Optimality of Zeroth Order Gradient Ascent for Nonlinear Bandit Optimization

Anonymous Author(s)
Affiliation
Address
email

Abstract

Much work has considered multi-armed bandit problems under linear (or kernelized) assumptions on the underlying reward function. This work studies PAC bounds for bandit problems where the unknown underlying reward function is structured polynomials, including a two-layer neural network with polynomial activations and a low-rank generalized linear bandit problem. For the low-rank generalized linear bandit problem, we provide a minimax optimal algorithm in the dimension, improving the results in [34, 27]. Our algorithms are based on a unified zeroth-order optimization paradigm that applies in general, and attain optimal rates in several structured polynomial settings (in the dimension). We further demonstrate the applicability of our algorithms in the RL with generative model setting, resulting in improved sample complexity over prior approaches. Finally, we show that the standard optimistic algorithms (e.g., UCB) are sub-optimal by dimension factors. In the neural net setting (with polynomial activation functions) with noiseless reward, we provide a bandit algorithm that has sample complexity equal to the intrinsic algebraic dimension. Again, we show that optimistic approaches have worse sample complexity, polynomial in the extrinsic dimension (which could be exponentially worse in the polynomial degree).

1 Introduction

Bandits [30] are a class of online decision making problems where an agent interacts with the environment and aims to learn the optimal reward it can get. It is also a simplified version of Reinforcement Learning (RL) problems. In many real world applications, bandit and RL problems are characterized with big or infinite (or continuous) action space. To encode the reward information associated with the action, function approximation for the reward function are proposed, with the simplest settings being the linear bandits [32]. Stochastic linear bandits assume the mean reward to be the inner product between the unknown model parameter and the feature vector associated with the action. This setting has received extensive studies, and people have derived algorithms with sample complexity matching the lower bound [30, 16, 2, 6]. Results show that there is no benefit of using adaptive algorithms in terms of PAC risk, as some non-adaptive algorithms can also achieve an optimal PAC risk [35].

However, linear bandits suffer from the limited power of representing the reward function in the complex reality. To take a step further beyond the linear setting, it becomes more challenging to understand the optimal rates. Unlike linear bandits, more sophisticated adaptive algorithms become necessary. For instance, a natural first step is to look at quadratic [28] and higher-order polynomials [22]. In the context of phase retrieval, which is a special case for the quadratic bandit, people have derived algorithms that achieve minimax risks among non-adaptive algorithms [5, 31, 8].

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

However, it doesn't match our lower bound among all adaptive algorithms on the dependence of problem dimension.

In bandit domain, existing analysis on the nonlinear setting includes eluder dimension [43], subspace elimination [34, 27], etc. Their results also suffer from some gap with the best known lower bound under their corresponding scenarios. (See Table 1 and Section 3 for a detailed illustration of these results.) Therefore in this paper, we are interested in investigating the following question:

What is the optimal sample complexity for non-linear bandit problems beyond eluder dimension, including quadratic and higher-order polynomials? Can we obtain the optimal rates?

Contribution: In this paper, we answer the questions and close the gap (in problem dimension) for various non-linear bandit problems. Rather than looking at the expected regret, we are concerned with the PAC bounds on the sample complexity to identify an ϵ-optimal arm [35, 15, 7].

First, we design some stochastic zeroth-order gradient ascent algorithms with phase elimination and attain minimax sample complexity for a class of structured polynomials. We consider bilinear and low-rank linear bandits and symmetric and asymmetric higher-order homogeneous polynomial bandits with action dimension d. It is challenging to recover the optimal arm of non-concave reward function even with first-order algorithms, not to mention zeroth-order gradient ascent. However, we manage to connect the optimization strategy with some linear algebraic techniques, namely noisy subspace iteration and robust tensor power method. Our algorithms are also computationally efficient, bringing more practical advantages. For low-rank settings, our algorithms match the lower bound we derived and thus settle the correct dependence on the problem dimension d.

For the noiseless polynomial bandit problems, we prove that solving polynomial equations achieves sample complexity equal to the intrinsic algebraic dimension of the underlying polynomial class, which is often linear in d for interesting cases. Furthermore, we provide a lower bound showing that all UCB algorithms have a worse sample complexity of $\Omega(d^p)$, where p is the degree of the polynomials.

1.1 Related Work

Linear Bandits. Linear bandit problems and their variants are studied in [12, 30, 1, 6, 2, 12, 38, 21, 20]. The matching upper bound and minimax lower bound achieves $O(d^2/\epsilon^2)$ sample complexity. Linear bandit with sparse arm is closely related to compressed sensing problems [10]. However in this setting, any adaptive strategy has no advantage than non-adaptive algorithms, as proven in [5].

Generalized Linear Bandits. [48][46] study bandits with function approximation as the Neural Tangent Kernels (NTKs) in a contextual setting. NTKs have limited expressivity and can be viewed as approximations to the neural networks in certain regimes [24]. They adopt the linear UCB methods in the kernelized space and exhibit a rate polynomial to the equivalent dimension or the RKHS norm of the representations. However, NTK methods requie d^p samples to express a degree p polynomial [17].

Bandit with Quadratic Rewards. An natural starting point to analyze quadratic bandit problem is to treat the reward function as generalized linear bandit [28, 16]. These algorithms suffer a $\widetilde{O}(d^4/\epsilon^2)$ sample complexity to achieve ϵ optimal reward. An improvement in the bilinear setting [27] and the low-rank linear setting [34] can be made in the low rank (k) case to achieve $\widetilde{O}(d^3k/\epsilon^2)$ complexity. Some other literature consider related but different settings that are not comparable to our results [26, 18, 29]. We note that the sample complexity on obtaining the optimal arm derived by any existing work is at least at least $\Omega(d^3)$. This includes the subspace exploration and refinement algorithms from [27, 34], existing phase retrieval techniques [9, 8], or from eluder dimension [43, 25].

For higher order polynomials existing work mainly focus on non-adaptive tensor regression problems [49, 19, 33]. Recently [22] considers online problem with a low-rank tensor associated with some axis-aligned set of arms. Our setting is much more general.

In the noiseless setting, there is some investigation in phase retrieval borrowing the tools from algebraic geometry (see e.g. [44]). In this paper, we will study the bandit problem with more general reward functions: neural nets with polynomial activation (structured polynomials).

Matrix/Tensor Power Method. As our analysis stems from noisy power method. We review the relevant literature. Robust power method and subspace iteration and tensor decomposition that tolerate

noise first appear in [23, 4]. Follow-up work attained the optimal rate for both gap-dependence and gap-free settings for matrix decomposition [37, 3]. An improvement on the problem dimension for tensor power method is established in [45]. Unlike prior work, [40] consider non-orthogonal tensor decomposable problems.

2 Preliminaries

2.1 Setup: Structured bandits

We study the problem of structured bandit where the reward function is unknown but chosen from a structured function class. A player play the bandit for T rounds, and in each round $t \in [T]$, the player chooses one action $\boldsymbol{a}_t$ from the feasible action set $\mathcal{A}$, and receives the reward y_t afterwards. We assume that the action set $\mathcal{A} \subset \mathbb{R}^{D_1}$ can be embedded in some Euclidean space with dimension D_1.

We consider both the stochastic case where $r_t = f_{\boldsymbol{\theta}}(\boldsymbol{a}_t) + \eta_t$ where η_t is the random noise, and the noiseless case $r_t = f_{\boldsymbol{\theta}}(\boldsymbol{a}_t)$. Specifically the function $f_{\boldsymbol{\theta}}$ is unknown to the player, but lies in a known function class $\mathcal{F}$ parameterizable by $\boldsymbol{\theta} \in \mathbb{R}^{D_2}$. $f_{\boldsymbol{\theta}}$ can be chosen adversarially in advance. We use the notation $\mathrm{vec}(\boldsymbol{M})$ to denote the vectorization of a matrix or a tensor $\boldsymbol{M}$, and $\boldsymbol{v}^{\otimes p}$ to denote the p-order tensor product of a vector $\boldsymbol{v}$. For vectors we use $\|\cdot\|_2$ or $\|\cdot\|$ to denote its ℓ_2 norm. For matrices $\|\cdot\|_2$ or $\|\cdot\|$ stands for its spectral norm, and $\|\cdot\|_F$ is Frobenius norm.

We now present the paper outline with the settings considered in the paper:

The stochastic low-rank linear reward case considers action sets on bounded matrices $\mathcal{A} = \{\boldsymbol{A} \in \mathbb{R}^{d\times d} : \|\boldsymbol{A}\|_{\mathrm{F}} \le 1\}$ in Section 3.1, and

$$\mathcal{F}_{\mathrm{LL}} = \left\{ \begin{array}{l} f_{\boldsymbol{\theta}}(\boldsymbol{A}) = \langle \boldsymbol{M}, \boldsymbol{A}\rangle = \mathrm{vec}(\boldsymbol{M})^\top \mathrm{vec}(\boldsymbol{A}), \\ \boldsymbol{M} \in \mathbb{R}^{d\times d}, \mathrm{rank}(\boldsymbol{M}) = k, \boldsymbol{M} = \boldsymbol{M}^\top, \|\boldsymbol{M}\|_{\mathrm{F}} \le 1, \boldsymbol{\theta} = \mathrm{vec}(\boldsymbol{M}) \in \mathbb{R}^{d^2} \end{array} \right\}.$$

The stochastic bilinear reward case considers action set $\mathcal{A} = \{\boldsymbol{a} \in \mathbb{R}^d : \|\boldsymbol{a}\|_2 \le 1\}$ as shown in Section 3.2, and

$$\mathcal{F}_{\mathrm{B}} = \left\{ \begin{array}{l} f_{\boldsymbol{\theta}}(\boldsymbol{a}) = \boldsymbol{a}^T \boldsymbol{M} \boldsymbol{a}, \boldsymbol{M} = \sum_{j=1}^k \lambda_j \boldsymbol{v}_j \boldsymbol{v}_j^\top, \text{ for orthonormal } \boldsymbol{v}_j \\ \boldsymbol{M} \in \mathbb{R}^{d\times d}, 1 \ge \lambda_1 \ge |\lambda_2| \ge \cdots \ge |\lambda_k|, \boldsymbol{\theta} = \mathrm{vec}(\boldsymbol{M}) \in \mathbb{R}^{d^2} \end{array} \right\}.$$

We illustrate how to apply the established bandit oracles to attain a better sample complexity for RL problems with simulator in Section 3.2.1.

The stochastic homogeneous polynomial reward case is presented in Section 3.3. For the symmetric case in Section 3.3.1, the action sets are $\mathcal{A} = \{\boldsymbol{a} \in \mathbb{R}^d : \|\boldsymbol{a}\|_2 \le 1\}$, and

$$\mathcal{F}_{\mathrm{HPS}} = \left\{ \begin{array}{l} f_{\boldsymbol{\theta}}(\boldsymbol{a}) = \sum_{j=1}^k \lambda_j (\boldsymbol{v}_j^\top \boldsymbol{a})^p \text{ for orthonormal } \boldsymbol{v}_j, \\ 1 \ge r^* = \lambda_1 \ge |\lambda_2| \ge \cdots \ge |\lambda_k|, \boldsymbol{\theta} = \mathrm{vec}(\sum_{j=1}^k \lambda_j \boldsymbol{v}_j^{\otimes p}) \in \mathbb{R}^{d^p} \end{array} \right\};$$

in the asymmetric case in Section 3.3.2, the action sets are
$\mathcal{A} = \{\boldsymbol{a} = \boldsymbol{a}(1) \otimes \boldsymbol{a}(2) \otimes \cdots \otimes \boldsymbol{a}(p) \in \mathbb{R}^{d^p} : \forall q, \|\boldsymbol{a}(q)\|_2 \le 1\}$, and

$$\mathcal{F}_{\mathrm{HPA}} = \left\{ \begin{array}{l} f_{\boldsymbol{\theta}}(\boldsymbol{a}) = \sum_{j=1}^k \lambda_j \prod_{q=1}^p (\boldsymbol{v}_j(q)^\top \boldsymbol{a}(q)) \text{ for orthonormal } \boldsymbol{v}_j(q) \text{ for each } q, \\ 1 \ge r^* = |\lambda_1| \ge |\lambda_2| \ge \cdots \ge |\lambda_k|, \\ \boldsymbol{\theta} = \mathrm{vec}(\sum_{j=1}^k \lambda_j \boldsymbol{v}_j(1) \otimes \boldsymbol{v}_j(2) \otimes \cdots \otimes \boldsymbol{v}_j(p) \in \mathbb{R}^{d^p}) \end{array} \right\}.$$

In the above settings, there is stochastic noise on the observed rewards. We also consider noiseless settings as below:
The noise-less polynomial reward case is presented in Section 3.4. The action sets $\mathcal{A}$ are subsets of $\mathbb{R}^d$, and

$$\mathcal{F}_{\mathrm{P}} = \left\{ \ f_{\boldsymbol{\theta}}(\boldsymbol{a}) = \langle \boldsymbol{\theta}, \widetilde{\boldsymbol{a}}^{\otimes p} \rangle : \boldsymbol{\theta} \in \mathcal{V}, \widetilde{\boldsymbol{a}} = [1, \boldsymbol{a}^\top]^\top, \mathcal{V} \subseteq (\mathbb{R}^{d+1})^{\otimes p} \text{ is an algebraic variety } \right\}.$$

Additionally, $\mathcal{F}$ need to be admissible (Definition 3.15). This class includes two-layer neural networks with polynomial activation (.e. structured polynomials). We study the fundamental limits of all UCB algorithms in Section 3.4.2 as they are $\Omega(d^{p-1})$ worse than our algorithm presented in Section 3.4.1.

			$\mathcal{F}_{\text{HPS}}$	$\mathcal{F}_{\text{HPA}}$	$\mathcal{F}_{\text{B}}$	$\mathcal{F}_{\text{LL}}$
Upper Bound from LinUCB/eluder			$\widetilde{O}(d^{p+1}k\epsilon^{-2})(k>p)$	$\widetilde{O}(d^{p+1}k\epsilon^{-2})$	$\widetilde{O}(d^3k\epsilon^{-2})$	$\widetilde{O}(d^3k\epsilon^{-2})$
Our Results	NPM	Gap-dependent	$\widetilde{O}(\kappa k d^p\Delta^{-2}\epsilon^{-2})$ (even p)	N/A	$\widetilde{O}(\kappa d^2\Delta^{-2}\epsilon^{-2})$	$\widetilde{O}(d^2r^2\lambda_k^{-2}\epsilon^{-2})$
		Gap-independent	$\widetilde{O}(d^pk\epsilon^{-2})$ (odd p)	$\widetilde{O}((Ck)^pd^p\epsilon^{-2})$	$\widetilde{O}(\min\{d^2\epsilon^{-5}, d^2k^5\epsilon^{-2}\})$	N/A
	Lower bounds		$\Omega(d^pp^{-2p}\epsilon^{-2})$	$\Omega(d^p\epsilon^{-2})$	$\Omega(d^2\epsilon^{-2})$	$\Omega(d^2r^2\epsilon^{-2})^*$

Table 1: Baselines and Our Main Results (for stochastic settings). Eigengap $\Delta = \lambda_1 - |\lambda_2|$, $\kappa = \lambda_1/\Delta$. C is a constant that depends on failure probability. The result with $*$ is from [34].

In the above settings, we are concerned with the sample complexity to learn an ϵ-optimal policy. We hope to find the smallest T such that in each subsequent round $t > T$, our algorithm gives an action $\boldsymbol{a}_t \in \mathcal{A}$ that $|f_{\boldsymbol{\theta}}(\boldsymbol{a}_t) - \arg\max_{\boldsymbol{a}\in\mathcal{A}} f_{\boldsymbol{\theta}}(\boldsymbol{a})| \leq \epsilon$ with high probability. We call such $\boldsymbol{a}_t$ an ϵ-optimal action. In all the stochastic settings above, we make the standard assumption on stochasticity that η_t is conditionally zero-mean 1-sub-Gaussian random variable with regards to the randomness before t.

2.2 Warm-up: Adapting Existing Algorithm

In all the above settings, the function class can be viewed as a generalized linear function in kernel spaces. Namely, there is fixed feature maps ψ, ϕ so that $f_{\boldsymbol{\theta}}(\boldsymbol{a}) = \psi(\boldsymbol{\theta})^T\phi(\boldsymbol{a})$. Thus it is straightforward to adapt linear bandit algorithms like the renowned LinUCB [32] to our settings. Furthermore, another baseline is given by the eluder dimension argument [43] which gives explicit upper bound for general function classes. We present the best upper bound sample complexity of learning an ϵ-optimal action (or policy) from adapting these methods as a baseline in Table 1, together with our newly-derived lower bound and upper bound in this paper.

The best-known statistical rates are given based on the following result.

Theorem 2.1 (Adaptation from [39], [30])**.** *With δ appropriately small, given the δ-covering-number N (under $\|\cdot\|_\infty$) and the $O(\epsilon)$-eluder-dimension d_E of the function class $\mathcal{F}$, and the dimension d_F of the feature space (the linear dimension spanned by the image of ϕ), OFUL with LinUCB achieves sample complexity $\frac{d_F \log N}{\epsilon^2}$, and UCB with eluder dimension arguments achieves $\frac{d_E \log N}{\epsilon^2}$, to find an ϵ-optimal action.*

In first row of table 1, we further elaborate the best results obtained from 2.1 in individual settings. More details can be found in the appendix.

3 Main results

We now present our main results. We consider 4 different stochastic settings (as summarized in Table 1) and one noiseless setting with structured polynomials. We next present the details in this section.

With stochastic noise, all our algorithms can be unified as zeroth-order optimization. At each stage with some candidate arm $\boldsymbol{a}$, we play $f_{\boldsymbol{\theta}}((1-\zeta)\boldsymbol{a} + \zeta\boldsymbol{z})\boldsymbol{z}$, with $\boldsymbol{z} \sim \mathcal{N}(0, \sigma^2)$ and some proper step-size ζ. Therefore $\mathbb{E}_{\boldsymbol{z}} f_{\boldsymbol{\theta}}((1-\zeta)\boldsymbol{a}+\zeta\boldsymbol{z})\boldsymbol{z} = \sigma^2\zeta\nabla f_{\boldsymbol{\theta}}((1-\zeta)\boldsymbol{a}) = \zeta(1-\zeta)^p\sigma^2\nabla f_{\boldsymbol{\theta}}(\boldsymbol{a})$ for p-th order polynomials. Therefore with enough samples, we are able to conduct noisy gradient ascent.

In the noiseless setting, our algorithm is to solve for the parameter $\boldsymbol{\theta}$ with randomly sampled arms $\{\boldsymbol{a}_t\}$ and the noiseless reward $\{f_{\boldsymbol{\theta}}(\boldsymbol{a}_t)\}$, and determine the optimal action afterwards.

3.1 Stochastic Low-rank Linear Bandits

Let the arm space is $\mathcal{A} = \{\boldsymbol{A} \in \mathbb{R}^{d\times d} | \|\boldsymbol{A}\|_F \leq 1\}$, matching the dimension of parameter $\boldsymbol{M}$. Reward function is $f(\boldsymbol{A}) = \langle \boldsymbol{A}, \boldsymbol{M}\rangle, r_t = f(\boldsymbol{A}) + \eta_t$. Without loss of generality we assume k, the rank of $\boldsymbol{M}$ satisfies $k \leq d/2$, since when k same order as d the known upper and lower bound are both d^4/ε^2 [34] and there is no room of improvement. For simplicity we assume $\|\boldsymbol{M}\|_F = 1$, therefore the optimal arm is $\boldsymbol{A}^* = \boldsymbol{M}$. We write $\boldsymbol{X}(s)$ to be the s-th column of $\boldsymbol{X}$. As presented in Algorithm 1, the zeroth order gradient ascent is similar to noisy subspace iteration. We start from some random initialization matrix $\boldsymbol{X}_0$, and iteratively estimate $\boldsymbol{M}\boldsymbol{X}_{l-1} \equiv \mathbb{E}[\boldsymbol{Y}_l]$, in order to recover the eigenspace $\boldsymbol{V}$ of $\boldsymbol{M}$ (Let $\boldsymbol{M} = \boldsymbol{V}\Sigma\boldsymbol{V}^\top$). With some proper concentration bound, we can apply the analysis of noisy power method [23] and get:

Algorithm 1 Subspace Iteration Exploration for Low-rank Linear Reward.

Input: Stochastic quadratic reward $f : \mathcal{X} \to \mathbb{R}$, failure probability δ, error ε.
Initialization: Set $k' = 2k$. Initial candidate matrix $\boldsymbol{X}_0 \in \mathbb{R}^{d\times k'}$, $\boldsymbol{X}_0(j) \in \mathbb{R}^d, j = 1, 2, \cdots k'$ is the j-th column of X_0 and are i.i.d sampled on the unit sphere $\mathbb{S}^{d-1}$ uniformly. Sample variance m,# sample per iteration n, total iteration L.
for Iteration l from 1 to L **do**
  for s from 1 to k' **do**
    Noisy subspace iteration:
    Sample $\boldsymbol{z}_i \sim \mathcal{N}(0, 1/mI_d), i = 1, 2, \cdots n_s$.
    Calculate tentative rank-1 arms $\widetilde{\boldsymbol{A}}_i = \boldsymbol{X}_{l-1}(s)\boldsymbol{z}_i^\top$.
    Conduct estimation $\boldsymbol{Y}_l(s) \leftarrow m/n \sum_{i=1}^n (\langle \boldsymbol{M}, \widetilde{\boldsymbol{A}}_i\rangle + \eta_i)\boldsymbol{z}_i$. ($\boldsymbol{Y}_l \in \mathbb{R}^{d\times k'}$)
  Let $\boldsymbol{Y}_l = \boldsymbol{X}_l\boldsymbol{R}_l$ be a QR-factorization of $\boldsymbol{Y}_l$
  Update target arm $\boldsymbol{A}_l \leftarrow \boldsymbol{Y}_l\boldsymbol{X}_l^\top$.

Theorem 3.1. *With Algorithm 1, $\boldsymbol{A}_L$ satisfies $\|\boldsymbol{A}_L - \boldsymbol{A}^*\|_F \le \varepsilon$. Therefore $\mathbb{E}[f(\boldsymbol{A}_l)] \ge 1 - 2\varepsilon$. Altogether to get an ε-optimal reward, one needs a total sample complexity of $T \ge \Theta(d^2 r^2 \lambda_k^{-2}\varepsilon^{-2})$.*

3.2 Stochastic Bilinear Reward

Now consider stochastic reward $f(\boldsymbol{a}) = \boldsymbol{a}^\top \boldsymbol{M}\boldsymbol{a} + \eta$, with arm set $\mathcal{A} = \{\boldsymbol{a} | \|\boldsymbol{a}\|_2 \le 1\}$ and $\boldsymbol{M} = \sum_{i=1}^r \lambda_i \boldsymbol{v}_i\boldsymbol{v}_i^\top$ is symmetric and satisfies $\|\boldsymbol{M}\|_2 = \lambda_1 > |\lambda_2| \ge |\lambda_3| \cdots \ge |\lambda_r|, \eta \sim \mathcal{N}(0, 1)$. Denote by $\boldsymbol{a}^*$ the optimal arm ($\pm\boldsymbol{v}_1$), the leading eigenvector of $\boldsymbol{M}$, $(\boldsymbol{a}^*)^\top \boldsymbol{M}\boldsymbol{a}^* = \lambda_1$. Let $\Delta = \lambda_1 - |\lambda_2| > 0$ be the eigengap. $\kappa := \lambda_1/\Delta$ is the condition number.

Remark 3.2 (Asymmetric matrix)**.** *We assume the leading eigenvalue λ_1 to be positive. Normally the reward function is always positive and thus $\boldsymbol{M}$ is positive definite. We state the general setting since it naturally extends to the asymmetric setting: $f(\boldsymbol{a}_1, \boldsymbol{a}_2) = \boldsymbol{a}_1^\top \widetilde{\boldsymbol{M}}\boldsymbol{a}_2$, where $\widetilde{\boldsymbol{M}} = \boldsymbol{U}\Sigma\boldsymbol{V}^\top$. This setting can be reduced to the case where $\boldsymbol{M} = [0, \widetilde{\boldsymbol{M}}^\top; \widetilde{\boldsymbol{M}}, 0] = [\boldsymbol{V}|\boldsymbol{V}; \boldsymbol{U}|-\boldsymbol{U}]\mathrm{diag}(\boldsymbol{\Sigma}, -\boldsymbol{\Sigma})[\boldsymbol{V}|\boldsymbol{V}; \boldsymbol{U}|-\boldsymbol{U}]^\top$ which is symmetric matrix with some negative eigenvalues. Our analysis on symmetric matrices also applies to the asymmetric setting and will equivalently depend on the gap between the top singular values of $\widetilde{\boldsymbol{M}}$.*

Algorithm. We note that by conducting zeroth order gradient estimate $f(\boldsymbol{a}/2 + \boldsymbol{z}/2)\boldsymbol{z}$ with step-size $1/2$, we get estimate for $\mathbb{E}[f(\boldsymbol{a}/2 + \boldsymbol{z}/2)\boldsymbol{z}] = \sigma^2/4\boldsymbol{M}\boldsymbol{a}$ when $\boldsymbol{z} \sim \mathcal{N}(0, \sigma^2)$. Therefore we are able to conduct noisy power method on recovering the top eigenvector. We present a more general result with Tensor power method in Algorithm 2 and get a gap-dependent risk bound:

Theorem 3.3 (PAC risk bound for noisy power method (NPM))**.** *For a fixed $\delta \in (0, 1), \varepsilon \in (0, 1/2)$, and with large enough universal constant C_0, C_1, C_2, for each stage s in Algorithm 2 for the $p = 2$ case, we have:*

$$\tan\theta(\boldsymbol{a}^*, \boldsymbol{a}_0^{(s+1)}) \le \widetilde{\varepsilon}_s,$$

for every s. Finally we have $\tan\theta(\boldsymbol{a}^*, \boldsymbol{a}_0^S) \le \varepsilon$. with probability $1 - \delta$. The total number of pulled arms is $\widetilde{O}(\frac{\kappa d^2}{\Delta^2\varepsilon^2})$.

When the matrix is ill-conditional, i.e., λ_1 is very close to λ_2, we can still get some gap-free versions with some modification: The first idea comes from our goal on finding higher reward instead of recovering the optimal arm. Therefore when λ_1 and λ_2 are very close (gap being smaller than desired accuracy ε), it is okay to find any direction in the span of $(\boldsymbol{v}_1, \boldsymbol{v}_2)$. More formally, we care about the convergence speed of identifying any arm in the space spanned by any top eigenvectors (whose associated eigenvalues are higher than $\lambda_1 - \varepsilon$). Therefore the gap can be viewed as ε instead of $\lambda_1 - \lambda_2$.

Lemma 3.4 (Gap-free PAC risk bound)**.** *In Algorithm 2, by some small parameter alternation of setting $\alpha = 1 - \varepsilon$, one can get ε-optimal reward with a total of $\widetilde{O}(d^2/\varepsilon^5)$ total samples.*

Another novel gap-free algorithm requires to identify any top eigenspace $\boldsymbol{V}_{1:l}, l \in [k]$: $\boldsymbol{V}_{1:l}$ is the column span of $\{v_1, v_2, \cdots v_l\}$. Notice that in traditional subspace iteration, the convergence rate of

Bilinear	LB	[27]	NPM	Gap-free NPM	Subspace Iteration
PAC sample	$\Omega(d^2\varepsilon^{-2})$	$\widetilde{O}(d^3k\lambda_k^{-2}\varepsilon^{-2})$	$\widetilde{O}(\kappa d^2\Delta^{-2}\varepsilon^{-2})$	$\widetilde{O}(d^2\varepsilon^{-5})$	$\widetilde{O}(\min\{d^2k^5\epsilon^{-2}, \widetilde{\kappa}^2d^2k^2\epsilon^{-2}\})$

Low-rank Linear	LB ([34])	UB ([34])	Subspace Iteration
PAC sample	$\Omega(d^2k^2\varepsilon^{-2})$	$\widetilde{O}(d^3k\varepsilon^{-2})^*$, or $\widetilde{O}(d^3k\lambda_k^{-2}\epsilon^{-2})$	$\widetilde{O}(d^2k^2\lambda_k^{-2}\varepsilon^{-2})$

Table 2: All red expressions are our results. LB,UB,NPM stands for lower bound, upper bound, and noisy power method respectively. Eigengap $\Delta = \lambda_1 - |\lambda_2|$ and condition numbers $\kappa = \lambda_1/\Delta$, $\widetilde{\kappa} = \lambda_1/|\lambda_k|$. The result with $*$ is not computationally tractable. Our upper bounds match the lower bound in terms of dimension and is better than existing algorithms that are computationally efficient.

recovering S_l depends on the eigengap $\Delta_l := |\lambda_l| - |\lambda_{l+1}|$. However, since $\sum_{l=1}^k \Delta_l = \lambda_1$, at least one eigengap is larger or equal to λ_1/k. Suppose $\Delta_{l^*} \geq \lambda_1 1/k$. Therefore we can set $\alpha = 1 - 1/k$ and recover the top l^* subspace up to $\lambda_1\epsilon$ error, which will give an ϵ-optimal reward in the end.

Lemma 3.5 ((Gap-free) Subspace Iteration)**.** *With subspace iteration specified in Algorithm 3, one can get ϵ-optimal reward with a total of $\widetilde{O}(d^2k^5/\epsilon^2)$ samples, or $\widetilde{O}(\widetilde{\kappa}^2d^2k^2/\epsilon^2)$ with some modifications, where $\widetilde{\kappa} = \lambda_1/|\lambda_k|$.*

The second rate is through directly identifying the whole eigenspace for $\boldsymbol{M}$ and therefore depends on its smallest eigenvalue. Compared to the low-rank setting of $\|\boldsymbol{M}\|_F d^2r^2/(\lambda_k^2\epsilon^2)$, the sample complexity for bilinear reward with similar subspace iteration is actually $\|\boldsymbol{M}\|_2 d^2r^2/(\lambda_k^2\epsilon^2)$. This is natural difference since the optimal arm for bilinear is measured by operator norm while we need $\|\boldsymbol{A}_L - \boldsymbol{A}^*\|_F \leq \epsilon$ in previous setting.

Remark 3.6 (Connection to Phase Retrieval and Eluder Dimension)**.** *For rank-1 $\boldsymbol{M} = \boldsymbol{x}\boldsymbol{x}^\top$, the bilinear bandits can be viewed as phase retrieval, where one observes $y_r = (\boldsymbol{a}_r^\top\boldsymbol{x})^2 = \boldsymbol{a}_r^\top\boldsymbol{M}\boldsymbol{a}_r$ plus some noise $\eta_r \sim \mathcal{N}(0, \sigma^2)$. The optimal (among non-adaptive algorithms) sample complexity to recover $\boldsymbol{x}$ is $\sigma^2 d/\epsilon^2$ [9, 8] where they play $\boldsymbol{a}$ from random Gaussian $\mathcal{N}(0, \boldsymbol{I})$. However, in bandit, we need to set the variance of $\boldsymbol{a}$ to at most $1/d$ to ensure $\|\boldsymbol{a}\| \leq 1$. Our problem is equivalent to observing y_r/d where their $\boldsymbol{a}/\sqrt{d} \sim \mathcal{N}(0, 1/d\boldsymbol{I})$ and noise level $\eta_r/d \sim \mathcal{N}(0, 1)$, i.e., $\sigma^2 = d^2$. Therefore one gets d^3/ϵ^2 even for the rank-1 problems. Thus our results also have fundamental improvements for phase retrieval problems by leveraging adaptivity. With eluder dimension the result is also $O(d_E \log(N)\epsilon^{-2}) = O(d^3r\epsilon^{-2})$ as presented in Theorem 2.1.*

3.2.1 Extension to Simulator RL: Q-function is Quadratic and Bellman Complete

In this section we interpret how our results analyzing bandit can apply to reinforcement learning with simulators. Let Bellman operator $\mathcal{T}_h$ applying to the Q-function Q_{h+1} is defined as:

$$\mathcal{T}_h(Q_{h+1})(s,a) = r_h(s,a) + \mathbb{E}_{s'\sim\mathbb{P}(\cdot|s,a)}[\max_{a'} Q_{h+1}(s',a')].$$

Definition 3.7 (Bellman complete)**.** Given MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathbb{P}, r, H)$, function class $\mathcal{F}_h : \mathcal{S}\times\mathcal{A} \mapsto \mathbb{R}, h \in [H]$ is called Bellman complete if for all $h \in [H]$ and $Q_{h+1} \in \mathcal{F}_{h+1}$, $\mathcal{T}_h(Q_{h+1}) \in \mathcal{F}_h$.

Assumption 3.8 (Bellman complete for low-rank quadratic reward)**.** We assume the function class $\mathcal{F}_h = \{f_M : f_M(s,a) = \phi(s,a)^\top\boldsymbol{M}\phi(s,a)$ with M at most rank-k, and $0 < \lambda_1(\boldsymbol{M})/\sigma_{\min}(\boldsymbol{M}) \leq \widetilde{\kappa}.\}$ is a class of quadratic function and is Bellman complete. Here $f_M(\phi(s,a)) = \sum_{j=1}^k \lambda_j(\boldsymbol{v}_j^\top\phi(s,a))^2$ when $\boldsymbol{M} = \sum_{j=1}^k \lambda_j\boldsymbol{v}_j\boldsymbol{v}_j^\top$. Write $Q_h^* = f_{\boldsymbol{M}_h^*} \in \mathcal{F}_h$.

Observation: When querying s_{h-1}, a_{h-1}, we observe $s_h' \sim \mathbb{P}(\cdot|s_{h-1}, a_{h-1})$ and reward $r_{h-1}(s_{h-1}, a_{h-1})$.

Oracle to recover tensor $\widehat{\boldsymbol{M}}$: Given $n \geq \widetilde{\Theta}(d^2k^2\widetilde{\kappa}^2(\boldsymbol{M})/\varepsilon^2)$, if one can play $\geq n$ samples $\boldsymbol{a}_i$ and observe $y_i \sim \boldsymbol{a}_i^\top\boldsymbol{M}\boldsymbol{a}_i + \eta_i, i \in [n]$ with 1-sub-gaussian and mean-zero noise η, we can recover $\widehat{\boldsymbol{M}} = \widehat{\boldsymbol{M}}(\{(\boldsymbol{a}_i, y_i)\})$ such that $\|\widehat{\boldsymbol{M}} - \boldsymbol{M}\|_2 \leq \epsilon$. This oracle is guaranteed with our analysis from the bandit setting.

With the oracle, at horizon H, we can estimate $\widehat{\boldsymbol{M}}_H$ that is ϵ/H close to $\boldsymbol{M}_H^*$ in spectral norm through noisy observations from the reward function with $\widetilde{O}(\kappa^2d^2H^2/\varepsilon^2)$ samples. Next, for each horizon $h = H-1, H-1, \cdots, 1$, sample $s_i' \sim \mathbb{P}(\cdot|s,a)$, we define $\eta_i = \max_{a'} f_{\widehat{\boldsymbol{M}}_{h+1}}(s_i', a') -$

$\mathbb{E}_{s'\sim\mathbb{P}(\cdot|s,a)} \max_{a'} f_{\widehat{\boldsymbol{M}}_{h+1}}(s',a')$. η_i is mean-zero and $O(1)$-sub-gaussian since it is bounded. Denote $\boldsymbol{M}_h$ as the matrix that satisfies $f_{\boldsymbol{M}_h} := \mathcal{T} f_{\widehat{\boldsymbol{M}}_{h+1}}$, which is well-defined due to Bellman completeness. We estimate $\widehat{\boldsymbol{M}}_h$ from the noisy observations $y_i = r_h(s,a) + \max_{a'} f_{\widehat{\boldsymbol{M}}_{h+1}}(s_i', a') = \mathcal{T} f_{\widehat{\boldsymbol{M}}_{h+1}} + \eta_i =: f_{\boldsymbol{M}_h} + \eta_i$. Therefore with the oracle, we can estimate $\widehat{\boldsymbol{M}}_h$ such that $\|\widehat{\boldsymbol{M}}_h - \boldsymbol{M}_h\|_2 \le \epsilon/H$ with $\Theta(\kappa^2 d^2 k^2 H^2/\epsilon^2)$ bandits. More details is deferred to Algorithm 4 and the Appendix. We state the theorem on sample complexity of finding ϵ-optimal policy here:

Theorem 3.9. *Suppose $\mathcal{F}$ is Bellmen complete associated with parameter $\widetilde{\kappa}$. With probability $1-\delta$, Algorithm 4 learns an ϵ-optimal policy π with $\widetilde{\Theta}(d^2k^2\widetilde{\kappa}^2H^3/\epsilon^2)$ samples.*

Existing approaches would require sample complexity $O(\frac{d^3H^2}{\epsilon^2})$, though they operate in the online RL setting [47, 14, 25], which is worse by a factor of the dimension.

3.3 Stochastic High-order Homogeneous Polynomial Reward

Next we move on to homogeneous high-order polynomials.

3.3.1 The symmetric setting

Let reward function be a p-th order stochastic polynomial function $f : S^{d-1} \to \mathbb{R}$, $f(\boldsymbol{a}) = \boldsymbol{T}(\boldsymbol{a}^{\otimes p}), r_t = f(\boldsymbol{a}) + \eta_t$, where $\boldsymbol{T} = \sum_{j=1}^k \lambda_j \boldsymbol{v}_j^{\otimes p}$ is an orthogonally decomposable rank-k tensor. $\{\boldsymbol{v}_1, \cdots \boldsymbol{v}_k\}$ form an orthonormal basis. Optimal reward r^* satisfies $1 \ge r^* = \lambda_1 \ge |\lambda_2| \cdots \ge |\lambda_k|$. Noise $\eta \sim \mathcal{N}(0,1)$.

In this setting, the problem is fundamentally more challenging than quadratic reward functions. On one hand, it has higher noise-to-signal ratio with larger p. One can tell from the rank-1 setting where $\boldsymbol{T} = \lambda_1 \boldsymbol{v}_1^{\otimes p}$. For a randomly generated arm $\boldsymbol{a}$ on the unit ball, $\mathbb{E}[\|\boldsymbol{a}^\top \boldsymbol{v}_1\|^2] = 1/d$. Therefore on average the signal strength is only $(\boldsymbol{a}^\top \boldsymbol{v}_1)^p \sim d^{-p/2}$, much smaller than the noise level 1. Intuitively this demonstrate why higher complexity is needed for high-order polynomials. On the other hand, it is also technically more challenging. Unlike matrix case, the expected zeroth-order gradient is no longer equal to any tensor product. Therefore existing tensor decomposition arguments do not apply. Fortunately, we find out that with a good initialization, zeroth-order optimization also still pushes the iterated arms toward the optimal arm with linear convergence. We now the bandit optimization procedure in Algorithm 2 and present the result in Theorem 3.10:

Theorem 3.10 (Staged Progress)**.** *For each stage s, with probability $1-\delta$, $\mathcal{A}_s$ is not empty. At least one arm $\boldsymbol{a} \in \mathcal{A}_s$ satisfies: $\tan\theta(\boldsymbol{a}, \boldsymbol{a}^*) \le \widetilde{\varepsilon}_s = 2^{-s}$. Meanwhile all arms left in stage s satisfy:*

$$\mathbb{E}[f(\boldsymbol{a})] \ge \lambda_1(1 - (1 + 1/p^d)\widetilde{\varepsilon}_s), \forall \boldsymbol{a} \in \mathcal{A}_s.$$

Remark 3.11 (Choice of step-size)**.** *Here we choose step-size $\zeta = 1/2p$. Note that $\mathbb{E}[\boldsymbol{y}] = \zeta(1-\zeta)^p\sigma^2\nabla_{\boldsymbol{a}} f(\boldsymbol{a})$. The scaling $(1-\zeta)^p \ge 1/\sqrt{e}$ ensures the signal to noise ratio not too small. The choice of weighted arm is a delicate balance between making progress in optimization and controlling the noise to signal ratio.*

Corollary 3.12. *Let $\kappa = \lambda_1/(\lambda_1 - |\lambda_2|)$ be the condition number. For small enough ε, the to get an ε-optimal arm from Algorithm 2, the total number of bandit arms satisfies:*
(1) For $p = 2$, $T \le \widetilde{O}\left(\frac{\kappa d^2}{\Delta^2\varepsilon^2}\right)$; *(2) For even $p \ge 4$, $T \le \widetilde{O}\left(\frac{\kappa k d^p}{\Delta^2\varepsilon^2}\right)$;*
(3) For odd $p \ge 3$, $T \le \widetilde{O}\left(\frac{k d^p}{\varepsilon^2}\right)$, [1] *with high probability.*

3.3.2 The asymmetric setting

Now we consider the asymmetric tensor problem with reward $r : \mathcal{X} \to \mathbb{R}$, the input space $\mathcal{X}$ consists of p vectors in a unit ball: $\vec{\boldsymbol{a}} = (\boldsymbol{a}(1), \boldsymbol{a}(2), \cdots \boldsymbol{a}(p)) \in \mathcal{X}, \|\boldsymbol{a}(s)\| \le 1, \forall s \in [p]$. $f(\vec{\boldsymbol{a}}) = \boldsymbol{T}(\otimes_{s=1}^p \boldsymbol{a}(s)) + \eta$. Tensor $\boldsymbol{T} = \sum_{j=1}^k \lambda_j \boldsymbol{v}_j(1) \otimes \boldsymbol{v}_j(2) \cdots \otimes \boldsymbol{v}_j(p)$. For each $s \in [p]$, $\{\boldsymbol{v}_1(s), \boldsymbol{v}_2(s), \cdots \boldsymbol{v}_k(s)\}$ are orthonormal vectors. We order the eigenvalues such that $\lambda_1 \ge$

[1] For cleaner presentation, in the main paper we use $\widetilde{O}$, $\widetilde{\Theta}$ or $\widetilde{\Omega}$ to hide universal constants, polynomial factors in p and polylog factors in dimension d, error ε, eigengap Δ or failure rate δ.

Algorithm 2 Phased elimination with zeroth order exploration.

Input: Stochastic reward $r : B_1^d \to \mathbb{R}$ of polynomial degree p, failure probability δ, error ε.
Initialization: $L_0 = C_L k \log(1/\delta)$ if $p \geq 3$, $L_0 = C_L \log(1/\delta)$ if $p = 2$; Set $\alpha = 0.5$ for odd p and $\alpha = |\lambda_2|/\lambda_1$ for even p; Total number of stages $S = C_S \lceil \log(1/\varepsilon) \rceil + 1$, $\mathcal{A}_0 = \{\boldsymbol{a}_0^{(1)}, \boldsymbol{a}_0^{(2)}, \cdots \boldsymbol{a}_0^{(L_0)}\}$ where each $\boldsymbol{a}_0^{(l)}$ is uniformly sampled on the unit sphere $\mathbb{S}^{d-1}$. $\widetilde{\varepsilon}_0 = 1$.
for s from 1 to S **do**
  $\widetilde{\varepsilon}_s \leftarrow \widetilde{\varepsilon}_{s-1}/2$, $n_s \leftarrow C_n d^p \log(d/\delta)/(\lambda_1^2(1-\alpha)^2\widetilde{\varepsilon}_s^2)$,$n_s \leftarrow n_s \cdot \log^3(n_s/\delta)$, $m_s \leftarrow C_m d \log(n/\delta)$, $\mathcal{A}_s = \varnothing$.
  for l from 1 to L_{s-1} **do**
    Zeroth order optimization:
    Locate current arm $\widetilde{\boldsymbol{a}} = \boldsymbol{a}_{s-1}^{(l)}$.
    for $\lceil (1/(1-\alpha)) \log(2d) \rceil$ times **do**
      Sample $\boldsymbol{z}_i \sim \mathcal{N}(0, 1/m_s I_d), i = 1, 2, \cdots n_s$.
      Calculate tentative arms $\boldsymbol{a}_i = (1 - \frac{1}{2p})\widetilde{\boldsymbol{a}} + \frac{1}{2p}\boldsymbol{z}_i$.
      Conduct estimation $\boldsymbol{y} \leftarrow 1/n_s \sum_{i=1}^{n_s} (T(\boldsymbol{a}_i) + \eta_i)\boldsymbol{z}_i$.
      Update the current arm $\widetilde{\boldsymbol{a}} \leftarrow \boldsymbol{y}/\|\boldsymbol{y}\|$.
    Estimate the expected reward for $\widetilde{\boldsymbol{a}}$ through n_s samples: $r_n = 1/n_s \sum_{i=1}^{n_s} T(\widetilde{\boldsymbol{a}}) + \eta_i)$.
    Candidate Elimination:
    if $r_n \geq \lambda_1(1 - p\widetilde{\varepsilon}_s)$ **then**
      Keep the arm $\mathcal{A}_s \leftarrow \mathcal{A}_s \cup \{\widetilde{\boldsymbol{a}}\}$
  Label the arms: $L_s = |\mathcal{A}_s|, \mathcal{A}_t =: \{\boldsymbol{a}_s^{(1)}, \cdots \boldsymbol{a}_s^{(L_s)}\}$.

$|\lambda_2| \cdots \geq |\lambda_k|$. Therefore the optimal reward is λ_1 and can be achieved by $\boldsymbol{a}^*(s) = \boldsymbol{v}_1(s), s \in [p]$. In this section we only consider $p \geq 3$ and leave the quadratic and low-rank matrix setting to the next section.

Theorem 3.13. *For $p \geq 3$, by conducting alternating power iteration, one can get an ε-optimal arm with a total $O\left((2k)^p \log^p(p/\delta) d^p/\varepsilon^2\right)$ bandits.*

This setting is actually much easier than the symmetric setting. Notice by replacing one slice of $\vec{\boldsymbol{a}}$ by random Gaussian $\boldsymbol{z}_i \sim \mathcal{N}(0, 2/d\log(d/\delta))$, one directly gets $\boldsymbol{T}(\boldsymbol{a}(1), \cdots \boldsymbol{a}(s-1), \boldsymbol{I}, \boldsymbol{a}(s+1), \cdots \boldsymbol{a}(p))$ on each slice with $1/n \sum_i f(\boldsymbol{a}(1), \cdots \boldsymbol{a}(s-1), \boldsymbol{z}_i, \boldsymbol{a}(s+1), \cdots \boldsymbol{a}(p))\boldsymbol{z}_i$ which is tensor product. We defer the proof to Appendix D.3.

3.3.3 Lower Bound for Stochastic Polynomial Bandits

In this section we show lower bound for stochastic polynomial bandits.

$$f(\boldsymbol{a}) = \prod_{i=1}^{p} (\boldsymbol{\theta}_i^\top \boldsymbol{a}) + \eta, \text{ where } \eta \sim N(0,1), \|\boldsymbol{a}\|_2 \leq 1 \text{ and } f(\boldsymbol{a}) \leq 1. \tag{1}$$

Theorem 3.14. *For all algorithms $\mathcal{A}$ that adaptively interact with bandit (Eq (1)) for T rounds and output a vector $a^{(T)} \in \mathbb{R}^d$, we have*

$$\inf_{\mathcal{A}} \sup_{(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_p)} \mathbb{E}_{(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_p)} \left[\prod_{i=1}^{p} (\boldsymbol{\theta}_i^\top \boldsymbol{a}^*) - \prod_{i=1}^{p} (\boldsymbol{\theta}_i^\top \boldsymbol{a}^{(T)}) \right] \geq \frac{1}{p^p} \sqrt{\frac{d^p}{T}}$$

From the theorem, we can tell even when the problem is rank-1, we need a sample complexity of $\Omega((d/p^2)^p/\epsilon^2)$ to attain ϵ-optimal reward. In the appendix we also show a complexity of $\Omega(d^p/\epsilon^2)$ for asymmetric setting, which matches our upper bound up to poly-logarithmic factors.

3.4 Noiseless Polynomial Reward

In this subsection, we study the sample complexity for learning bandits with **noiseless** polynomial rewards. First we present the definition of admissible polynomial families.

Definition 3.15 (Admissible Polynomial Family)**.** For $\boldsymbol{a} \in \mathbb{R}^d$, define $\widetilde{\boldsymbol{a}} = [1, \boldsymbol{a}^\top]^\top$. For a algebraic variety $\mathcal{V} \subseteq (\mathbb{R}^{d+1})^{\otimes p}$, define $\mathcal{R}_\mathcal{V} := \{r_{\boldsymbol{\theta}}(\boldsymbol{a}) = \langle \boldsymbol{\theta}, \widetilde{\boldsymbol{a}}^{\otimes p} \rangle : \boldsymbol{\theta} \in \mathcal{V}\}$ as the polynomial family with parameters in $\mathcal{V}$. We define the dimension of the family $\mathcal{R}_\mathcal{V}$ as the algebraic dimension of $\mathcal{V}$. Next, define $\mathcal{X} := \left\{\widetilde{\boldsymbol{a}}^{\otimes p} : \boldsymbol{a} \in \mathbb{R}^d\right\}$. An polynomial family $\mathcal{R}_\mathcal{V}$ is said to be admissible[2] w.r.t. $\mathcal{X}$ if for any $\boldsymbol{\theta} \in \mathcal{V}$, $\dim(\mathcal{X} \cap \{X \in \mathcal{X} : \langle X, \boldsymbol{\theta} \rangle = 0\}) < \dim(\mathcal{X}) = d$.

3.4.1 Upper Bound via Solving Polynomial Equations

We show that if the action set $\mathcal{A}$ is of positive measure with respect to the Lebesgue measure μ, then by playing actions randomly, we can uniquely solve for the ground-truth reward function $r_{\boldsymbol{\theta}}(\boldsymbol{a})$ almost surely with samples of size that scales with the intrinsic algebraic dimension of $\mathcal{V}$, provided that $\mathcal{V}$ is an admissible algebraic variety.

Theorem 3.16. *Assume that the reward function class is an admissible polynomial family $\mathcal{R}_\mathcal{V}$. If $\mu(\mathcal{A}) > 0$, where μ is the Lebesgue meaure, then by randomly sample actions $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_T$ from $\mathbb{P}_{\boldsymbol{a} \sim \mathcal{N}(0, I_d)}(\cdot | \boldsymbol{a} \in \mathcal{A})$, when $T \geq 2\dim(\mathcal{V})$, we can uniquely solve for the ground-truth $\boldsymbol{\theta}$ and thus determine the optimal action almost surely.*

We state two important examples of admissible polynomial families with $O(d)$ dimensions.

Example 3.17. (Low-rank Polynomial) The function class $\mathcal{R}_\mathcal{V}$ of possibly inhomogeneous degree-p polynomials with k summands $\mathcal{R}_\mathcal{V} = \{r(\boldsymbol{a}) = \sum_{i=1}^k \lambda_i \langle \boldsymbol{v}_i, \boldsymbol{a} \rangle^{p_i} \mid \lambda_i \in \mathbb{R}, \boldsymbol{v}_i \in \mathbb{R}^d\}$ is admissible with $\dim(\mathcal{R}_\mathcal{V}) \leq dk$, where $p = \max\{p_i\}$. Neural network with monomial/polynomial activation functions are low-rank polynomials.

Example 3.18. The function class $\mathcal{R}_\mathcal{V} = \{r(\boldsymbol{a}) = q(\boldsymbol{U}\boldsymbol{a}) \mid \boldsymbol{U} \in \mathbb{R}^{k \times d}, \deg q(\cdot) \leq p\}$ is admissible with $\dim(\mathcal{V}) \leq d(k+1)^p$ [11].

3.4.2 Lower bounds with UCB Algorithms

In this subsection, we construct a hard bandit problem where the rewards are noiseless degree-p polynomial, and shows that any UCB algorithm needs at least $\Omega(d^p)$ actions to learn an ϵ-optimal action. On the contrary, Theorem 3.16 shows that by playing actions randomly, we only need $2d(p+1)^p) = O(d)$ actions.

Hard case construction Let e_i denotes the i-th standard orthonormal basis of $\mathbb{R}^d$, i.e., e_i has only one 1 at the i-th entry and 0's for other entries. We define a p-th multi-indices set Λ as $\Lambda = \{(\alpha_1, \ldots, \alpha_p) | 1 \leq \alpha_1 < \cdots < \alpha_p \leq d\}$. For an $\alpha = (\alpha_1, \ldots, \alpha_p) \in \Lambda$, denote $M_\alpha = e_{\alpha_1} \otimes \cdots \otimes e_{\alpha_p}$. Then the model space $\mathcal{M}$ is is defined as $\mathcal{M} = \left\{M_\alpha | \alpha \in \Lambda\right\}$, which a subset of rank-1 p-th order tensors. The action set $\mathcal{A}$ is defined as $\mathcal{A} = \mathrm{conv}(\{e_{\alpha_1} + \cdots + e_{\alpha_p} | \alpha \in \Lambda\})$. Assume that the ground-truth parameter is $M^* = M_{\alpha^*} \in \mathcal{M}$. The noiseless reward $r_t = r(M^*, a_t) = \langle M^*, (a_t)^{\otimes p} \rangle = \prod_{i=1}^p \langle e_{\alpha_i^*}, a_t \rangle$ is a polynomial of a_t and falls into the case of Example 3.18.

UCB Algorithms The UCB algorithms sequentially maintain a confidence set $\mathcal{C}_t$ after playing actions $a_1, \ldots, a_t$. Then UCB algorithms play $a_{t+1} \in \arg\max_{a \in \mathcal{A}} \mathrm{UCB}_t(a)$, where $\mathrm{UCB}_t(a) = \max_{M \in \mathcal{C}_t} \langle M, (a)^{\otimes p} \rangle$.

Theorem 3.19. *Assume that for each $t \geq 0$, the confidence set $\mathcal{C}_t$ contains the ground-truth model, i.e., $M^* \in \mathcal{C}_t$. Then for the noiseless degree-p polynomial bandits, any UCB algorithm needs to play at least $\binom{d}{p} - 1$ actions to distinguish models in $\mathcal{M}$. Therefore, in the worst-case, the sample complexity to learn an ϵ-optimal policy for any $\epsilon < 1$ is lower bounded by $\binom{d}{p} - 1 = \Omega(d^p)$.*

Theorem 3.19 shows the failure of the optimistic mechanism, which forbids the algorithm to play an informative action that is known to be low reward for all models in the confidence set. On the contrary, the reward function class falls into the form of $q(\boldsymbol{U}\boldsymbol{a})$, therefore, by playing actions randomly[3], we only need $2d(p+1)^p = O(d)$ actions as Theorem 3.16 suggests.

[2] Intuitively, admissibility means the dimension of $\mathcal{X}$ decreases by one when there is an additional linear constraint $\langle \boldsymbol{\theta}, X \rangle = 0$

[3] Careful readers may notice that $\mathcal{A}$ is of measure zero in this setting. However, since the reward function is a homogenous polynomial of degree p, we can actually obtain the rewards on $\mathrm{conv}(\mathcal{A}, \mathbf{0})$, which is of positive measure.

References

[1] Yasin Abbasi-Yadkori, Peter L Bartlett, and Alan Malek. Linear programming for large-scale Markov decision problems. *arXiv preprint arXiv:1402.6763*, 2014.

[2] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *NIPS*, volume 11, pages 2312–2320, 2011.

[3] Zeyuan Allen-Zhu and Yuanzhi Li. First efficient convergence for streaming k-pca: a global, gap-free, and near-optimal rate. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 487–492. IEEE, 2017.

[4] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of machine learning research*, 15:2773–2832, 2014.

[5] Ery Arias-Castro, Emmanuel J Candes, and Mark A Davenport. On the fundamental limits of adaptive sensing. *IEEE Transactions on Information Theory*, 59(1):472–481, 2012.

[6] Sebastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1), 2012.

[7] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *International conference on Algorithmic learning theory*, pages 23–37. Springer, 2009.

[8] T Tony Cai, Xiaodong Li, Zongming Ma, et al. Optimal rates of convergence for noisy sparse phase retrieval via thresholded wirtinger flow. *The Annals of Statistics*, 44(5):2221–2251, 2016.

[9] Emmanuel J Candes, Xiaodong Li, and Mahdi Soltanolkotabi. Phase retrieval via wirtinger flow: Theory and algorithms. *IEEE Transactions on Information Theory*, 61(4):1985–2007, 2015.

[10] Alexandra Carpentier and Rémi Munos. Bandit theory meets compressed sensing for high dimensional stochastic linear bandit. In *Artificial Intelligence and Statistics*, pages 190–198. PMLR, 2012.

[11] Sitan Chen and Raghu Meka. Learning polynomials in few relevant dimensions. In *Conference on Learning Theory*, pages 1161–1227. PMLR, 2020.

[12] Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. 2008.

[13] Simon S Du, Wei Hu, Sham M Kakade, Jason D Lee, and Qi Lei. Few-shot learning via learning the representation, provably. *arXiv preprint arXiv:2002.09434*, 2020.

[14] Simon S Du, Sham M Kakade, Jason D Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in RL. *arXiv preprint arXiv:2103.10897*, 2021.

[15] Eyal Even-Dar, Shie Mannor, and Yishay Mansour. Pac bounds for multi-armed bandit and markov decision processes. In *International Conference on Computational Learning Theory*, pages 255–270. Springer, 2002.

[16] Sarah Filippi, Olivier Cappe, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *NIPS*, volume 23, pages 586–594, 2010.

[17] Behrooz Ghorbani, Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Linearized two-layers neural networks in high dimension. *The Annals of Statistics*, 49(2):1029–1054, 2021.

[18] Aditya Gopalan, Odalric-Ambrym Maillard, and Mohammadi Zaki. Low-rank bandits with latent mixtures. *arXiv preprint arXiv:1609.01508*, 2016.

[19] Rajarshi Guhaniyogi, Shaan Qamar, and David B Dunson. Bayesian tensor regression. *The Journal of Machine Learning Research*, 18(1):2733–2763, 2017.

[20] Botao Hao, Tor Lattimore, Csaba Szepesvári, and Mengdi Wang. Online sparse reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pages 316–324. PMLR, 2021.

[21] Botao Hao, Tor Lattimore, and Mengdi Wang. High-dimensional sparse linear bandits. *arXiv preprint arXiv:2011.04020*, 2020.

[22] Botao Hao, Jie Zhou, Zheng Wen, and Will Wei Sun. Low-rank tensor bandits. *arXiv preprint arXiv:2007.15788*, 2020.

[23] Moritz Hardt and Eric Price. The noisy power method: A meta algorithm with applications. *Advances in neural information processing systems*, 27:2861–2869, 2014.

[24] Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, 2018.

[25] Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of rl problems, and sample-efficient algorithms. *arXiv preprint arXiv:2102.00815*, 2021.

[26] Nicholas Johnson, Vidyashankar Sivakumar, and Arindam Banerjee. Structured stochastic linear bandits. *arXiv preprint arXiv:1606.05693*, 2016.

[27] Kwang-Sung Jun, Rebecca Willett, Stephen Wright, and Robert Nowak. Bilinear bandits with low-rank structure. In *International Conference on Machine Learning*, pages 3163–3172. PMLR, 2019.

[28] Sumeet Katariya, Branislav Kveton, Csaba Szepesvari, Claire Vernade, and Zheng Wen. Stochastic rank-1 bandits. In *Artificial Intelligence and Statistics*, pages 392–401. PMLR, 2017.

[29] Sahin Lale, Kamyar Azizzadenesheli, Anima Anandkumar, and Babak Hassibi. Stochastic linear bandits with hidden low rank structure. *arXiv preprint arXiv:1901.09490*, 2019.

[30] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[31] Guillaume Lecué and Shahar Mendelson. Minimax rate of convergence and the performance of erm in phase recovery. *arXiv preprint arXiv:1311.5024*, 2013.

[32] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

[33] Xiaoshan Li, Da Xu, Hua Zhou, and Lexin Li. Tucker tensor regression and neuroimaging analysis. *Statistics in Biosciences*, 10(3):520–545, 2018.

[34] Yangyi Lu, Amirhossein Meisami, and Ambuj Tewari. Low-rank generalized linear bandit problems. In *International Conference on Artificial Intelligence and Statistics*, pages 460–468. PMLR, 2021.

[35] Shie Mannor and John N Tsitsiklis. The sample complexity of exploration in the multi-armed bandit problem. *Journal of Machine Learning Research*, 5(Jun):623–648, 2004.

[36] James S. Milne. Algebraic geometry (v6.02), 2017. Available at www.jmilne.org/math/.

[37] Cameron Musco and Christopher Musco. Randomized block krylov methods for stronger and faster approximate singular value decomposition. *arXiv preprint arXiv:1504.05477*, 2015.

[38] Paat Rusmevichientong and John N Tsitsiklis. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

[39] Dan Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In *Advances in Neural Information Processing Systems*, pages 2256–2264, 2013.

[40] Maziar Sanjabi, Sina Baharlouei, Meisam Razaviyayn, and Jason D Lee. When does non-orthogonal tensor decomposition have no spurious local minima? *arXiv preprint arXiv:1911.09815*, 2019.

[41] Igor R Shafarevich. *Basic Algebraic Geometry 1: Varieties in Projective Space*. Springer Science & Business Media, 2013.

[42] Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12(4):389–434, 2012.

[43] Ruosong Wang, Russ R Salakhutdinov, and Lin Yang. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. *Advances in Neural Information Processing Systems*, 33:6123–6135, 2020.

[44] Yang Wang and Zhiqiang Xu. Generalized phase retrieval: measurement number, matrix recovery and beyond. *Applied and Computational Harmonic Analysis*, 47(2):423–446, 2019.

[45] Yining Wang and Animashree Anandkumar. Online and differentially-private tensor decomposition. *arXiv preprint arXiv:1606.06237*, 2016.

[46] Pan Xu, Zheng Wen, Handong Zhao, and Quanquan Gu. Neural contextual bandits with deep representation and shallow exploration. *arXiv preprint arXiv:2012.01780*, 2020.

[47] Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. In *International Conference on Machine Learning*, 2020.

[48] Dongruo Zhou, Lihong Li, and Quanquan Gu. Neural contextual bandits with ucb-based exploration. In *International Conference on Machine Learning*, pages 11492–11502. PMLR, 2020.

[49] Hua Zhou, Lexin Li, and Hongtu Zhu. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] Our abstract and introduction accurately reflect our main contribution, which is also summarized in Table 1.
   (b) Did you describe the limitations of your work? [Yes] In the Preliminary section we clearly state the settings we considered and when our results apply.
   (c) Did you discuss any potential negative societal impacts of your work? [No] Our work is theoretical and generally will not have negative social impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] The assumptions are clearly included in every theorem or lemma.
   (b) Did you include complete proofs of all theoretical results? [Yes] All the proofs can be found in the appendix
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]