among models to diminish as the posterior concentrates. Based on the sampled model $\widetilde{m}$, we pull the optimal arm $\mathbf{a}_t$ given the observed context $\boldsymbol{x}_t$. At the end of each time step, we perturb observed rewards for all M models to diversify the ensemble. Our `tensor ensemble sampling` can be viewed as an extension of ensemble sampling (Lu and Van Roy, 2017) for contextual bandits problem. Note that (18) is a non-convex optimization problem, and there is no assurance of achieving the global optimizer. However, the optimization problem in (18) is bi-convex, meaning that the loss function is convex with respect to one set of parameters while fixing the other sets. This attractive property guarantees that the algorithm will always converge, though possibly to a local optimum (Xu and Yin, 2013). Whether the algorithm can reach the global optimum depends on how close the initialization value is to the true value. The same holds for other similar low-rank tensor estimation problems (Sun et al., 2017; Cai et al., 2021; Xia et al., 2021). In all of our experiments, we have observed that the `tensor ensemble sampling` method performs well with the random initialization utilized in Algorithm 3. It is challenging to analytically quantify how local solutions to (18) affect the `tensor ensemble sampling` method and we leave a comprehensive theoretical investigation to future work. Moreover, our choices of Gaussian prior distribution and Gaussian perturbation noise follow from the existing ensemble sampling literature (Lu and Van Roy, 2017; Osband et al., 2018; Kveton et al., 2020; Dwaracherla et al., 2022; Qin et al., 2022) due to their successful empirical

performance and ease in computation in practice.

Although `tensor ensemble sampling` is motivated by contextual tensor bandit problems, it can also be used to solve tensor bandits without context. In this case, the context dimension $d_0 = 0$ and an arm $\mathcal{A}_t$ consists of all decisions to be made. While `tensor ensemble sampling` performs well empirically, its theoretical investigation is very challenging due to the nature of the ensemble sampling framework (Lu and Van Roy, 2017) and the non-convex optimization in low-rank tensor problems. In Section S.4 of the supplement, we present some preliminary Bayes regret analysis of a general approximate Thompson sampling (TS) algorithm for tensor bandits. Notably, our `tensor ensemble sampling` can be considered as a specific instance of an approximate TS algorithm. Since approximate TS is a Bayesian algorithm, following the literature in this field (Russo and Van Roy, 2016; Qin et al., 2022), we develop a Bayes regret bound, rather than a frequentist regret bound in (4). The Bayes regret is defined as

$$\mathrm{BR}_n = \mathbb{E}\left[\mathrm{R}_n\right] = \mathbb{E}\left[\sum_{t=1}^{n} \langle \mathcal{X}, \mathcal{A}^* \rangle - \sum_{t=1}^{n} \langle \mathcal{X}, \mathcal{A}_t \rangle\right],$$

where the expectation is taken over the reward tensor $\mathcal{X}$ under the prior distribution P_0. Different from the frequentist regret bound in (4), the Bayes regret has an additional expectation over the reward tensor $\mathcal{X}$. Theorem 3 in Section S.4.2 provides a Bayes regret bound for a general approximate TS, $\mathrm{BR}_n \leq \widetilde{\mathcal{O}}\left(\sqrt{p^d\, \mathbb{H}(A^*) n} + \sum_{t=1}^{n} \mathbb{E}\left[\mathbf{d}_{\mathrm{H}}(P_t^* \| \bar{P}_t)\right]\right)$, where $\mathbb{H}(A^*)$ is the entropy of optimal action A^* and the second term measures the distance between its action sampling distribution $\bar{P}_t(\cdot) = \texttt{sample}(\cdot \mid P_0, \mathcal{D}_{t-1})$, and that of the standard Thompson sampling algorithm, $P_t^*(\cdot) = \Pr(A^* \in \cdot | \mathcal{D}_{t-1})$. See Section S.4.2 for more details.

It is important to note that the above Bayesian regret bound is based on a preliminary information-theoretic analysis and we expect its dependence on p, the dimension of the tensor, can be further improved. Specifically, our analysis has not fully exploited the low-rank structure of the reward tensor $\mathcal{X}$. The question of how to incorporate this low-rank structure into the information-theoretic analysis of approximate Thompson sampling remains

an open problem that is particularly challenging. We believe that addressing this problem will necessitate novel insights into Bayesian inference in low-rank tensors and potentially require additional assumptions about the prior distribution P_0. Even in the low-rank matrix case, this issue is not well understood, and we see it as an interesting but very challenging direction for future research. Moreover, to derive an explicit regret bound for the `tensor ensemble sampling` algorithm, we need to further bound the Hellinger distance term $\mathbf{d}_{\mathrm{H}}(P_t^* \| \bar{P}_t)$ for the ensemble sampling algorithm. We believe that this is also a challenging problem that requires better understanding of how perturbed rewards and priors affect the non-convex tensor decomposition formulation (see equation (18)), as well as their connections to Bayesian inferences in low-rank tensors. It is worth mentioning that Qin et al. (2022) has provided a Bayes regret bound for ensemble sampling in a special linear Gaussian bandits; however, their techniques highly depend on the structure of Gaussian linear bandits and cannot be applied to low-rank matrix or tensor bandits. This is another interesting future direction.

5 Simulations

We carry out some preliminary experiments to compare the numerical performance of `tensor epoch-greedy`, `tensor elimination` and `tensor ensemble sampling` with two competitive methods: `vectorized UCB` which unfolds the tensor into a long vector and then implements standard UCB (Auer, 2002) for multi-armed bandits, and `matricized ESTR` (Jun et al., 2019) which unfolds the tensor into a matrix along an arbitrary mode and implements ESTR for low-rank matrix bandits.

We first describe the way to generate an order-three true reward tensor ($d = 3$) according to Tucker decomposition in (5). The tensor dimensions are set to be same, i.e., $p_1 = p_2 = p_3 = p$. The triplet of tensor Tucker rank is fixed to be $r_1 = r_2 = r_3 = r = 2$. Denote $\widetilde{U}_j \in \mathbb{R}^{p_j \times r_j}$ as i.i.d standard Gaussian matrices. Then we apply QR decomposition on $\widetilde{U}_j$, and assign the Q part as the singular vectors U_j. The core tensor $\mathcal{S} \in \mathbb{R}^{r \times r \times r}$ is constructed as a diagonal

tensor with $\mathcal{S}_{iii} = wp^{1.5}$, for $1 \leq i \leq r$. Here, $wp^{1.5}$ indicates the signal strength (Zhang and Xia, 2018). The random noise ϵ_t is generated i.i.d from a standard Gaussian distribution.

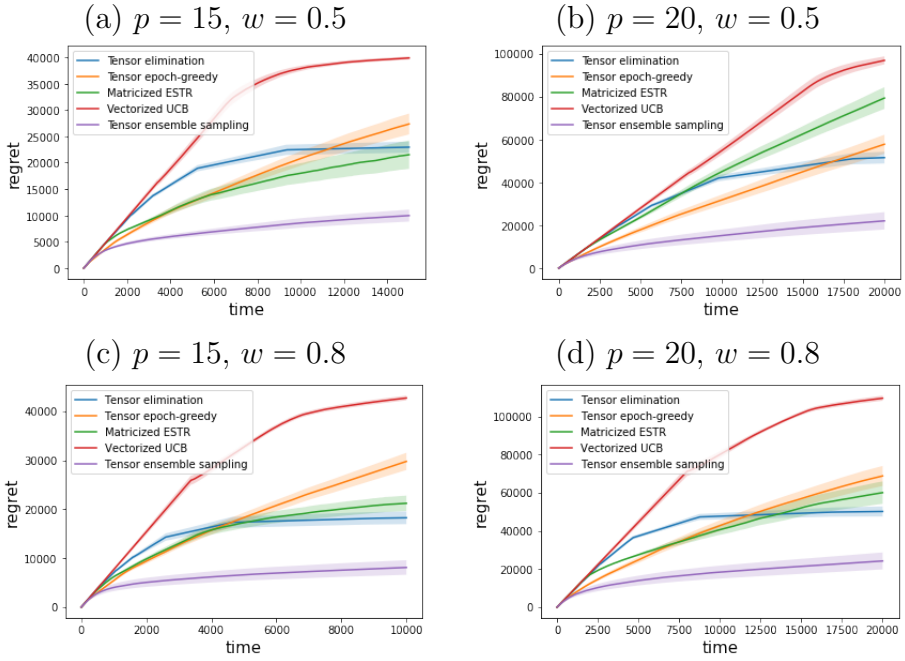


Figure 2: Cumulative regrets with varying dimension p and signal strength w. The shaded areas represent the confidence bands.

All algorithms involve some hyperparameters, such as the length of initial explorations, width of confidence intervals, the number of rounds of pure explorations and etc. In Section S.5 of the appendix, we discuss the choice of hyper-parameters for `tensor elimination`, `tensor epoch-greedy`, `tensor ensemble sampling`, and `matricized ESTR` respectively. In Figure 2, we report the cumulative regrets of all five algorithms for four settings with $w \in \{0.5, 0.8\}$ and $p \in \{15, 20\}$. All the results are based on 30 replications. Figure 2 shows that `tensor ensemble sampling` outperforms all other methods in different settings. `Tensor elimination` does not perform as well as `tensor ensemble sampling` but is better than other methods for a long time horizon. It aligns with our theoretical findings that `tensor elimination` has a better overall regret bound for long time horizon, while `tensor epoch-greedy` is more competitive for small time horizon. When the tensor dimension p increases, the advantage of `tensor epoch-greedy` in early stage is more apparent. This

result agrees with our theoretical finding in that the regret bound of `tensor epoch-greedy` has a lower dependency on dimension compared with other methods.

6 Applications to Online Advertising

Two real data analysis studies are conducted in the field of online advertising to assess the proposed algorithms. The first study focuses on a contextual tensor bandit problem, while the second study examines a non-contextual tensor bandit problem.

Our first data set comes from a major internet company and contains the impressions for advertisements displayed on the company's webpages over four weeks in May to June, 2016. The impression is the number of times the advertisement has been displayed. It is a crucial measure to evaluate the effectiveness of an advertisement campaign, and plays an important role in digital advertising pricing. Studying online advertisement recommendation not only brings opportunities for advertisers to increase their ad exposures but also allows them to efficiently study individual-level behavior.

The impressions of 20 advertisements were recorded for 20 most active users. In order to understand the user behavior over different days of a week, the data were aggregated by days of a week. Thus, the data forms an order-three tensor of dimension $20 \times 7 \times 20$ where each entry in the tensor corresponds to the impression for the given combination of user, day of week and advertisement. The goal of this real application is to recommend advertisement to a selected user on a specific day to achieve maximum reward (impression). The user mode and the day-of-week mode are both contextual information and the agent recommends the corresponding optimal advertisement. `Tensor elimination` and `matricized ESTR` can only handle the setting where the agent chooses arms without contextual information. `Tensor epoch-greedy` is for context-free tensor bandits in our theory but it can also be extended to tensor bandit with contextual information. Therefore, we compare the performance of `tensor epoch-greedy`, `tensor ensemble sampling` and `vectorized UCB` in this contextual tensor

bandits problem. The cumulative regrets of all these algorithms are shown in Figure 3.

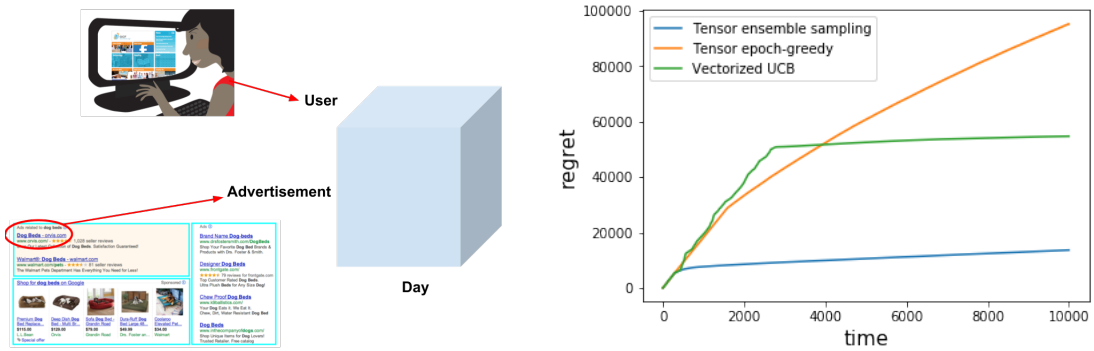


Figure 3: The left plot illustrates the reward tensor formulation in our online advertising data. The right plot shows cumulative regrets of `tensor epoch-greedy`, `tensor ensemble sampling` and `vectorized UCB` in the contextual tensor bandit real data.

From the right plot of Figure 3, we can observe that `tensor ensemble sampling` achieves the lowest regret for a long time horizon. Comparing `tensor epoch-greedy` and `vectorized UCB`, the former is better for a short time horizon. At the last time horizon, `tensor ensemble sampling` is 75% lower than that of `vectorized UCB` and is 85.6% lower than that of `tensor epoch-greedy`. The t-test of difference between the mean of final regret for `tensor epoch-greedy` and `tensor ensemble sampling` indicates that the two means are significantly different (t-statistic is 1191.37 and p-value is 0). The t-test between `tensor ensemble sampling` and `vectorized UCB` also shows significantly improvement is achieved by `tensor ensemble sampling` (t-statistic is 1770.33 and p-value is 0). The success of `tensor ensemble sampling` helps advertisers to better optimize the allocation of ad resources for different users on different days. By tracking users' behavior on ad exposures and conversions over time, advertises can make personalized recommendation based on individual-level data. Besides, our models are maintained and updated based on users' feedback. Such interactive models can be applied to other dynamic and online learning real problems.

In addition to the aforementioned contextual tensor bandit problem, we further consider a real data analysis on non-contextual tensor bandits. The tensor data used in this study

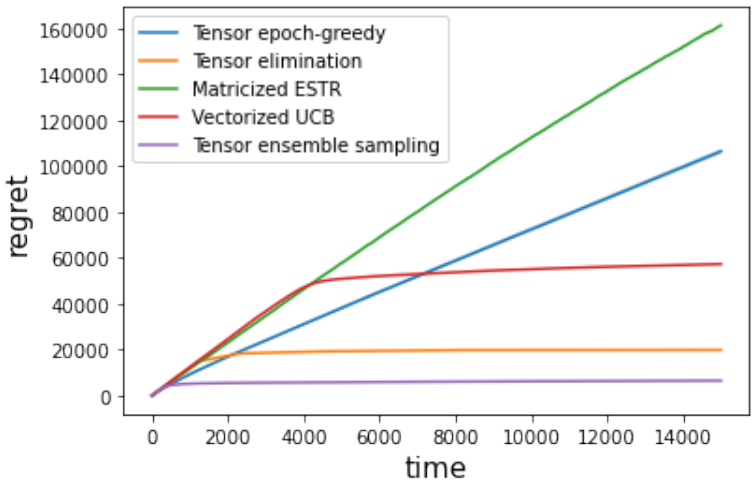


Figure 4: Cumulative regrets in non-contextual tensor bandit real data.

is a third-order tensor that collects information on ad clicks across 20 advertisements, 10 publishers, and 7 days of the week. A *publisher* refers to a specific webpage on the online company's website, such as the main homepage, a page dedicated to financial news, or one dedicated to sports news. Each entry in the tensor corresponds to the number of clicks for a particular combination of advertisement, publisher, and day. The goal of this analysis is to identify the optimal combination of advertisement, publisher, and day that results in the highest reward for behavioral targeting purposes (Choi et al., 2020; Rafieian and Yoganarasimhan, 2021). For example, if we discover that a particular type of customer prefers ad i on publisher j on day k of the week, we can use this information to target this customer segment in future advertising campaigns by displaying ad i on publisher j on day k of the week to maximize the reward. Since all three modes represent actions, this is a non-contextual tensor bandit problem. We conducted a comparison of three proposed algorithms with two baseline models on non-contextual tensor data, and the cumulative regrets of all these algorithms are shown in Figure 4. It is seen that both `tensor ensemble sampling` and `tensor elimination` yielded low regret over a long time horizon, with `tensor ensemble sampling` performing slightly better. `Matricized ESTR` has the worst performance. When the time horizon is short, `tensor epoch greedy` performs better in comparison to `vectorized`

UCB. These findings are consistent with those from our simulation studies.

Acknowledgment

The authors thank the editor Professor Jane-Ling Wang, the associate editor and two anonymous reviewers for their valuable comments and suggestions which led to a much improved paper. Will Wei Sun's research was partially supported by NSF-SES grant (2217440). Any opinions, findings, and conclusions expressed in this material are those of the authors and do not reflect the views of the National Science Foundation. The authors report there are no competing interests to declare.

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. (2011), "Improved algorithms for linear stochastic bandits," *Advances in Neural Information Processing Systems*, 24, 2312–2320.

Ahn, D., Kim, S., and Kang, U. (2021), "Accurate Online Tensor Factorization for Temporal Tensor Streams with Missing Values," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2822–2826.

Allen, G. (2012), "Sparse Higher-Order Principal Components Analysis," *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, 22, 27–36.

Audibert, J.-Y., Munos, R., and Szepesvári, C. (2009), "Exploration–exploitation tradeoff using variance estimates in multi-armed bandits," *Theoretical Computer Science*, 410, 1876–1902.

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

Auer, P., Cesa-Bianchi, N., and Fischer, P. (2002), "Finite-time analysis of the multiarmed bandit problem," *Machine learning*, 47, 235–256.

Bi, X., Adomavicius, G., Li, W., and Qu, A. (2022), "Improving Sales Forecasting Accuracy: A Tensor Factorization Approach with Demand Awareness," *INFORMS Journal on Computing*.

Bi, X., Qu, A., Shen, X., et al. (2018), "Multilayer tensor factorization with applications to recommender systems," *The Annals of Statistics*, 46, 3308–3333.

Bi, X., Tang, X., Yuan, Y., Zhang, Y., and Qu, A. (2021), "Tensors in statistics," *Annual review of statistics and its application*, 8, 345–368.

Bubeck, S. and Cesa-Bianchi, N. (2012), "Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems," *Foundations and Trends in Machine Learning*, 5, 1–122.

Cai, C., Li, G., Poor, H. V., and Chen, Y. (2021), "Nonconvex Low-Rank Tensor Completion from Noisy Data," *Operations Research*.

Choi, H., Mela, C. F., Balseiro, S. R., and Leary, A. (2020), "Online display advertising markets: A literature review and future directions," *Information Systems Research*, 31, 556–575.

Dwaracherla, V., Wen, Z., Osband, I., Lu, X., Asghari, S. M., and Van Roy, B. (2022), "Ensembles for Uncertainty Estimation: Benefits of Prior Functions and Bootstrapping," *arXiv preprint arXiv:2206.03633*.

Friedland, S. and Lim, L.-H. (2017), "Nuclear norm of higher-order tensors," *Mathematics of Computation*, 87, 1255–1281.

Frolov, E. and Oseledets, I. (2017), "Tensor methods and recommender systems," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 7.

Garivier, A. and Cappé, O. (2011), "The KL-UCB algorithm for bounded stochastic bandits and beyond," in *Proceedings of the 24th annual conference on learning theory*, JMLR Workshop and Conference Proceedings, pp. 359–376.

Han, R., Willett, R., and Zhang, A. R. (2022), "An optimal statistical and computational framework for generalized tensor estimation," *The Annals of Statistics*, 50, 1–29.

Idé, T., Murugesan, K., Bouneffouf, D., and Abe, N. (2022), "Targeted Advertising on Social Networks Using Online Variational Tensor Regression," *arXiv preprint arXiv:2208.10627*.

Jain, P. and Oh, S. (2014), "Provable tensor factorization with missing data," *Advances in Neural Information Processing Systems*, 1431–1439.

Jun, K., Willett, R., Nowak, R., and Wright, S. (2019), "Bilinear Bandits with Low-Rank Structure," *36st International Conference on Machine Learning*, 97, 3163–3172.

Katariya, S., Kveton, B., Szepesvári, C., Vernade, C., and Wen, Z. (2017a), "Bernoulli rank-1 bandits for click feedback," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 2001–2007.

Katariya, S., Kveton, B., Szepesvari, C., Vernade, C., and Wen, Z. (2017b), "Stochastic Rank-1 Bandits," in *Artificial Intelligence and Statistics*, pp. 392–401.

Kolda, T. and Bader, B. (2009), "Tensor Decompositions and Applications," *SIAM Review*, 51, 455–500.

Kveton, B., Szepesvari, C., Rao, A., Wen, Z., Abbasi-Yadkori, Y., and Muthukrishnan, S. (2017), "Stochastic low-rank bandits," *arXiv preprint arXiv:1712.04644*.

Kveton, B., Zaheer, M., Szepesvari, C., Li, L., Ghavamzadeh, M., and Boutilier, C. (2020), "Randomized exploration in generalized linear bandits," in *International Conference on Artificial Intelligence and Statistics*, PMLR, pp. 2066–2076.

Lai, T. L. and Robbins, H. (1985), "Asymptotically efficient adaptive allocation rules," *Advances in applied mathematics*, 6, 4–22.

Langford, J. and Zhang, T. (2007), "The epoch-greedy algorithm for contextual multi-armed bandits," in *Proceedings of the 20th International Conference on Neural Information Processing Systems*, Citeseer, pp. 817–824.

Lattimore, T. and Szepesvári, C. (2020), *Bandit algorithms*, Cambridge University Press.

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

Lu, X. and Van Roy, B. (2017), "Ensemble Sampling," *Advances in Neural Information Processing Systems*.

Lu, X., Wen, Z., and Kveton, B. (2018), "Efficient online recommendation via low-rank ensemble sampling," *Proceedings of the 12th ACM Conference on Recommender Systems*, 460–464.

Lu, Y., Meisami, A., and Tewari, A. (2021), "Low-rank generalized linear bandit problems," in *International Conference on Artificial Intelligence and Statistics*, PMLR, pp. 460–468.

Negahban, S. and Wainwright, M. J. (2012), "Restricted strong convexity and weighted matrix completion: Optimal bounds with noise," *The Journal of Machine Learning Research*, 13, 1665–1697.

Osband, I., Aslanides, J., and Cassirer, A. (2018), "Randomized prior functions for deep reinforcement learning," *Advances in Neural Information Processing Systems*, 31.

Qin, C., Wen, Z., Lu, X., and Van Roy, B. (2022), "An Analysis of Ensemble Sampling," in *Advances in Neural Information Processing Systems*, vol. 35.

Rafieian, O. and Yoganarasimhan, H. (2021), "Targeting and privacy in mobile advertising," *Marketing Science*, 40, 193–218.

Richard, E. and Montanari, A. (2014), "A statistical model for tensor PCA," *Advances in Neural Information Processing Systems*, 2897–2905.

Russo, D. and Van Roy, B. (2016), "An information-theoretic analysis of thompson sampling," *The Journal of Machine Learning Research*, 17, 2442–2471.

Russo, D. J., Van Roy, B., Kazerouni, A., Osband, I., and Wen, Z. (2018), "A Tutorial on Thompson Sampling," *Foundations and Trends® in Machine Learning*, 11, 1–96.

Sen, R., Shanmugam, K., Kocaoglu, M., Dimakis, A., and Shakkottai, S. (2017), "Contextual Bandits with Latent Confounders: An NMF Approach," *Artificial Intelligence and Statistics*, 518–527.

Song, Q., Ge, H., Caverlee, J., and Hu, X. (2019), "Tensor completion algorithms in big data analytics," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13, 1–48.

Sun, W. W., Lu, J., Liu, H., and Cheng, G. (2017), "Provable sparse tensor decomposition," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79, 899–916.

Trinh, C., Kaufmann, E., Vernade, C., and Combes, R. (2020), "Solving Bernoulli Rank-One Bandits with Unimodal Thompson Sampling," *31st International Conference on Algorithmic Learning Theory*, 1–28.

Tsybakov, A. B. (2009), *Introduction to Nonparametric Estimation*, Springer series in statistics, Springer.

Valko, M., Munos, R., Kveton, B., and Kocák, T. (2014), "Spectral bandits for smooth graph functions," in *International Conference on Machine Learning*, pp. 46–54.

Xia, D., Yuan, M., and Zhang, C.-H. (2021), "Statistically optimal and computationally efficient low rank tensor completion from noisy entries," *The Annals of Statistics*, 49, 76–99.

Xu, Y. and Yin, W. (2013), "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM Journal on imaging sciences*, 6, 1758–1789.

Yu, R., Cheng, D., and Liu, Y. (2015), "Accelerated online low rank tensor learning for multivariate spatiotemporal streams," in *International conference on machine learning*, PMLR, pp. 238–247.

Yuan, M. and Zhang, C.-H. (2016), "On tensor completion via nuclear norm minimization," *Foundations of Computational Mathematics*, 16, 1031–1068.

Zhang, A. and Xia, D. (2018), "Tensor SVD: Statistical and computational limits," *IEEE Transactions on Information Theory*, 64, 7311–7338.

Zhang, A. et al. (2019), "Cross: Efficient low-rank tensor completion," *The Annals of Statistics*, 47, 936–964.

Supplementary Materials

"Stochastic Low-rank Tensor Bandits for Multi-dimensional Online Decision Making"

In the appendix, we provide detailed proofs of Theorems 1-2 in Section S.1, proof of the main lemma in Section S.2, the equivalent formulation of tensor bandits in Section S.3.1, and the algorithm for low-rank tensor completion in Section S.3.2. Section S.4 contains a general approximate Thompson sampling algorithm and its Bayesian regret bound, and Section S.5 includes the implementation details of all algorithms in the experiments.

S.1 Proofs of Main Theorems

S.1.1 Proof of Theorem 1

From Lemma S3 and the assumption $\|\mathcal{X}\|_\infty \le 1$, we know that with probability at least $1-p^{-10}$,

$$\left\|\widehat{\mathcal{X}}_{n_1}-\mathcal{X}\right\|_F \le C_1\sqrt{\frac{p^{d+1}r\log(p)}{n_1}}.$$

By definitions, $U_i, \widehat{\mathbf{U}}_i$ are left singular vectors of $\mathcal{M}_i(\mathcal{X})$ and $\mathcal{M}_i(\widehat{\mathcal{X}}_{n_1})$, respectively. Here, the matricization operator $\mathcal{M}(\cdot)$ is defined in (1). Then we can verify

$$U_iU_i^\top\mathcal{M}_i(\mathcal{X}) = U_iU_i^\top U_i\Sigma V_i^\top = U_i^\top\Sigma V_i^\top = \mathcal{M}_i(\mathcal{X}).$$

Let $\widehat{\mathbf{U}}_{i\perp}\in\mathbb{R}^{p\times(p-r)}$ be the orthogonal complement of $\widehat{\mathbf{U}}_i$ for $i\in[d]$. For an orthogonal matrix U and an arbitary matrix X, Y, we have $\|UX\|_F \le \|U\|_2\|X\|_F = \|X\|_F$ and $\|XY\|_F \ge$

$\|X\|\sigma_{\min}(Y)$. Suppose σ_i is the r-th singular value of $\mathcal{M}_i(\mathcal{X})$. Using the above fact, we have

$$\begin{aligned}
&\|\mathcal{M}_i(\widehat{\mathcal{X}}_{n_1}) - \mathcal{M}_i(\mathcal{X})\|_F \\
\geq &\|\widehat{\mathbf{U}}_{i\perp}^\top(\mathcal{M}_i(\widehat{\mathcal{X}}_{n_1}) - U_iU_i^\top\mathcal{M}_i(\mathcal{X}))\|_F \\
= &\|\widehat{\mathbf{U}}_{i\perp}^\top U_iU_i^\top\mathcal{M}_i(\mathcal{X})\|_F \\
\geq &\|\widehat{\mathbf{U}}_{i\perp}^\top U_i\|_F\sigma_r(U_i^\top\mathcal{M}_i(\mathcal{X})) = \|\widehat{\mathbf{U}}_{i\perp}^\top U_i\|_F\sigma_i.
\end{aligned}$$

Therefore we have,

$$\|\widehat{\mathbf{U}}_{i\perp}^\top U_i\|_F \leq \frac{\|\mathcal{M}_i(\mathcal{X}) - \mathcal{M}_i(\widehat{\mathcal{X}}_{n_1})\|_F}{\sigma_i} = \frac{\|\mathcal{X} - \widehat{\mathcal{X}}_{n_1}\|_F}{\sigma_i} \leq \frac{C_1}{\sigma_i}\sqrt{\frac{p^{d+1}r\log(p)}{n_1}}, \tag{S1}$$

with probability at least $1 - p^{-\alpha}$. As discussed in Section 3.1, we reformulate original tensor bandits into a stochastic linear bandits with finitely many arms. Recall that $\boldsymbol{\beta} = \mathrm{vec}(\mathcal{Y})$ with

$$\mathcal{Y} = \mathcal{X} \times_1 [\widehat{\mathbf{U}}_1; \widehat{\mathbf{U}}_{1\perp}] \cdots \times_d [\widehat{\mathbf{U}}_d; \widehat{\mathbf{U}}_{d\perp}] \in \mathbb{R}^{p_1\times\cdots\times p_d},$$

and the corresponding action set

$$\mathbb{A} := \Big\{\mathrm{vec}\Big([\widehat{\mathbf{U}}_1; \widehat{\mathbf{U}}_{1\perp}]^\top \boldsymbol{e}_{\mathbf{i}_1} \circ \cdots \circ [\widehat{\mathbf{U}}_d; \widehat{\mathbf{U}}_{d\perp}]^\top \boldsymbol{e}_{\mathbf{i}_d}\Big), \mathbf{i}_1 \in [p_1], \ldots, \mathbf{i}_d \in [p_d]\Big\}.$$

From Eq. (S1), we have

$$\begin{aligned}
\big\|\boldsymbol{\beta}_{(q+1):p^d}\big\|_2 &\leq \prod_{i=1}^d \|\widehat{\mathbf{U}}_{i\perp}^\top U_i\|_F\|\mathcal{S}\|_F \\
&\leq \frac{\|\widehat{X}_{n_1} - X\|_F^d}{\Pi_{i=1}^d\sigma_i}\|\mathcal{S}\|_F \\
&\leq \frac{r^{d/2}}{\Pi_{i=1}^d\sigma_i}\frac{C_1^d r^{d/2} p^{\frac{d^2+d}{2}}\log^{d/2}(p)}{n_1^{d/2}},
\end{aligned} \tag{S2}$$

with probability at least $1 - dp^{-\alpha}$. Thus it is equivalent to consider the following linear bandit problem:

$$y_t = \langle A_t, \boldsymbol{\beta}\rangle + \epsilon_t,$$

where $\|\boldsymbol{\beta}_{(q+1):p^d}\|_2$ satisfies Eq. (S2) and A_t is pulled from action set $\mathbb{A}$. To better utilize the information coming from low-rank tensor completion, we present the following regret bound for the elimination-based algorithm for stochastic linear bandits with finitely-many arms. The detailed proof is deferred to Section S.2.

Lemma S2. *Consider the the elimination-based algorithm in Algorithm 1 with* $\lambda_2 = n/(k\log(1+n/\lambda_1))$ *and* $\lambda_1 > 0$. *With the choice of* $\xi = 2\sqrt{14\log(2/\delta)} + \sqrt{\lambda_1}\|\boldsymbol{\beta}_{1:q}\|_2 + \sqrt{\lambda_2}\|\boldsymbol{\beta}_{(q+1):p^d}\|_2$, *the upper bound of cumulative regret of* n *rounds satisfies*

$$R_n \leq 8\Big(2\sqrt{14\log(2\log(n)p^d/\delta)} + \sqrt{\lambda_1}\|\boldsymbol{\beta}_{1:q}\|_2\Big)\sqrt{2qn\log(1+\frac{n}{\lambda_1})} + 8\sqrt{2}n\|\boldsymbol{\beta}_{(q+1):p^d}\|_2.$$

with probability at least $1-\delta$, *where* $q = p^d - (p-r)^d$.

Overall, we can decompose the pseudo regret Eq. (4) into two parts:

$$\mathrm{R}_n = R_{1n} + R_{2n} + R_{3n},$$

where R_{1n} quantifies the regret during initialization phase, R_{2n} quantifies the regret during exploration phase and R_{3n} quantifies the regret during commit phase (linear bandits reduction). Note that $q \leq C_1 p^{d-1}$ for sufficient large C_1. Denote

$$\delta_{p,r} = \frac{r^d}{\Pi_{i=1}^d \sigma_i} p^{\frac{d^2+d}{2}} \log^{d/2}(p),$$

such that $\|\boldsymbol{\beta}_{(q+1):p^d}\|_2 \leq \delta_{p,r}/n_1^{d/2}$ from Eq. (S2). Applying the result in Lemma S2 to bound R_{3n} and properly choosing $0 < \lambda_1 \leq 1/p^d$, we have the following holds with probability at least $1 - dp^{-10} - 1/n$,

$$\mathrm{R}_n \leq C\Big(\underbrace{r^{d/2}p^{d/2}}_{R_{1n}} + \underbrace{n_1}_{R_{2n}} + \underbrace{\delta_{p,r}n_2/n_1 + \sqrt{\log(\log(n_2)) + \log(n_2p^d)}\sqrt{p^{d-1}n_2\log(n_2p^d)}}_{R_{3n}}\Big),$$

where $n_2 = n - n_1 - Cr^{d/2}p^{d/2}$ and $C > 0$ is an universal constant. Here, R_{3n} is due to the fact that we run elimination-based algorithm for the rest n_2 rounds. For simplicity, we bound all n_2 by n as usually did for the proof of explore-then-commit type algorithm.

We optimize with respect to n_1 such that

$$n_1 = (n\delta_{p,r})^{\frac{2}{d+2}}.$$

It implies the following bound holds with probability at least $1 - dp^{-10} - 1/n$,

$$\begin{aligned}
\mathrm{R}_n &\le C\Big(r^{d/2}p^{d/2} + \Big(\frac{r^d}{\Pi_{i=1}^d \sigma_i} p^{\frac{d^2+d}{2}} \log^{d/2}(p)\Big)^{\frac{2}{d+2}} n^{\frac{2}{d+2}} \\
&\quad + \sqrt{\log(\log(n)) + \log(np^d)}\sqrt{p^{d-1}n\log(np^d)}\Big) \\
&\le C\Big(r^{d/2}p^{d/2} + \Big(\frac{r^d}{\Pi_{i=1}^d \sigma_i} \log^{d/2}(p)\Big)^{\frac{2}{d+2}} p^{\frac{d^2+d}{d+2}} n^{\frac{2}{d+2}} + \sqrt{(d\log(p) + \log(n))^2 p^{d-1} n}\Big).
\end{aligned}$$

This ends the proof. ■

S.1.2 Proof of Theorem 2

The proof uses the trick that couples epoch-greedy algorithm with explore-then-commit algorithm with an optimal tuning.

Step 1. We decompose the pseudo regret defined in (4) as:

$$\begin{aligned}
\mathrm{R}_n &= \sum_{t=1}^{n} \langle \mathcal{A}^* - \mathcal{A}_t, \mathcal{X} \rangle \\
&= \sum_{t=1}^{s_1} \langle \mathcal{A}^* - \mathcal{A}_t, \mathcal{X} \rangle + \sum_{t=s_1+1}^{n} \langle \mathcal{A}^* - \mathcal{A}_t, \mathcal{X} \rangle,
\end{aligned}$$

where s_1 is the number of initialization steps. After initialization phase, from the definition of exploration time index set in (14), the algorithm actually proceeds in phases and each phase contains $(1 + \lceil s_{2k} \rceil)$ steps: one step random exploration plus $\lceil s_{2k} \rceil$ steps greedy actions. By algorithm, at phase k, the greedy action $\mathcal{A}_t$ is taken to maximize $\langle \mathcal{A}_t, \widehat{\mathcal{X}}_{k+s_1} \rangle$ where $\widehat{\mathcal{X}}_{k+s_1}$ is the low-rank tensor completion estimator at phase k based on $(k + s_1)$ random samples. Therefore, we have $\langle \mathcal{A}_t - \mathcal{A}^*, \widehat{\mathcal{X}}_{k+s_1} \rangle \ge 0$ and

$$\langle \mathcal{A}^* - \mathcal{A}_t, \mathcal{X} \rangle \le \langle \mathcal{A}^* - \mathcal{A}_t, \mathcal{X} - \widehat{\mathcal{X}}_{k+s_1} \rangle.$$

By Lemma S3 and the choice of s_{2k} in (15), it is sufficient to guarantee

$$\big\| \widehat{\mathcal{X}}_{k+s_1} - \mathcal{X} \big\|_F \le 1/s_{2k},$$

holds with probability at least $1 - p^{-\alpha}$ from Lemma S3 for any $\alpha > 1$. By the Cauchy-Schwarz inequality, we have

$$\langle \mathcal{A}^* - \mathcal{A}_t, \mathcal{X} \rangle \le \big\| \mathcal{A}^* - \mathcal{A}_t \big\|_F \big\| \widehat{\mathcal{X}}_{k+s_1} - \mathcal{X} \big\|_F \le 2/s_{2k},$$

where for the second inequality we use the fact that both tensors $\mathcal{A}^*$ and $\mathcal{A}_t$ have only one entry equal to 1 and others are 0. Denote $n_2 = n - s_1$ and $K^* = \min\{K : \sum_{k=1}^{K}(1 + \lceil s_{2k} \rceil) \geq n_2\}$. Since we assume $\|\mathcal{X}\|_\infty \leq 1$, the maximum gap $\Delta_{\max}$ is bounded by 2. Then we have

$$\begin{aligned} \mathrm{R}_n &\leq s_1 \Delta_{\max} + \sum_{k=1}^{K^*} \Big(1 \cdot \Delta_{\max} + \lceil s_{2k} \rceil 2 / s_{2k} \Big) \\ &\leq (s_1 + K^*)\Delta_{\max} + 2K^* \leq 2s_1 + 4K^*, \end{aligned} \tag{S3}$$

with probability at least $1 - K^* p^{-\alpha}$.

Step 2. We will derive an upper bound for K^*. Let $n_2^* = \operatorname{argmin}_{u \in [0, n_2]} [u + (n_2 - u)/s_{2u}]$. Consider the following two cases.

1. If $n_2^* \geq K^*$, it is obvious that

$$K^* \leq n_2^* + (n_2 - n_2^*)/s_{2n_2^*}.$$

2. If $n_2^* \leq K^* - 1$, it holds that

$$\sum_{k=1}^{K^*-1} s_{2k} \geq \sum_{k=n_2^*}^{K^*-1} s_{2k} \geq (K^* - n_2^*) s_{2n_2^*},$$

where the second inequality is from the fact that s_{2k} is monotone increasing. By the definition of K^*, it holds that

$$n_2 - 1 \geq \sum_{k=1}^{K^*-1} \big(1 + \lceil s_{2k} \rceil\big) \geq \sum_{k=1}^{K^*-1} \big(1 + s_{2k}\big) \geq K^* - 1 + \big(K^* - n_2^*\big) s_{2n_2^*},$$

which implies

$$K^* \leq n_2^* + (n_2 - n_2^*)/s_{2n_2^*}.$$

Overall, K^* is upper bounded by $n_2^* + (n_2 - n_2^*)/s_{2n_2^*}$.

Step 3. From (S3), the cumulative regret can be bounded by

$$\mathrm{R}_n \leq 2s_1 + 4 \min_{u \in [0, n_2]} \Big(u + (n_2 - u)/s_{2u} \Big).$$

The second term above is essentially the regret for explore-then-comment type algorithm with the optimal tuning for the length of exploration. Plugging the definition of s_{2u} in (15)

and letting $u = n/s_{2u}$, we have

$$K^*/2 \le n_2^* \le n^{2/3} p^{\frac{d+1}{3}} (r \log p)^{\frac{1}{3}}.$$

Thus, we choose $\alpha = \log(2n^{2/3} p^{\frac{d+1}{3}} (r \log p)^{\frac{1}{3}} p)$ such that $K^* p^{-\alpha} \le 1/p$. Plugging in $s_1 = C_0 r^{d/2} p^{d/2}$, we have

$$\mathrm{R}_n \le C_0 r^{d/2} p^{d/2} + 8\Big(n^{2/3} p^{\frac{d+1}{3}} (r \log p)^{\frac{1}{3}}\Big),$$

with probability at least $1 - 1/p$. This ends the proof. ■

S.2 Proof of Lemma S2

Before we prove it, we introduce some notations first. For a vector x and matrix V, we define $\|x\|_V = \sqrt{x^\top V x}$ as the weighted ℓ_2-norm and $\det(V)$ as its determinant. Let $K = \lfloor \log_2(n) \rfloor$ and $t_k = 2^{k-1}$. Denote $x^* = \mathrm{argmax}_{a \in \mathbb{A}} \langle a, \boldsymbol{\beta} \rangle$.

We have the following regret decomposition by phases:

$$\begin{aligned}
\mathrm{R}_n &= \sum_{t=1}^{n} \langle x^* - A_t, \boldsymbol{\beta} \rangle = \sum_{k=0}^{K} \sum_{t=t_k}^{t_{k+1}-1} \langle x^* - A_t, \boldsymbol{\beta} \rangle \\
&= \sum_{k=0}^{K} \sum_{t=t_k}^{t_{k+1}-1} \Big(\langle x^* - A_t, \widehat{\boldsymbol{\beta}}_k \rangle - \langle x^* - A_t, \widehat{\boldsymbol{\beta}}_k - \boldsymbol{\beta} \rangle \Big),
\end{aligned}$$

where $\widehat{\boldsymbol{\beta}}_k$ is the ridge estimator only based on the sample collected in the current phase, defined in Eq. (8). According to Lemma 7 in (Valko et al., 2014), for any fixed $x \in \mathbb{R}^p$ and any $\delta > 0$, we have, at phase k,

$$\mathbb{P}\Big(|x^\top (\widehat{\boldsymbol{\beta}}_k - \boldsymbol{\beta})| \le \|x\|_{V_k^{-1}} \xi \Big) \ge 1 - \delta, \tag{S4}$$

where $\xi = 2\sqrt{14 \log(2/\delta)} + \sqrt{\lambda_1} \|\boldsymbol{\beta}_{1:q}\|_2 + \sqrt{\lambda_2} \|\boldsymbol{\beta}_{(q+1):p^d}\|_2$. Applying Eq. (S4) for x^* and A_t, we have with probability at least $1 - K p^d \delta$,

$$\mathrm{R}_n \le \sum_{k=0}^{K} \sum_{t=t_k}^{t_{k+1}-1} \langle x^* - A_t, \widehat{\boldsymbol{\beta}}_k \rangle + \sum_{k=0}^{K} (t_{k+1} - t_k) \Big(\|x^*\|_{V_k^{-1}} + \|A_t\|_{V_k^{-1}} \Big) \xi.$$

By step (7) in Algorithm 1, we have

$$\langle x^* - A_t, \widehat{\boldsymbol{\beta}}_k \rangle \le \Big(\|x^*\|_{V_k^{-1}} + \|A_t\|_{V_k^{-1}} \Big) \xi.$$

According to Lemma 8 in Valko et al. (2014), for all the actions $x \in \mathbb{A}_k$ defined in Eq. (7),

$$\|x\|^2_{V_k^{-1}} \leq \frac{1}{t_k - t_{k-1}} \sum_{t=t_{k-1}+1}^{t_k} \|x_t\|^2_{V_{t-1}^{-1}}.$$

Then using the elliptical potential lemma (Lemma 19.4 in Lattimore and Szepesvári (2020)), with probability at least $1 - Kp^d\delta$, we have

$$\begin{aligned}
\mathrm{R}_n &\leq 2\sum_{k=0}^{K}(t_{k+1} - t_k)\Big(\|x^*\|_{V_k^{-1}} + \|A_t\|_{V_k^{-1}}\Big)\xi \\
&\leq 4\sum_{k=0}^{K}(t_{k+1} - t_k)\sqrt{\frac{1}{t_k - t_{k-1}}\log\Big(\frac{\det(V_k)}{\det(\Lambda)}\Big)}\xi,
\end{aligned}$$

where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_1, \lambda_2, \ldots, \lambda_2)$. According to Lemma 5 in (Valko et al., 2014), we have

$$\log\Big(\frac{\det(V_k)}{\det(\Lambda)}\Big) \leq k\log(1 + \frac{n}{\lambda_1}) + \sum_{i=k+1}^{p^d}\log(1 + \frac{t_i}{\lambda_2}),$$

where $\sum_{i=k+1}^{p^d} t_i \leq T$. With the choice of λ_2,

$$\log\Big(\frac{\det(V_k)}{\det(\Lambda)}\Big) \leq k\log(1 + \frac{n}{\lambda_1}) + \sum_{i=k+1}^{p^d}\frac{t_i}{\lambda_2} \leq 2k\log(1 + \frac{n}{\lambda_1}).$$

We know that $t_{k+1} - t_k = 2^{k-1}$ and $t_k - t_{k-1} = 2^{k-2}$. Then one can have

$$\sum_{k=0}^{K}(t_{k+1} - t_k)\frac{1}{\sqrt{t_k - t_{k-1}}} = \sum_{k=0}^{K} 2^{k/2} \leq \sqrt{n}.$$

Overall, with probability at least $1 - Kp^d\delta$, we have

$$\begin{aligned}
\mathrm{R}_n &\leq 8\sqrt{2kn\log(1 + \frac{n}{\lambda_1})}\Big(2\sqrt{14\log(2/\delta)} + \sqrt{\lambda_1}\|\boldsymbol{\beta}_{1:k}\|_2 + \sqrt{\lambda_2}\|\boldsymbol{\beta}_{(k+1):p^d}\|_2\Big) \\
&= 8(2\sqrt{14\log(2\log(n)p^d/\delta)} + \sqrt{\lambda_1}\|\boldsymbol{\beta}_{1:k}\|_2)\sqrt{2kn\log(1 + \frac{n}{\lambda_1})} + 8\sqrt{2}n\|\boldsymbol{\beta}_{(k+1):p^d}\|_2.
\end{aligned}$$

This ends the proof. ■

S.3 Auxiliary Results

S.3.1 An equivalent formulation of tensor bandits

We write $\widehat{\mathbf{U}}_{1\perp}, \ldots, \widehat{\mathbf{U}}_{d\perp}$ as the orthogonal basis of the complement subspaces of $\widehat{\mathbf{U}}_1, \ldots \widehat{\mathbf{U}}_d$. By definitions, $[\widehat{\mathbf{U}}_j \widehat{\mathbf{U}}_{j\perp}]$ is an orthogonal matrix for all $j \in [d]$ such that

$$[\widehat{\mathbf{U}}_j \widehat{\mathbf{U}}_{j\perp}][\widehat{\mathbf{U}}_j \widehat{\mathbf{U}}_{j\perp}]^\top = [\widehat{\mathbf{U}}_j \widehat{\mathbf{U}}_{j\perp}]^\top[\widehat{\mathbf{U}}_j \widehat{\mathbf{U}}_{j\perp}] = \mathbb{I}_{d\times d}.$$

Denote a rotated true reward tensor as

$$\mathcal{Y} = \mathcal{X} \times_1 [\widehat{\mathbf{U}}_1; \widehat{\mathbf{U}}_{1\perp}] \cdots \times_d [\widehat{\mathbf{U}}_d; \widehat{\mathbf{U}}_{d\perp}] \in \mathbb{R}^{p_1 \times \cdots \times p_d},$$

where $\times_1$ is the marginal multiplication defined in Eq. (2). Denote

$$\mathcal{E}_1 = [\widehat{\mathbf{U}}_1; \widehat{\mathbf{U}}_{1\perp}]^\top \boldsymbol{e}_{i_{1t}} \circ \cdots \circ [\widehat{\mathbf{U}}_d; \widehat{\mathbf{U}}_{d\perp}]^\top \boldsymbol{e}_{i_{dt}}, \mathcal{E}_2 = \mathbf{e}_{i_{1t}} \circ \cdots \circ \mathbf{e}_{i_{dt}}.$$

We want to prove

$$\langle \mathcal{Y}, \mathcal{E}_1 \rangle = \langle \mathcal{X}, \mathcal{E}_2 \rangle.$$

To see this, we use a fact of the Kronecker product (see details in Section 2.6 in (Kolda and Bader, 2009)). Let $\mathcal{Z}_1 \in \mathbb{R}^{I_1 \times \cdots I_N}$ and $A^{(n)} \in \mathbb{R}^{J_n \times I_n}$ for all $n \in [N]$. Then, for any $n \in [N]$, we have

$$\begin{aligned} &\mathcal{Z}_2 = \mathcal{Z}_1 \times_1 A^{(1)} \cdots \times_N A^{(N)} \\ \Leftrightarrow\ &\mathcal{M}_n(\mathcal{Z}_2) = A^{(n)} \mathcal{M}_n(\mathcal{Z}_1) \Big(A^{(N)} \otimes \ldots \otimes A^{(n+1)} \otimes A^{(n-1)} \otimes \cdots \otimes A^{(1)} \Big)^\top, \end{aligned}$$

where $\mathcal{M}_n(\mathcal{Z})$ is the mode-n matricization and $\otimes$ is a Kronecker product. Denote $H = [\widehat{\mathbf{U}}_2 \widehat{\mathbf{U}}_{2\perp}] \otimes \cdots \otimes [\widehat{\mathbf{U}}_d; \widehat{\mathbf{U}}_{d\perp}]$. By a matricization of $\mathcal{Y}, \mathcal{E}$ along the first mode, we have

$$\begin{aligned} \langle \mathcal{Y}, \mathcal{E}_1 \rangle &= \langle \mathcal{M}_1(\mathcal{Y}), \mathcal{M}_1(\mathcal{E}_1) \rangle \\ &= \langle [\widehat{\mathbf{U}}_1; \widehat{\mathbf{U}}_{1\perp}] \mathcal{M}_1(\mathcal{X}) H^\top, [\widehat{\mathbf{U}}_1; \widehat{\mathbf{U}}_{1\perp}] \mathcal{M}_1(\mathcal{E}_2) H^\top \rangle \\ &= \operatorname{trace}\Big(H \mathcal{M}_1(\mathcal{X})^\top [\widehat{\mathbf{U}}_1; \widehat{\mathbf{U}}_{1\perp}]^\top [\widehat{\mathbf{U}}_1; \widehat{\mathbf{U}}_{1\perp}] \mathcal{M}_1(\mathcal{E}_2) H^\top \Big) \\ &= \operatorname{trace}\Big(H \mathcal{M}_1(\mathcal{X})^\top \mathcal{M}_1(\mathcal{E}_2) H^\top \Big) \\ &= \big\langle \mathcal{X} \times_1 \mathbb{I}_{d \times d} \times_2 [\widehat{\mathbf{U}}_2 \widehat{\mathbf{U}}_{2\perp}] \cdots \times_d [\widehat{\mathbf{U}}_d; \widehat{\mathbf{U}}_{d\perp}], \boldsymbol{e}_{\mathbf{i}_{1t}} \circ \cdots \circ [\widehat{\mathbf{U}}_d; \widehat{\mathbf{U}}_{d\perp}]^\top \boldsymbol{e}_{\mathbf{i}_{dt}} \big\rangle. \end{aligned}$$

Recursively using the above arguments along each mode, we reach our conclusion.

S.3.2 Tensor completion algorithm and guarantee

For the sake of completeness, we state the tensor completion algorithm in (Xia et al., 2021). The goal is to estimate the true tensor $\mathcal{X} \in \mathbb{R}^{p_1 \times \cdots \times p_d}$ from

$$y_t = \langle \mathcal{X}, \mathcal{A}_t \rangle + \epsilon_t, t = 1, \ldots, T,$$

where $\mathcal{A}_t = \boldsymbol{e}_{\mathbf{i}_{1t}} \circ \ldots \circ \boldsymbol{e}_{\mathbf{i}_{dt}}$. This is a standard tensor completion with uniformly random missing data. The algorithm consists of two stages: spectral initialization and power iteration.

Spectral initialization. We first construct an unbiased estimator $\mathcal{X}_{\text{ini}}$ for $\mathcal{X}$ as follows:

$$\mathcal{X}_{\text{ini}} = \frac{p_1 \cdots p_d}{T} \sum_{t=1}^{n} y_t \mathcal{A}_t.$$

For each $j \in [d]$, we construct the following U-statistic:

$$\widehat{R}_j = \frac{(p_1 \cdots p_d)^2}{T(T-1)} \sum_{1 \le t \ne t' \le T} y_t y_{t'} \mathcal{M}_j(\mathcal{A}_t) \mathcal{M}_j(\mathcal{A}_t')^\top,$$

where $\mathcal{M}_j$ is the mode-j matricization defined in Eq. (1). Compute the eigenvectors of $\{\widehat{R}_j\}_{j=1}^d$ with eigenvalues greater than δ, and denote them by $\{\widehat{\mathbf{U}}_j^{(0)}\}_{j=1}^d$.

Power iteration. Given $\{\widehat{\mathbf{U}}_j^{(l-1)}\}_{j=1}^d$, $\mathcal{X}_{\text{ini}}$ can be denoised via projections to j-th mode. For $l = 1, 2, \ldots$, we alternatively update $\{\widehat{\mathbf{U}}_j^{(l-1)}\}_{j=1}^d$ as follows,

$$\widehat{\mathbf{U}}_j^{(l)} = \text{ first } r_j \text{ left singular vectors of } \mathcal{M}_j\Big(\mathcal{X}_{\text{ini}} \times_{j'<j} (\widehat{\mathbf{U}}_{j'}^{(l-1)})^\top \times_{j'>j} (\widehat{\mathbf{U}}_{j'}^{(l-1)})^\top\Big).$$

The iteration is stopped when either the increment is no more than the tolerance ε, i.e.,

$$\Big\|\mathcal{X}_{\text{ini}} \times_1 (\widehat{\mathbf{U}}_1^{(l)})^\top \cdots \times_d (\widehat{\mathbf{U}}_d^{(l)})^\top\Big\|_{\text{F}} - \Big\|\mathcal{X}_{\text{ini}} \times_1 (\widehat{\mathbf{U}}_1^{(l-1)})^\top \cdots \times_d (\widehat{\mathbf{U}}_d^{(l-1)})^\top\Big\|_{\text{F}} \le \varepsilon, \tag{S5}$$

or the maximum number of iterations is reached. With the final estimates $\widehat{\mathbf{U}}_1, \ldots, \widehat{\mathbf{U}}_d$, it is natural to estimate $\mathcal{S}$ and $\mathcal{X}$ as $\widehat{\mathcal{S}} = \mathcal{X}_{\text{ini}} \times_1 \widehat{\mathbf{U}}_1^\top \cdots \times_d \widehat{\mathbf{U}}_d^\top, \widehat{\mathcal{X}} = \widehat{\mathcal{S}} \times_1 \widehat{\mathbf{U}}_1 \cdots \times_d \widehat{\mathbf{U}}_d$. ■

Lemma S3. *Suppose Assumptions 1-2 holds. Suppose $\widehat{\mathcal{X}}_T$ is the low-rank tensor estimator constructed from T uniformly random samples by Algorithm 1 in Xia et al. (2021). Then for any $\alpha > 1$, if the number of samples $T \ge C_0 \alpha^3 r^{(d-2)/2} p^{d/2}$ for sufficiently large constant C_0, the following holds with probability at least $1 - p^{-\alpha}$,*

$$\frac{\|\widehat{\mathcal{X}}_T - \mathcal{X}\|_F}{\|\mathcal{X}\|_F} \le C_1 \sqrt{\frac{\alpha r p \log p}{T}}, \tag{S6}$$

where C_1 is an absolute constant.

Lemma S3 is a direct application of Corollary 2 in Xia et al. (2021) with some constant terms ignored.

S.4 Approximate Thompson Sampling and Its Bayesian Regret Bound

This section presents analysis of a broad approximate Thompson sampling (TS) algorithm for tensor bandits. Notably, our `tensor ensemble sampling` in Algorithm 3 can be considered as a specific instance of an approximate TS algorithm. The approximate TS algorithm is introduced in Section S.4.1, followed by its Bayesian regret bound in Section S.4.2. A detailed proof of the regret bound is provided in Section S.4.3.

S.4.1 Approximate Thompson Sampling

The general approximate Thompson sampling algorithm is described in Algorithm 4. Note that we define $\mathcal{D}_{t-1}$ as the "history" (i.e. the action-reward trajectory) at the beginning of time t. Algorithm 4 takes two inputs: the first input is a prior distribution P_0 over the reward tensor $\mathcal{X}$, and the second input is an action sampling oracle `sample`. In particular, the action sampling oracle maps a prior distribution P_0 and a "history" $\mathcal{D}_{t-1}$ to a probability distribution over the actions. At each time step t, Algorithm 4 proceeds as follows: it first samples an action A_t based on `sample`, then it pulls arm A_t and receives reward y_t, and finally it updates the "history" based on the action-reward pair (A_t, y_t).

Algorithm 4 `Approximate TS for tensor bandits`

Input: prior P_0 over the reward tensor $\mathcal{X}$, an action sampling oracle `sample`
Initialize $\mathcal{D}_0$ as the empty sequence
for $t = 1, 2, \cdots$ **do**
  Sample arm $A_t \sim$ `sample`$(\cdot \mid P_0, \mathcal{D}_{t-1})$
  Pull arm A_t and receive reward y_t
  Update $\mathcal{D}_t =$ `append`$(\mathcal{D}_{t-1}, (A_t, y_t))$
end for

Note that many algorithms can be viewed as a special case of Algorithm 4 with a particular choice of `sample`. For instance, let $\Pr(A^* \in \cdot | \mathcal{D}_{t-1})$ denote the posterior distribution over the optimal action A^*, then if we choose `sample`$(\cdot \mid P_0, \mathcal{D}_{t-1}) = \Pr(A^* \in \cdot | \mathcal{D}_{t-1})$, then Algorithm 4 reduces to the standard Thompson sampling algorithm. Hence, Algorithm 4 can be viewed

as a generalization of the standard Thompson sampling algorithm. Moreover, as we will show later, we can bound the performance of Algorithm 4 based on the "distance" between its action sampling distribution $\texttt{sample}(\cdot \mid P_0, \mathcal{D}_{t-1})$, and that of the standard Thompson sampling algorithm, $\Pr(A^* \in \cdot | \mathcal{D}_{t-1})$. That is why it is referred to as the approximate Thompson sampling algorithm. It is also seen that the tensor ensemble sampling algorithm (Algorithm 3) can be viewed as another special case of Algorithm 4. In particular, Algorithm 3 implicitly defines an action sampling function via an ensemble of tensors.

S.4.2 Bayes Regret Bound

We now establish a general regret bound for Algorithm 4. To simplify the exposition, we make the following assumption:

Assumption 4 (Bounded reward)**.** *For all $\mathcal{X}' \in \texttt{support}(P_0)$, and all action $\mathcal{A}$, we assume that $y = \langle \mathcal{X}', \mathcal{A} \rangle + \epsilon \in [0, 1]$ with probability* 1*.*

Note that this bounded reward assumption is non-essential, and it is assumed to simplify the exposition. In particular, it is satisfied by assuming the noises are sub-Gaussian as in Assumption 1 and the boundedness of tensor $\mathcal{X}'$ in Assumption 2.

Following the literature in this field (Russo and Van Roy, 2016; Qin et al., 2022), we develop a Bayes regret bound for Algorithm 4. The Bayes regret is defined as

$$\mathrm{BR}_n = \mathbb{E}\left[R_n\right] = \mathbb{E}\left[\sum_{t=1}^n \langle \mathcal{X}, \mathcal{A}^* \rangle - \sum_{t=1}^n \langle \mathcal{X}, \mathcal{A}_t \rangle\right], \tag{S7}$$

where the expectation is over the reward tensor $\mathcal{X}$ under the prior distribution P_0. Different from the frequentist regret bound in (4), the Bayes regret has an additional expectation over the reward tensor $\mathcal{X}$. To simplify the exposition, we define

$$P_t^*(\cdot) = \Pr(A^* \in \cdot | \mathcal{D}_{t-1}) \quad \text{and} \quad \bar{P}_t(\cdot) = \texttt{sample}(\cdot | P_0, \mathcal{D}_{t-1}). \tag{S8}$$

Recall that the *Hellinger distance* between P_t^* and $\bar{P}_t$ is defined as

$$\mathbf{d}_{\mathrm{H}}(P_t^* \| \bar{P}_t) = \sqrt{\sum_a \left(\sqrt{P_t^*(a)} - \sqrt{\bar{P}_t(a)}\right)^2}. \tag{S9}$$

Note that the Hellinger distance is symmetric. Moreover, Lemma 2.4 in Tsybakov (2009) shows that the Hellinger distance can be bounded by KL divergences in both directions: for any P_t^* and any $\bar{P}_t$, we have $\mathbf{d}_{\mathrm{H}}^2(P_t^* \| \bar{P}_t) \leq \min\left\{\mathbf{d}_{\mathrm{KL}}(P_t^* \| \bar{P}_t), \mathbf{d}_{\mathrm{KL}}(\bar{P}_t \| P_t^*)\right\}$. Then we have the following Bayes regret bound for Algorithm 4:

Theorem 3. *Assume that $p_1 = p_2 = \cdots = p_d = p$, and the bounded reward assumption (Assumption 4) holds, then under Algorithm 4, we have*

$$BR_n \leq \sqrt{p^d \, \mathbb{H}(A^*) n/2} + 2 \sum_{t=1}^{n} \mathbb{E}\left[\mathbf{d}_{\mathrm{H}}(P_t^* \| \bar{P}_t)\right],$$

where $\mathbb{H}(A^*)$ is the entropy of A^* under the prior distribution.

The proof for Theorem 3 is provided in Section S.4.3. Note that under the prior distribution P_0, $\mathcal{X}$ is a random variable. Consequently, A^*, which is the index of a maximum element[1] of $\mathcal{X}$, is also a random variable. Since $\mathbb{H}(A^*) \leq d \log p$, Theorem 3 immediately implies that

$$\mathrm{BR}_n \leq \widetilde{\mathcal{O}}\left(\sqrt{dp^d n} + \sum_{t=1}^{n} \mathbb{E}\left[\mathbf{d}_{\mathrm{H}}(P_t^* \| \bar{P}_t)\right]\right).$$

Finally, note that in the standard Thompson sampling algorithm, we have $P_t^*(\cdot) = \bar{P}_t(\cdot)$, thus, $\mathrm{BR}_n \leq \widetilde{\mathcal{O}}\left(\sqrt{dp^d n}\right)$, which is sublinear in n and matches the regret bound of vectorized UCB. Note that the analysis in Theorem 3 does not exploit the possible low-rank structure of the reward tensor $\mathcal{X}$. Incorporating this low-rank structure into this information-theoretic analysis of approximate Thompson sampling is very challenging, and we believe that it will require novel insights on Bayesian inference in low-rank tensors, and possibly additional assumptions on the prior distribution P_0. To the best of our knowledge, this issue is not well understood even in the low-rank matrix case. This is an interesting but challenging direction for future work.

Theorem 3 is a general Bayes regret bound for approximate Thompson sampling algorithms, based on the quality of the action sampling distribution. To derive an explicit regret bound for

[1]When there are multiple maximum elements in $\mathcal{X}$, we assume that there is a fixed tie-breaking rule to choose A^*.

the tensor ensemble sampling algorithm (Algorithm 3), we need to further bound the Hellinger distance term for the ensemble sampling algorithm. This is also challenging and it requires better understanding of how perturbed rewards and priors affect the tensor decomposition (see equation (18)), as well as their connections to Bayesian inferences in low-rank tensors. This is another interesting direction for future research. It is worth mentioning that Qin et al. (2022) has provided a Bayes regret bound for ensemble sampling in linear bandits; however, their techniques highly depend on the structure of Gaussian linear bandits and cannot be applied to low-rank matrix or tensor bandits.

S.4.3 Proof for Theorem 3

Our proof utilizes the information-theoretic tool in Qin et al. (2022) which provided a Bayes regret bound for ensemble sampling in linear bandits.

We start by defining some notations. For any time t and any action a, define $y_{t,a}$ as the observed reward if the agent takes action a at time t. Then, by definition, we have

$$\mathrm{BR}_n = \sum_{t=1}^{n} \mathbb{E}\left[y_{t,A^*} - y_{t,A_t}\right] = \sum_{t=1}^{n} \mathbb{E}\left[\mathbb{E}_t\left[y_{t,A^*} - y_{t,A_t}\right]\right],$$

where $\mathbb{E}_t[\cdot]$ is a shorthand notation for $\mathbb{E}[\cdot \,|\, \mathcal{D}_{t-1}]$. Note that by definition of the Bayes regret, the expectation is also over the reward tensor $\mathcal{X}$. Following Lemma 1 in Qin et al. (2022), we can decompose BR_n into a "main regret term" and an "approximation error term". Specifically, for any $t = 1, 2, \ldots, n$, we have

$$\mathbb{E}_t\left[y_{t,A^*} - y_{t,A_t}\right] = G_t + J_t,$$

where G_t is the main regret term defined as

$$G_t = \sum_a \sqrt{P_t^*(a)\bar{P}_t(a)}\left(\mathbb{E}_t\left[y_{t,a}|A^* = a\right] - \mathbb{E}_t\left[y_{t,a}\right]\right) \tag{S10}$$

and J_t is the "approximation error term" defined as

$$J_t = \sum_a \left(\sqrt{P_t^*(a)} - \sqrt{\bar{P}_t(a)}\right)\left(\sqrt{P_t^*(a)}\mathbb{E}_t\left[y_{t,a}|A^* = a\right] + \sqrt{\bar{P}_t(a)}\mathbb{E}_t\left[y_{t,a}\right]\right) \tag{S11}$$

The following lemma bounds G_t based on the information gain in A^*.

Lemma S4. *For each time $t = 1, 2, \ldots, n$, with probability 1, we have*

$$G_t \leq \sqrt{p^d \, \mathbb{I}_t \left(A^*; (A_t, y_{t,A_t})\right) / 2},$$

where $\mathbb{I}_t \left(A^*; (A_t, y_{t,A_t})\right) = \mathbb{I} \left(A^*; (A_t, y_{t,A_t}) \,|\, \mathcal{D}_{t-1} = \mathcal{D}_{t-1}\right)$ is the conditional mutual information between A^* and (A_t, y_{t,A_t}) conditioning on the given history $\mathcal{D}_{t-1}$.

Proof. We follow the proof of Lemma 2 in Qin et al. (2022). In particular,

$$\begin{aligned}
\mathbb{I}_t \left(A^*; (A_t, y_{t,A_t})\right) &\overset{(a)}{=} \mathbb{I}_t(A^*; A_t) + \mathbb{I}_t(A^*; y_{t,A_t} | A_t) \\
&\overset{(b)}{=} \mathbb{I}_t(A^*; y_{t,A_t} | A_t) \\
&= \sum_a \bar{P}_t(a) \mathbb{I}_t(A^*; y_{t,a} | A_t = a) \\
&\overset{(c)}{=} \sum_a \bar{P}_t(a) \mathbb{I}_t(A^*; y_{t,a}) \\
&= \sum_{a, a^*} \bar{P}_t(a) P_t^*(a^*) \mathbf{d}_{\mathrm{KL}} \left(\Pr_t(y_{t,a} \in \cdot \,|\, A^* = a^*) \,\|\, \Pr_t(y_{t,a} \in \cdot)\right), \qquad \text{(S12)}
\end{aligned}$$

where (a) follows from the chain rule of mutual information; (b) follows from the fact that A^* and A_t are conditionally independent given $\mathcal{D}_{t-1}$ so that $\mathbb{I}_t(A^*; A_t) = 0$; and (c) follows from the fact that A_t is conditionally independent of A^* and $y_{t,a}$ given $\mathcal{D}_{t-1}$. Since $y_{t,a} \in [0, 1]$, from Pinsker's inequality, we have

$$\mathbb{E}_t[y_{t,a} | A^* = a^*] - \mathbb{E}_t[y_{t,a}] \leq \sqrt{\frac{1}{2} \mathbf{d}_{\mathrm{KL}} \left(\Pr_t(y_{t,a} \in \cdot \,|\, A^* = a^*) \,\|\, \Pr_t(y_{t,a} \in \cdot)\right)}.$$

Consequently we have

$$\mathbb{I}_t \left(A^*; (A_t, y_{t,A_t})\right) \geq 2 \sum_{a, a^*} \bar{P}_t(a) P_t^*(a^*) \left(\mathbb{E}_t[y_{t,a} | A^* = a^*] - \mathbb{E}_t[y_{t,a}]\right)^2.$$

On the other hand, recall that

$$G_t = \sum_a \sqrt{P_t^*(a) \bar{P}_t(a)} \left(\mathbb{E}_t \left[y_{t,a} | A^* = a\right] - \mathbb{E}_t \left[y_{t,a}\right]\right).$$

Without loss of generality, we index the actions as $a = 1, 2, \ldots, p^d$. Following Russo and Van Roy (2016); Qin et al. (2022), we define the $p^d \times p^d$ matrix $\mathbf{M}$ as

$$\mathbf{M}_{a, a^*} = \sqrt{\bar{P}_t(a) P_t^*(a^*)} \left(\mathbb{E}_t \left[y_{t,a} | A^* = a^*\right] - \mathbb{E}_t \left[y_{t,a}\right]\right)$$

for all $a, a^* = 1, 2, \ldots, p^d$, where $\mathbf{M}_{a,a^*}$ is the (a, a^*)-th element in $\mathbf{M}$. Hence, $G_t = \text{trace}(\mathbf{M})$, while $\mathbb{I}_t\left(A^*; (A_t, y_{t,A_t})\right) \geq 2\|\mathbf{M}\|_F^2$. Hence, we have

$$\frac{G_t^2}{\mathbb{I}_t\left(A^*; (A_t, y_{t,A_t})\right)} \leq \frac{\text{trace}(\mathbf{M})^2}{2\|\mathbf{M}\|_F^2} \overset{(a)}{\leq} \frac{\text{rank}(\mathbf{M})}{2} \overset{(b)}{\leq} \frac{p^d}{2},$$

where (a) follows from $\text{trace}(\mathbf{M}) \leq \sqrt{\text{rank}(\mathbf{M})}\|\mathbf{M}\|_F$ (Fact 10 in Russo and Van Roy (2016)), and (b) follows from $\text{rank}(\mathbf{M}) \leq p^d$ since $\mathbf{M}$ is a $p^d \times p^d$ matrix. This concludes the proof. □

Based on Lemma S4, and following Lemma 3 in Qin et al. (2022), we can show that

$$\sum_{t=1}^{n} \mathbb{E}[G_t] \leq \sqrt{p^d \mathbb{H}(A^*) n/2},$$

which is based on Cauchy-Schwarz inequality and the chain rule of the mutual information. Finally, we bound the approximation error term:

Lemma S5. *For all $t = 1, 2, \ldots, n$, we have*

$$\mathbb{E}\left[J_t\right] \leq 2\mathbb{E}\left[\mathbf{d}_{\text{H}}(P_t^* \| \bar{P}_t)\right].$$

Summing over t gives the second term in the regret bound of Theorem 3.

Proof. We follow the proof of Lemma 4 in Qin et al. (2022). Recall that

$$\begin{aligned}
J_t &= \sum_a \left(\sqrt{P_t^*(a)} - \sqrt{\bar{P}_t(a)}\right)\left(\sqrt{P_t^*(a)}\mathbb{E}_t\left[y_{t,a} | A^* = a\right] + \sqrt{\bar{P}_t(a)}\mathbb{E}_t\left[y_{t,a}\right]\right) \\
&\overset{(a)}{\leq} \sqrt{\sum_a \left(\sqrt{P_t^*(a)} - \sqrt{\bar{P}_t(a)}\right)^2}\left[\sqrt{\sum_a P_t^*(a)\mathbb{E}_t^2\left[y_{t,a} | A^* = a\right]} + \sqrt{\sum_a \bar{P}_t(a)\mathbb{E}_t^2\left[y_{t,a}\right]}\right] \\
&\overset{(b)}{\leq} \sqrt{\sum_a \left(\sqrt{P_t^*(a)} - \sqrt{\bar{P}_t(a)}\right)^2}\left[\sqrt{\sum_a P_t^*(a)} + \sqrt{\sum_a \bar{P}_t(a)}\right] \\
&= 2\mathbf{d}_{\text{H}}(P_t^* \| \bar{P}_t).
\end{aligned} \tag{S13}$$

Note that inequality (a) follows from the Cauchy–Schwarz inequality, and inequality (b) follows from $y_{t,a} \in [0, 1]$. Taking the expectation, we have $\mathbb{E}\left[J_t\right] \leq 2\mathbb{E}\left[\mathbf{d}_{\text{H}}(P_t^* \| \bar{P}_t)\right]$. Since $\sum_{t=1}^{n} \mathbb{E}[G_t] \leq \sqrt{p^d \mathbb{H}(A^*) n/2}$, and $\mathbb{E}\left[J_t\right] \leq 2\mathbb{E}\left[\mathbf{d}_{\text{H}}(P_t^* \| \bar{P}_t)\right]$, Theorem 3 is proved. □

S.5 Implementation Details in Simulations

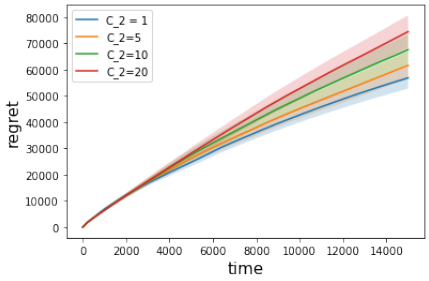
Before discussing the choices of hyperparameters in the experiments, we would like to mention that parameter tuning in bandit problems is uniquely challenging, as decisions are made in real time and are based on rewards observed from the past. In a bandit environment, once a parameter is used on partial datasets and a decision is made based on it, the regret resulting from that decision is irreversible. Hence, it is not feasible to select hyperparameters using traditional offline methods such as cross validation.

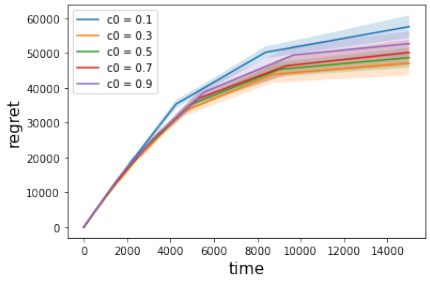
Next we discuss the choice of hyper-parameters for our `tensor elimination`, `tensor epoch-greedy`, `tensor ensemble sampling` in the experiments, and also conducted sensitivity tests on the choice of these parameters. Finally, we discuss how to select hyper-parameters in the competitive methods.

- The algorithm `tensor epoch-greedy` has two hyperparameters, a positive constant C_0 that determines the length of the initialization phase $s_1 = C_0 r^{(d-2)/2} p^{(d-2)}$ and a positive constant C_2 that determines the length of the exploitation phase $s_{2k} = \left\lceil C_2 p^{-\frac{d+1}{2}} r^{-\frac{1}{2}} (\log p)^{-\frac{1}{2}} (k+s_1)^{\frac{1}{2}} \right\rceil$, both are derived in Theorem 2. In our theoretical analysis, the specific choices for C_0 and C_2 do not affect the order of the derived regret bound. We let $C_0 = 1$ and found that it gave enough number of steps in the random initialization phase. For C_2, we conducted a sensitivity analysis to evaluate the performance of `tensor epoch-greedy` with regard to varying values of C_2. As shown in Figure 5(a), the regret of `tensor epoch-greedy` is not sensitive to different values of C_2. Hence, we have chosen to fix $C_2 = 1$ in all numerical experiments.

- The algorithm `tensor elimination` has one hyperparameter c_0 used to determine the number of the exploration steps $c_0 n_1$, where n_1 follows the theoretical value defined in Theorem 1 and $c_0 > 0$ is a small constant. For c_0, we carried out a sensitivity analysis to evaluate the performance of `tensor elimination` with regard to varying values of c_0. From Figure 5(b), there is no significant difference between cumulative regrets

(a) `tensor epoch-greedy` parameter C_2 (b) `tensor elimination` parameter c_0





(c) `tensor ensemble` parameter $\widetilde{\sigma}^2$

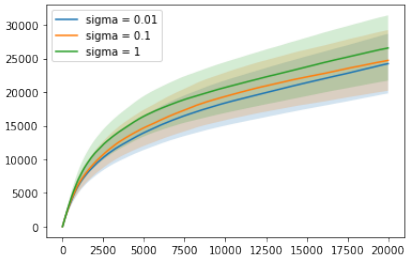


Figure 5: Top left: Cumulative regrets of different constant multiplier C_2 in `tensor epoch-greedy`. Top right: Cumulative regrets of different exploration length constant multiplier c_0 in `tensor epoch-elimination`. Bottom left: Cumulative regrets of different variance of perturbation noise $\widetilde{\sigma}^2$ in `tensor ensemble sampling`. The shaded areas represent the confidence bands. The simulation setting is same as that in Section 5 with dimension $p_1 = p_2 = p_3 = 20$ and $w = 0.8$.

under different values of c_0. Hence, we have chosen to fix $c_0 = 0.5$ in all numerical experiments.

- The algorithm `tensor ensemble sampling` has two hyperparameters including the ensemble size M and the variance of perturbation noise $\widetilde{\sigma}^2$. We set M as a relative large number $M = 100$ to better approximate posterior distribution and found that it gave a good performance. For $\widetilde{\sigma}^2$, we performed a sensitivity analysis to evaluate the performance of `tensor ensemble sampling` with regard to varying values of $\widetilde{\sigma}^2$. As shown in Figure 5(c), the regret of `tensor ensemble sampling` is not sensitive to different values of $\widetilde{\sigma}^2$. We have chosen to fix $\widetilde{\sigma}^2 = 0.1$ in all numerical experiments.

Similar to `tensor elimination`, the competitive method `matricized ESTR` also has a parameter c_0 in the initial exploration length. In our experiments, we selected the parameters

$c_0 \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ that resulted in the lowest cumulative regret for `matricized ESTR`, making the comparison favorable to `matricized ESTR`. Besides, we set the ridge regularization parameter $\lambda_1 = 0.1$ for both `tensor elimination` and `matricized ESTR`.

Finally, determining the appropriate rank is still an unresolved issue even in traditional low-rank tensor models, and existing theoretical studies usually assume prior knowledge of the true rank (Sun et al., 2017; Zhang and Xia, 2018; Zhang et al., 2019; Xia et al., 2021; Cai et al., 2021; Han et al., 2022). In this paper, we adopt this convention and assume prior knowledge of the true ranks for all experiments. However, in practice, one can employ some ad-hoc methods to determine the ranks using uniformly collected samples in the initialization and exploration stages.