Learning with Good Feature Representations in Bandits and in RL with a Generative Model

Tor Lattimore [1] **Csaba Szepesvári** [2 3] **Gellért Weisz** [1]

Abstract

The construction by Du et al. (2019) implies that even if a learner is given linear features in $\mathbb{R}^d$ that approximate the rewards in a bandit with a uniform error of ε, then searching for an action that is optimal up to $O(\varepsilon)$ requires examining essentially all actions. We use the Kiefer–Wolfowitz theorem to prove a positive result that by checking only a few actions, a learner can always find an action that is suboptimal with an error of at most $O(\varepsilon\sqrt{d})$. Thus, features are useful when the approximation error is small relative to the dimensionality of the features. The idea is applied to stochastic bandits and reinforcement learning with a generative model where the learner has access to d-dimensional linear features that approximate the action-value functions for all policies to an accuracy of ε. For linear bandits, we prove a bound on the regret of order $\sqrt{dn\log(k)} + \varepsilon n\sqrt{d}\log(n)$ with k the number of actions and n the horizon. For RL we show that approximate policy iteration can learn a policy that is optimal up to an additive error of order $\varepsilon\sqrt{d}/(1-\gamma)^2$ and using $d/(\varepsilon^2(1-\gamma)^4)$ samples from a generative model. [0]

1. Introduction

Du et al. (2019) ask whether "good feature representation" is sufficient for efficient reinforcement learning and suggest a negative answer. Efficiency here means learning a good policy with a small number of interactions either with the environment (on-line learning), or with a simulator (planning). A linear feature representation is called "good" if it approximates the value functions of *all* policies with a small uniform error. The same question can also be asked for learning in bandits. The ideas by Du et al. (2019) suggest that the answer is also negative in finite-armed bandits with a misspecified linear model. All is not lost, however. By relaxing the objective, we will show that one can obtain positive results showing that efficient learning *is* possible in interactive settings with good feature representations.

The rest of this article is organised as follows. First we introduce the problem of learning to identify a near-optimal action with side information about the possible reward (Section 2). We then adapt the argument of Du et al. (2019) to show that no algorithm can find an $O(\varepsilon)$-optimal action without examining nearly all the actions, even when the rewards lie within an ε-vicinity of a subspace spanned by some features available to the algorithm (Section 3). The negative result is complemented by a positive result showing that there exists an algorithm such that for any feature map of dimension d, the algorithm is able to find an action with suboptimality gap of at most $O(\varepsilon\sqrt{d})$ where ε is the maximum distance between the reward and the subspace spanned by the features in the max-norm. The algorithm only needs to investigate the reward at $O(d\log\log d)$ well-chosen actions. The main idea is to use the Kiefer-Wolfowitz theorem with a least-squares estimator of the reward function. Finally, we apply the idea to stochastic bandits (Section 5) and reinforcement learning with a generative model (Section 6).

Related work Despite its importance, the problem of identifying near-optimal actions when rewards follow a misspecified linear model has only recently received attention. Of course, there is the recent paper by Du et al. (2019), whose negative result inspired this work and is summarised for our setting in Section 3. A contemporaneous work also addressing the issues raised by Du et al. (2019) is by Van Roy and Dong (2019), who make a connection to the Eluder dimension (Russo and Van Roy, 2013) and prove a variation on our Proposition 4.2. The setting studied here in Section 3 is closely related to the query complexity of exactly maximising a function in a given function class, which was studied by Amin et al. (2011).

[0] January 16, 2021: Correction to Lemma 2.2: $-1/k$ was missing in the first version. Nothing after Corollary 3.3 is impacted.

[1] DeepMind, London [2] DeepMind, Edmonton [3] University of Alberta. Correspondence to: Tor Lattimore <lattimore@google.com>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

They introduced the haystack dimension as a hardness measure for exact maximisation. Unfortunately, their results for infinite classes are generally not tight and no results for misspecified linear models were provided. Another related area is pure exploration in bandits, which was popularised in the machine learning community by Even-Dar et al. (2006); Audibert and Bubeck (2010). The standard problem is to identify a (near)-optimal action in an unstructured bandit. Soare et al. (2014) study pure exploration in linear bandits, but do not address the case where the model is misspecified. More general structured settings have also been considered by Degenne et al. (2019) and others. The algorithms in these works begin by playing every action once, which is inconsequential in an asymptotic sense. Our focus, however, is on the finite-time regime where the number of actions is very large, which makes these algorithms unusable. We discuss the related work on linear bandits and RL in the relevant sections later.

Notation Given a matrix $A \in \mathbb{R}^{n\times m}$, the set of rows is denoted by $\mathrm{rows}(A)$ and its range is $\mathrm{range}(A) = \{A\theta : \theta \in \mathbb{R}^m\}$. When A is positive semi-definite, we define $\|x\|_A^2 = x^\top Ax$. The Minkowski sum of sets $U, V \subset \mathbb{R}^d$ is $U + V = \{u + v : u \in U, v \in V\}$. The standard basis vectors are $e_1, \dots, e_d$. There will never be ambiguity about deducing the dimension.

2. Problem setup

We start with an abstract problem that is reminiscent of pure exploration in bandits, but without noise. Fix $\delta > 0$ and consider the problem of identifying a δ-optimal action out of k actions with the additional information that the unknown reward vector $\mu \in \mathbb{R}^k$ belongs to a known hypothesis set $\mathcal{H} \subset \mathbb{R}^k$. An action $j \in [k] = \{1, \dots, k\}$ is δ-optimal for $\mu = (\mu_i)_{i=1}^k$ if $\mu_j > \max_i \mu_i - \delta$. The learner sequentially queries actions $i \in [k]$ and observes the reward μ_i. At some point the learner should stop and output both an estimated optimal action $\hat{a} \in [k]$ and an estimation vector $\hat{\mu} \in \mathbb{R}^k$. There is no noise, so the learner has no reason to query the same action twice. Of course, if the learner queries all the actions, then it knows both μ and the optimal action. The learner is permitted to randomise. Two objectives are considered. The first only measures the quality of the outputted action $\hat{a}$, while the second depends on $\hat{\mu}$.

Definition 2.1. A learner is called sound for $(\mathcal{H}, \delta)$ if $\|\hat{\mu} - \mu\|_\infty < \delta$ almost surely for all $\mu \in \mathcal{H}$. It is called max-sound for $(\mathcal{H}, \delta)$ if $\mu_{\hat{a}} > \max_a \mu_a - \delta$ almost surely for all $\mu \in \mathcal{H}$.

Denote by $q_\delta(\mathscr{A}, \mu)$ the expected number of queries learner $\mathscr{A}$ executes when interacting with the environment specified by μ and let

$$c_\delta^{\max}(\mathcal{H}) = \inf_{\mathscr{A}:\mathscr{A}\text{ is }(\mathcal{H},\delta)\text{-max-sound}} \sup_{\mu\in\mathcal{H}} q_\delta(\mathscr{A}, \mu)$$
$$c_\delta^{\mathrm{est}}(\mathcal{H}) = \inf_{\mathscr{A}:\mathscr{A}\text{ is }(\mathcal{H},\delta)\text{-sound}} \sup_{\mu\in\mathcal{H}} q_\delta(\mathscr{A}, \mu)\,,$$

which are the minimax query complexities for max-sound/sound learners respectively when interacting with reward vectors in $\mathcal{H}$ and with error tolerance δ. Both complexity measures are increasing as the hypothesis class is larger in the sense that if $U \subset V$, then $c_\delta^{\max}(U) \le c_\delta^{\max}(V)$, and similarly for c_δ^{est}. If a learner is sound for $(\mathcal{H}, \delta)$ and $\hat{a} = \arg\max \hat{\mu}$, then clearly it is also max-sound for $(\mathcal{H}, 2\delta)$, which shows that

$$c_{2\delta}^{\max}(\mathcal{H}) \le c_\delta^{\mathrm{est}}(\mathcal{H})\,. \tag{1}$$

Our primary interest is to understand $c_\delta^{\max}(\mathcal{H})$. Upper bounds, however, will be proven using Eq. (1) and by controlling $c_\delta^{\mathrm{est}}(\mathcal{H})$. Furthermore, in Section 4 we provide a simple characterisation of $c_\delta^{\mathrm{est}}(\mathcal{H})$, while $c_\delta^{\max}(\mathcal{H})$ is apparently more subtle. Later we need the following intuitive result, which says that the complexity of finding a near-optimal action when the hypothesis set consists of the unit vectors is linear in the number of actions. The proof is given in Section A.

Lemma 2.2. $c_1^{\max}(\{e_1, \dots, e_k\}) = (k+1)/2 - 1/k$.

It follows from the aforementioned monotonicity that if $\{e_1, \dots, e_k\} \subseteq \mathcal{H}$, then $c_1^{\max}(\mathcal{H}) \ge (k+1)/2 - 1/k$.

3. Negative result

Let $\Phi \in \mathbb{R}^{k\times d}$. The rows of Φ should be thought of as feature vectors assigned to each of the k actions; accordingly we call Φ a feature matrix. Furthermore, when $\mu \in \mathbb{R}^k$ and $a \in \mathrm{rows}(\Phi)$, we abuse notation by writing μ_a for the value of vector μ at the index of row a in Φ. Our interest lies in the regime where k is much larger than d and where $\exp(d)$ is large.

We consider hypothesis classes where the true reward lies within an ε-vicinity of $\mathrm{range}(\Phi)$ as measured in max-norm. Let $\mathcal{H}_\Phi^\varepsilon = \mathrm{range}(\Phi) + B_\infty(\varepsilon)$, where $B_\infty(\varepsilon) = [-\varepsilon, \varepsilon]^k$ is a k-dimensional hypercube. How large is $c_\delta^{\max}(\mathcal{H}_\Phi^\varepsilon)$ for different regimes of δ and ε and feature matrices? As we shall see, for $\delta = \Omega(\varepsilon\sqrt{d})$ one can keep the complexity small, while for smaller δ there exist feature matrices for which the complexity can be as high as the large dimension, k.

The latter result follows from the core argument of the recent paper by Du et al. (2019). The next lemma is the key tool, and is a consequence of the Johnson–Lindenstrauss lemma. It shows that there exist matrices $\Phi \in \mathbb{R}^{k\times d}$ with k much larger than d where rows have unit length and all non-equal rows are almost orthogonal.

Lemma 3.1. *For any $\varepsilon > 0$ and $d \in [k]$ such that $d \geq \lceil 8\log(k)/\varepsilon^2 \rceil$, there exists a feature matrix $\Phi \in \mathbb{R}^{k\times d}$ with unique rows such that for all $a, b \in \mathrm{rows}(\Phi)$ with $a \neq b$, $\|a\|_2 = 1$ and $|a^\top b| \leq \varepsilon$.*

Lemmas 2.2 and 3.1 together imply the promised result:

Proposition 3.2. *For any $\varepsilon > 0$ and $d \in [k]$ with $d \geq \lceil 8\log(k)/\varepsilon^2 \rceil$, there exists a feature matrix $\Phi \in \mathbb{R}^{k\times d}$ such that $c_1^{\max}(\mathcal{H}_\Phi^\varepsilon) \geq (k+1)/2 - 1/k$.*

Proof. Let Φ be the matrix from Lemma 3.1 with $\mathrm{rows}(\Phi) = (a_i)_{i=1}^k$. We want to show that $e_i \in \mathcal{H}_\Phi^\varepsilon$ for all $i \in [k]$ and then apply Lemma 2.2. If $\theta = a_i$, then $\Phi\theta = (a_1^\top a_i, \ldots, a_i^\top a_i, \ldots, a_k^\top a_i)^\top$. By the choice of Φ the ith component is one and the others are all less than ε in absolute value. Hence, $\|\Phi\theta - e_i\|_\infty \leq \varepsilon$, which completes the proof. □

The proposition has a worst-case flavour. Not all feature matrices have a high query complexity. For a silly example, the query complexity of the zero matrix $\Phi = \mathbf{0}$ satisfies $c_1^{\max}(\mathcal{H}_\Phi^\varepsilon) = 0$ provided $\varepsilon < 1$. That said, the matrix witnessing the claims in Lemma 3.1 can be found with non-negligible probability by sampling the rows of Φ uniformly from the surface of the $(d-1)$-dimensional sphere. There is another way of writing this result, emphasising the role of the dimension rather than the number of actions.

Corollary 3.3. *For all $\delta > \varepsilon$, there exists a feature matrix $\Phi \in \mathbb{R}^{k\times d}$ with suitably large k such that*

$$c_\delta^{\max}(\mathcal{H}_\Phi^\varepsilon) \geq \frac{1}{2}\exp\left(\frac{d-1}{8}\left(\frac{\varepsilon}{\delta}\right)^2\right).$$

The proof follows by rescaling the features in Proposition 3.2 and is given in Appendix B.

4. Positive result

The negative result of the previous section is complemented with a positive result showing that the query complexity can be bounded independently of k whenever $\delta = \Omega(\varepsilon\sqrt{d})$. For the remainder of the article we make the following assumption:

Assumption 4.1. $\Phi \in \mathbb{R}^{k\times d}$ has unique rows and the span of $\mathrm{rows}(\Phi)$ is all of $\mathbb{R}^d$.

We discuss the relationship between this result and Proposition 3.2 at the end of the section.

Proposition 4.2. *Let $\Phi \in \mathbb{R}^{k\times d}$ and $\delta > 2\varepsilon(1+\sqrt{2d})$. Then, $c_\delta^{\max}(\mathcal{H}_\Phi^\varepsilon) \leq 4d\log\log d + 16$.*

The proof relies on the Kiefer–Wolfowitz theorem, which we now recall. Given a probability distribution $\rho : \mathrm{rows}(\Phi) \to [0,1]$, let $G(\rho) \in \mathbb{R}^{d\times d}$ and $g(\rho) \in \mathbb{R}$ be given by

$$G(\rho) = \sum_{a\in\mathrm{rows}(\Phi)} \rho(a)aa^\top, \quad g(\rho) = \max_{a\in\mathrm{rows}(\Phi)} \|a\|^2_{G(\rho)^{-1}}.$$

Theorem 4.3 (Kiefer and Wolfowitz 1960)**.** *The following are equivalent:*

1. *ρ^* is a minimiser of g.*
2. *ρ^* is a maximiser of $f(\rho) = \log\det G(\rho)$.*
3. *$g(\rho^*) = d$.*

Furthermore, there exists a minimiser ρ^* of g such that the support of ρ^* has cardinality at most $|\mathrm{supp}(\rho)| \leq d(d+1)/2$.

The distribution ρ^* is called an (optimal) experimental design and the elements of its support are called its core set. Intuitively, when covariates are sampled from ρ, then $g(\rho)$ is proportional to the maximum variance of the corresponding least-squares estimator over all directions in $\mathrm{rows}(\Phi)$. Hence, minimising g corresponds to minimising the worst-case variance of the resulting least-squares estimator. A geometric interpretation is that the core set lies on the boundary of the central ellipsoid of minimum volume that contains $\mathrm{rows}(\Phi)$. The next theorem shows that there exists a near-optimal design with a small core set. The proof follows immediately from part (ii) of lemma 3.9 in the book by Todd (2016), which also provides an algorithm for computing such a distribution in roughly order kd^2 computation steps.

Theorem 4.4. *There exists a probability distribution ρ such that $g(\rho) \leq 2d$ and the core set of ρ has size at most $4d\log\log(d) + 16$.*

The proof of Proposition 4.2 is a corollary of the following more general result about least-squares estimators over near-optimal designs.

Proposition 4.5. *Let $\mu \in \mathcal{H}_\Phi^\varepsilon$ and $\eta \in [-\beta,\beta]^k$. Suppose that ρ is a probability distribution over $\mathrm{rows}(\Phi)$ satisfying the conclusions of Theorem 4.4. Then $\|\Phi\hat{\theta} - \mu\|_\infty \leq \varepsilon + (\varepsilon+\beta)\sqrt{2d}$, where*

$$\hat{\theta} = G(\rho)^{-1}\sum_{a\in\mathrm{rows}(\Phi)} \rho(a)(\mu_a + \eta_a)a.$$

The problem can be reduced to the case where $\eta = \mathbf{0}$ by noting that $\mu + \eta \in \mathcal{H}_\Phi^{\varepsilon+\beta}$ The only disadvantage is that this leads to an additional additive dependence on β.

Proof. Let $\mu = \Phi\theta + \Delta$ where $\|\Delta\|_\infty \leq \varepsilon$. The difference

between $\hat{\theta}$ and θ can be written as

$$\begin{aligned}\hat{\theta} - \theta &= G(\rho)^{-1} \sum_{a \in \mathrm{rows}(\Phi)} \rho(a) a \left(a^\top \theta + \Delta_a + \eta_a\right) - \theta \\ &= G(\rho)^{-1} \sum_{a \in \mathrm{rows}(\Phi)} \rho(a)(\Delta_a + \eta_a) a \,.\end{aligned}$$

Next, for any $b \in \mathrm{rows}(\Phi)$,

$$\begin{aligned}\langle b, \hat{\theta} - \theta \rangle &= \left\langle b, G(\rho)^{-1} \sum_{a \in \mathrm{rows}(\Phi)} \rho(a)(\Delta_a + \eta_a) a \right\rangle \\ &= \sum_{a \in \mathrm{rows}(\Phi)} \rho(a)(\Delta_a + \eta_a) \langle b, G(\rho)^{-1} a \rangle \\ &\le (\varepsilon + \beta) \sum_{a \in \mathrm{rows}(\Phi)} \rho(a) |\langle b, G(\rho)^{-1} a \rangle| \\ &\le (\varepsilon + \beta) \sqrt{\sum_{a \in \mathrm{rows}(\Phi)} \rho(a) \langle b, G(\rho)^{-1} a \rangle^2} \\ &= (\varepsilon + \beta) \sqrt{\sum_{a \in \mathrm{rows}(\Phi)} \rho(a) b^\top G(\rho)^{-1} a a^\top G(\rho)^{-1} b} \\ &= (\varepsilon + \beta) \sqrt{\|b\|^2_{G(\rho)^{-1}}} \le (\varepsilon + \beta) \sqrt{g(\rho)} \le (\varepsilon + \beta) \sqrt{2d} \,,\end{aligned}$$

where the first inequality follows from Hölder's inequality and the fact that $\|\Delta\|_\infty \le \varepsilon$, the second by Jensen's inequality and the last two by our choice of ρ and Theorem 4.4. Therefore

$$\langle b, \hat{\theta} \rangle \le \langle b, \theta \rangle + (\varepsilon + \beta)\sqrt{2d} \le \mu_b + \varepsilon + (\varepsilon + \beta)\sqrt{2d} \,.$$

A symmetrical argument completes the proof. □

Proof of Proposition 4.2. Let ρ be a probability distribution over $\mathrm{rows}(\Phi)$ satisfying the conclusions of Theorem 4.4. Consider the algorithm that evaluates μ on each point of the support of ρ and computes the least-squares estimator defined in Proposition 4.5 and predicts $\hat{a} = \arg\max_{a \in \mathrm{rows}(\Phi)} \langle a, \hat{\theta} \rangle$. Let $a^* = \arg\max_{a \in \mathrm{rows}(\Phi)} \mu_a$ be the optimal action. Then by Proposition 4.5 with $\eta = \mathbf{0}$,

$$\begin{aligned}\mu_{\hat{a}} &\ge \langle \hat{a}, \hat{\theta} \rangle - \varepsilon \left(1 + \sqrt{2d}\right) \ge \langle a^*, \hat{\theta} \rangle - \varepsilon \left(1 + \sqrt{2d}\right) \\ &\ge \mu_{a^*} - 2\varepsilon \left(1 + \sqrt{2d}\right) > \mu_{a^*} - \delta \,.\end{aligned}$$ □

Discussion Corollary 3.3 shows that the query complexity is exponential in d when δ is not much larger than ε, but is benign when $\delta = \Omega(\varepsilon\sqrt{d})$. The positive result shows that in the latter regime the complexity is more or less linear in d. Precisely,

$$\min\left\{\delta : c^{\max}_\delta(\mathcal{H}^\varepsilon_\Phi) \le 4d \log\log(d) + 16\right\} = O(\varepsilon\sqrt{d}) \,.$$

The message is that there is a sharp tradeoff between query complexity and error. The learner pays dearly in terms of query complexity if they demand an estimation error that is close to the approximation error. By sacrificing a factor of $\sqrt{d}$ in estimation error, the query complexity is practically just linear in d.

Comparison to supervised learning As noted by Du et al. (2019), the negative result does not hold in supervised learning, where the learner is judged on its average prediction error with respect to the data generating distribution. Suppose that $a, a_1, \ldots, a_n$ are sampled i.i.d. from some distribution P on $\mathrm{rows}(\Phi)$ and the learner observes $(a_t)_{t=1}^n$ and $(\mu_{a_t})_{t=1}^n$.

$$\hat{\theta} = \left(\sum_{t=1}^n a_t a_t^\top\right)^{-1} \sum_{t=1}^n a_t \mu_{a_t} \,.$$

Then, by making reasonable boundedness and span assumptions on $\mathrm{rows}(\Phi)$, and by combining the results in chapters 13 and 14 of (Wainwright, 2019), with high probability,

$$\mathbb{E}\left[(a^\top \hat{\theta} - \mu_a)^2 \,\middle|\, \hat{\theta}\right] = O\left(\frac{d}{n} + \varepsilon^2\right) \,.$$

Notice, there is no d multiplying the dependence on the approximation error. The fundamental difference is that a is sampled from P. The quantity $\max_{a \in \mathrm{rows}(\Phi)} (a^\top \hat{\theta} - \mu_a)^2$ behaves quite differently, as the lower bound shows.

Feature-dependent bounds The negative result in Section 3 shows that there *exist* feature matrices for which the learner must query exponentially many actions or suffer an estimation error that expands the approximation error by a factor of $\sqrt{d}$. On the other hand, Proposition 4.2 shows that for *any* feature matrix, there exists a learner that queries $O(d \log\log(d))$ actions for an estimation error of $\varepsilon\sqrt{d}$, roughly matching the lower bound. One might wonder whether or not there exists a feature-dependent measure that characterises the blowup in estimation error in terms of the feature matrix and query budget. One such measure is given here. Given a set $C \subseteq [k]$ with $|C| = q$, let $\Phi_C \in \mathbb{R}^{q \times d}$ be the matrix obtained from Φ by restricting to those rows indexed by C. Define

$$\lambda_q(\Phi) = \min_{C \subset [k], |C| = q} \max_{v \in \mathbb{R}^d \setminus \{0\}} \frac{\|\Phi v\|_\infty}{\|\Phi_C v\|_\infty} \,.$$

Proposition 4.6. *Let $1 \le q < k$ and $\delta_1 = \varepsilon(1 + \lambda_q(\Phi))$ and $\delta_2 > \varepsilon(1 + 2\lambda_q(\Phi))$. Then,*

$$c^{\mathrm{est}}_{\delta_1}(\mathcal{H}^\varepsilon_\Phi) > q \ge c^{\mathrm{est}}_{\delta_2}(\mathcal{H}^\varepsilon_\Phi) \,.$$

The proof is supplied in Appendix C. By (1), it also holds that $c^{\max}_{2\delta_2}(\mathcal{H}^\varepsilon_\Phi) \le q$. Currently we do not have a corresponding lower bound, however.

5. Misspecified linear bandits

Here we consider the classic stochastic bandit where the mean rewards are nearly a linear function of their associated features. We assume for simplicity that no two actions have the same features. In case this does not hold, a representative action can be chosen for each feature without changing the main theorem. Let $\Phi \in \mathbb{R}^{k\times d}$ and $\mu \in \mathcal{H}_\Phi^\varepsilon$. In rounds $t \in [n]$, the learner chooses actions $(X_t)_{t=1}^n$ with $X_t \in \mathrm{rows}(\Phi)$ and the reward is $Y_t = \mu_{X_t} + \eta_t$ where $(\eta_t)_{t=1}^n$ is a sequence of independent 1-subgaussian random variables. The optimal action has expected reward $\mu^* = \max_{a\in\mathcal{A}} \mu_a$ and the expected regret is $R_n = \mathbb{E}[\sum_{t=1}^n \mu^* - \mu_{X_t}]$. The idea is to use essentially the same elimination algorithm as (Lattimore and Szepesvári, 2019, chapter 22), which is summarised in Algorithm 1. In each episode, the algorithm computes a near-optimal design over a subset of the actions that are plausibly optimal. It then chooses each action in proportion to the optimal design and eliminates arms that appear sufficiently suboptimal.

Proposition 5.1. *When $\alpha = 1/(kn)$ and C is a suitably large universal constant and $\max_a \mu_a - \min_a \mu_a \le 1$, Algorithm 1 satisfies*

$$R_n \le C\left[\sqrt{dn\log(nk)} + \varepsilon n\sqrt{d}\log(n)\right].$$

In Appendix F, we show that the bound in Proposition 5.1 is tight up to logarithmic factors in the interesting regime where k is comparable to n.

Proof. Let $\mu = \Phi\theta + \Delta$ with $\|\Delta\|_\infty \le \varepsilon$, which exists by the assumption that $\mu \in \mathcal{H}_\Phi^\varepsilon$. We only analyse the behaviour of the algorithm within an episode, showing that the least-squares estimator is guaranteed to have sufficient accuracy so that (a) arms that are sufficiently suboptimal are eliminated and (b) some near-optimal arms are retained. Fix any $b \in \mathcal{A}$. Using the notation in Algorithm 1,

$$\begin{aligned}\langle b, \hat\theta - \theta\rangle &= \left|b^\top G^{-1}\sum_{s=1}^u \Delta_{X_s} X_s + b^\top G^{-1}\sum_{s=1}^u X_s\eta_s\right| \\ &\le \left|b^\top G^{-1}\sum_{a\in\mathcal{A}} u(a)a\Delta_a\right| + \left|b^\top G^{-1}\sum_{s=1}^u X_s\eta_s\right|. \quad (2)\end{aligned}$$

The first term is bounded using Jensen's inequality as before:

$$\begin{aligned}\left|b^\top G^{-1}\sum_{a\in\mathcal{A}} u(a)\Delta_a a\right| &\le \varepsilon\sum_{a\in\mathcal{A}} u(a)\left|b^\top G^{-1}a\right| \\ &\le \varepsilon\sqrt{\left(\sum_{a\in\mathcal{A}} u(a)\right) b^\top\sum_{a\in\mathcal{A}} u(a)G^{-1}aa^\top G^{-1}b} \\ &= \varepsilon\sqrt{\sum_{a\in\mathcal{A}} u(a)\|b\|_{G^{-1}}^2} \le \varepsilon\sqrt{\frac{2du}{m}} \le 2\varepsilon\sqrt{d},\end{aligned}$$

where the first inequality follows form Hölder's inequality, the second is Jensen's inequality and the last follows from the exploration distribution that guarantees $\|b\|_{G^{-1}}^2 \le 2d/m$. The second term in Eq. (2) is bounded using standard concentration bounds. Preciesly, by eq. (20.2) of (Lattimore and Szepesvári, 2019), with probability at least $1-2\alpha$,

$$\begin{aligned}\left|b^\top G^{-1}\sum_{s=1}^u X_s\eta_s\right| &\le \|b\|_{G^{-1}}\sqrt{2\log\left(\frac{1}{\alpha}\right)} \\ &\le \sqrt{\frac{4d}{m}\log\left(\frac{1}{\alpha}\right)}\end{aligned}$$

and $|\langle b, \hat\theta - \theta\rangle| \le 2\varepsilon\sqrt{d} + \sqrt{\frac{4d}{m}\log\left(\frac{1}{\alpha}\right)}$. Continuing with standard calculations, provided in Appendix D, one gets that the expected regret satisfies

$$R_n \le C\left[\sqrt{dn\log(nk)} + \varepsilon n\sqrt{d}\log(n)\right]$$

where $C > 0$ is a suitably large universal constant. The logarithmic factor in the second term is due to the fact that in each of the logarithmically many episodes the algorithm may eliminate the best remaining arm, but keep an arm that is at most $O(\varepsilon\sqrt{d})$ worse than the best remaining arm. □

Algorithm 1 PHASED ELIMINATION

INPUT $\Phi \in \mathbb{R}^{k\times d}$ and confidence level $\alpha \in (0,1)$

(1) Set $m = \lceil 4d\log\log d\rceil + 16$ and $\mathcal{A} = \mathrm{rows}(\Phi)$

(2) Find design $\rho : \mathcal{A} \to [0,1]$ with $g(\rho) \le 2d$ and $|\operatorname{supp}(\rho)| \le 4d\log\log(d) + 16$

(3) Compute $u(a) = \lceil m\rho(a)\rceil$ and $u = \sum_{a\in\mathcal{A}} u(a)$

(4) Take each action $a \in \mathcal{A}$ exactly $u(a)$ times with corresponding features $(X_s)_{s=1}^u$ and rewards $(Y_s)_{s=1}^u$

(5) Calculate the vector $\hat\theta$:

$$\hat\theta = G^{-1}\sum_{s=1}^u X_s Y_s \quad \text{with} \quad G = \sum_{a\in\mathcal{A}} u(a)aa^\top$$

(6) Update active set:

$$\mathcal{A} \leftarrow \left\{a \in \mathcal{A} : \max_{b\in\mathcal{A}}\langle\hat\theta, b-a\rangle \le 2\sqrt{\frac{4d}{m}\log\left(\frac{1}{\alpha}\right)}\right\}.$$

(7) $m \leftarrow 2m$ and GOTO (1)

Remark 5.2. When the active set contains fewer than d actions, then the conditions of Kiefer-Wolfowitz are not satisfied because $\mathcal{A}$ cannot span $\mathbb{R}^d$. Rest assured, however, since in these cases one can simply work in the smaller space spanned by $\mathcal{A}$ and the analysis goes through without further changes.

Known approximation error The logarithmic factor in the second term in the regret bound can be removed when ε is known by modifying the elimination criteria so that with high probability the optimal action is never eliminated, as explained in Remark D.1.

Infinite action sets The logarithmic dependence on k follows from the choice of α, which is needed to guarantee the concentration holds for all actions. When $k = \Omega(\exp(d))$, the union bound can be improved by a covering argument or using the argument in the next section. This leads to a bound of $O(d\sqrt{n\log(n)} + \varepsilon n\sqrt{d}\log(n))$, which is independent of the number of arms.

Other approaches We are not the first to consider misspecified linear bandits. Ghosh et al. (2017) consider the same setting and show that in the favourable case when one can cheaply test linearity, there exist algorithms for which the regret has order $\min(d, \sqrt{k})\sqrt{n}$ up to logarithmic factors. While such results a certainly welcome, our focus is on the case where k has the same order of magnitude as n and hence the dependence of the regret on ε is paramount. Another way to obtain a similar result to ours is to use the Eluder dimension (Russo and Van Roy, 2013), which should first be generalised a little to accommodate the need to use an accuracy threshhold that does not decrease with the horizon. Then the Eluder dimension can be controlled using either our techniques or the alternative argument by Van Roy and Dong (2019).

Contextual linear bandits Algorithms based on phased elimination are not easily adapted to the contextual case, which is usually addressed using optimistic methods. You might wonder whether or not LinUCB (Abbasi-Yadkori et al., 2011) serendipitously adapts to misspecified models in the contextual case. Gopalan et al. (2016) have shown that LinUCB *is* robust in the non-contextual case when ε is very small. Their conditions, however, depend the structure of the problem, and in particular on having good control of the 2-norm of Δ, which may scale like $\Omega(\varepsilon\sqrt{k})$ and is too big for large action sets. We provide a negative result in the supplementary material, as well as a modification that corrects the algorithm, but requires knowledge of the approximation error. The modification is a data-dependent refinement of the bonus used by Jin et al. (2019). An open question is to find an algorithm for contextual linear bandits for which the regret similar to Proposition 5.1 and where the algorithm does not need to know the approximation error.

6. Reinforcement learning

We now consider discounted reinforcement learning with a generative model, which means the learner can sample next-states and rewards for any state-action pair of their choice. The notation is largely borrowed from (Szepesvári, 2010). Fix an MDP with state space $[S]$, action space $[A]$, transition kernel P, reward function $r : [S] \times [A] \to [0, 1]$ and discount factor $\gamma \in (0, 1)$. The finiteness of the state space is assumed only for simplicity. As usual, V^π and Q^π refer to the value and action-value functions for policy π (e.g., $V^\pi(s)$ is the total expected discounted reward incurred while following policy π in the MDP) and V^* and Q^* the same for the optimal policy. The learner is given a feature matrix $\Phi \in \mathbb{R}^{SA\times d}$ such that $Q^\pi \in \mathcal{H}^\varepsilon_\Phi$ for all policies π and where Q^π is vectorised in the obvious way. The notation $\Phi(s, a) \in \mathbb{R}^d$ denotes the feature associated with state-action pair (s, a).

The main idea is the observation that if Q^* were known with reasonable accuracy on the support of an approximately optimal design ρ on the set of vectors $(\Phi(s, a) : s, a \in [S] \times [A])$, then least-squares in combination with our earlier arguments would provide a good estimation of the optimal state-action value function. Approximating Q^* on the core set $\mathcal{C} = \mathrm{supp}(\rho) \subset [S] \times [A]$ is possible using approximate policy iteration. For the remainder of this section let ρ be a design with $g(\rho) \le 2d$ and with support $\mathcal{C}$ and $G(\rho) = \sum_{(s,a)\in\mathcal{C}} \rho(s, a)\Phi(s, a)\Phi(s, a)^\top$.

Related work The idea to extrapolate a value function by sampling from a few anchor state/action pairs has been used before in a few works. The recent work by Zanette et al. (2019) consider approximate value iteration in the episodic setting and do not make a connection to optimal design. The challenge in the finite-horizon setting is that one must learn one parameter vector for each layer and, at least naively, errors propagate multiplicatively. For this reason using the anchor pairs from the support of an experimental design would not make the algorithm proposed by the aforementioned paper practical. Yang and Wang (2019) assume the transition matrix has a linear structure and also use least-squares with data from a pre-selected collection of anchor state/action pairs. Their assumption is that the features of all state-action pairs can be written as a convex combination of the anchoring features, which means the number of anchors is the number of corners of the polytope spanned by $\mathrm{rows}(\Phi)$ and may be much larger than d. One special feature of their paper is that the dependency on the horizon of the sample complexity is cubic in $1/(1-\gamma)$, while in our theorem it is quartic. Earlier, Lakshminarayanan et al. (2018) described how anchor states (with some lag allowed) can be used to reduce the number of constraints in the approximate linear programming approach to approximate planning in MDPs, while maintaining error bounds.

Approximate policy iteration Let π_1 be an arbitrary policy and define a sequence of policies $(\pi_k)_{k=1}^\infty$ inductively

using the following procedure. From each state-action pair $(s,a) \in \mathcal{C}$ take m roll-outs of length n following policy π_k and let $\hat{Q}_k(s,a)$ be the empirical average, which is only defined on the core set $\mathcal{C}$. The estimation of Q^{π_k} is then extended to all state-action pairs using the features and least-squares

$$\hat{\theta}_k = G(\rho)^{-1} \sum_{(s,a)\in\mathcal{C}} \rho(s,a)\Phi(s,a)\hat{Q}_k(s,a) \quad Q_k = \Phi\hat{\theta}_k \,.$$

Then π_{k+1} is chosen to be the greedy policy with respect to Q_k and the process is repeated. The following theorem shows that for suitable choices of roll-out length n, roll-out number m and iterations k, the policy π_{k+1} is nearly optimal with high probability. Significantly, the choice of parameters ensures that the total number of samples from the generative model is independent of S and A.

Theorem 6.1. *Suppose that approximate policy iteration is run with*

$$k = \frac{\log\left(\frac{1}{\varepsilon\sqrt{d}}\right)}{1-\gamma} \quad m = \frac{\log\left(\frac{2k|\mathcal{C}|}{\alpha}\right)}{2\varepsilon^2(1-\gamma)^2} \quad n = \frac{\log\left(\frac{1}{\varepsilon(1-\gamma)}\right)}{1-\gamma} \,.$$

Then there exists a universal constant C such that with probability at least $1-\alpha$, the policy π_{k+1} satisfies

$$\max_{s\in[S]} \left(V^*(s) - V^{\pi_{k+1}}(s)\right) \le C\varepsilon\sqrt{d}/(1-\gamma)^2 \,,$$

When ρ is chosen using Theorem 4.4 so that $|\mathcal{C}| \le 4d\log\log(d) + 16$, then the number of samples from the generative model is $kmn|\mathcal{C}|$, which is

$$O\left(\frac{\log\left(\frac{1}{\varepsilon(1-\gamma)}\right)\log\left(\frac{2k|\mathcal{C}|}{\alpha}\right)\log\left(\frac{1}{\varepsilon\sqrt{d}}\right) d\log\log(d)}{\varepsilon^2(1-\gamma)^4}\right) \,.$$

Before the proof we need two lemmas. The first controls the propagation of errors in policy iteration when using Q_k rather than Q^{π_k}. For policy π, let $P^\pi : \mathbb{R}^{[S]\times[A]} \to \mathbb{R}^{[S]\times[A]}$ defined by $(P^\pi Q)(s,a) = \sum_{s'} P(s'|s,a)Q(s',\pi(s'))$.

Lemma 6.2. *Let $\delta_i = Q_i - Q^{\pi_i}$ and $E_i = P^{\pi_{i+1}}(I - \gamma P^{\pi_{i+1}})^{-1}(I - \gamma P^{\pi_i}) - P^{\pi^*}$. Then, $Q^* - Q^{\pi_k} \le (\gamma P^{\pi^*})^k(Q^* - Q^{\pi_0}) + \gamma\sum_{i=0}^{k-1}(\gamma P^{\pi^*})^{k-i-1}E_i\delta_i$.*

Proof. This is stated as Eq. (7) in the proof of part (b) of Theorem 3 of (Farahmand et al., 2010) and ultimately follows from Lemma 4 of Munos (2003). □

The second lemma controls the value of the greedy policy with respect to a Q function in terms of the quality of the Q function.

Lemma 6.3 (Singh and Yee (1994), corollary 2)**.** *Let π be greedy with respect to an action-value function Q. Then for any state $s \in [S]$, $V^\pi(s) \ge V^*(s) - \frac{2}{1-\gamma}\|Q - Q^*\|_\infty$.*

Proof of Theorem 6.1. Hoeffding's bound and the definition of the roll-out length shows that for any $(s,a) \in \mathcal{C}$, with probability at least $1-\alpha$,

$$\left|\hat{Q}_i(s,a) - Q^{\pi_i}(s,a)\right| \le \frac{1}{1-\gamma}\sqrt{\frac{1}{2m}\log\left(\frac{2}{\alpha}\right)} + \varepsilon = 2\varepsilon \,.$$

At the end we analyse the failure probability of the algorithm, but for now assume the above inequality holds for all $i \le k$ and $(s,a) \in \mathcal{C}$. Let $\theta_i = \arg\min_\theta \|Q^{\pi_i} - \Phi\theta\|_\infty$. Then, by Proposition 4.5 with $\beta = 2\varepsilon$,

$$\|Q_i - Q^{\pi_i}\|_\infty = \|\Phi\hat{\theta}_i - Q^{\pi_i}\|_\infty \le 3\varepsilon\sqrt{2d} + \varepsilon \doteq \delta \,.$$

Since the rewards belong to the unit interval, taking the maximum norm of both sides in Lemma 6.2 shows that $\|Q^* - Q^{\pi_k}\|_\infty \le 2\delta/(1-\gamma) + \gamma^k/(1-\gamma)$. Then, by the triangle inequality,

$$\begin{aligned}\|Q_k - Q^*\|_\infty &\le \|Q_k - Q^{\pi_k}\|_\infty + \|Q^* - Q^{\pi_k}\|_\infty \\ &\le \frac{3\delta}{1-\gamma} + \frac{\gamma^k}{1-\gamma} \,.\end{aligned}$$

Next, by Lemma 6.3, for any state $s \in [S]$,

$$\begin{aligned}V^{\pi_{k+1}}(s) &\ge V^*(s) - \frac{2}{1-\gamma}\|Q_k - Q^*\|_\infty \\ &\ge V^*(s) - \frac{2}{(1-\gamma)^2}\left(3\delta + \gamma^k\right) \,.\end{aligned}$$

All that remains is bounding the failure probability, which follows immediately from a union bound over all iterations $i \le k$ and state-action pairs $(s,a) \in \mathcal{C}$. □

7. Conclusions

Are good representations sufficient for efficient learning in bandits or in RL with a generative model? The answer depends on whether one accepts a blowup of the approximation error by a factor of $\sqrt{d}$, and is positive if and only if this blowup is acceptable. The implication is that the role of bias/prior information is more pronounced than in supervised learning where the blowup does not appear. One may wonder whether the usual changes to the learning problem, such as considering sparse approximations, could reduce the blowup. Since sparsity is of little help even in the realisable setting (Lattimore and Szepesvári, 2019, chapter 23), we are only modestly optimistic in this regard. Note also that in reinforcement learning, the blowup is even harsher: in the discounted case we see that a factor of $1/(1-\gamma)^2$ also appears, which we believe is not improvable.

The analysis in both the bandit and reinforcement learning settings can be decoupled into two components. The first is to control the query complexity of identifying a near-optimal action and the second is estimating the value of an action/policy using roll-outs. This view may be prove fruitful when analysing (more) non-linear classes of reward function.

There are many open questions. First, in order to compute an approximate optimal design, the algorithm needs to examine all features. Second, the argument in Section 6 heavily relies on the uniform contraction property of the various operators involved. It remains to be seen whether similar arguments hold for other settings, such as the finite horizon setting or the average cost setting. Another interesting open question is whether a similar result holds for the online setting when the learner needs to control its regret.

Acknowledgements

Csaba Szepesvári gratefully acknowledges funding from the Canada CIFAR AI Chairs Program, Amii and NSERC.

8. Bibliography

Y. Abbasi-Yadkori, D. Pál, and Cs. Szepesvári. Improved algorithms for linear stochastic bandits. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, NIPS, pages 2312–2320. Curran Associates, Inc., 2011.

K. Amin, M. Kearns, and U. Syed. Bandits, query learning, and the haystack dimension. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 87–106, 2011.

J.-Y. Audibert and S. Bubeck. Best arm identification in multi-armed bandits. In *Proceedings of Conference on Learning Theory (COLT)*, 2010.

R. Degenne, W. M. Koolen, and P. Ménard. Non-asymptotic pure exploration by solving games. In *Advances in Neural Information Processing Systems*, pages 14465–14474, 2019.

S. S. Du, S. M. Kakade, R. Wang, and L. F. Yang. Is a good representation sufficient for sample efficient reinforcement learning?, 2019.

E. Even-Dar, S. Mannor, and Y. Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*, 7(Jun):1079–1105, 2006.

A-m. Farahmand, R. Munos, and Cs. Szepesvári. Error propagation for approximate policy and value iteration (extended version). In *NIPS*, 2010.

V. V. Fedorov. Theory of optimal experiments. *Academic Press, New York*, 1972.

A. Ghosh, S. R. Chowdhury, and A. Gopalan. Misspecified linear bandits. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

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

C. Jin, Z. Yang, Z. Wang, and M. I. Jordan. Provably efficient reinforcement learning with linear function approximation. *arXiv preprint arXiv:1907.05388*, 2019.

J. Kiefer and J. Wolfowitz. The equivalence of two extremum problems. *Canadian Journal of Mathematics*, 12(5):363–365, 1960.

C. Lakshminarayanan, S. Bhatnagar, and Cs. Szepesvári. A linearly relaxed approximate linear program for Markov decision processes. *IEEE Transactions on Automatic Control*, 63(4):1185–1191, 2018.

T. Lattimore and Cs. Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2019. draft.

R. Munos. Error bounds for approximate policy iteration. *19th International Conference on Machine Learning*, pages 560–567, 2003.

D. Russo and B. Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, NIPS, pages 2256–2264. Curran Associates, Inc., 2013.

S. P. Singh and R. C. Yee. An upper bound on the loss from approximate optimal-value functions. *Machine Learning*, 16(3):227–233, 1994.

M. Soare, A. Lazaric, and R. Munos. Best-arm identification in linear bandits. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, NIPS, pages 828–836. Curran Associates, Inc., 2014.

Cs. Szepesvári. *Algorithms for Reinforcement Learning*. Morgan & Claypool, 2010.

M. J. Todd. *Minimum-volume ellipsoids: Theory and algorithms*, volume 23. SIAM, 2016.

Ben Van Roy and Shi Dong. Comments on the Du-Kakade-Wang-Yang lower bounds, 2019.

M. J. Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

L. Yang and M. Wang. Sample-optimal parametric Q-learning using linearly additive features. In *International Conference on Machine Learning*, pages 6995–7004, 2019.

A. Zanette, A. Lazaric, M. J. Kochenderfer, and E. Brunskill. Limiting extrapolation in linear approximate value iteration. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5616–5625. Curran Associates, Inc., 2019.

A. Proof of Lemma 2.2

Recall that Lemma 2.2 states that

$$c_1^{\max}(\{e_1,\dots,e_k\}) = (k+1)/2 - 1/k\,.$$

For the upper bound consider, an algorithm that queries each coordinate in a random order, stopping as soon as it receives a nonzero reward, at which point the algorithm can return $\hat{\mu} = \mu$ and $\hat{a} = \arg\max \mu$. Clearly, this algorithm is sound. Let $S \in \mathrm{Perm}([k])$ be a (uniform) random permutation of $[k]$, which represents the order of the algorithm's queries. Fix $i \in [k]$ and let $T = \min\{t \geq 1 : S(t) = i\}$ be the number of queries when $r = e_i$. Note that $T = S^{-1}(i)$ and as such that $\mathbb{P}(T = t) = \mathbb{P}(S(t) = i) = 1/k$ for any $t \in [k]$, i.e., T is uniformly distributed over $\{1,\dots,k\}$. Now consider the slightly improved algorithm, which, using the principle of elimination queries at most $k-1$ items. The runtime of this algorithms on instance $r = e_i$ is $\tilde{T} = \min(T, k-1)$ and we have $\mathbb{E}[\tilde{T}] = \sum_{t=1}^{k-1} t\mathbb{P}(\tilde{T} = t) = \sum_{t=1}^{k-1} t\mathbb{P}(T = t) + (k-1)\mathbb{P}(T = k)$. Plugging in $\mathbb{P}(T = t) = 1/k$ and algebra gives the result.

The proof of the lower bound is based on a similar calculation. First, note that by Yao's principle it suffices to show that there exist a distribution P over the problem instances such that any *deterministic* algorithm needs to ask at least $(k+1)/2 - 1/k$ queries on expectation when the instance that the algorithm runs on is chosen randomly from the distribution. Let P be the uniform distribution. Note that any deterministic algorithm $\mathcal{A}$ can be identified with a fixed sequence $s_1, s_2, \dots \in [k]$ of queries. Call $\mathcal{A}$ dominated if some other algorithm $\mathcal{A}'$ achieves better query complexity on any input instance while the query complexity of $\mathcal{A}'$ is never worse than that of $\mathcal{A}$ on any other instance. Clearly, it suffices to consider nondominated algorithms. Hence, $s_1, s_2, \dots, s_k$ cannot have repeated elements, i.e., $(s_1, s_2, \dots, s_k)$ is a permutation of $[k]$. Further, $\mathcal{A}$ must stop as soon as it receives a nonzero answer or it would be dominated. This implies that the number of queries issued by $\mathcal{A}$ when run on e_i is $\min(k-1, t(i))$. (Since $\mathcal{A}$ is sound and $\delta \leq 1$, $\mathcal{A}$ cannot stop before time $t(i)$ when running on e_i and if stopped later, it would be dominated by the algorithm that stops at time $t(i)$). Since $(s_i)_i$ is a permutation of $[k]$, so is $(t(i))_{i\in[k]}$. Let $I \sim P$, i.e., I is uniformly distributed on $\{1,\dots,k\}$. We have $\mathbb{P}(T = t(I)) = 1/k$ and thus, with the same calculation as before, $\mathbb{E}[\min(k-1, T)] = (k+1)/2 - 1/k$.

B. Proof of Corollary 3.3

By the proof of Lemma 2.2, it should be clear that if $\delta e_1, \dots, \delta e_k \in \mathcal{H}$, then $c_\delta^{\max}(\mathcal{H}) \geq (k+1)/2 - 1/k$. Let

$$k = \left\lceil \exp\left(\frac{d-1}{8}\left(\frac{\varepsilon}{\delta}\right)^2\right)\right\rceil .$$

Then, by Lemma 3.1, there exists a feature matrix $\Phi' \in \mathbb{R}^{k\times d}$ such that for all $a \neq b \in \mathrm{rows}(\Phi')$, $\|a\| = 1$ and $\langle a, b\rangle \leq \varepsilon/\delta$. Let $\mathrm{rows}(\Phi') = \{a_1,\dots,a_k\}$. Then, $\|\delta e_i - \Phi'(\delta a_i)\| \leq \varepsilon$ and hence $\delta e_i \in \mathcal{H}_\Phi^\varepsilon$ for all $i \in [k]$. Thus, $c_\delta^{\max}(\mathcal{H}_\Phi^\varepsilon) \geq (k+1)/2 - 1/k$. The result follows from the definition of k.

C. Proof of Proposition 4.6

Algorithm 2 Optimal estimation algorithm

INPUT: $\Phi \in \mathbb{R}^{k\times d}$ and $q \in [k]$ and $\varepsilon \geq 0$

(1) Find $C \subset [k]$ with $|C| = q$ minimising

$$\max\left\{\|\Phi u\|_\infty : u \in \mathbb{R}^d, \|\Phi_C u\|_\infty \leq 1\right\}$$

(2) Probe μ on C

(3) Find $\hat{\theta} \in \mathbb{R}^d$ such that $\|\Phi_C\hat{\theta} - \mu_C\|_\infty \leq \varepsilon$

(4) Return $\hat{\mu} = \Phi\hat{\theta}$

Upper bound The upper bound is realised by Algorithm 2. Since $\mu \in \mathcal{H}_\Phi^\varepsilon$, there exists a $\theta \in \mathbb{R}^d$ and $\Delta \in B_\infty(\varepsilon)$ such that $\mu = \Phi\theta + \Delta$. By the definition of the algorithm, $\|\hat{\mu}_C - \mu_C\|_\infty \leq \varepsilon$, which implies that

$$\begin{aligned}\|\Phi_C(\theta - \hat{\theta})\|_\infty &= \|\mu_C - \Delta_C - \hat{\mu}_C\|_\infty \\ &\leq \varepsilon + \|\mu_C - \hat{\mu}_C\|_\infty \\ &\leq 2\varepsilon\,.\end{aligned}$$

Next, using the definition of λ_q,

$$\begin{aligned}\|\mu - \hat{\mu}\|_\infty &= \|\Phi(\theta - \hat{\theta}) + \Delta\|_\infty \\ &\leq \|\Phi(\theta - \hat{\theta})\|_\infty + \varepsilon \\ &\leq \|\Phi_C(\theta - \hat{\theta})\|_\infty \max_{v\in\mathbb{R}^d\setminus\{0\}} \frac{\|\Phi v\|_\infty}{\|\Phi_C v\|_\infty} + \varepsilon \\ &\leq 2\varepsilon\lambda_q(\Phi) + \varepsilon \\ &< \delta_2\,.\end{aligned}$$

Lower bound Suppose an algorithm is sound with respect to $(\mathcal{H}_\Phi^\varepsilon, \delta_1)$. It suffices to show that there exists a $\mu \in \mathcal{H}_\Phi^\varepsilon$ such that whenever the algorithm halts with non-zero probability, it has made more than q queries. Let $\mu = \mathbf{0}$ and suppose the algorithm halts having queried μ on $C \subset [k]$ with non-zero probability and $|C| \le q$. Let $\mathcal{R} = \{\mu \in \mathcal{H}_\Phi^\varepsilon : \mu_C = \mathbf{0}\}$ be the set of plausible rewards consistent with the observation. By the assumption that the algorithm is sound with respect to $(\mathcal{H}_\Phi^\varepsilon, \delta_1)$, it must be that

$$2 \max_{\nu \in \mathcal{R}} \|\nu\|_\infty \le \max_{\nu,\xi \in \mathcal{R}} \|\nu - \xi\|_\infty \le 2\delta_1 \,,$$

where the first inequality is true since for all $\nu \in \mathcal{R}$ we have $-\nu \in \mathcal{R}$. Then,

$$\begin{aligned}
&\max_{\nu \in \mathcal{R}} \|\nu\|_\infty \\
&= \max\left\{\|\Phi\theta + \Delta\|_\infty : \|\Phi_C \theta\|_\infty \le \varepsilon, \theta \in \mathbb{R}^d, \Delta \in B_\infty(\varepsilon)\right\} \\
&= \varepsilon + \max\left\{\|\Phi\theta\|_\infty : \|\Phi_C\theta\|_\infty \le \varepsilon, \theta \in \mathbb{R}^d\right\} \\
&\ge \varepsilon + \varepsilon \lambda_q(\Phi) \\
&\ge \delta_1 \,,
\end{aligned}$$

which contradicts soundness and hence $|C| > q$.

D. Details for proof of Proposition 5.1

The proof is completed in two steps. First we summarise what has already been established about the within-episode behaviour of the algorithm. Then, in the second step, the regret is summed over the episodes.

In-episode behaviour Let $\mathscr{F}$ be the σ-algebra generated by the history up to the start of a given episode and $\hat{\mu}$ be the least-squares estimate of μ computed by the algorithm in that episode. Similarly, let m, u and $\mathcal{A}$ be the quantities defined in the given episode of the algorithm. Let $a^* = \arg\max_{a \in \mathcal{A}} \mu_a$ and $\hat{a} = \arg\max_{a \in \mathcal{A}} \hat{\mu}_a$. Now, for $b \in \mathcal{A}$, let $\mathcal{E}_b$ be the event

$$\mathcal{E}_b = \left\{ \left| \langle b, \hat{\theta} - \theta \rangle \right| \le 2\varepsilon\sqrt{d} + \sqrt{\frac{4d}{m} \log\left(\frac{1}{\alpha}\right)} \right\} .$$

We have shown that $\mathbb{P}(\mathcal{E}_b \mid \mathscr{F}) \ge 1 - \alpha$, which by a union bound implies that

$$\mathbb{P}(\cup_{b \in \mathcal{A}} \mathcal{E}_b \mid \mathscr{F}) \ge 1 - k\alpha \,.$$

By the definition of the algorithm, an action $a \in \mathcal{A}$ is not eliminated at the end of the episode if

$$\langle \hat{\theta}, \hat{a} - a \rangle \le 2\sqrt{\frac{4d}{m} \log\left(\frac{1}{\delta}\right)} \,,$$

which implies that

$$\begin{aligned}
2\sqrt{\frac{4d}{m} \log\left(\frac{1}{\delta}\right)} &\ge \langle \hat{\theta}, \hat{a} - a \rangle \\
&\ge \langle \theta, a^* - a \rangle - 2\sqrt{\frac{4d}{m} \log\left(\frac{1}{\alpha}\right)} - 4\varepsilon\sqrt{d} \,.
\end{aligned}$$

Hence, if $a \in \mathcal{A}$ is not eliminated, then

$$\begin{aligned}
\mu_a &\ge \langle a, \theta \rangle - \varepsilon \\
&\ge \langle a^*, \theta \rangle - \varepsilon - 4\sqrt{\frac{4d}{m} \log\left(\frac{1}{\alpha}\right)} - 4\varepsilon\sqrt{d} \\
&\ge \mu_{a^*} - \varepsilon - 4\sqrt{\frac{4d}{m} \log\left(\frac{1}{\alpha}\right)} - 4\varepsilon\sqrt{d} \,.
\end{aligned}$$

Because the condition for eliminating arms does not depend on ε, it is not possible to prove that a^* is not eliminated with high probability. What we can show is that at least one near-optimal action is retained. Suppose that a^* is eliminated, then, using the definition of the algorithm,

$$\begin{aligned}
2\sqrt{\frac{4d}{m} \log\left(\frac{1}{\alpha}\right)} &< \langle \hat{\theta}, \hat{a} - a^* \rangle \\
&\le \langle \theta, \hat{a} - a^* \rangle + 2\sqrt{\frac{4d}{m} \log\left(\frac{1}{\alpha}\right)} + 4\varepsilon\sqrt{d} \,.
\end{aligned}$$

Rearranging shows that

$$\begin{aligned}
\mu_{\hat{a}} &\ge \langle \theta, \hat{a} \rangle - \varepsilon \\
&> \langle \theta, a^* \rangle - \varepsilon - 4\varepsilon\sqrt{d} \\
&\ge \mu_{a^*} - 2\varepsilon(1 + 2\sqrt{d}) \,. \qquad (3)
\end{aligned}$$

Of course, $\hat{a}$ is not eliminated, which means that either a^* is retained, or an action with nearly the same reward is. What we have shown is that arms are eliminated if they are much worse than a^* and that some arm with mean close to a^* is retained. We now combine these results.

Combining the episodes Let L be the number of episodes and δ_ℓ be the suboptimality of the best arm in the active set at the start of episode ℓ. By the previous part, with probability at least $1 - k\alpha L$, the good events occur in all episodes. Suppose for a moment that this happens. Then, by Eq. (3),

$$\delta_\ell \le 2\varepsilon(\ell - 1)(1 + 2\sqrt{d}) \,.$$

Then, letting $m_\ell = 2^{\ell-1}\lceil 4d\log\log(d)+16\rceil$, the regret is bounded by

$$\begin{aligned}
R_n &= \sum_{\ell=2}^{L}\sum_{a\in\mathcal{A}_\ell} u_\ell(a)(\mu^* - \mu_a) \\
&\le m_1 + \sum_{\ell=2}^{L} m_\ell\left[\delta_\ell + 2\sqrt{\frac{4d}{m_{\ell-1}}\log\left(\frac{1}{\alpha}\right)} + 4\varepsilon\sqrt{d}\right] \\
&\le C\left[\sqrt{dm_L\log\left(\frac{1}{\alpha}\right)} + \varepsilon m_L\log(m_L)\sqrt{d}\right] \\
&\le C\left[\sqrt{dn\log\left(\frac{1}{\alpha}\right)} + \varepsilon n\sqrt{d}\log(n)\right] .
\end{aligned}$$

The result follows because the regret due to failing confidence intervals is at most $\alpha knL \le L \le \log_2(n)$, which is negligible relative to the above term.

Remark D.1. When ε is known, the elimination condition can be changed to

$$\mathcal{A} \leftarrow \left\{a\in\mathcal{A} : \max_{b\in\mathcal{A}} \frac{\langle\hat\theta, b-a\rangle}{4} \le \sqrt{\frac{d}{m}\log\left(\frac{1}{\delta}\right)} + \varepsilon\sqrt{d}\right\} .$$

Repeating the analysis now shows that the optimal action is never eliminated with high probability, which eliminates the logarithmic dependence in the second term of Proposition 5.1.

E. Linear contextual bandits

In the contextual version of the misspecified linear bandit problem, the feature matrix changes from round to round. Let $(k_t)_{t=1}^n$ be a sequence of natural numbers. At the start of round t the learner observes a matrix $\Phi_t \in \mathbb{R}^{k_t\times d}$, chooses an action $X_t \in \mathrm{rows}(\Phi_t)$ and receives a reward $Y_t = \langle X_t, \theta\rangle + \eta_t + \Delta(X_t)$ where $\Delta : \mathbb{R}^d \to \mathbb{R}$ satisfies $\|\Delta\|_\infty \le \varepsilon$. Elimination algorithms are not suitable for such problems. Here we show that if ε is known, then a simple modification of LinUCB (Abbasi-Yadkori et al., 2011) can be effective. You might wonder whether or not this algorithm works well without modification. The answer is sadly negative.

Let $G_t = I + \sum_{s=1}^t X_s X_s^\top$ and define the regularised least-squares estimator based on data from the first t rounds by

$$\hat\theta_t = G_t^{-1}\sum_{s=1}^t X_s Y_s .$$

Assume for all $a \in \cup_{t=1}^n \mathrm{rows}(\Phi_t)$ that $\|a\|_2 \le 1$ and $|\langle a,\theta\rangle| \le 1$. The standard version of LinUCB chooses

$$X_{t+1} = \operatorname*{arg\,max}_{a\in\mathrm{rows}(\Phi_{t+1})} \langle a, \hat\theta_t\rangle + \|a\|_{G_t^{-1}}\beta_t , \tag{4}$$

$$\text{with } \beta_t = 1 + \sqrt{2\log(n) + d\log\left(1+\frac{n}{d}\right)} . \tag{5}$$

The modification chooses

$$X_{t+1} = \operatorname*{arg\,max}_{a\in\mathrm{rows}(\Phi_{t+1})} \langle a, \hat\theta_t\rangle + \|a\|_{G_t^{-1}}\beta_t + \varepsilon\sum_{s=1}^t |a^\top G_t^{-1} X_s| . \tag{6}$$

This modification is reminiscent of the algorithm by Jin et al. (2019), who use a bonus of $\Omega(t^{1/2})$ when using upper confidence bounds for least-squares estimators for linear dynamics in reinforcement learning.

Theorem E.1. *The regret of the algorithm defined by Eq. (6) satisfies*

$$R_n = O\left(d\sqrt{n}\log(n) + n\varepsilon\sqrt{d\log(n)}\right) .$$

Proof sketch. The main point is that the additional bonus term ensures optimism. Then, the standard regret calculation shows that

$$R_n = O\left(d\sqrt{n}\log(n) + \varepsilon\mathbb{E}\left[\sum_{t=1}^n\sum_{s=1}^{t-1} |X_t^\top G_{t-1}^{-1} X_s|\right]\right) .$$

The latter term is bounded by

$$\begin{aligned}
\sum_{t=1}^n\sum_{s=1}^{t-1} |X_t^\top G_{t-1}^{-1} X_s| &\le n\sqrt{\sum_{t=1}^n\sum_{s=1}^{t-1} (X_t^\top G_{t-1}^{-1} X_s)^2} \\
&\le n\sqrt{\sum_{t=1}^n \|X_t\|^2_{G_{t-1}^{-1}}} \\
&= O\left(n\sqrt{d\log(n)}\right) .
\end{aligned}$$

Hence the regret of this algorithm satisfies

$$R_n = O\left(d\sqrt{n}\log(n) + \varepsilon n\sqrt{d\log(n)}\right) . \qquad \square$$

Remark E.2. As far as we know, there is no algorithm obtaining a similar bound when ε is unknown.

Failure of unmodified algorithm That the algorithm defined by Eq. (5) is not good for contextual bandits follows from the following example. Let $\eta_t = 0$ for all rounds t and

$$\Phi_{\text{odd}} = \begin{pmatrix}\varepsilon & 0\end{pmatrix} \quad \Phi_{\text{even}} = \begin{pmatrix}0 & \varepsilon\end{pmatrix} \quad \Phi_{\text{large}} = \begin{pmatrix}2 & 1\\ 0 & 0\end{pmatrix} .$$

Now suppose for odd rounds $t \le n/2$ the feature matrix is $\Phi_t = \Phi_{\text{odd}}$ in odd rounds and $\Phi_t = \Phi_{\text{even}}$ in even rounds.

For rounds $t > n/2$ the feature matrix is Φ_{large}. Then let $\theta = (1/2, -1/2)$ and $\Delta((\varepsilon, 0)) = -\varepsilon$ and $\Delta((0, \varepsilon)) = \varepsilon$. Hence for $t = n/2$,

$$G_t = \begin{pmatrix} 1 + n\varepsilon^2/4 & 0 \\ 0 & 1 + n\varepsilon^2/4 \end{pmatrix}$$
$$\hat{\theta}_t = \left(-\frac{n\varepsilon^2/8}{1 + n\varepsilon^2/4}, \frac{n\varepsilon^2/8}{1 + n\varepsilon^2/4} \right) .$$

Therefore $\|(2,1)\|^2_{G_t^{-1}} \leq 20/(n\varepsilon^2)$ and

$$\langle \hat{\theta}_t, (2,1) \rangle = -\frac{n\varepsilon^2/8}{1 + n\varepsilon^2/4} \leq \frac{4}{n\varepsilon^2} - 1 .$$

Hence

$$\langle \hat{\theta}_t, (2,1) \rangle + \|(2,1)\|_{G_t^{-1}} \beta_t = -1 + O\left(\sqrt{\frac{d \log(n)}{n\varepsilon^2}} \right)$$

and this for every suitably large n the algorithm will choose $(0,0)$ for all rounds $t \geq n/2$ and suffer regret at least $n/2$. Thus, if $R_n(\varepsilon)$ is the regret on the above problem,

$$\sup_{n, \varepsilon > 0} \frac{R_n(\varepsilon)}{\varepsilon n \sqrt{\log(n)}} = \infty ,$$

while for the modified algorithm,

$$\sup_{n, \varepsilon > 0} \frac{R_n(\varepsilon)}{\varepsilon n \sqrt{\log(n)}} < +\infty .$$

F. Lower bounds for linear bandits

The upper bound in Section 5 cannot be improved in the most interesting regimes, as the following theorem shows:

Theorem F.1. *There exists a feature matrix $\Phi \in \mathbb{R}^{k \times d}$ such that for any algorithm there is a mean reward vector $\mu \in \mathcal{H}^{\varepsilon}_{\Phi}$ for which*

$$R_n \geq \varepsilon \min(n, (k-1)/2) \sqrt{\frac{d-1}{8 \log(k)}} .$$

Proof. By the negative result, we may choose $\Phi \in \mathbb{R}^{k \times d}$ such that

$$\langle a, a \rangle = 1 \text{ for all } a \in \text{rows}(\Phi)$$
$$\langle a, b \rangle \leq \sqrt{\frac{8 \log(k)}{d-1}} \text{ for all } a, b \in \text{rows}(\Phi) \text{ with } a \neq b .$$

Next, let $a^* \in \text{rows}(\Phi)$ and

$$\theta = \delta a^* \qquad \text{with } \delta = \sqrt{\frac{d-1}{8 \log(k)}} ,$$

which is chosen so that $\mu \in \mathcal{H}^{\varepsilon}_{\Phi}$, where

$$\mu_a = \begin{cases} \delta & \text{if } a = a^* \\ 0 & \text{otherwise} . \end{cases}$$

Let $\tau = \max\{t \leq n : A_s \neq a^* \forall s \leq t\}$. Then $\mathbb{E}[R_n] \geq \delta \mathbb{E}[\tau]$. Since the law of the rewards is independent of a^* for $t \leq \tau$, it follows from the randomisation hammer that $\mathbb{E}[\tau] \geq \min(n, (k-1)/2)$ and the result follows. □

G. Computation complexity

We briefly discuss the computation complexity of our algorithms here. Both the bandit and RL algorithms rely on computing a near-optimal design, which is addressed first.

Computing a near-optimal design The standard method for computing a near-optimal design is Frank–Wolfe, which in this setting is often attributed to Fedorov (1972). With this algorithm and an appropriate initialisation constant factor approximation of the optimal design can be computed in $O(kd^2 \log\log(d))$ computations. For more details we recommend chapter 3 of the book by Todd (2016), which also describes a number of improvements, heuristics and practical guidance.

Bandit algorithm computations Algorithm 1 has at most $O(\log(n))$ episodes. In each episode it needs to (a) compute a near-optimal design and (b) collect data and find the least-squares estimator and (c) perform action elimination. The computation is dominated by finding the near optimal design and computing the covariance matrix G, which leads to a total computation of $O(kd^2 \log\log(d) \log(n) + nd^2)$.

RL computations The algorithm described in Section 6 operates in episodes over k episodes. In each episode it computes an approximate design and performs m roll-outs of length n from each action in the core set. Assuming sampling from the generative model is $O(1)$, the total computation, ignoring logarithmic factors, is

$$\tilde{O}\left(\frac{dA}{\varepsilon^2 (1-\gamma)^4} + \frac{SAd^2}{1-\gamma} \right) .$$

Dishearteningly, the size of the state space appears in the computation of the optimal design. Hence, while the sample complexity of our algorithm is independent of the state space, the computation complexity is not.