Reinforcement Learning in Feature Space: Matrix Bandit, Kernels, and Regret Bound

Lin F. Yang [1] **Mengdi Wang** [2]

Abstract

Exploration in reinforcement learning (RL) suffers from the curse of dimensionality when the state-action space is large. A common practice is to parameterize the high-dimensional value and policy functions using given features. However existing methods either have no theoretical guarantee or suffer a regret that is exponential in the planning horizon H. In this paper, we propose an online RL algorithm, namely the MatrixRL, that leverages ideas from linear bandit to learn a low-dimensional representation of the probability transition model while carefully balancing the exploitation-exploration tradeoff. We show that MatrixRL achieves a regret bound $O\big(H^2 d \log T\sqrt{T}\big)$ where d is the number of features, independent with the number of state-action pairs. MatrixRL has an equivalent kernelized version, which is able to work with an arbitrary kernel Hilbert space without using explicit features. In this case, the kernelized MatrixRL satisfies a regret bound $O\big(H^2 \widetilde{d} \log T\sqrt{T}\big)$, where $\widetilde{d}$ is the effective dimension of the kernel space.

1. Introduction

Reinforcement learning (RL) is about learning to make sequential decisions in an unknown environment through trial and error. It finds wide applications in robotics (Kober et al., 2013), autonomous driving (Shalev-Shwartz et al., 2016), game AI (Silver et al., 2017) and beyond. We consider a basic RL model - the Markov decision process (MDP). In the MDP, an agent at a state $s \in \mathcal{S}$ is able to play an action $a \in \mathcal{A}$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces. Then the system transitions to another state $s' \in \mathcal{S}$ according to an unknown probability $P(s' \mid s, a)$, while returning an immediate reward $r(s, a) \in [0, 1]$. The goal of the agent is to obtain the maximal possible return after playing for a period of time - even though she has no knowledge about the transition probabilities at the beginning.

The performance of a learning algorithm is measured by "regret". Regret is the difference between the cumulative reward obtained using the best possible policy and the cumulative reward obtained by the learning algorithm. Any algorithm that is capable of learning requires to have a regret sublinear in T, the number of time steps. In the tabular setting where $\mathcal{S}$ and $\mathcal{A}$ are finite sets, there exist algorithms that achieve asymptotic regret $\sim \mathrm{poly}(|\mathcal{S}||\mathcal{A}|) \cdot \sqrt{T}$ (e.g. (Jaksch et al., 2010; Dann & Brunskill, 2015; Osband & Van Roy, 2016; Osband et al., 2017; Agrawal & Jia, 2017; Azar et al., 2017; Dann et al., 2018; Jin et al., 2018)). However, the aforementioned regret bound depends polynomially on $|\mathcal{S}|$ and $|\mathcal{A}|$, sizes of the state and action space, which can be very large or even infinite. For instance, the game of Go has 3^{361} unique states, and a robotic arm has infinitely many continuous-valued states. In the most general sense, the regret $\mathrm{poly}(|\mathcal{S}||\mathcal{A}|) \cdot \sqrt{T}$ is nonimprovable in the worst case (Jaksch et al., 2010). This issue is more generally known as the "curse of dimensionality" of control and dynamic programming (Bellman, 1966).

To tackle the dimensionality, a common practice is to use features to parameterize high-dimensional value and policy functions in compact presentations, with the hope that the features can capture leading structures of the MDP. In fact, there are phenomenal empirical successes of reinforcement learning using explicit features and/or neural networks as implicit features (see e.g., (Mnih et al., 2015)). However, there is a lack of theoretical understanding about using features for exploration in RL and its learning complexity. In this paper, we are interested in the following theoretical question:

How to use features for provably efficient exploration in reinforcement learning?

Furthermore, we consider online RL in a reproducing ker-

[1]Department of Electrical and Computer Engineering, University of California, Los Angeles [2]Department of Electrical Engineering and Center for Statistics and Machine Learning, Princeton University. Correspondence to: Lin F. Yang <linyang@ee.ucla.edu>, Mengdi Wang <mengdiw@princeton.edu>.

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

nel space. Kernel methods are well known to be powerful to capture nonlinearity and high dimensionality in many machine learning tasks (Shawe-Taylor et al., 2004). We are interested in using *kernel methods* to capture nonlinearity in the state-transition dynamics of MDP. A kernel space may consist of infinitely many implicit feature functions. We study the following questions: How to use kernels in online reinforcement learning? Can one achieve low regret even though the kernel space is infinite-dimensional? The goal of this paper is to answer the aforementioned questions affirmatively. In particular, we would like to design algorithms that take advantages of given features and kernels to achieve efficient exploration.

1.1. Our Approach and Main Results

Consider episodic reinforcement learning in finite-horizon MDP. The agent learns through episodes, and each episode consists of H time steps. Here H is also called the *planning horizon*. Let $\phi(\cdot) \in \mathbb{R}^d, \psi(\cdot) \in \mathbb{R}^{d'}$ be given feature functions, where we usally assume $d' = O(d)$. We focus on the case that the probability transition model $P(\cdot \mid \cdot)$ can be fully embedded in the feature space (Assumption 1), i.e., there exists some core matrix M^* such that

$$P(\cdot \mid \cdot) = \phi(\cdot)^\top M^* \psi(\cdot).$$

In the kernel setting, this condition is equivalent to that the transition probability model P belongs to the product space of the reproducing kernel spaces. This condition is essentially equivalent to using the features ϕ to represent value functions (Parr et al., 2008). When the probability transition model P cannot be fully embedded using ϕ, then value function approximation using ϕ may lead to arbitrarily large Bellman error (Yang & Wang, 2019). Note that the core matrix model is related to many statistic models, e.g., latent state model (Singh et al., 1995), Lumpable Markov Chain (Weinan et al., 2008), Markov process with rich observations (Azizzadenesheli et al., 2016), and so on. For completeness, we list their relationships to the core matrix model in Section A. Moreover, the importance of similar models have also been emphasized in (Yang & Wang, 2019) and (Jin et al., 2019).

We propose an algorithm, which is referred to as MatrixRL, that actively explores the state-action space by estimating the core matrix via ridge regression. The algorithm balances the exploitation-exploration tradeoff by constructing a confidence ball of core matrix for optimistic dynamic programming. It can be thought of as a "matrix bandit" algorithm which generalizes the idea of linear bandit (e.g. (Dani et al., 2008; Li et al., 2010; Chu et al., 2011)). It is proved to achieve the regret bound either

$$\widetilde{O}(H^2 d^{3/2}\sqrt{T}) \quad \text{or} \quad \widetilde{O}(H^2 d\sqrt{T})^1,$$

depending on regularity properties of the features. MatrixRL can be implemented efficiently in space $O(d^2)$. Each step can be carried out in closed form. Next we extend the MatrixRL to work with the kernel spaces with $k_\phi(\cdot,\cdot) = \langle \phi(\cdot), \phi(\cdot) \rangle$ and $k_\psi(\cdot,\cdot) = \langle \psi(\cdot), \psi(\cdot) \rangle$, and show that it admits a kernelized version. The kernelized MatrixRL achieves a regret bound of

$$\widetilde{O}(H^2 \widetilde{d}\sqrt{T})$$

where $\widetilde{d}$ is the *effective dimension* of kernel space, even if there may be infinitely many features. The regret bounds using features or kernels do not depend on sizes of the state and action spaces, making efficient exploration possible in high dimensions. Very recently, (Du et al., 2019a) show that even if the value-function of the MDP admits an approximate linear representation with features, there is still an $\Omega(2^H)$ regret lower bound. Our results complement this lower bound by showing that feature representations of the transition model effectively reduces the learning complexity.

Note that for linear bandit, the regret lower bound is known to be $\widetilde{\Omega}(d\sqrt{T})$ (Dani et al., 2008). Since linear bandit is a special case of RL, our regret bounds match the lower bound up to polylog factors in d and T. To our best knowledge, for reinforcement learning using features/kernels, our result gives the first regret bound that is *simultaneously near-optimal in time T, polynomial in the planning horizon H, and the feature dimension d.*

1.2. Related Literature

In the tabular case where there are finitely many states and actions without any structural knowledge, complexity and regret for RL has been extensively studied. For H-horizon episodic RL, efficient methods typically achieve regret that scale asymptotically as $O(\sqrt{HSAT})$ (see for examples (Jaksch et al., 2010; Osband & Van Roy, 2016; Osband et al., 2017; Agrawal & Jia, 2017; Azar et al., 2017; Dann et al., 2018; Jin et al., 2018)). In particular, (Jaksch et al., 2010) provided a regret lower bound $\Omega(\sqrt{HSAT})$ for H-horizon MDP. There is also a line of works studying the sample complexity of obtaining a value or policy that is at most ϵ-suboptimal (Kakade, 2003; Strehl et al., 2006; 2009; Szita & Szepesvári, 2010; Lattimore & Hutter, 2014; Azar et al., 2013; Dann & Brunskill, 2015; Sidford et al., 2018). The optimal sample complexity for finding an ϵ-optimal policy is $O\big(|\mathcal{S}||\mathcal{A}|(1-\gamma)^{-2}\epsilon^{-2}\big)$ (Sidford et al., 2018) for a discounted MDP with discount factor γ. The optimal lower bound has been proven in (Azar et al., 2013).

There is also a line of works on solving MDPs with a function approximation. For instance (Baird, 1995; Tsitsiklis & Van Roy, 1997; Parr et al., 2008; Mnih et al., 2013; 2015; Silver et al., 2017; Yang & Wang, 2019). There

[1] $\widetilde{O}(\cdot)$ hides poly log factors of the input.

are also phenomenal empirical successes in deep reinforcement learning as well (e.g., (Silver et al., 2017)). However there are not many works on the regret analysis of RL with function approximators. Very recently, (Azizzadenesheli et al., 2018) studied the regret bound for linear function approximator. However their bound has factor that can be exponential in H. (Chowdhury & Gopalan, 2019) considers the regret bound for kernelized MDP. However, they need a Gaussian process prior and assumes that the transition is deterministic with some controllable amount of noise – a very restrictive setting. Another work (Modi & Tewari, 2019) also considers the linear setting for RL. However, the regret bound is linearly depending on the number of states. More recently, there are several works released after the first arXiv version of this paper, e.g. (Du et al., 2019b; Jin et al., 2019; Zanette et al., 2020). They study similar models as ours but with different algorithms and from different perspectives. In particular, (Jin et al., 2019; Zanette et al., 2020) are considering a similar low-rank MDP model. The major distinction of our results to theirs is that ours is a model-based one whereas theirs are model-free. Algorithm-wise, these results together form nice toolkits for RL on low-rank MDPs. Another important distinction is that we require the features ψ to be known in priori. However, as learning the representations ϕ and ψ from data have the same complexity in unsupervised learning, such a requirement is not a strong requirement. To the best of our knowledge, we are not aware of other works that achieve regret bound for RL with function approximators that is simultaneously near optimal in T, polynomial in H, and has no dependence with the state-action space size.

Our results are also related to the literature of linear bandits. Bandit problems can be viewed as a special case as Markov decision problems. There is a line of works on linear bandit problems and their regret analysis (Dani et al., 2008; Rusmevichientong & Tsitsiklis, 2010; Li et al., 2010; Abbasi-Yadkori et al., 2011; Chu et al., 2011). For a more detailed survey, please refer to (Bubeck et al., 2012). Part of our results are inspired by the kernelization for the linear bandit problems, e.g. (Valko et al., 2013; Chowdhury & Gopalan, 2017), who studied the regret bound when the features of each arm lies in some reproducing kernel Hilbert space.

2. Problem Formulation

In an *episodic Markov decision process* (MDP for short), there is a set of *states* $\mathcal{S}$ and a set of *actions* $\mathcal{A}$, which are not necessarily finite. At any state $s \in \mathcal{S}$, an agent is allowed to play an action $a \in \mathcal{A}$. She receives an immediate reward $r(s,a) \in [0,1]$ after playing a at s, the process will transition to the next state $s' \in \mathcal{S}$ with probability $P(s' \mid s,a)$, where P is the collection of *transition distributions*. After H time steps, the system restarts at a prespecified state s_0. The full instance of an MDP can be described by the tuple $M = (\mathcal{S}, \mathcal{A}, P, r, s_0, H)$. The agent would like to find a *policy* $\pi : \mathcal{S} \times [H] \to \mathcal{A}$ that maximizes the long-term expected reward starting from every state s and every stage $h \in [H]$, i.e.,

$$V_h^\pi(s) := \mathbb{E}\bigg[\sum_{h'=h}^{H} r(s^t, \pi_h(s^t)) | s^0 = s\bigg].$$

We call $V^\pi : \mathcal{S} \times [H] \to \mathbb{R}$ the *value function* of policy π. A policy π^* is said to be *optimal* if it attains the maximal possible value at every state-stage pair (s,h). We denote V^* as the optimal value function. We also denote the optimal action-value function (or Q-function) as

$$\forall h \in [H-1]: \quad Q_h^*(s,a) = r(s,a) + P(\cdot|s,a)^\top V_{h+1}^*,$$

and $Q_H^*(s,a) = r(s,a)$.

In the online RL setting, the learning algorithm interacts with the environment episodically. Each episode starts from state s_0 takes H steps to finish. We let n denote the current number of episodes and denote $t = (n-1)H + h$ the current time step. We equalize t and (n,h) and may switch between the two nations. We use the following definition of regret.

Definition 1. *Suppose we run algorithm $\mathcal{K}$ in the online environment of an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, s_0, H)$ for $T = NH$ steps. We define the regret for algorithm $\mathcal{K}$ as*

$$\mathrm{Regret}_{\mathcal{K}}(T) = \mathbb{E}_{\mathcal{K}}\bigg[\sum_{n=1}^{N}\Big(V^*(s_0) - \sum_{h=1}^{H} r\Big(s_t, a_t\Big)\Big)\bigg], \tag{1}$$

where $\mathbb{E}_{\mathcal{K}}$ is taken over the random path of states under the control of algorithm $\mathcal{K}$.

Throughout this paper, we focus on RL problems where the probability transition kernel $P(\cdot \mid \cdot)$ can be fully embedded in a given feature space.

Assumption 1 (Feature Embedding of Transition Model)**.** *For each $(s,a) \in \mathcal{S} \times \mathcal{A}, \widetilde{s} \in \mathcal{S}$, feature vectors $\phi(s,a) \in \mathbb{R}^d, \psi(\widetilde{s}) \in \mathbb{R}^{d'}$ are given as a priori. There exists an unknown matrix $M^* \in \mathbb{R}^{d \times d'}$ such that*

$$P(\widetilde{s} \mid s,a) = \phi(s,a)^\top M^* \psi(\widetilde{s}).$$

Here, we call the matrix M^* as a transition core.

Note that when ϕ, ψ are features associated with two reproducing kernel spaces $\mathcal{H}_1$ and $\mathcal{H}_2$, this assumption requires that P belong to their product kernel space $\mathcal{H}_1 \times \mathcal{H}_2$.

For simplicity of representation, we assume throughout that the reward function r is known. This is in fact without

loss of generality because learning about the environment P is much harder than learning about r. In the case if r is unknown but satisfies $\mathbb{E}[r(s,a) \mid s,a] = \phi(s,a)^\top \theta^*$ for some unknown $\theta^* \in \mathbb{R}^d$, we can extend our algorithm by adding a step of optimistic reward estimation like in LinUCB (Dani et al., 2008; Chu et al., 2011). This would generate an extra $\widetilde{O}(d\sqrt{T})$ regret, which is a low order term compared to our current regret bounds.

3. RL Exploration in Feature Space

In this section, we study the near optimal way to balance exploration and exploitation in RL using a given set of features. We aim to develop an online RL algorithm with regret that depends only on the feature size d but not on the size of the state-action space. Our algorithm is inspired by the LinUCB algorithm (Chu et al., 2011) and its variants (Dani et al., 2008) and can be viewed as a "matrix bandit" method.

3.1. The MatrixRL Algorithm

The high level idea of the algorithm is to approximate the unknown transition core M^* using data that has been collected so far. Suppose at the time step $t = (n,h)$ (i.e. episode $n \le N$ and stage $h \le H$), we obtain the following state-action-state transition triplet: $(s_t, a_t, \widetilde{s}_t)$, where $\widetilde{s}_t := s_{t+1}$. For simplicity, we denote the associated features by

$$\phi_t := \phi(s_t, a_t) \in \mathbb{R}^d \quad \text{and} \quad \psi_t := \psi(\widetilde{s}_t) \in \mathbb{R}^{d'}.$$

Estimating the core matrix. Let $K_\psi := \sum_{\widetilde{s} \in \mathcal{S}} \psi(\widetilde{s})\psi(\widetilde{s})^\top$, which we assume is given along with the features ψ. We then construct our estimator of M^* as:

$$M_n = [A_n]^{-1} \sum_{n'<n, h \le H} \phi_{n',h} \psi_{n',h}^\top K_\psi^{-1}, \tag{2}$$

where

$$A_n = I + \sum_{n'<n, h \le H} \phi_{n',h}\phi_{n',h}^\top.$$

Let us explain the intuition of M_n. Note that

$$\begin{aligned}
&\mathbb{E}\Big[\phi_{n,h}\psi_{n,h}^\top K_\psi^{-1} \mid s_{n,h}, a_{n,h}\Big] \\
&\quad = \sum_{\widetilde{s}} \phi_{n,h} P(\widetilde{s} \mid s_{n,h}, a_{n,h}) \psi(\widetilde{s})^\top K_\psi^{-1} \\
&\quad = \sum_{\widetilde{s}} \phi_{n,h}\phi_{n,h}^\top M^* \psi(\widetilde{s})\psi(\widetilde{s})^\top K_\psi^{-1} \\
&\quad = \phi_{n,h}\phi_{n,h}^\top M^*.
\end{aligned}$$

Therefore M_n is the approximate solution to the following ridge regression problem:

$$M_n = \arg\min_M \sum_{n'<n, h \le H} \left\| \psi_{n',h}^\top K_\psi^{-1} - \phi_{n',h}^\top M \right\|_2^2 + \|M\|_F^2. \tag{3}$$

Upper confidence RL using a matrix ball. In online RL, a critical step is to estimate future value of the current state and action use dynamic programming. To better balance exploitation and exploration, we use a matrix ball to construct optimistic value function estimator. At episode n:

$$\begin{aligned}
& w_{n,H+1} = 0 \in \mathbb{R}^{d'} \quad \text{and} \\
\forall h \in [H]: \quad & w_{n,h} = \alpha_\psi \cdot \sum_{j \in [m_w]} \frac{\psi(s_j)}{\|\psi(s_j)\|_1} \cdot V_{n,h}(s_j)
\end{aligned} \tag{4}$$

where $\alpha_\psi = \sum_s \|\psi(s)\|_1$ is also given along with ψ, m_w is a number to be determined, and s_j are picking from the distribution generated by $p(s) = \|\psi(s)\|_1/\alpha_\psi$, which is also given with ψ, and, for all (s,a), we define,

$$\begin{aligned}
V_{n,h}(s) &:= \Pi_{[0,H]}\big[\max_a Q_{n,h}(s,a)\big], \\
Q_{n,h}(s,a) &:= r(s,a) + \max_{M \in B_n} \phi(s,a)^\top M w_{n,h+1}.
\end{aligned} \tag{5}$$

Here the matrix ball $B_n = B_n^{(1)}$ is constructed as

$$B_n^{(1)} := \Big\{ M \in \mathbb{R}^{d \times d'} : \quad \|(A_n)^{1/2}(M - M_n)\|_{2,1} \le \sqrt{d\beta_n} \Big\} \tag{6}$$

where β_n is a parameter to be determined later, and $\|Y\|_{2,1} = \sum_i \sqrt{\sum_j Y(i,j)^2}$. At time (n,h), suppose the current state is $s_{n,h}$, we play the optimistic action $a_{n,h} = \arg\max_a Q_{n,h}(s_{n,h}, a)$.

Remark 1 (Computation in Continuous Space)**.** *Later we will show that m_w is polynomial in the number of episodes. Consequently, our algorithm computes $w_{n,h}$ efficiently in each round. In the continuous space, the only issue is to solve the* max *problem in* (4)*. In this case, we assume there exists an* optimization oracle *that solves* $\sup_a Q_{n,h}(s,a)$ *in time* $O(1)$*. Note that the time complexity of the oracle depends subtly on the feature space structure, and there is no general way of expressing it analytically.*

The full algorithm is given in Algorithm 1.

3.2. Regret Bounds for MatrixRL

Let $\Psi = [\psi(s_1), \psi(s_2), \ldots, \psi(s_{|\mathcal{S}|})]^\top \in \mathbb{R}^{\mathcal{S} \times d'}$ be the matrix of all ψ features. We first introduce some regularity conditions of the features space.

Algorithm 1 Upper Confidence Matrix Reinforcement Learning (UC-MatrixRL)

Input: An episodic MDP environment $M = (\mathcal{S}, \mathcal{A}, P, r, s_0, H)$;
  Features $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$ and $\psi : \mathcal{S} \to \mathbb{R}^{d'}$;
  Trotal number of episodes N;
Initialize: $A_1 \leftarrow I \in \mathbb{R}^{d\times d}$, $M_1 \leftarrow 0 \in \mathbb{R}^{d\times d'}$;
for episode $n = 1, 2, \ldots, N$ **do**
  Let $\{Q_{n,h}\}$ be given in (4) using M_n, β_n;
  for stage $h = 1, 2, \ldots, H$ **do**
    Let the current state be $s_{n,h}$;
    Play action $a_{n,h} = \arg\max_{a\in\mathcal{A}} Q_{n,h}(s_{n,h}, a)$
    Record the next state $s_{n,h+1}$;
  end for
  $A_{n+1} \leftarrow A_n + \sum_{h\le H} \phi_{n,h}\phi_{n,h}^\top$;
  Compute M_{n+1} using (2);
end for

Assumption 2 (Feature Regularity)**.** *Let C_M, C_ϕ, C_ψ and C'_ψ be positive parameters.*

1. $\|M^*\|_F^2 \le C_M \cdot d$;
2. $\forall\ (s,a) \in \mathcal{S}\times\mathcal{A}: \quad \|\phi(s,a)\|_2^2 \le C_\phi d$;
3. $\forall v \in \mathbb{R}^{\mathcal{S}} : \|\Psi^\top v\|_\infty \le C_\psi \|v\|_\infty$, *and* $\|\Psi K_\psi^{-1}\|_{2,\infty} \le C'_\psi$.[2]

With these conditions we are ready to provide the regret bound.

Theorem 1. *Suppose Assumption 1 and Assumption 2 hold, and $m_w = \mathrm{poly}(T)$. Then after $T = NH$ steps, Algorithm 1 achieves regret bound:*

$$\mathrm{Regret}(T) \le O\Big[\sqrt{C_\psi(C_M + {C'_\psi}^2)}\cdot\ln(C_\phi T)\Big]\cdot H^2\cdot\sqrt{d^2d'T},$$

if we let $\beta_n = c\cdot(C_M + {C'_\psi}^2)\cdot\ln(NHC_\phi)\cdot d$, for some absolute constant $c > 0$. Moreover, for each new episode, Algorithm 1 takes time $\mathrm{poly}(T)$ to complete.

Consider the case where C_M, C_ϕ, C_ψ and C'_ψ are absolute constants. For example, we may let Ψ, Φ, M^* be probability matrices. Then Assumption 2 automatically holds with $C_M = C_\psi = C_\psi = 1$. In this case, our regret bound is simply $O(d^{3/2}H^2\log(T)\sqrt{T})$. The d-dependence in such a regret bound is consistent with the regret of the ℓ_1-ball algorithm for linear bandit (Dani et al., 2008).

Further, if the feature space Ψ admits a tighter bound for value function in this space, we can slightly modify our algorithm to achieve sharper regret bound. To do this, we need to slightly change our Assumption 2 to Assumption 2′.

Assumption 2′ (Stronger Feature Regularity)**.** *Let C_M, C_ϕ, C_ψ and C'_ψ be positive parameters.*

1. $\|M^*\|_F^2 \le C_M \cdot d$;
2. $\forall\ (s,a) \in \mathcal{S}\times\mathcal{A}: \quad \|\phi(s,a)\|_2^2 \le C_\phi d$;
3. $\forall v \in \mathbb{R}^{\mathcal{S}} : \|\Psi^\top v\|_2 \le C_\psi \|v\|_\infty$, *and* $\|\Psi K_\psi^{-1}\|_{2,\infty} \le C'_\psi$.

We modify the algorithm slightly by using a Frobenious-norm matrix ball instead of the 2-1 norm and computing sharper confidence bounds. Let $B_n = B_n^{(2)}$ in (4), where

$$B_n^{(2)} := \Big\{M \in \mathbb{R}^{d\times d'} : \|(A_n)^{1/2}(M - M_n)\|_F \le \sqrt{\beta_n}\Big\} \tag{7}$$

Then a sharper regret bound can be established.

Theorem 2. *Suppose Assumption 1 and Assumption 2′ hold, and $m_w = \mathrm{poly}(T)$. Then after $T = NH$ steps, Algorithm 1, with $B_n = B_n^{(2)}$ applied in (4), achieves regret*

$$\mathrm{Regret}(T) \le O\Big[\sqrt{C_\psi(C_M + {C'_\psi}^2)}\cdot\ln(C_\phi T)\Big]\cdot dH^2\cdot\sqrt{T},$$

provided $\beta_n = c\cdot(C_M + {C'_\psi}^2)\cdot\ln(NHC_\phi)\cdot d$, for some absolute constant $c > 0$. Moreover, for each new episode, Algorithm 1 takes time $\mathrm{poly}(T)$ to complete.

The only stronger condition needed by Assumption 2′ is $\|\Psi^\top v\|_2 \le C_\psi\|v\|_\infty$. It can be satisfied if Ψ is a set of sparse features, or if Ψ is a set of highly concentrated features. Note that a key difference of our analysis to that of (Jin et al., 2019) is that they apply an ϵ-net argument to bound the regret. Such an union bound gives a regret bound proportional to $\sqrt{d^3}$. In contrast, we use a model-based method to decouple the dependence between samples and apply a martingale concentration argument, which saves an additional d factor under Assumption 2′.

We remark that in Theorem 1 and Theorem 2, we need to know the value N in β_n before the algorithm runs. In the

[2] Here $\|Y\|_{2,\infty} := \max_i \sqrt{\sum_j Y^2(i,j)}$ is the operator $2 \to \infty$ norm.

case when N is unknown, one can use the doubling trick to learn N adaptively: first we run the algorithm by picking $N = 2$, then for $N = 4, 8, \ldots, 2^i$ until the true N is reached. It is standard knowledge that this trick increase the overall regret by only a constant factor (e.g. (Besson & Kaufmann, 2018)).

Proof Sketch. The proof consists of two parts. We show that when the core matrix M^* belongs to the sequence of constructed balls $\{B_n\}$, the estimated Q-functions provide optimistic estimates of the optimal values, therefore the algorithm's regret can be bounded using the sum of confidence bounds on the sample path. The second part constructs a martingale difference sequence by decomposing a matrix into an iterative sum and uses a probabilistic concentration argument to show that the "good" event happens with sufficiently high probability. In the next section, we present the full proof for Theorems 1. The full proof of Theorems 2 is deferred to the appendix.

Near optimality of regret bounds? The regret bound in Theorem 2 matches the optimal regret bound $\widetilde{O}(d\sqrt{T})$ for linear bandit (Dani et al., 2008). In fact, linear bandit is a special case of RL: the planning horizon H is 1. However the hard instance for linear bandit may not be necessarily constructable using the core matrix model.

Closed-form confidence bounds. Equation (4) requires maximization over a matrix ball. However, it is not necessary to solve this maximization problem explicitly. The algorithm only requires an optimistic Q value. In fact, we can use a closed-form *confidence bound* instead of searching for the optimal M in the *confidence ball*. It can be verified that Theorem 1 still holds (by following the same proofs of the theorem) if we replace the second equation of (4) as the following equation (see the proof of Theorem 2)

$$\forall h \in [H]: \quad Q_{n,h}(s,a) = r(s,a) + \phi(s,a)^\top M_n \Psi^\top V_{n,h+1} + 2C_\psi H\sqrt{d\beta_n} \cdot w_{n,h}, \tag{8}$$

where $w_{n,h} := \sqrt{\phi_{n,h}^\top (A_n)^{-1} \phi_{n,h}}$. Similarly, Theorem 2 still holds if we replace the the second equation of 4 with

$$\forall h \in [H]: \quad Q_{n,h}(s,a) = r(s,a) + \phi(s,a)^\top M_n \Psi^\top V_{n,h+1} + 2C_\psi \sqrt{\beta_n} \cdot w_{n,h}. \tag{9}$$

Equations (8) and (9) can be computed easily. They can be viewed as the "dualization" of (4).

3.3. Analysis and Proofs

In this section we will focus on proving Theorem 1. In the proof we will also establish all the necessary analytical tools for proving Theorem 2 and Theorem 9. We provide the proofs of the last two theorems in the appendix.

To begin, we first investigate the Monte-Carlo step used in (4). For each step, we let $\widetilde{w}_{n,h} = \Psi^\top V_{n,h}$. For $w_{n,h}$, we have the following guarantee, which is a simple consequence of Hoeffding's inequality.

Proposition 1. *With probability at least $1-\delta$, we have*

$$\|\widetilde{w}_{n,h} - w_{n,h}\|_\infty \le C \cdot \sqrt{\log \delta^{-1}/m_w}$$

for some generic constant C depends on the properties of Ψ.

Notice that the randomness used in computing the above equation is independent with that from other parts of the algorithm, and can the error can be made arbitrarily small by picking $m_w = \mathrm{poly}(T)$. The regret incurred by this Monto-Carlo step can be at most

$$T \cdot \frac{1}{\sqrt{m_w}} \ll \sqrt{T}.$$

For the sake of presentation, we assume $w_{n,h} = \widetilde{w}_{n,h}$ in the rest of the proof.

The proof of Theorem 1 consists of two steps: (a) We first show that if the true transition core M^* is always in the confidence ball B_n, defined in Equation 6, we can then achieve the desired regret bound; (b) We then show that with high probability, the event required by (a) happens. We formalize the event required by step (a) as follows.

Definition 2 (Good Estimator Event)**.** *For all $n \in [N]$, we denote $E_n = 1$ if $M^* \in B_{n'}$ for all $n' \in [n]$ and otherwise $E_n = 0$.*

Note that E_n is completely determined by the game history up to episode n. In Section B.2, we show that $E_n = 1$ happens with high probability for all n. In the next subsection, we show (a).

3.4. Regret Under Good Event

To better investigate the regret formulation (1), we rewrite it according to Algorithm 1. Note that conditioning on the history before episode n, the algorithm plays a fixed policy π_n for episode n. Therefore, we have

$$\mathrm{Regret}(NH) = \sum_{n=1}^{N} \mathbb{E}[R(n)], \tag{10}$$

where $R(n) = V_1^*(s_0) - V_1^{\pi_n}(s_0)$. We now show that the algorithm always plays an optimistic action (an action with value estimated greater than the optimal value of the state). All missing proofs can be found in Section B.1.

Lemma 3 (Optimism)**.** *Suppose for $n \in [N]$, we have the good estimator event, $E_n = 1$, happens. Then for $h \in [H]$ and $(s,a) \in \mathcal{S} \times \mathcal{A}$, we have*

$$Q_h^*(s,a) \le Q_{n,h}(s,a).$$

Next we show that the confidence ball B_n actually gives a strong upper bound for the estimation error: the estimation error is "along" the direction of the exploration.

Lemma 4. *For any $M \in B_n$ we have*

$$\|\phi_{s,a}^\top (M - M_n)\|_1 \le \sqrt{d'\beta_n \cdot \phi(s,a)^\top (A_n)^{-1} \phi(s,a)}.$$

Proof.

$$\begin{aligned}\|\phi_{s,a}^\top (M - M_n)\|_1 &= \|\phi_{s,a}^\top (A_n)^{-1/2} (A_n)^{1/2} (M - M_n)\|_1 \\ &\le \|\phi_{s,a}^\top (A_n)^{-1/2}\|_2 \cdot \|((A_n)^{1/2}(M - M_n))^\top\|_{2,1} \\ &\le \sqrt{d'\beta_n} \cdot \|\phi_{s,a}^\top (A_n)^{-1/2}\|_2\end{aligned}$$

as desired. □

Next we show that the value iteration per-step does not introduce too much error.

Lemma 5. *Suppose for $n \in [N]$, $E_n = 1$. Then for $h \in [H]$, we have*

$$\begin{aligned}Q_{n,h}(s_{n,h}, a_{n,h}) - \Big[r(s_{n,h}, a_{n,h}) + P(\cdot|s_{n,h}, a_{n,h})^\top V_{n,h+1}\Big] \\ \le 2C_\psi H \sqrt{d'\beta_n} \cdot w_{n,h}\end{aligned}$$

where

$$w_{n,h} := \sqrt{\phi_{n,h}^\top (A_n)^{-1} \phi_{n,h}}. \tag{11}$$

We are now ready to show the regret bound, whose proof is presented in the appendix.

Lemma 6. *Suppose Assumption 2 holds, $1 \le \beta_1 \le \beta_2 \le \ldots \beta_N$, then,*

$$\begin{aligned}\mathrm{Regret}(NH) &\le 2C_\psi H \sqrt{d'\beta_N} \\ &\cdot \mathbb{E}\Big[\sum_{n=1}^N \sum_{h=1}^H \sqrt{\min(1, w_{n,h}^2)}\Big] + \sum_{n=1}^N H\mathbb{P}[E_n \ne 1]\end{aligned}$$

It remains to bound

$$\sum_{n=1}^N \sum_{h=1}^H \sqrt{\min(1, w_{n,h}^2)} \le \sqrt{HN \cdot \sum_{n=1}^N \sum_{h=1}^H \min(1, w_{n,h}^2)}.$$

We provide the following lemma.

Lemma 7.

$$\begin{aligned}\sum_{n=1}^N \sum_{h=1}^H \min(1, w_{n,h}^2) &\le 2H \ln\det(A_{N+1}) \\ &\le 2Hd \ln(NHC_\phi + 1).\end{aligned}$$

The proof of this lemma is rather technical and requires some new notations, we postpone it to Section B.3. We are now ready to state the regret bound.

Lemma 8. *Suppose $1 \le \beta_1 \le \beta_2 \le \ldots \beta_N$ and $C_\psi \ge 1$, then*

$$\begin{aligned}\mathrm{Regret}(T) &\le 2C_\psi H \sqrt{d'\beta_N} \cdot \sqrt{HN \cdot 2Hd \cdot \ln[NHC_\phi + 1]} \\ &+ \sum_{n=1}^N H\mathbb{P}[E_n = 0].\end{aligned}$$

We are now ready to prove Theorem 1.

Proof of Theorem 1. We let $\delta \le 1/(NH)$. By Lemma 13 and 15, we pick

$$\begin{aligned}\beta_n &\ge cd \cdot [C_M + {C'_\psi}^2 \cdot \ln(nHC_\phi) + c \cdot {C'_\psi}^2 \cdot \ln(nHC_\phi/\delta)] \\ &= \Theta(C_M + {C'_\psi}^2) \cdot \ln(nHC_\phi) \cdot d.\end{aligned}$$

Then the following is guaranteed,

$$\Pr[\forall n \le N: \quad E_n = 1] \ge 1 - \delta.$$

Then by Lemma 8, we have,

$$\begin{aligned}\mathrm{Regret}(T) &\le 2C_\psi H \sqrt{d'\beta_N} \\ &\cdot \sqrt{HN \cdot 2Hd \cdot \ln[NHC_\phi + 1]} + O(1) \\ &= O\Big[C_\psi \sqrt{C_M + {C'_\psi}^2} \cdot \ln(NHC_\phi)\Big] \cdot \sqrt{d^2 d' H^3 T}.\end{aligned}$$

□

4. RL Exploration in Kernel Space

In this section, we transform MatrixRL to work with kernels instead of explicitly given features. Suppose we are given two reproducing kernel Hilbert spaces $\mathcal{H}_\phi, \mathcal{H}_\psi$ with kernel functions $k_\phi : (\mathcal{S} \times \mathcal{A}) \times (\mathcal{S} \times \mathcal{A}) \to \mathbb{R}$ and $k_\psi : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$, respectively. There exists implicit features ϕ, ψ such that $k_\phi(x,y) = \phi(x)^\top \phi(y)$ and $k_\psi(x,y) = \psi(x)^\top \psi(y)$, but the learning algorithm can access the kernel functions only.

4.1. Kernelization of MatrixRL

The high level idea of Kernelized MatrixRL is to represent all the features used in Algorithm 1 with their corresponding kernel representations. We first introduce some notations. For episode n and $T = nH$, we denote $\mathbf{K}_{\phi_n} \in$

$\mathbb{R}^{T\times T}$ and $\mathbf{K}_{\psi_n}\in\mathbb{R}^{T\times T}$ as the Gram matrix, respectively, i.e., for all $t_1=(n_1,h_1), t_2=(n_2,h_2)\in[n]\times[H]$,

$$\mathbf{K}_{\phi_n}[t_1,t_2]=k_\phi[(s_{t_1},a_{t_1}),(s_{t_2},a_{t_2})],$$
$$\mathbf{K}_{\psi_n}[t_1,t_2]=k_\psi(\widetilde{s}_{t_1},\widetilde{s}_{t_2}),$$

where $\widetilde{s}_t := s_{t+1}$. We denote $\overline{\mathbf{K}}_{\psi_n}\in\mathbb{R}^{T\times|\mathcal{S}|}$ and $\mathbf{k}_{\Phi_n,s,a}\in\mathbb{R}^T$ by

$$\overline{\mathbf{K}}_{\psi_n}[t_1,s]=k_\psi(\widetilde{s}_{t_1},s),$$
$$\mathbf{k}_{\Phi_n,s,a}[t]=k_\phi[(s_t,a_t),(s,a)].$$

We are now ready to kernelize Algorithm 1. The full algorithm of Kernelized MatrixRL is given in Algorithm 2. Note that the new Q function estimator (12) is the dualization form of (4). Therefore Algorithm 2 is more general but essentially equivalent to Algorithm 1 if we let $k_\phi(x,y) := \phi(x)^\top\phi(y)$ and $k_\psi(x,y) := \psi(x)^\top\psi(y)$. Note that the computation of $(\lambda I+\overline{\mathbf{K}}_{\Psi_{n-1}}\overline{\mathbf{K}}_{\Psi_{n-1}}^\top)^{-1}\overline{\mathbf{K}}_{\Psi_n}$ can again be performed using Monte-Carlo efficiently (as in (4)). For the sake of representation, we assume we have an exact computation. See Section C for the proof.

4.2. Regret Bound for Kernelized MatrixRL

We define the *effective dimension* of the kernel space $\mathcal{H}_\phi$ as

$$\widetilde{d}=\sup_{t\le NH}\sup_{X\subset\mathcal{S}\times\mathcal{A},|X|=t}\frac{\log\det[I+\mathbf{K}_X]}{\log(1+t)},$$

where $X=\{x_j\}_{j\in[t]}$, $\mathbf{K}_X\in\mathbb{R}^{t\times t}$ with $\mathbf{K}_X[i,j]=k_\phi(x_i,x_j)$ is the Gram matrix over data set X. Note that $\widetilde{d}$ captures the effective dimension of the space spanned by the features of state-action pairs. Consider the case when ϕ are d-dimensional unit vectors, then $\mathcal{H}_\phi$ has dimension at most d. It can be verified that $\widetilde{d}\le d$. A similar notation of effective dimension was introduced by (Valko et al., 2013) for analyzing kernelized contextual bandit.

Further, we need regularity assumptions for the kernel space.

Assumption 3. *Let $\mathcal{H}_\psi$ be generated by orthonormal basis on $\mathcal{S}$, i.e., there exists ψ such that $\sum_{s\in\mathcal{S}}\psi(s)\psi(s)^\top=I$ and $k_\psi(s,s')=\psi(s)^\top\psi(s')$. There exists a constant C_ψ such that $\forall v\in\mathcal{H}_\psi:\quad\|v\|_{\mathcal{H}_\psi}\le C_\psi\|v\|_\infty$, where $\|\cdot\|_{\mathcal{H}_\psi}$ denotes the Hilbert space norm.*

The formal guarantee of Kernelized MatrixRL is presented as follows.

Theorem 9. *Suppose the probability transition kernel $P(\cdot\mid\cdot)$ belongs to the product Hilbert spaces, i.e., $P\in\mathcal{H}_\phi\times\mathcal{H}_\psi$. Let Assumption 3 hold. Then after $T=NH$ time steps, the regret of Algorithm 2 satisfies*

$$\text{Regret}(T)\le O\Big(C_\psi\cdot\|P\|_{\mathcal{H}_\phi\times\mathcal{H}_\psi}\cdot\log(T)\cdot\widetilde{d}\cdot H^2\cdot\sqrt{T}\Big)$$

provided $\eta_n=2C_\psi H\sqrt{\beta_n}$ and $\beta_n=\Theta(\|P\|_{\mathcal{H}_\phi\times\mathcal{H}_\psi}\cdot\ln(NH)\cdot\widetilde{d})$.

Note that in Assumption 3, we can additionally relax the assumption on the orthogonality of ψ. Similar regret bound can be proved with Assumption 2′. The proof of Theorem 9 is very similar to that of Theorem 2. Although Kernelized MatrixRL does not access the features, the proof is based on the underlying features and the equivalence between kernel representation and feature representation. We postpone it to Section C.

Remark. Similar as MatrixRL, Kernelized MatrixRL can be generalized to deal with unknown reward function by using the Kernelized Bandit (Valko et al., 2013). Again, since linear bandit problems are special cases of kernel RL with $H=1$, our results match the linear bandit bound on $\widetilde{d}$ and $\sqrt{T}$. The computation time of Kernelized MatrixRL scales with time T as T^2 (by applying randomized algorithms, e.g. (Dani et al., 2008), in dealing with $\overline{\mathbf{K}}_\Psi$ matrices), still polynomial in T. We can apply the random features or sketching techniques for kernel to additionally accelerate the computation (e.g. (Rahimi & Recht, 2008; Yang et al., 2017)).

5. Summary

This paper provided the algorithm MatrixRL for episodic reinforcement learning in high dimensions. It also provides the first regret bounds that are near-optimal in time T and feature dimension d and polynomial in the planning horizon H. MatrixRL uses given features (or kernels) to estimate a core transition matrix and its confidence ball, which is used to compute optimistic Q-functions for balancing the exploitation-exploration tradeoff. We prove that the regret of MatrixRL is bounded by $O\big(H^2d\log T\sqrt{T}\big)$ where d is the number of features, provided that the feature space satisfies some regularity conditions. MatrixRL has an equivalent kernel version, which does not require explicit features. The kernelized MatrixRL satisfies a regret bound $O\big(H^2\widetilde{d}\log T\sqrt{T}\big)$, where $\widetilde{d}$ is the effective dimension of the kernel space. For future work, it remains open if the regularity condition can be relaxed and if there is a more efficient way for constructing confidence balls in order to further reduce the regret.

Acknowledgment

Mengdi Wang gratefully acknowledges funding from the U.S. National Science Foundation (NSF) grant CMMI-1653435, Air Force Office of Scientific Research (AFOSR) grant FA9550-19-1-020, and C3.ai DTI.

Algorithm 2 KernelMatrixRL: Reinforcement Learning with Kernels

Input: An episodic MDP environment $M = (\mathcal{S}, \mathcal{A}, P, s_0, r, H)$, kernel functions k_ϕ, k_ψ;
Total number of episodes N;
Initialize: empty reply buffer $\mathcal{B} = \{\}$;
for episode $n = 1, 2, \ldots, N$ **do**
For $(s, a) \in \mathcal{S} \times \mathcal{A}$, let

$$w_n(s,a) := \sqrt{k_\phi[(s,a),(s,a)] - \mathbf{k}_{\Phi_{n-1},s,a}^\top (I + \mathbf{K}_{\Phi_{n-1}})^{-1} \mathbf{k}_{\Phi_{n-1},s,a}};$$

$$x_n(s,a) := \mathbf{k}_{\Phi_{n-1},s,a}^\top (I + \mathbf{K}_{\Phi_{n-1}})^{-1} \mathbf{K}_{\Psi_{n-1}} (\lambda I + \overline{\mathbf{K}}_{\Psi_{n-1}} \overline{\mathbf{K}}_{\Psi_{n-1}}^\top)^{-1} \overline{\mathbf{K}}_{\Psi_n};$$

where $\lambda > 0$ is a parameter to be adjusted;
Let $\{Q_{n,h}\}$ be defined as follows:

$$\begin{aligned} \forall (s,a) \in \mathcal{S} \times \mathcal{A}: \quad & Q_{n,H+1}(s,a) := 0 \quad \text{and} \\ \forall h \in [H]: \quad & Q_{n,h}(s,a) := r(s,a) + x_n(s,a)^\top V_{n,h+1} + \eta_n w_n(s,a), \end{aligned} \tag{12}$$

where

$$V_{n,h}(s) = \Pi_{[0,H]} \big[\max_a Q_{n,h}(s,a) \big] \qquad \forall s, a, n, h;$$

and η_n is a parameter to be determined;
for stage $h = 1, 2, \ldots, H$ **do**
Let the current state be $s_{n,h}$;
Play action $a_{n,h} = \arg\max_{a \in \mathcal{A}} Q_{n,h}(s_{n,h}, a)$;
Record the next state $s_{n,h+1}$: $\mathcal{B} \leftarrow \mathcal{B} \cup \{(s_{n,h}, a_{n,h}, s_{n,h+1})\}$;
end for
end for

References

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

Agrawal, S. and Jia, R. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. In *Advances in Neural Information Processing Systems*, pp. 1184–1194, 2017.

Azar, M. G., Munos, R., and Kappen, H. J. Minimax pac bounds on the sample complexity of reinforcement learning with a generative model. *Machine learning*, 91 (3):325–349, 2013.

Azar, M. G., Osband, I., and Munos, R. Minimax regret bounds for reinforcement learning. *arXiv preprint arXiv:1703.05449*, 2017.

Azizzadenesheli, K., Lazaric, A., and Anandkumar, A. Reinforcement learning in rich-observation mdps using spectral methods. *arXiv preprint arXiv:1611.03907*, 2016.

Azizzadenesheli, K., Brunskill, E., and Anandkumar, A. Efficient exploration through bayesian deep q-networks. In *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–9. IEEE, 2018.

Baird, L. Residual algorithms: Reinforcement learning with function approximation. In *Machine Learning Proceedings 1995*, pp. 30–37. Elsevier, 1995.

Bellman, R. Dynamic programming. *Science*, 153(3731): 34–37, 1966.

Besson, L. and Kaufmann, E. What doubling tricks can and can't do for multi-armed bandits. *arXiv preprint arXiv:1803.06971*, 2018.

Bubeck, S., Cesa-Bianchi, N., et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

Chowdhury, S. R. and Gopalan, A. On kernelized multi-armed bandits. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 844–853. JMLR. org, 2017.

Chowdhury, S. R. and Gopalan, A. Online learning in kernelized markov decision processes. In Chaudhuri, K. and Sugiyama, M. (eds.), *Proceedings of Machine Learning Research*, volume 89 of *Proceedings of Machine Learning Research*, pp. 3197–3205. PMLR, 16–18 Apr 2019. URL http://proceedings.mlr.press/v89/chowdhury19a.html.

Chu, W., Li, L., Reyzin, L., and Schapire, R. E. Contextual Bandits with Linear Payoff Functions. Technical report, 2011. URL http://proceedings.mlr.press/v15/chu11a/chu11a.pdf.

Dani, V., Hayes, T. P., and Kakade, S. M. Stochastic linear optimization under bandit feedback. 2008.

Dann, C. and Brunskill, E. Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 2818–2826, 2015.

Dann, C., Li, L., Wei, W., and Brunskill, E. Policy certificates: Towards accountable reinforcement learning. *arXiv preprint arXiv:1811.03056*, 2018.

Du, S. S., Kakade, S. M., Wang, R., and Yang, L. F. Is a good representation sufficient for sample efficient reinforcement learning? *arXiv preprint arXiv:1910.03016*, 2019a.

Du, S. S., Luo, Y., Wang, R., and Zhang, H. Provably efficient q-learning with function approximation via distribution shift error checking oracle. In *Advances in Neural Information Processing Systems*, pp. 8058–8068, 2019b.

Freedman, D. A. On Tail Probability for Martigales. *The Annals of Probability*, 3(1):100–118, 1975.

Hsu, D., Kakade, S. M., and Zhang, T. A spectral algorithm for learning hidden markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480, 2012.

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

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pp. 4863–4873, 2018.

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

Kakade, S. M. *On the sample complexity of reinforcement learning*. PhD thesis, University of London London, England, 2003.

Kober, J., Bagnell, J. A., and Peters, J. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

Lattimore, T. and Hutter, M. Near-optimal pac bounds for discounted mdps. *Theoretical Computer Science*, 558: 125–143, 2014.

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

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., and Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518 (7540):529, 2015.

Modi, A. and Tewari, A. Contextual markov decision processes using generalized linear models. *arXiv preprint arXiv:1903.06187*, 2019.

Osband, I. and Van Roy, B. On lower bounds for regret in reinforcement learning. *arXiv preprint arXiv:1608.02732*, 2016.

Osband, I., Van Roy, B., Russo, D., and Wen, Z. Deep exploration via randomized value functions. *arXiv preprint arXiv:1703.07608*, 2017.

Parr, R., Li, L., Taylor, G., Painter-Wakefield, C., and Littman, M. L. An analysis of linear models, linear value-function approximation, and feature selection for reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 752–759. ACM, 2008.

Rahimi, A. and Recht, B. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pp. 1177–1184, 2008.

Rusmevichientong, P. and Tsitsiklis, J. N. Linearly parameterized bandits. *Mathematics of Operations Research*, 35(2):395–411, 2010.

Shalev-Shwartz, S., Shammah, S., and Shashua, A. Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*, 2016.

Shawe-Taylor, J., Cristianini, N., et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

Sidford, A., Wang, M., Wu, X., Yang, L. F., and Ye, Y. Near-optimal time and sample complexities for for solving discounted markov decision process with a generative model. *arXiv preprint arXiv:1806.01492*, 2018.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

Singh, S. P., Jaakkola, T., and Jordan, M. I. Reinforcement learning with soft state aggregation. In *Advances in neural information processing systems*, pp. 361–368, 1995.

Strehl, A. L., Li, L., Wiewiora, E., Langford, J., and Littman, M. L. Pac model-free reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*, pp. 881–888. ACM, 2006.

Strehl, A. L., Li, L., and Littman, M. L. Reinforcement learning in finite MDPs: PAC analysis. *Journal of Machine Learning Research*, 10(Nov):2413–2444, 2009.

Szita, I. and Szepesvári, C. Model-based reinforcement learning with nearly tight exploration complexity bounds. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 1031–1038, 2010.

Tsitsiklis, J. N. and Van Roy, B. Analysis of temporal-diffference learning with function approximation. In *Advances in neural information processing systems*, pp. 1075–1081, 1997.

Valko, M., Korda, N., Munos, R., Flaounas, I., and Cristianini, N. Finite-time analysis of kernelised contextual