Learning Infinite-horizon Average-reward MDPs with Linear Function Approximation

Chen-Yu Wei chenyu.wei@usc.edu
Mehdi Jafarnia-Jahromi mjafarni@usc.edu
Haipeng Luo haipengl@usc.edu
Rahul Jain rahul.jain@usc.edu
University of Southern California

Abstract

We develop several new algorithms for learning Markov Decision Processes in an infinite-horizon average-reward setting with linear function approximation. Using the optimism principle and assuming that the MDP has a linear structure, we first propose a computationally inefficient algorithm with optimal $\widetilde{\mathcal{O}}(\sqrt{T})$ regret and another computationally efficient variant with $\widetilde{\mathcal{O}}(T^{\frac{3}{4}})$ regret, where T is the number of interactions. Next, taking inspiration from adversarial linear bandits, we develop yet another efficient algorithm with $\widetilde{\mathcal{O}}(\sqrt{T})$ regret under a different set of assumptions, improving the best existing result by Hao et al. (2021) with $\widetilde{\mathcal{O}}(T^{\frac{2}{3}})$ regret. Moreover, we draw a connection between this algorithm and the Natural Policy Gradient algorithm proposed by Kakade (2002), and show that our analysis improves the sample complexity bound recently given by Agarwal et al. (2020).

1 Introduction

Reinforcement learning with value function approximation has gained significant empirical success in many applications. However, the theoretical understanding of these methods is still quite limited. Recently, some progress has been made for Markov Decision Processes (MDPs) with a transition kernel and a reward function that are both linear in a fixed state-action feature representation (or more generally with a value function that is linear in such a feature representation). For example, Jin et al. (2020) develop an optimistic variant of the Least-squares Value Iteration (LSVI) algorithm (Bradtke and Barto, 1996; Osband et al., 2016) for the finite-horizon episodic setting with regret $\widetilde{\mathcal{O}}(\sqrt{d^3T})$, where d is the dimension of the features and T is the number of interactions. Importantly, the bound has no dependence on the number of states or actions.

However, the understanding of function approximation for the *infinite-horizon average-reward* setting, even under the aforementioned linear conditions, remains underexplored. Compared to the finite-horizon setting, the infinite-horizon model is often a better fit for real-world problems such as server operation optimization or stock market decision making which last for a long time or essentially never end. On the other hand, compared to the discounted-reward model, maximizing the long-term average reward also has its advantage in the sense that the transient behavior of the learner does not really matter for the latter case. Indeed, the infinite-horizon average-reward setting for the tabular case (that is, no function approximation) is a heavily-studied topic in the literature. Several recent works start to investigate function approximation for this setting, albeit under strong assumptions (Abbasi-Yadkori et al., 2019a,b; Hao et al., 2021).

Motivated by this fact, in this work we significantly expand the understanding of learning MDPs in the infinite-horizon average-reward setting with linear function approximation. We develop three new algorithms, each with different pros and cons. Our first two algorithms provably ensure low regret for MDPs with linear transition and reward, which are the *first* for this setting to the best of our knowledge. More specifically, the first algorithm Fixed-point OPtimization with Optimism (FOPO) is based on the principle of "optimism in the face of uncertainty" applied in a novel way. FOPO aims to find a weight vector (parametrizing the estimated value function) that maximizes the average reward under a fixed-point constraint akin to the LSVI update involving the observed data and an optimistic term. The constraint is non-convex and we do not know of a way to efficiently solve it. FOPO also relies on a lazy update schedule similar to (Abbasi-Yadkori et al., 2011) for stochastic linear bandits, which is only for the purpose of saving computation in their work but critical for our regret

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130.

Table 1: Summary of our results and comparisons to prior work. Our first two algorithms are the first results for infinite-horizon average-reward MDPs under Assumptions 1 and 2, while our third algorithm improves over the best existing results in a setting with a different set of assumptions. These two set of assumptions are incomparable, in the sense that Assumption 1 is weaker than Assumptions 3 and 5, while Assumption 2 is stronger than Assumption 4.

Algorithm	**Regret**	**Assumptions**	
		Explorability	Structure
FOPO (Algorithm 1)	$\widetilde{\mathcal{O}}(\sqrt{T})$	Bellman optimality equation (Assumption 1)	linear MDP (Assumption 2)
OLSVI.FH (Algorithm 2)	$\widetilde{\mathcal{O}}(T^{\frac{3}{4}})$		
MDP-EXP2 (Algorithm 3)	$\widetilde{\mathcal{O}}(\sqrt{T})$	uniform mixing (Assumption 3) uniformly excited features (Assumption 5)	linear bias function (Assumption 4)
Politex (Abbasi-Yadkori et al., 2019a)	$\widetilde{\mathcal{O}}(T^{\frac{3}{4}})$		
AAPI (Hao et al., 2021)	$\widetilde{\mathcal{O}}(T^{\frac{2}{3}})$		

guarantee. We prove that FOPO enjoys $\widetilde{\mathcal{O}}(\sqrt{d^3T})$ regret with high probability, which is optimal in T. (Section 2)

Our second algorithm OLSVI.FH addresses the computational inefficiency issue of FOPO with the price of having larger regret. Specifically, it combines two ideas: 1) solving an infinite-horizon problem via an artificially constructed finite-horizon problem, which is new as far as we know, and 2) the optimistic LSVI algorithm of Jin et al. (2020) for the finite-horizon setting. OLSVI.FH can be implemented efficiently and is shown to achieve $\widetilde{\mathcal{O}}((dT)^{\frac{3}{4}})$ regret. (Section 3)

Our third algorithm MDP-EXP2 takes a very different approach and is inspired by another algorithm called MDP-OOMD from Wei et al. (2020). MDP-OOMD runs a particular adversarial multi-armed bandit algorithm for each state to obtain $\widetilde{\mathcal{O}}(\sqrt{T})$ regret (ignoring dependence on other parameters) for the tabular case under an ergodic assumption. We generalize the idea and apply a particular *adversarial linear bandit* algorithm known as EXP2 (Dani et al., 2008; Bubeck et al., 2012) for each state (only conceptually — the algorithm can still be implemented efficiently). Under the same set of assumptions made in Hao et al. (2021) (which does not necessarily require linear transition and reward), we improve their regret bound from $\widetilde{\mathcal{O}}(T^{\frac{2}{3}})$ to $\widetilde{\mathcal{O}}(\sqrt{T})$. In Appendix F, we also describe the connection of this algorithm with the Natural Policy Gradient algorithm proposed by Kakade (2002), whose sample complexity bound is recently formalized by Agarwal et al. (2020). We argue that under the setting considered in Section 4, their analysis translates to a sub-optimal regret bound of $\widetilde{\mathcal{O}}(T^{\frac{3}{4}})$, and that our improvement over theirs comes from the way we construct the gradient estimates.

We summarize our results and the comparisons to previous work in Table 1.

Related work. For the tabular case with finite state and action space in the infinite-horizon average-reward setting, the works (Bartlett and Tewari, 2009; Jaksch et al., 2010) are among the first to develop algorithms with provable sublinear regret. Over the years, numerous improvements have been proposed, see for example (Ortner, 2020; Fruit et al., 2018; Talebi and Maillard, 2018; Fruit et al., 2020; Zhang and Ji, 2019; Wei et al., 2020). In particular, the recent work of Wei et al. (2020) develops two model-free algorithms for this problem. We refer the reader to (Wei et al., 2020, Table 1) for comparisons of existing algorithms. As mentioned, our algorithm MDP-EXP2 is inspired by the MDP-OOMD algorithm of Wei et al. (2020). Also note that their Optimistic Q-learning algorithm reduces an infinite-horizon average-reward problem to a discounted-reward problem. For technical reasons, we are not able to generalize this idea to the linear function approximation setting (see Section 3.2). Instead, our OLSVI.FH reduces the problem to a finite-horizon version, which is new to the best of our knowledge and might be of independent interest.

The work of Chen et al. (2018) considers learning in infinite-horizon average-reward MDPs with linear function approximation, under the assumption that the learner has access to a sampling oracle from which the learner can sample states and actions under any given distribution. The assumptions they make for the MDP is similar to the ones in our Section 4, and the sample complexity bound they obtain is $\widetilde{\mathcal{O}}\left(1/\epsilon^2\right)$. However, since the oracle assumption is rather strong, it is not clear how to extend their algorithm to the online setting.

The works of Abbasi-Yadkori et al. (2019a,b); Hao et al. (2021) are among the first to consider the infinite-horizon average-reward setting with function approximation and provable regret guarantees in the online setting. Their results all depend on some uniformly mixing and uniformly excited feature conditions. As mentioned, under the same assumption, our MDP-EXP2 algorithm with $\widetilde{\mathcal{O}}(\sqrt{T})$ regret

improves the best existing result by Hao et al. (2021) with $\widetilde{\mathcal{O}}(T^{\frac{2}{3}})$ regret. Moreover, our other two algorithms ensure low regret for linear MDPs without these extra assumptions, which do not appear before.

Provable function approximation has gained growing research interest in other settings as well (finite-horizon or discounted-reward). See recent works (Liu et al., 2019; Wang et al., 2019; Yang and Wang, 2020; Jin et al., 2020; Zanette et al., 2020; Dong et al., 2020; Wang et al., 2020) for example. In particular, our FOPO algorithm shares some similarity with the algorithm of Zanette et al. (2020), which also relies on solving an optimization problem under a constraint akin to LSVI, with no efficient implementation.

Adversarial linear bandit is also known as bandit linear optimization. The EXP2 algorithm (Bubeck et al., 2012), on top of which our MDP-EXP2 algorithm is built, is also known as Geometric Hedge (Dani et al., 2008) or ComBand (Cesa-Bianchi and Lugosi, 2012) in the literature. A concurrent work by Neu and Olkhovskaya (2020) proposes an algorithm called MDP-LINEXP3 for the linear function approximation setting that is also based on the adversarial linear bandit framework. However, their result is incomparable to ours because they focus on finite-horizon MDPs with adversarial reward, and they assume that the learner has access to a sampling oracle.

2 Preliminaries

We consider infinite-horizon average-reward Markov Decision Processes (MDPs) described by $(\mathcal{X}, \mathcal{A}, r, p)$ where $\mathcal{X}$ is a Borel state space with possibly infinite number of elements, $\mathcal{A}$ is a finite action set, $r : \mathcal{X} \times \mathcal{A} \to [-1, 1]$ is the (unknown) reward function, and $p(\cdot|x, a)$ is the (unknown) transition kernel induced by x, a, satisfying $\int_{\mathcal{X}} p(\mathrm{d}x'|x, a) = 1$ (following integral notation from Hernández-Lerma (2012)).

The learning protocol is as follows. A learner interacts with the MDP through T steps, starting from an arbitrary initial state $x_1 \in \mathcal{X}$. At each step t, the learner decides an action a_t, and then observes the reward $r(x_t, a_t)$ as well as the next state x_{t+1} which is a sample drawn from $p(\cdot|x_t, a_t)$. The goal of the learner is to be competitive against any fixed stationary policy. Specifically, a stationary policy is a mapping $\pi : \mathcal{X} \to \Delta_{\mathcal{A}}$ with $\pi(a|x)$ specifying the probability of selecting action a at state x. The long-term average reward of a stationary policy π starting from state $x \in \mathcal{X}$ is naturally defined as:

$$J^\pi(x) \triangleq \liminf_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} r(x_t, a_t) \;\middle|\; x_1 = x,\ \ \forall t \geq 1,\ a_t \sim \pi(\cdot|x_t),\ x_{t+1} \sim p(\cdot|x_t, a_t)\right].$$

The performance measure of the learner, known as regret, is then defined as $\mathrm{Reg}_T := \max_\pi \sum_{t=1}^{T}(J^\pi(x_1) - r(x_t, a_t))$, which is the difference between the total rewards of the best stationary policy and that of the learner.

However, in contrast to the finite-horizon episodic setting where ensuring sublinear regret is always possible, it is known that in our setting a necessary condition is that the optimal policy has a long-term average reward that is independent of the initial state (Bartlett and Tewari, 2009). To this end, throughout the paper we only consider a broad subclass of MDPs where a certain form of Bellman optimality equation holds (Hernández-Lerma, 2012):

Assumption 1 (Bellman optimality equation)**.** *There exist $J^* \in \mathbb{R}$ and bounded measurable functions $v^* : \mathcal{X} \to \mathbb{R}$ and $q^* : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ such that the following holds for all $x \in \mathcal{X}$ and $a \in \mathcal{A}$:*

$$\begin{aligned} J^* + q^*(x, a) &= r(x, a) + \mathbb{E}_{x'\sim p(\cdot|x,a)}[v^*(x')], \\ v^*(x) &= \max_{a\in\mathcal{A}} q^*(x, a). \end{aligned} \tag{1}$$

Indeed, under this assumption, the claim is that a policy π^* that deterministically selects an action from $\operatorname{argmax}_a q^*(x, a)$ at each state x is the optimal policy, with $J^{\pi^*}(x) = J^*$ for all x. To see this, note that for any policy π, using the Bellman optimality equation we have

$$\begin{aligned} J^\pi(x) &= \liminf_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\left(J^* + \sum_{a\in\mathcal{A}} q^*(x_t, a)\cdot\pi(a|x_t) - v^*(x_{t+1})\right)\right] \\ &\leq \liminf_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}(J^* + v^*(x_t) - v^*(x_{t+1}))\right] = J^*, \end{aligned}$$

with equality attained by π^*, proving the claim. Consequently, under Assumption 1 we simply write the regret as $\mathrm{Reg}_T := \sum_{t=1}^{T}(J^* - r(x_t, a_t))$.

All existing works on regret minimization for infinite-horizon average-reward MDPs make this assumption, either explicitly or through even stronger assumptions which imply this one. In the tabular case with a finite state space, weakly communicating MDPs is the broadest class to study regret minimization in the literature, and is known to satisfy Assumption 1 (see (Puterman, 2014)). More generally, Assumption 1 holds under many other common conditions; see (Hernández-Lerma, 2012, Section 3.3).

Note that $v^*(x)$ and $q^*(x, a)$ quantify the *relative advantage* of starting with x and starting with (x, a) respectively and then acting optimally in the MDP. Therefore, v^* is sometimes called the *state bias function* and q^* is called the *state-action bias function*.

For a bounded function $v : \mathcal{X} \to \mathbb{R}$, we define its span as $\mathrm{sp}(v) \triangleq \sup_{x,x'\in\mathcal{X}} |v(x) - v(x')|$. Notice that if (v^*, q^*) is

a solution of Eq. (1), then a translated version (v^*-c, q^*-c) for any constant c is also a solution. In the remaining of the paper, we let (v^*, q^*) be an arbitrary solution pair of Eq. (1) with a small span $\mathrm{sp}(v^*)$ in the sense that $\mathrm{sp}(v^*) \le 2\,\mathrm{sp}(v')$ for any other solution (v', q'). We also assume without loss of generality $|v^*(x)| \le \frac{1}{2}\,\mathrm{sp}(v^*)$ for any x because we can perform the above translation and center the values of v^* around zero. Similarly to previous works (e.g. (Wei et al., 2020)), $\mathrm{sp}(v^*)$ is assumed to be known to the learner.

3 Optimism-based Algorithms

In this section, we present two optimism-based algorithms with sublinear regret, under only one extra assumption that the MDP is *linear* (also known as low-rank MDPs). We emphasize that earlier works for linear MDPs in the finite-horizon average-reward setting all require extra strong assumptions (Abbasi-Yadkori et al., 2019a,b; Hao et al., 2021).

Specifically, a linear MDP has a transition kernel and a reward function both linear in some state-action feature representation, formally summarized as:

Assumption 2 (Linear MDP)**.** *There exist a known d-dimensional feature mapping $\Phi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$, d unknown measures $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_d)$ over $\mathcal{X}$, and an unknown vector $\boldsymbol{\theta} \in \mathbb{R}^d$ such that for all $x, x' \in \mathcal{X}$ and $a \in \mathcal{A}$,*

$$p(x' \mid x, a) = \Phi(x,a)^\top \boldsymbol{\mu}(x'), \qquad r(x,a) = \Phi(x,a)^\top \boldsymbol{\theta}.$$

Without loss of generality, we further assume that for all $x \in \mathcal{X}$ and $a \in \mathcal{A}$, $\|\Phi(x,a)\| \le \sqrt{2}$, the first coordinate of $\Phi(x,a)$ is fixed to 1, and that $\|\boldsymbol{\mu}(\mathcal{X})\| \le \sqrt{d}$, $\|\boldsymbol{\theta}\| \le \sqrt{d}$, where we use $\boldsymbol{\mu}(\mathcal{X})$ to denote the vector $(\mu_1(\mathcal{X}), \ldots, \mu_d(\mathcal{X}))$ and $\mu_i(\mathcal{X}) \triangleq \int_{\mathcal{X}} \mathrm{d}\mu_i(x)$ is the total measure of $\mathcal{X}$ under μ_i. (All norms are 2-norm.)

In (Jin et al., 2020), the same assumption is made except for a different rescaling: $\|\Phi(x,a)\| \le 1$, $\|\boldsymbol{\mu}(\mathcal{X})\| \le \sqrt{d}$, and $\|\boldsymbol{\theta}\| \le \sqrt{d}$. The reason that this is without loss of generality is not justified in (Jin et al., 2020), and for completeness we prove this in Appendix A. With this scaling, clearly one can augment the feature $\Phi(x,a)$ with a constant coordinate of value 1 and augment $\boldsymbol{\mu}(x)$ and $\boldsymbol{\theta}$ with a constant coordinate of value 0, such that the linear structure is preserved while the scaling specified in Assumption 2 holds.

Under Assumption 2, one can show that the state-action bias function q^* is in fact also linear in the features.

Lemma 1. *Under Assumption 1 and Assumption 2, there exists a fixed weight vector $w^* \in \mathbb{R}^d$ such that $q^*(x,a) = \Phi(x,a)^\top w^*$ for all $x \in \mathcal{X}$ and $a \in \mathcal{A}$, and furthermore, $\|w^*\| \le (2 + \mathrm{sp}(v^*))\sqrt{d}$.*

Based on this lemma, a natural idea emerges: at time t, build an estimator w_t of w^* using observed data, then act according to the estimated long-term reward of each action given by $\Phi(x_t, a)^\top w_t$. While the idea is intuitive, how to construct the estimator and, perhaps more importantly, how to incorporate the optimism principle well known to be important for learning with partial information, are highly non-trivial. In the next two subsections, we describe two different ways of doing so, leading to our two algorithms FOPO and OLSVI.FH.

Algorithm 1 Fixed-point OPtimization with Optimism (FOPO)

Parameters: $0 < \delta < 1$, $\lambda = 1$, $\beta = 20(2 + \mathrm{sp}(v^*))d\sqrt{\log(T/\delta)}$
Initialize: $\Lambda_1 = \lambda I$ where $I \in \mathbb{R}^{d\times d}$ is the identity matrix
for $t = 1, \ldots, T$ **do**
  if $t = 1$ *or* $\det(\Lambda_t) \ge 2\det(\Lambda_{s_{t-1}})$ **then**
    Set $s_t = t$ ▷ s_t records the most recent update
    Let w_t be the solution of the optimization problem:

$$\max_{w_t, b_t \in \mathbb{R}^d, J_t \in \mathbb{R}} \quad J_t$$
$$\text{s.t.} \quad w_t = \Lambda_t^{-1}\sum_{\tau=1}^{t-1}\Big(\Phi(x_\tau,a_\tau)(r(x_\tau,a_\tau) - J_t + v_t(x_{\tau+1})) + b_t\Big) \tag{2}$$
$$q_t(x,a) = \Phi(x,a)^\top w_t, \quad v_t(x) = \max_a q_t(x,a)$$
$$\|b_t\|_{\Lambda_t} \le \beta, \quad \|w_t\| \le (2 + \mathrm{sp}(v^*))\sqrt{d}$$

  else
    $(w_t, J_t, b_t, v_t, q_t, s_t) = (w_{t-1}, J_{t-1}, b_{t-1}, v_{t-1}, q_{t-1}, s_{t-1})$
  Play $a_t = \mathrm{argmax}_a\, q_t(x_t, a)$
  Observe $r(x_t, a_t)$ and x_{t+1}
  Update $\Lambda_{t+1} = \Lambda_t + \Phi(x_t,a_t)\Phi(x_t,a_t)^\top$

3.1 Fixed-point OPtimization with Optimism (FOPO)

We present our first algorithm FOPO which is computationally inefficient but achieves regret $\widetilde{\mathcal{O}}(\mathrm{sp}(v^*)\sqrt{d^3T})$. This is optimal in T since even in the tabular case $\mathcal{O}(\sqrt{T})$ is unimprovable (Jaksch et al., 2010). See Algorithm 1 for the complete pseudocode.

As mentioned, the key part lies in how the estimator w_t is constructed. In Algorithm 1, this is done by solving an optimization problem over certain constraints. To understand the first constraint Eq. (2), recall that $q^*(x,a) = \Phi(x,a)^\top w^*$ satisfies the Bellman optimality equation:

$$\Phi(x,a)^\top w^* = r(x,a) - J^* + \int_{\mathcal{X}} v^*(x')p(\mathrm{d}x' \mid x,a)$$
$$= r(x,a) - J^* + \int_{\mathcal{X}} \Big(\max_{a'} \Phi(x',a')^\top w^*\Big)\, p(\mathrm{d}x' \mid x,a).$$

While p and r are unknown, we do observe samples

$x_1, \ldots, x_{t-1}$ and $r(x_1, a_1), \ldots, r(x_{t-1}, a_{t-1})$. If for a moment we assume J^* was known, then it is natural to try to find w_t such that $\forall \tau = 1, \ldots, t-1$,

$$\Phi(x_\tau, a_\tau)^\top w_t \approx r(x_\tau, a_\tau) - J^* + \max_{a'} \Phi(x_{\tau+1}, a')^\top w_t. \tag{3}$$

In common variants of Least-squares Value Iteration (LSVI) update, the w_t on the right hand side of Eq. (3) would be replaced with another already computed weight vector w'_t that is either from the last iteration (i.e, w_{t-1}) or from the next layer in the case of episodic MDPs. Then solving a least-squares problem with regularization $\lambda \|w_t\|^2$ gives a natural estimate of w_t:

$$\Lambda_t^{-1} \sum_{\tau=1}^{t-1} \Phi(x_\tau, a_\tau) \left(r(x_\tau, a_\tau) - J^* + \max_{a'} \Phi(x_{\tau+1}, a')^\top w'_t \right)$$

where $\Lambda_t = \lambda I + \sum_{\tau<t} \Phi(x_\tau, a_\tau) \Phi(x_\tau, a_\tau)^\top$ is the empirical covariance matrix. Based on this formula, what we propose in Algorithm 1 are the following three modifications. First, instead of using an already computed weight w'_t, we directly set it back to w_t (and thus $\max_{a'} \Phi(x_{\tau+1}, a')^\top w'_t = v_t(x_{\tau+1})$), making the formula a fixed-point equation now. Second, to incorporate uncertainty, we introduce a slack variable b_t with a bounded quadratic norm $\|b_t\|_{\Lambda_t} \triangleq \sqrt{b_t^\top \Lambda_t b_t} \le \beta$ (for a parameter β) that controls the amount of uncertainty. Last, to deal with the fact that J^* is unknown, we replace it with a variable J_t (arriving at Eq. (2) finally), and apply the well-known principle of *optimism in the face of uncertainty* — we maximize the long-term average reward J_t (over w_t, b_t and J_t) under the aforementioned constraints and also $\|w_t\| \le (2 + \mathrm{sp}(v^*))\sqrt{d}$ in light of Lemma 1.

With the vector w_t and the corresponding bias function q_t, the algorithm simply plays $a_t = \mathrm{argmax}_a \, q_t(x_t, a)$ greedily. Note that w_t is only updated when the determinant of Λ_t doubles compared to that of $\Lambda_{s_{t-1}}$ where s_{t-1} is the time step with the most recent update before time t. This can happen at most $\mathcal{O}(d \log T)$ times. Similar ideas are used in e.g., (Abbasi-Yadkori et al., 2011) for stochastic linear bandits. However, while they use this lazy update only to save computation, here we use it to make sure that w_t does not change too often, which is critical for our regret analysis.

We point out that the closest existing algorithm we are aware of is the one from a recent work (Zanette et al., 2020) for the finite-horizon setting. Just like theirs, our algorithm also does not admit an efficient implementation due to the complicated nature of the optimization problem. However, it can be shown that the constraint set is non-empty with $(w_t, b_t, J_t) = (w^*, b, J^*)$ for some b being a feasible solution (with high probability). This fact also immediately implies that J_t is indeed an optimistic estimator of J^* in the following sense:

Algorithm 2 OLSVI.FH

Parameters: $0 < \delta < 1$, $\lambda = 1$, $\beta = 40dH\sqrt{\log(T/\delta)}$, $H = \max\left\{ \frac{\sqrt{\mathrm{sp}(v^*)} T^{1/4}}{d^{3/4}}, \left(\frac{\mathrm{sp}(v^*) T}{d^2}\right)^{1/3} \right\}$

Initialize: $\Lambda_1 = \lambda I$ where $I \in \mathbb{R}^{d \times d}$ is the identity matrix

Define: $x_h^k = x_t$ and $a_h^k = a_t$, for $t = (k-1)H + h$

for $k = 1, \ldots, T/H$ **do**

  Define $V_{H+1}^k(x) = 0$ for all x.

  for $h = H, \ldots, 1$ **do**

    Compute

$$w_h^k = \Lambda_k^{-1} \sum_{k'=1}^{k-1} \sum_{h'=1}^{H} \Phi(x_{h'}^{k'}, a_{h'}^{k'}) \Big(r(x_{h'}^{k'}, a_{h'}^{k'}) + V_{h+1}^k(x_{h'+1}^{k'}) \Big)$$

    Define

$$\widehat{Q}_h^k(x,a) = w_h^k \cdot \Phi(x,a) + \beta \sqrt{\Phi(x,a)^\top \Lambda_k^{-1} \Phi(x,a)}$$
$$Q_h^k(x,a) = \min\left\{ \widehat{Q}_h^k(x,a), H \right\}$$
$$V_h^k(x) = \max_a Q_h^k(x,a)$$

  for $h = 1, \ldots, H$ **do**

    Play $a_h^k = \mathrm{argmax}_a \, Q_h^k(x_h^k, a)$

    Observe x_h^k and $r(x_h^k, a_h^k)$

  Update $\Lambda_{k+1} = \Lambda_k + \sum_{h=1}^{H} \Phi(x_h^k, a_h^k) \Phi(x_h^k, a_h^k)^\top$

Lemma 2. *With probability at least* $1 - \delta$, *Algorithm 1 ensures* $J_t \ge J^*$ *for all* t.

With the help of this lemma, we prove the following regret bound of FOPO with optimal (in T) rate.

Theorem 3. *Under Assumptions 1 and 2,* FOPO *guarantees with probability at least* $1 - 3\delta$*:*

$$\mathrm{Reg}_T = \mathcal{O}\left(\mathrm{sp}(v^*) \log(T/\delta) \sqrt{d^3 T} \right).$$

3.2 Finite-Horizon Optimistic Least-Square Value Iteration (OLSVI.FH)

Next, we present another optimism-based algorithm which can be implemented efficiently, albeit with a suboptimal regret guarantee. The high-level idea is still based on LSVI. However, since we do not know how to efficiently solve a fixed-point problem as in Algorithm 1, we "open the loop" by solving a finite-horizon problem instead. More specifically, we divide the T rounds into T/H episodes each with H rounds, and run a finite-horizon optimistic LSVI algorithm over the episodes as in (Jin et al., 2020).

The resulted algorithm is shown in Algorithm 2. For simplicity, we replace the time index t with a combination of an *episode index* k and a *step index* h within the episode. This gives the relation $t = (k-1)H + h$, and (x_t, a_t) is written as (x_h^k, a_h^k). At the beginning of each episode k, the learner computes a set of Q-function parameters $w_1^k, \dots, w_H^k$ by backward calculation using all historical data (Line 3 to Line 5). Note that Line 4 is now simply an assignment step (as opposed to a fixed-point problem) since V_{h+1}^k is computed already when in step h. In Line 5, we introduce optimism by incorporating a bonus term $\beta\|\Phi(x,a)\|_{\Lambda_k^{-1}}$ into the definition of $\widehat{Q}_h^k(x,a)$, and hence $Q_h^k(x,a)$. Then in step h of episode k, the learner simply follows the greedy choice suggested by $Q_h^k(x_h^k, \cdot)$ (Line 7).

Note that Algorithm 2 is slightly different from the version in (Jin et al., 2020): they maintain a different covariance matrix Λ_h^k separately for each step h, but we only maintain a single Λ_k for all h. Similarly, their w_h^k is computed using only data related to step h from all previous episodes, while ours is computed using all previous data. This is because in our problem, the steps within an episode share the same transition and reward functions, and consequently they can be learned jointly, which eventually reduces the sample complexity.

Clearly, this reduction ensures that the learner has low regret against the best policy for the finite-horizon problem that we create. However, since our original problem is about average-reward over infinite horizon, we need to argue that the best finite-horizon policy also performs well under the infinite-horizon criteria. Indeed, we show that the sub-optimality gap of the best finite-horizon policy is bounded by some quantity governed by $\mathrm{sp}(v^*)/H$, which is intuitive since the larger H is, the smaller the gap becomes (see Lemma 13).

In our analysis, for a fixed episode we define $\pi = (\pi_1, \dots, \pi_H)$ as the finite-horizon policy (i.e., a length-H sequence of policies), where each π_h is a mapping $\mathcal{X} \to \Delta_{\mathcal{A}}$. For any such finite-horizon policy π, we define $Q_h^\pi(x,a)$ and $V_h^\pi(x)$ as the value functions for the finite-horizon problem we create, which satisfy: $V_{H+1}^\pi(x) = 0$ and for $h = H, \dots, 1$,

$$\begin{aligned} Q_h^\pi(x,a) &= r(x,a) + \mathbb{E}_{x' \sim p(\cdot|x,a)}[V_{h+1}^\pi(x')], \\ V_h^\pi(x) &= \mathbb{E}_{a \sim \pi_h(\cdot|x)} Q_h^\pi(x,a). \end{aligned} \tag{4}$$

The analysis of the algorithm relies on the following key lemma, which shows that $Q_h^k(x,a)$ upper bounds $Q_h^\pi(x,a)$ for any π.

Lemma 4. *With probability at least $1-\delta$, Algorithm 2 ensures for any finite-horizon policy π that $\forall x, a, k, h$.*

$$\begin{aligned} &0 \le Q_h^k(x,a) - Q_h^\pi(x,a) \\ &\le \mathbb{E}_{x' \sim p(\cdot|x,a)} \left[V_{h+1}^k(x') - V_{h+1}^\pi(x') \right] + 2\beta \|\Phi(x,a)\|_{\Lambda_k^{-1}}. \end{aligned}$$

With the help of Lemma 4, we prove the final regret bound of OLSVI.FH stated in the next theorem (proof deferred to the appendix).

Theorem 5. *Under Assumptions 1 and 2,* OLSVI.FH *guarantees with probability at least $1-3\delta$:*

$$\mathrm{Reg}_T = \widetilde{\mathcal{O}}\left(\sqrt{\mathrm{sp}(v^*)} (dT)^{\frac{3}{4}} + (\mathrm{sp}(v^*) dT)^{\frac{2}{3}} \right).$$

Note that although our bound is suboptimal, OLSVI.FH is the first efficient algorithm with sublinear regret for this setting under only Assumptions 1 and 2.

4 The MDP-EXP2 Algorithm

There are two disadvantages of the optimism-based algorithms introduced in the last section. First, they require the transition kernel and reward function to be both linear in the feature (Assumption 2), which is restrictive and might not hold especially when d is small. Second, even for the polynomial-time algorithm OLSVI.FH, it is still computationally intensive because in Line 4 of the algorithm, V_{h+1}^k is applied to all previous states, and every evaluation of V_{h+1}^k requires computing $\|\Phi(x,a)\|_{\Lambda_k}$. Since this is done for every k, the total computational cost of the algorithm is super-linear in T. In fact, all existing optimism-based algorithms with linear function approximation suffer the same issue Yang and Wang (2020); Jin et al. (2020); Zanette et al. (2020).

To this end, we propose yet another algorithm based on very different ideas. It is computationally less intensive and it enjoys $\widetilde{\mathcal{O}}(\sqrt{T})$ regret, albeit under a different (and non-comparable) set of assumptions compared to those in Section 3. Note that these are the same assumptions made in (Abbasi-Yadkori et al., 2019a; Hao et al., 2021). Below, we start with stating these assumptions, followed by the description of our algorithm.

The first assumption we make is that the MDP is uniformly mixing.

Assumption 3 (Uniform Mixing)**.** *There exists a constant $t_{mix} \ge 1$ such that for any policy π, and any distributions $\nu_1, \nu_2 \in \Delta_{\mathcal{X}}$ over the state space,*

$$\|\mathbb{P}^\pi \nu_1 - \mathbb{P}^\pi \nu_2\|_{\mathrm{TV}} \le e^{-1/t_{mix}} \|\nu_1 - \nu_2\|_{\mathrm{TV}},$$

where $(\mathbb{P}^\pi \nu)(x') = \int_{\mathcal{X}} \sum_{a \in \mathcal{A}} \pi(a|x) p(x'|x,a) \mathrm{d}\nu(x)$ and $\|\cdot\|_{\mathrm{TV}}$ is the total variation.

Under this uniform mixing assumption, we are able to define the stationary state distribution under a policy π as $\nu^\pi = (\mathbb{P}^\pi)^\infty \nu_1$ for an arbitrary initial distribution ν_1. Also, now we not only have the Bellman optimality equation (1) (that is, Assumption 3 implies Assumption 1), but also a Bellman equation for every policy π, as shown in the following lemma.

Lemma 6. *Suppose Assumption 3 holds. For any π, its long-term average reward $J^\pi(x)$ is independent of the initial state x, thus denoted as J^π. Also, the following Bellman equation holds:*

$$J^\pi + q^\pi(x,a) = r(x,a) + \mathbb{E}_{x'\sim p(\cdot|x,a)}[v^\pi(x')],$$
$$v^\pi(x) = \sum_{a\in\mathcal{A}} \pi(a|x) q^\pi(x,a)$$

for some measurable functions $v^\pi : \mathcal{X} \to [-4t_{mix}, 4t_{mix}]$ and $q^\pi : \mathcal{X}\times\mathcal{A} \to [-6t_{mix}, 6t_{mix}]$ with $\int_{\mathcal{X}} v^\pi(x)\mathrm{d}\nu^\pi(x) = 0$.

On the other hand, with this assumption (stronger than Assumption 1), we can replace Assumption 2 (linear MDP) with the following weaker one that only requires the bias function q^π to be linear.

Assumption 4 (Linear bias function)**.** *There exists a known d-dimensional feature mapping $\Phi : \mathcal{X}\times\mathcal{A} \to \mathbb{R}^d$ such that for every policy π, $q^\pi(x,a)$ can be written as $\Phi(x,a)^\top w^\pi$ for some weight vector $w^\pi \in \mathbb{R}^d$. Again, without loss of generality (justified in Appendix A), we assume that for all x, a, $\|\Phi(x,a)\| \le \sqrt{2}$ holds, the first coordinate of $\Phi(x,a)$ is fixed to 1, and for all π, $\|w^\pi\| \le 6t_{mix}\sqrt{d}$.*

In Lemma 14 in the appendix, we show that this is indeed weaker than the linear MDP assumption. Note that there are indeed practical examples where Assumption 4 holds but Assumption 2 does not (see the queueing network example of (De Farias and Van Roy, 2003)).

The last assumption we make is uniformly excited features, which intuitively guarantees that every policy is explorative in the feature space.

Assumption 5 (Uniformly excited features)**.** *There exists $\sigma > 0$ such that for any π,*

$$\lambda_{\min}\left(\int_{\mathcal{X}} \left(\sum_a \pi(a|x)\Phi(x,a)\Phi(x,a)^\top\right)\mathrm{d}\nu^\pi(x)\right) \ge \sigma,$$

where $\lambda_{\min}$ denotes the smallest eigenvalue.

This assumption is needed due to the nature of our algorithm that only performs local search of the parameters. It can potentially be weakened if we combine our algorithm with the idea of Abbasi-Yadkori et al. (2019b) (details omitted).

4.1 Algorithm and guarantees

We are now ready to present our MDP-EXP2 algorithm, shown in Algorithm 3. It extends the idea of running an adversarial bandit algorithm at each state from the tabular case Neu et al. (2013); Wei et al. (2020) to the continuous state case, by using an adversarial linear bandit algorithm EXP2 Bubeck et al. (2012).

Specifically, MDP-EXP2 proceeds in *epochs* of equal length $B = \widetilde{\mathcal{O}}(dt_{\text{mix}}/\sigma)$. In each epoch k, the algorithm executes a fixed policy π_k (explained later), and collects $\frac{B}{2N}$ disjoint trajectories, each of length $N = \widetilde{\mathcal{O}}(t_{\text{mix}})$. Between every two consecutive trajectories, there is a window of length N in which the algorithm does not collect any samples, so that the correlation of samples from different trajectories is reduced. See Figure 1 in the appendix for an illustration.

In the analysis, we show that the expected total reward of a trajectory is roughly $q^\pi(x_\tau, a_\tau) + NJ^\pi$ (Lemma 15), where π is the policy used to collect that trajectory and τ is the first step of the trajectory. By Assumption 4 we have $q^\pi(x_\tau,a_\tau) + NJ^\pi = \Phi(x_\tau,a_\tau)^\top(w^\pi + NJ^\pi \mathbf{e}_1)$. This observation allows us to draw a connection between this problem and adversarial linear bandits. To see this, first note that the regret is roughly $B\sum_{k=1}^{T/B}(J^* - J^{\pi_k})$. By the standard value difference lemma (Kakade, 2003, Lemma 5.2.1),

Algorithm 3 MDP-EXP2

Parameter: $N = 8t_{\text{mix}}\log T$, $B = 32N\log(dT)\sigma^{-1}$, $\eta = \min\left\{\sqrt{1/(Tt_{\text{mix}})}, \sigma/(24N)\right\}$.

for $k = 1, \dots, T/B$ **do** ▷ k indexes an epoch

  Define policy π_k such that for every $x \in \mathcal{X}$:

$$\pi_k(a|x) \propto \exp\left(\eta \sum_{j=1}^{k-1} \Phi(x,a)^\top w_j\right)$$

  Execute π_k in the entire epoch:

  for $t = (k-1)B+1, \dots, kB$ **do**

    Play $a_t \sim \pi_k(\cdot|x_t)$, observe $r_t(x_t,a_t)$ and x_{t+1}

  for $m = 1, \dots, B/2N$ **do** ▷ m indexes a trajectory

    Define

$$\tau_{k,m} = (k-1)B + 2N(m-1) + N + 1,$$

    the first step of the m-th trajectory

    Compute

$$R_{k,m} = \sum_{t=\tau_{k,m}}^{\tau_{k,m}+N-1} r(x_t,a_t),$$

    the total reward of the m-th trajectory

  Compute

$$M_k = \sum_{m=1}^{\frac{B}{2N}} \sum_a \pi_k(a|x_{\tau_{k,m}})\Phi(x_{\tau_{k,m}},a)\Phi(x_{\tau_{k,m}},a)^\top,$$

  if $\lambda_{\min}(M_k) \ge \frac{B\sigma}{24N}$ **then**

    Set $w_k = M_k^{-1}\sum_{m=1}^{\frac{B}{2N}} \Phi(x_{\tau_{k,m}}, a_{\tau_{k,m}})R_{k,m}$

  else

    Set $w_k = \mathbf{0}$

we have

$$\sum_{k=1}^{T/B}(J^*-J^{\pi_k})= \int_{\mathcal{X}}\left(\sum_{k=1}^{T/B}\sum_a(\pi^*(a|x)-\pi_k(a|x))\,q^{\pi_k}(x,a)\right)\mathrm{d}\nu^{\pi^*}(x)$$

where according to the previous observation and the fact $\sum_a(\pi^*(a|x)-\pi_k(a|x))NJ^{\pi_k}=0$, the term in the parentheses with respect to a fixed state x can be further written as $\sum_{k=1}^{T/B}\sum_a(\pi^*(a|x)-\pi_k(a|x))\,\Phi(x,a)^\top(w^{\pi_k}+NJ^{\pi_k}\mathbf{e}_1)$. This is exactly the regret of a standard online learning problem over a set of actions $\{\Phi(x,a)\}_{a\in\mathcal{A}}$ with linear reward functions parameterized by a weight vector $(w^{\pi_k}+NJ^{\pi_k}\mathbf{e}_1)$ at step k. Moreover, since we do not observe this weight but have access to the reward of a trajectory whose mean is roughly $\Phi(x,a)^\top(w^{\pi_k}+NJ^{\pi_k}\mathbf{e}_1)$ as mentioned, we are in the so-called bandit setting. In fact, since the weight can generally change arbitrarily over time (because π_k is changing), this is an adversarial linear bandit problem.

With this connection in mind, the idea behind MDP-EXP2 is clear — it conceptually runs a variant of the linear bandit algorithm EXP2 for each state. Specifically, in epoch k the algorithm constructs an estimator w_k for the reward vector $w^{\pi_k}+NJ^{\pi_k}\mathbf{e}_1$. The construction mostly follows the idea of EXP2, with the only difference being the way of controlling the variance — in the original EXP2, a particular exploration scheme is enforced, while in our case, we average multiple trajectories as done in Line 10 making use of the uniformly excited feature assumption (to make sure that $\|w_k\|$ is not too large, we also set it to $\mathbf{0}$ if $\lambda_{\min}(M_k)$ is too small, where $\lambda_{\min}$ denotes the minimum eigenvalue). Finally, with these estimators, the policy for epoch k is computed by a standard exponential weight update rule (see Line 2).

We emphasize that MDP-EXP2 does not actually need to maintain an instance of EXP2 for each state, but instead only needs to maintain the estimators w_k and calculate $\pi(\cdot|x_t)$ on the fly for each x_t, which is even more efficient than optimism-based algorithms. It also enjoys a favorable regret guarantee of order $\widetilde{\mathcal{O}}(\sqrt{T})$, as shown below. Once again, the best existing result under the same set of assumptions is $\widetilde{\mathcal{O}}(T^{2/3})$ from (Hao et al., 2021).

Theorem 7. *Under Assumptions 3, 4, 5,* MDP-EXP2 *ensures* $\mathbb{E}[\mathrm{Reg}_T]=\widetilde{\mathcal{O}}\left(\frac{1}{\sigma}\sqrt{t_{mix}^3T}\right)$.

Note that while the bound in Theorem 7 seemingly does not depend on d, the dependence is in fact implicit because $\frac{1}{\sigma}=\Omega(d)$ always holds by the definition of σ (see Remark 1 in the appendix). We provide a proof for this fact along with the proof of Theorem 7 in the appendix.

Unknown t_{mix} and σ. To decide the epoch length and the trajectory length, MDP-EXP2 requires the prior knowledge of t_{mix} and σ. However, if such knowledge is not available, we can still get a slightly worsened asymptotic regret bound of $\widetilde{\mathcal{O}}(T^{1/2+\xi})$, with an additional constant regret of $C^{1/\xi}$ for some constant C that is related to t_{mix} and σ. The idea is to slowly increase epoch length and trajectory length with time, and make sure that they exceed the required amount in the long run. The details are provided in Appendix E.

Comparison to POLITEX and AAPI. Our algorithm MDP-EXP2 is closely related to the POLITEX algorithm of (Abbasi-Yadkori et al., 2019a) and its improved version AAPI (Hao et al., 2021). The key difference is the way we construct the estimator w_k, which at a high level provides a better bias-variance trade-off. More concretely, our construction is almost unbiased (see Lemma 16), but with a larger variance, while the construction for POLITEX/AAPI has a larger bias. Because of this large bias, POLITEX/AAPI uses a much longer epoch to ensure that the error of w_k is small (i.e., using $\Theta(1/\epsilon^2)$ samples to construct w_k to ensure that w_k's error is $O(\epsilon)$); this results in less frequent update of the policy. In contrasts, MDP-EXP2 only uses a constant (in terms of t_{mix} and σ) samples to construct w_k, and update policies more often. In short, although the w_k of MDP-EXP2 is noisier, it allows faster updates of the policies and the noise of w_k is amortized over epochs. This finally leads to a better regret bound compared to POLITEX/AAPI.

Connections to Natural Policy Gradient. Finally, we remark that although MDP-EXP2 is based on an linear bandit algorithm EXP2, it is related to the (in fact much earlier) reinforcement learning algorithm Natural Policy Gradient (NPG) Kakade (2002) under softmax parameterization. The connection between softmax-parameterized NPG and the exponential weight update was formalized in a recent work by Agarwal et al. (2020). In Appendix F, we first restate the connection. Then we compare the implementation details of MDP-EXP2 and the NPG algorithm in Agarwal et al. (2020), showing that MDP-EXP2 improves the sample complexity bound of Agarwal et al. (2020) under the considered setting.

5 Conclusions and Open Problems

In this work, we provide three new algorithms for learning infinite-horizon average-reward MDPs with linear function approximation, significantly extending and improving previous works. One key open question is how to achieve the optimal $\widetilde{\mathcal{O}}(\sqrt{T})$ regret efficiently under the linear MDP assumption. In Appendix F, we also discuss another open question related to weakening Assumption 5 while maintaining a similar regret bound.

Acknowledgement

HL and CYW are supported by NSF Awards IIS1755781 and IIS-1943607, and a Google Faculty Research Award. RJ and MJ are supported by NSF grants ECCS-1810447, CCF-1817212 and ONR grant N00014-20-1-2258.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

Yasin Abbasi-Yadkori, Peter Bartlett, Kush Bhatia, Nevena Lazic, Csaba Szepesvari, and Gellért Weisz. Politex: Regret bounds for policy iteration using expert prediction. In *International Conference on Machine Learning*, pages 3692–3702, 2019a.

Yasin Abbasi-Yadkori, Nevena Lazic, Csaba Szepesvari, and Gellert Weisz. Exploration-enhanced politex. *arXiv preprint arXiv:1908.10479*, 2019b.

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in markov decision processes. In *Conference on Learning Theory*, 2020.

Keith Ball et al. An elementary introduction to modern convex geometry. *Flavors of geometry*, 31:1–58, 1997.

Peter L Bartlett and Ambuj Tewari. Regal: A regularization based algorithm for reinforcement learning in weakly communicating mdps. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 35–42. AUAI Press, 2009.

Steven J Bradtke and Andrew G Barto. Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1-3):33–57, 1996.

Sébastien Bubeck, Nicolo Cesa-Bianchi, and Sham M Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *Conference on Learning Theory*, pages 41–1, 2012.

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

Yichen Chen, Lihong Li, and Mengdi Wang. Scalable bilinear π learning using state and action features. In *International Conference on Machine Learning*, pages 834–843, 2018.

Varsha Dani, Sham M Kakade, and Thomas P Hayes. The price of bandit information for online optimization. In *Advances in Neural Information Processing Systems*, 2008.

Daniela Pucci De Farias and Benjamin Van Roy. The linear programming approach to approximate dynamic programming. *Operations research*, 51(6):850–865, 2003.

Kefan Dong, Jian Peng, Yining Wang, and Yuan Zhou. $\sqrt{n}$-regret for learning in markov decision processes with function approximation and low bellman rank. In *Conference on Learning Theory*, 2020.

Yoav Freund and Robert E Schapire. A desicion-theoretic generalization of on-line learning and an application to boosting. In *European conference on computational learning theory*, pages 23–37. Springer, 1995.

Ronan Fruit, Matteo Pirotta, Alessandro Lazaric, and Ronald Ortner. Efficient bias-span-constrained exploration-exploitation in reinforcement learning. In *International Conference on Machine Learning*, pages 1573–1581, 2018.

Ronan Fruit, Matteo Pirotta, and Alessandro Lazaric. Improved analysis of ucrl2 with empirical bernstein inequality. *arXiv preprint arXiv:2007.05456*, 2020.

Botao Hao, Nevena Lazic, Yasin Abbasi-Yadkori, Pooria Joulani, and Csaba Szepesvari. Provably efficient adaptive approximate policy iteration. In *Artificial Intelligence and Statistics*, 2021.

Nick Harvey. Matrix chernoff bounds. In *https://www.cs.ubc.ca/~nickhar/Cargese2.pdf*.

Elad Hazan and Zohar Karnin. Volumetric spanners: An efficient exploration basis for learning. *Journal of Machine Learning Research*, 17(119):1–34, 2016.

Onésimo Hernández-Lerma. *Adaptive Markov control processes*, volume 79. Springer Science & Business Media, 2012.

Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11(Apr):1563–1600, 2010.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I Jordan. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, 2020.

Sham M Kakade. A natural policy gradient. In *Advances in neural information processing systems*, pages 1531–1538, 2002.

Sham Machandranath Kakade. *On the sample complexity of reinforcement learning*. PhD thesis, University College London, 2003.

Boyi Liu, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural proximal/trust region policy optimization attains globally optimal policy. In *Advances in Neural Information Processing Systems*, 2019.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

Gergely Neu and Julia Olkhovskaya. Online learning in mdps with linear function approximation and bandit feedback. *arXiv preprint arXiv:2007.01612*, 2020.

Gergely Neu, András György, Csaba Szepesvári, and András Antos. Online markov decision processes under bandit feedback. *IEEE Transactions on Automatic Control*, 59:676–691, 2013.

Gergely Neu, Anders Jonsson, and Vicenç Gómez. A unified view of entropy-regularized markov decision processes. *arXiv preprint arXiv:1705.07798*, 2017.

Ronald Ortner. Regret bounds for reinforcement learning via markov chain concentration. *Journal of Artificial Intelligence Research*, 67:115–128, 2020.

Ian Osband, Benjamin Van Roy, and Zheng Wen. Generalization and exploration via randomized value functions. In *International Conference on Machine Learning*, pages 2377–2386, 2016.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Mohammad Sadegh Talebi and Odalric-Ambrym Maillard. Variance-aware regret bounds for undiscounted reinforcement learning in mdps. In *Algorithmic Learning Theory*, pages 770–805, 2018.

Lingxiao Wang, Qi Cai, Zhuoran Yang, and Zhaoran Wang. Neural policy gradient methods: Global optimality and rates of convergence. In *International Conference on Learning Representations*, 2019.

Ruosong Wang, Ruslan Salakhutdinov, and Lin F Yang. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. In *Advances in Neural Information Processing Systems*, 2020.

Chen-Yu Wei, Mehdi Jafarnia-Jahromi, Haipeng Luo, Hiteshi Sharma, and Rahul Jain. Model-free reinforcement learning in infinite-horizon average-reward markov decision processes. In *International Conference on Machine Learning*, 2020.

Lin F Yang and Mengdi Wang. Reinforcement leaning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning*, 2020.

Andrea Zanette, Alessandro Lazaric, Mykel Kochenderfer, and Emma Brunskill. Learning near optimal policies with low inherent bellman error. In *International Conference on Machine Learning*, 2020.

Zihan Zhang and Xiangyang Ji. Regret minimization for reinforcement learning by evaluating the optimal bias function. In *Advances in Neural Information Processing Systems*, 2019.