Provably Efficient Reinforcement Learning for Discounted MDPs with Feature Mapping

Dongruo Zhou[1] Jiafan He[1] Quanquan Gu[1]

Abstract

Modern tasks in reinforcement learning have large state and action spaces. To deal with them efficiently, one often uses predefined feature mapping to represent states and actions in a low dimensional space. In this paper, we study reinforcement learning for discounted Markov Decision Processes (MDPs), where the transition kernel can be parameterized as a linear function of certain feature mapping. We propose a novel algorithm which makes use of the feature mapping and obtains a $\widetilde{O}(d\sqrt{T}/(1-\gamma)^2)$ regret, where d is the dimension of the feature space, T is the time horizon and γ is the discount factor of the MDP. To the best of our knowledge, this is the first polynomial regret bound without accessing to a generative model or making strong assumptions such as ergodicity of the MDP. By constructing a special class of MDPs, we also show that for any algorithms, the regret is lower bounded by $\Omega(d\sqrt{T}/(1-\gamma)^{1.5})$. Our upper and lower bound results together suggest that the proposed reinforcement learning algorithm is near-optimal up to a $(1-\gamma)^{-0.5}$ factor.

1. Introduction

Designing efficient algorithms that learn and plan in sequential decision-making tasks with large state and action spaces has become the central goal of modern reinforcement learning (RL) in recent years. Due to numerous possible states and actions, traditional tabular reinforcement learning methods (Watkins, 1989; Jaksch et al., 2010; Azar et al., 2017) which directly access each state-action pair are computationally intractable. A common method to design reinforcement learning algorithms for large-scale state and action spaces is to make use of feature mappings such as linear functions or neural networks to map states and actions to a low-dimensional space and solve the decision-making problem in the feature space. Despite the empirical success of feature mapping based reinforcement learning methods (Singh et al., 1995; Bertsekas, 2018), the theoretical understanding and the fundamental limits of these methods remain largely understudied.

In this paper, we aim to develop provable reinforcement learning algorithms with feature mapping for discounted Markov Decision Processes (MDPs). Discounted MDP is one of the most widely used models to formulate the modern reinforcement learning tasks such as Atari games (Mnih et al., 2015) and deep recommendation system (Zheng et al., 2018). With feature mapping, a series of recent work (Yang & Wang, 2019a; Lattimore et al., 2020; Bhandari et al., 2018; Zou et al., 2019) have proposed provably efficient algorithms along with theoretical guarantees. However, these existing results either rely on a special oracle called *generative model* (Kakade et al., 2003) that allows an algorithm to query any possible state-action pairs and return both the reward and the next state (Yang & Wang, 2019a; Lattimore et al., 2020), or needs strong assumptions such as uniform ergodicity (Bhandari et al., 2018; Zou et al., 2019) on the underlying MDP. A natural question arises:

Can we design provably efficient RL algorithms with feature mapping for discounted MDPs under mild assumptions?

We answer this question affirmatively. To be more specific, we consider a special class of discounted MDPs called *linear kernel MDP*, where the transition probability kernel can be represented as a linear function of a predefined d-dimensional feature mapping. A similar model has been studied in earlier work Jia et al. (2020); Ayoub et al. (2020) for finite horizon episodic MDPs, where the authors call it *linear mixture model*. Linear kernel MDP is a rich MDP class, which covers many classes of MDPs proposed in previous work (Yang & Wang, 2019b; Modi et al., 2019) as special cases. We propose a novel provably efficient algorithm namely Upper-Confidence Linear Kernel reinforcement learning (UCLK) to solve this MDP. We prove both upper and lower regret bounds and show that our algorithm is near-optimal under the linear kernel MDP setting.

[1] Department of Computer Science, University of California, Los Angeles, CA 90095, USA. Correspondence to: Quanquan Gu <qgu@cs.ucla.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

Our contributions are summarized as follows.

- We propose a novel algorithm UCLK to learn the optimal value function with the help of predefined feature mapping. We show that the regret (See Definition 3.5) for UCLK to learn the optimal value function is $\widetilde{O}(d\sqrt{T}/(1-\gamma)^2)$. It is worth noting that the regret is independent of the cardinality of the state and action spaces, which suggests that UCLK is efficient for large-scale RL problems. To the best of our knowledge, this is the first feature-based reinforcement learning algorithm that attains a polynomial regret bound for discounted MDPs without accessing the generative model or making strong assumptions on MDPs such as ergodicity[1].
- We also show that for any reinforcement learning algorithms, the regret to learn the optimal value function in linear kernel MDP is at least $\Omega(d\sqrt{T}/(1-\gamma)^{1.5})$. This lower bound result suggests that UCLK is optimal concerning feature mapping dimension d and time horizon T, and it is near-optimal concerning the discount factor up to $(1-\gamma)^{-0.5}$. Our proof is based on a specially constructed linear kernel MDP, which could be of independent interest.

After we posted the first version of this paper online, we were informed that the linear kernel MDP setting is the same as the so-called *parameterized transition model* or *linear mixture model* in earlier work (Jia et al., 2020; Ayoub et al., 2020).

The remainder of this paper is organized as follows. In Section 2, we review the related work in the literature. We introduce preliminaries in in Section 3, and our algorithm in Section 4. In Section 5, we present our main theoretical results including both upper and lower regret bounds, followed by a proof sketch of the main theory in Section 6. Finally, we conclude this paper in Section 7. The detailed proofs are deferred to the supplementary material.

Notation We use lower case letters to denote scalars, and use lower and upper case bold face letters to denote vectors and matrices respectively. Let $\mathbb{1}(\cdot)$ denote the indicator function. For a vector $\mathbf{x} \in \mathbb{R}^d$ and matrix $\mathbf{\Sigma} \in \mathbb{R}^{d\times d}$, we denote by $\|\mathbf{x}\|_2$ the Euclidean norm and denote by $\|\mathbf{x}\|_{\mathbf{\Sigma}} = \sqrt{\mathbf{x}^\top \mathbf{\Sigma}\mathbf{x}}$. For two sequences $\{a_n\}$ and $\{b_n\}$, we write $a_n = O(b_n)$ if there exists an absolute constant C such that $a_n \le Cb_n$, and we write $a_n = \Omega(b_n)$ if there exists an absolute constant C such that $a_n \ge Cb_n$. We use $\widetilde{O}(\cdot)$ to further hide the logarithmic factors.

2. Related Work

Finite-horizon MDPs with feature mappings. There is a series of work focusing on solving finite-horizon MDP using RL with function approximation (Jin et al., 2019; Yang & Wang, 2019b; Wang et al., 2019; Modi et al., 2019; Jiang et al., 2017; Zanette et al., 2020; Du et al., 2019). For instance, Jin et al. (2019) assumed the underlying transition kernel and reward function are linear functions of a d-dimensional feature mapping and proposed an RL algorithm with $\widetilde{O}(\sqrt{d^3H^3T})$ regret, where H is the length of an episode. Yang & Wang (2019b) assumed the probability transition kernel is bilinear in two feature mappings in dimension d and d', and proposed an algorithm with $\widetilde{O}(dH^2\sqrt{T})$ regret. Wang et al. (2019) assumed the Bellman backup of any value function is a generalized linear function of certain feature mapping and proposed an algorithm with a regret guarantee. Modi et al. (2019) assumed the underlying MDP can be represented as a linear combination of several base models and proposed an RL algorithm to solve it with a provable guarantee. Jiang et al. (2017) assumed the underlying MDP is of low inherent Bellman error and proposed an algorithm with polynomial PAC bounds. Jia et al. (2020) studied the linear mixture model and proposed a UCRL-VTR algorithm for finite-horizon MDPs which achieves a $\widetilde{O}(d\sqrt{H^3T})$ regret, where H is the episode length. Ayoub et al. (2020) considered the same model but with general function approximation, and proved a regret bound depending on Eluder dimension (Russo & Van Roy, 2013). Jia et al. (2020); Ayoub et al. (2020) also proved a lower bound of regret by considering the hard tabular MDP firstly proposed in Jaksch et al. (2010). Zanette et al. (2020) studied a similar MDP as Jin et al. (2019) and proposed an algorithm with tighter regret bound. Du et al. (2019) suggested that the sample complexity to learn the optimal policy can be exponential if the approximation error to the value function is moderate. More discussions and insights regarding these negative results can be found in Van Roy & Dong (2019); Lattimore et al. (2020).

Discounted MDPs with a generative model. For tabular discounted MDPs, many work focuses on RL with the help of a generative model (or called a simulator) (Kakade et al., 2003). To learn the optimal value function, Azar et al. (2013) proposed Empirical QVI, which learns an ϵ-suboptimal value function with $\widetilde{O}(|\mathcal{S}||\mathcal{A}|/((1-\gamma)^3\epsilon^2))$ optimal sample complexity. To learn the optimal policy, Kearns & Singh (1999) proposed Phased Q-Learning which learns an ϵ-suboptimal policy with $\widetilde{O}(|\mathcal{S}||\mathcal{A}|/((1-\gamma)^7\epsilon^2))$ sample complexity. Sidford et al. (2018b) proposed a Sublinear Randomized Value Iteration algorithm which achieves a $\widetilde{O}(|\mathcal{S}||\mathcal{A}|/((1-\gamma)^4\epsilon^2))$ sample complexity. Sidford et al. (2018a) further proposed Variance-Reduced QVI algorithm which achieves the optimal $\widetilde{O}(|\mathcal{S}||\mathcal{A}|/((1-\gamma)^3\epsilon^2))$ sample complexity. For discounted MDPs with function approxima-

[1] Without a generative model (simulator) or further assumptions on MDP, some states may never be visited starting from certain initial states, which makes it impossible to find a near-optimal policy on them. Therefore, it is not meaningful to consider the sample complexity of UCLK to find an ϵ-optimal policy.

tion, Yang & Wang (2019a) assumed the probability transition kernel can be parameterized by a d-dimensional feature mapping and proposed a Phased Parametric Q-Learning algorithm which learns an ϵ-suboptimal policy with the optimal $\widetilde{O}(d/((1-\gamma)^3\epsilon^2))$ sample complexity. Lattimore et al. (2020) considered a similar setting to Yang & Wang (2019a) and proposed a Phased Elimination algorithm with $\widetilde{O}(d/((1-\gamma)^4\epsilon^2))$ sample complexity.

Discounted MDPs without a generative model. Another line of work aims at learning the discounted MDP without accessing to the generative model. Szita & Szepesvári (2010) proposed an MoRmax algorithm which achieves $\widetilde{O}(|\mathcal{S}||\mathcal{A}|/((1-\gamma)^6\epsilon^2))$ sample complexity of exploration. Lattimore & Hutter (2012) proposed UCRL algorithm which achieves $\widetilde{O}(|\mathcal{S}|^2|\mathcal{A}|/((1-\gamma)^3\epsilon^2))$ sample complexity of exploration. Strehl et al. (2006) proposed delay-Q-learning with $\widetilde{O}(|\mathcal{S}||\mathcal{A}|/((1-\gamma)^8\epsilon^4))$ sample complexity of exploration. Dong et al. (2019) proposed Infinite Q-learning with UCB which achieves $\widetilde{O}(|\mathcal{S}||\mathcal{A}|/((1-\gamma)^7\epsilon^2))$ sample complexity of exploration. Liu & Su (2020) proposed the regret definition for discounted MDPs and presented Double Q-Learning to achieve $\widetilde{O}(\sqrt{|\mathcal{S}||\mathcal{A}|T}/(1-\gamma)^{2.5})$ regret. Our work falls into this category, and also uses regret to characterize the performance of RL.

3. Preliminaires

We consider infinite-horizon discounted Markov Decision Processes (MDPs), which is denoted by a tuple $M(\mathcal{S},\mathcal{A},\gamma,r,\mathbb{P})$. Here $\mathcal{S}$ is a countable state space (may be infinite), $\mathcal{A}$ is the action space, $\gamma : 0 \leq \gamma < 1$ is the discount factor, $r : \mathcal{S}\times\mathcal{A} \to [0,1]$ is the reward function. For simplicity, we assume the reward function r is *deterministic* and *known*. $\mathbb{P}(s'|s,a)$ is the transition probability function which denotes the probability for state s to transfer to state s' given action a. A (nonstationary) policy π is a collection of policies π_t, where each $\pi_t : \{\mathcal{S}\times\mathcal{A}\}^{t-1}\times\mathcal{S}\to\mathcal{A}$ maps history $s_1, a_1, \ldots, s_{t-1}, a_{t-1}, s_t$ to an action a. Let $\{s_t, a_t\}_{t=1}^{\infty}$ are states and actions deduced by $\mathbb{P}$ and π. We denote the action-value function $Q_t^{\pi}(s,a)$ and value function $V_t^{\pi}(s,a)$ as follows

$$Q_t^{\pi}(s,a) = \mathbb{E}\bigg[\sum_{i=0}^{\infty}\gamma^i r(s_{t+i}, a_{t+i})\bigg| s_1,\ldots,s_t = s, a_t = a\bigg],$$

$$V_t^{\pi}(s) = \mathbb{E}\bigg[\sum_{i=0}^{\infty}\gamma^i r(s_{t+i}, a_{t+i})\bigg| s_1,\ldots,s_t = s\bigg].$$

We define the optimal value function V^* and the optimal action-value function Q^* as $V^*(s) = \sup_{\pi} V_1^{\pi}(s)$ and $Q^*(s,a) = \sup_{\pi} Q_1^{\pi}(s,a)$. For simplicity, for any function $V : \mathcal{S}\to\mathbb{R}$, we denote $[\mathbb{P}V](s,a) = \mathbb{E}_{s'\sim\mathbb{P}(\cdot|s,a)}V(s')$. Therefore we have the following Bellman equation, as well as the Bellman optimality equation:

$$Q_t^{\pi}(s_t,a_t) = r(s_t,a_t) + \gamma[\mathbb{P}V_{t+1}^{\pi}](s_t,a_t),$$
$$Q^*(s_t,a_t) = r(s_t,a_t) + \gamma[\mathbb{P}V^*](s_t,a_t).$$

In this work, we consider a special class of MDPs called *linear kernel MDPs*, where the transition probability function can be represented as a linear function of a given feature mapping $\boldsymbol{\phi} : \mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\mathbb{R}^d$. It is worth noting that this is essentially the same MDP class as *linear mixture model* considered in Jia et al. (2020); Ayoub et al. (2020). Formally speaking, we have the following assumption for a linear kernel MDP.

Definition 3.1. $M(\mathcal{S},\mathcal{A},\gamma,r,\mathbb{P})$ is called a linear kernel MDP if there exist a *known* feature mapping $\boldsymbol{\phi}(s'|s,a) : \mathcal{S}\times\mathcal{A}\times\mathcal{S}\to\mathbb{R}^d$ and an *unknown* vector $\boldsymbol{\theta}\in\mathbb{R}^d$ with $\|\boldsymbol{\theta}\|_2 \leq \sqrt{d}$, such that

- For any state-action-state triplet $(s,a,s')\in\mathcal{S}\times\mathcal{A}\times\mathcal{S}$, we have $\mathbb{P}(s'|s,a) = \langle\boldsymbol{\phi}(s'|s,a),\boldsymbol{\theta}\rangle$;
- For any bounded function $V : \mathcal{S}\to[0,R]$ and any tuple $(s,a)\in\mathcal{S}\times\mathcal{A}$, we have $\|\boldsymbol{\phi}_V(s,a)\|_2 \leq \sqrt{d}R$, where $\boldsymbol{\phi}_V(s,a) = \sum_{s'}\boldsymbol{\phi}(s'|s,a)V(s')\in\mathbb{R}^d$.

We denote the linear kernel MDP by $M_{\boldsymbol{\theta}}$ for simplicity.

As we will show in the following examples, linear kernel MDPs cover several MDPs studied in previous work as special cases.

Example 3.2 (Tabular MDPs)**.** For an MDP $M(\mathcal{S},\mathcal{A},\gamma,r,\mathbb{P})$ with $|\mathcal{S}|,|\mathcal{A}| \leq \infty$, the transition probability function can be parameterized by $|\mathcal{S}|^2|\mathcal{A}|$ *unknown* parameters. The tabular MDP is a special case of linear kernel MDPs with the following feature mapping and parameter vector: $d = |\mathcal{S}|^2|\mathcal{A}|$, $\boldsymbol{\phi}(s'|s,a) = \mathbf{e}_{(s,a,s')}\in\mathbb{R}^d$, $\boldsymbol{\theta} = [\mathbb{P}(s'|s,a)]\in\mathbb{R}^d$, where $\mathbf{e}_{(s,a,s')}$ denotes the corresponding natural basis in the d-dimensional Euclidean space.

Example 3.3 (Linear combination of base models (Modi et al., 2019))**.** For an MDP $M(\mathcal{S},\mathcal{A},\gamma,r,\mathbb{P})$, suppose there exist m base transition probability functions $\{p_i(s'|s,a)\}_{i=1}^m$, a feature mapping $\boldsymbol{\psi}(s,a) : \mathcal{S}\times\mathcal{A}\to\Delta^{d'}$ where $\Delta^{d'}$ is a $(d'-1)$-dimensional simplex, and an *unknown* matrix $\mathbf{W}\in\mathbb{R}^{m\times d'}\in[0,1]^{m\times d'}$ such that $\mathbb{P}(s'|s,a) = \sum_{k=1}^m[\mathbf{W}\boldsymbol{\psi}(s,a)]_k p_k(s'|s,a)$. Then it is a special case of linear kernel MDPs with feature mapping and parameter vector defined as follows: $d = md'$, $\boldsymbol{\phi}(s'|s,a) = \mathrm{vec}(\mathbf{p}(s'|s,a)\boldsymbol{\psi}(s,a)^{\top})\in\mathbb{R}^d$, $\boldsymbol{\theta} = \mathrm{vec}(\mathbf{W})\in\mathbb{R}^d$, where $\mathrm{vec}(\cdot)$ is the vectorization operator, and $\mathbf{p}(s'|s,a) = [p_k(s'|s,a)]\in\mathbb{R}^m$.

Example 3.4 (Feature embedding of a transition model (Yang & Wang, 2019b)**).** For an MDP $M(\mathcal{S}, \mathcal{A}, \gamma, r, \mathbb{P})$, suppose that there exist feature mappings $\boldsymbol{\psi}_1(s,a) : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^{d_1}$ satisfying $\|\boldsymbol{\psi}_1(s,a)\|_2 \leq \sqrt{d_1}$, $\boldsymbol{\psi}_2(s') : \mathcal{S} \to \mathbb{R}$ satisfying for any $V : \mathcal{S} \to [0, R]$, $\| \sum_s V(s)\boldsymbol{\psi}_2(s)\|_2 \leq R$ and an *unknown* matrix $\mathbf{M} \in \mathbb{R}^{d_1 \times d_2}$ satisfying $\|\mathbf{M}\|_F \leq \sqrt{d_1}$ such that $\mathbb{P}(s'|s,a) = \boldsymbol{\psi}_1(s,a)^\top \mathbf{M} \boldsymbol{\psi}_2(s')$. Then it is a special case of linear kernel MDPs with the following feature mapping and parameter vector $d = d_1 d_2$, $\boldsymbol{\phi}(s'|s,a) = \text{vec}\big(\boldsymbol{\psi}_2(s')\boldsymbol{\psi}_1(s,a)^\top\big) \in \mathbb{R}^d$, $\boldsymbol{\theta} = \text{vec}(\mathbf{M}) \in \mathbb{R}^d$.

Comparison with linear MDPs. Yang & Wang (2019a); Jin et al. (2019) studied the so-called linear additive model or *linear MDP*, which assumes the probability transition function can be represented as $\mathbb{P}(\cdot|s,a) = \langle \boldsymbol{\psi}(s,a), \boldsymbol{\mu}(\cdot)\rangle$, where $\boldsymbol{\psi}(s,a)$ is a known feature mapping and $\mu(\cdot)$ is an unknown measure. It is worth noting that linear kernel MDPs studied in our paper and linear MDPs (Yang & Wang, 2019a; Jin et al., 2019) are two different classes of MDPs since they are based on different feature mappings, i.e., $\boldsymbol{\phi}(s'|s,a)$ versus $\boldsymbol{\psi}(s,a)$. One cannot be covered by the other. For instance, some MDPs only fit linear MDPs such as $\mathbb{P}(s'|s,a) = \sum_{i=1}^d \phi_i(s,a)\mu_i(s')$ satisfying $\phi_i(s,a) > 0, \sum_{i=1}^d \phi_i(s,a) = 1$ and $\mu_i(s')$ is an unknown measure of s'. Some MDPs only fit linear kernel MDPs such as $\mathcal{S} = \mathbb{R}$, $\mathcal{A} = \mathbb{R}/\{0\}$, $\mathbb{P}(s'|s,a) = \sum_{i=1}^d \theta_i p_i(s'|s,a)$, $p_i(s'|s,a) = \exp(-(s'-s-i)^2/(2a^2))/\sqrt{2\pi a^2}$. It is not a linear MDP because $p_i(s'|s,a)$ can not be decomposed as $\phi_i(s,a) \cdot \mu_i(s')$. In the rest of this paper, we assume the underlying linear kernel MDP is parameterized by $\boldsymbol{\theta}^*$ and denote it by $M_{\boldsymbol{\theta}^*}$.

In the online learning setting, the environment picks the starting state s_1 at the beginning. The goal is to design a nonstationary policy π such that the expected discounted return at step t, $V_t^\pi(s_t)$, is close to the optimal expected return $V^*(s_t)$. We formalize this goal as minimizing the regret, which can be defined as follows, inspired by Liu & Su (2020).

Definition 3.5. For any policy π, we define its regret on MDP $M(\mathcal{S}, \mathcal{A}, \gamma, r, \mathbb{P})$ in the first T rounds as the sum of the suboptimality Δ_t for $t = 1, \ldots, T$, i.e.,

$$\text{Regret}(\pi, M, T) = \sum_{t=1}^T \Delta_t, \text{ where } \Delta_t = V^*(s_t) - V_t^\pi(s_t),$$

Due to the optimality of the optimal value function V^*, we know that $\Delta_t \geq 0$ for any policy π. This fact suggests that Regret(T) can be regarded as a cumulative error for π to learn the optimal value function of MDP M.

Relation to sample complexity of exploration. A related quantity widely used for discounted MDPs is called the *sample complexity of exploration* $N(\epsilon, \delta)$ (Szita & Szepesvári, 2010; Lattimore & Hutter, 2012; Dong et al., 2019), which is defined as the number of rounds t where Δ_t is greater than ϵ with probability at least $1 - \delta$. Note that algorithms with smaller regret make fewer mistakes in total (for instance, for any $\epsilon > 0$, there does not exist $N_\epsilon > 0$ such that for any $t > N_\epsilon$, $\Delta_t \geq \epsilon$), but they could make several severe mistakes (for instance, $\Delta_t = 1$ may happen infinite times). In comparison, algorithms with smaller sample complexity of exploration do not make severe mistakes (since there are only finite number of Δ_t satisfying $\Delta_t \geq \epsilon$), but they may suffer from a infinite number of 'less severe' mistakes (for instance, Δ_t satisfying $\epsilon > \Delta_t \geq \epsilon/2$) in total. Therefore, these two quantities are not directly comparable. For any algorithm with $\widetilde{O}(C\epsilon^{-a})$ sample complexity of exploration, where C is a problem dependent constant (e.g., it may depend on $|\mathcal{S}|, |\mathcal{A}|, \gamma, d$), we can do a conversion and show that the algorithm also enjoys a $\widetilde{O}(C^{1/(a+1)}(1-\gamma)^{-1/(a+1)}T^{a/(a+1)})$ regret for the first T rounds. The proof is deferred to Appendix A. More comparisons and discussions can also be found in Liu & Su (2020) for the tabular setting.

4. The Proposed Algorithm

In this section, we propose an algorithm namely UCLK to learn the linear kernel MDP, which is illustrated in Algorithm 1. UCLK is essentially a multi-epoch algorithm inspired by Jaksch et al. (2010); Lattimore & Hutter (2012). Specifically, the k-th epoch of Algorithm 1 starts at round t_k and ends at round $t_{k+1} - 1$. The length of each epoch is not prefixed but depends on previous observations. In each epoch, UCLK uses Extended Value Iteration (EVI) function to compute the estimated optimal action-value function Q_k and selects the greedy policy according to the function. The reason for using adaptive epoch length is that it can control the amount of "switching error" which occurs when the policy is updated. Each epoch of UCLK can be divided into two phases, which we will discuss in detail in the sequel.

Planning phase (Line 4 to 6) Planning phase is executed at the beginning of each epoch. In this phase, UCLK first computes $\widehat{\boldsymbol{\theta}}_k$ as the estimate of $\boldsymbol{\theta}^*$, which is the minimizer of the following regularized least-square problem:

$$\widehat{\boldsymbol{\theta}}_k \leftarrow \mathop{\text{argmin}}_{\boldsymbol{\theta} \in \mathbb{R}^d} \sum_{j=0}^{k-1} \sum_{i=t_j}^{t_{j+1}-1} \big[\big\langle \boldsymbol{\theta}, \boldsymbol{\phi}_{V_j}(s_i, a_i)\big\rangle - V_j(s_{i+1})\big]^2 + \lambda\|\boldsymbol{\theta}\|_2^2, \tag{4.2}$$

which has a closed-form solution as shown in Line 4. Then Algorithm 1 computes the confidence set of $\boldsymbol{\theta}^*$ as $\mathcal{C}_k$ based on the confidence radius parameter β. Based on the confidence set $\mathcal{C}_k$, Algorithm 1 selects Algorithm 2 to compute the next action-value functions Q_k for the next steps.

Extended value iteration Algorithm 1 makes use of EVI

Algorithm 1 Upper-Confidence Linear Kernel Reinforcement Learning (UCLK)

```
Require: Regularization parameter $\lambda$, confidence radius $\beta$, number of value iteration rounds $U$
 1: Receive $s_1$
 2: Set $t \leftarrow 1$, $\mathbf{\Sigma}_1 \leftarrow \lambda \mathbf{I}$, $\mathbf{b}_1 = \mathbf{0}$
 3: for $k = 0, \ldots$ do
 4:     Set $t_k \leftarrow t$, $\widehat{\boldsymbol{\theta}}_k \leftarrow \mathbf{\Sigma}_{t_k}^{-1}\mathbf{b}_{t_k}$
 5:     Set $\mathcal{C}_k$ and $Q_k(\cdot,\cdot)$ as follows:
            $$\mathcal{C}_k = \{\boldsymbol{\theta} : \|\mathbf{\Sigma}_{t_k}^{1/2}(\boldsymbol{\theta} - \widehat{\boldsymbol{\theta}}_k)\|_2 \le \beta\},\ Q_k(\cdot,\cdot) \leftarrow \text{EVI}(\mathcal{C}_k, U)$$
 6:     Set $V_k(\cdot) \leftarrow \max_{a\in\mathcal{A}} Q_k(\cdot, a)$
 7:     repeat
 8:         Set $\pi_t(\cdot) \leftarrow \text{argmax}_a\, Q_k(\cdot, a)$, take action $a_t \leftarrow \pi_t(s_t)$, receive $s_{t+1} \sim \mathbb{P}(\cdot|s_t, a_t)$
 9:         Set $\mathbf{\Sigma}_{t+1} \leftarrow \mathbf{\Sigma}_t + \boldsymbol{\phi}_{V_k}(s_t, a_t)\boldsymbol{\phi}_{V_k}(s_t, a_t)^\top$
10:         Set $\mathbf{b}_{t+1} \leftarrow \mathbf{b}_t + \boldsymbol{\phi}_{V_k}(s_t, a_t)V_k(s_{t+1})$
11:         $t \leftarrow t + 1$
12:     until det$(\mathbf{\Sigma}_t) > 2$det$(\mathbf{\Sigma}_{t_k})$
13: end for
```

Algorithm 2 Extended Value Iteration: EVI$(\mathcal{C}, U)$

```
Require: Confidence set $\mathcal{C}$, number of value iteration rounds $U$
Let $Q^{(0)}(\cdot,\cdot) = 1/(1-\gamma)$.
$Q(\cdot,\cdot) \leftarrow Q^{(0)}(\cdot,\cdot)$
if $\mathcal{C} \cap \mathcal{B} \neq \emptyset$ then
    for $u = 1, \ldots, U$ do
        Let $V^{(u-1)}(\cdot) = \max_{a\in\mathcal{A}} Q^{(u-1)}(\cdot, a)$ and
            $$Q^{(u)}(\cdot,\cdot) \leftarrow r(\cdot,\cdot) + \gamma \max_{\boldsymbol{\theta}\in\mathcal{C}\cap\mathcal{B}} \big\langle \boldsymbol{\theta}, \boldsymbol{\phi}_{V^{(u-1)}}(\cdot,\cdot)\big\rangle \qquad (4.1)$$
    end for
    Let $Q(\cdot,\cdot) \leftarrow Q^{(U)}(\cdot,\cdot)$
end if
Ensure: $Q(\cdot,\cdot)$
```

in Algorithm 2 to compute the action-value function corresponding to the near-optimal MDP among all the plausible MDPs $\mathcal{M}_k$ induced by $\mathcal{C}_k$. In detail, besides $\mathcal{C}_k$, EVI needs to access an additional set $\mathcal{B}$ defined as follows:

$$\mathcal{B} = \Big\{\boldsymbol{\theta} : \forall (s,a),\ \langle \boldsymbol{\phi}(\cdot|s,a), \boldsymbol{\theta}\rangle \text{ is a probability distribution}\Big\}.$$

The intuition of introducing set $\mathcal{B}$ is that since $\boldsymbol{\theta}^* \in \mathcal{B}$, then $\mathcal{C}_k \cap \mathcal{B}$ is a tighter confidence set of $\boldsymbol{\theta}^*$. In addition, $\mathcal{B}$ is a convex set since it is easy to verify that: for any $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathcal{B}$, and any $\alpha \in [0,1]$, we have $\alpha\boldsymbol{\theta}_1 + (1-\alpha)\boldsymbol{\theta}_2$ belongs to $\mathcal{B}$. $\mathcal{B}$ contains all possible $\boldsymbol{\theta}^*$, which can be uniquely decided by the MDP class $\mathcal{M}$. For instance, when $\mathcal{M}$ is the global convex combination MDP class (Modi et al., 2019), $\mathcal{B}$ is a d-dimensional simplex. At each iteration of Algorithm 2, to obtain the new action-value function $Q^{(u)}$, EVI performs one-step optimal value iteration (4.1) by selecting the best possible MDP $\widetilde{M}$ among $\mathcal{M}$ to maximize the Bellman backup over the previous value function $V^{(u-1)}$. This can be illustrated as follows:

$$\begin{aligned} Q^{(u)}(\cdot,\cdot) &\leftarrow r(\cdot,\cdot) + \gamma \max_{\boldsymbol{\theta}\in\mathcal{C}\cap\mathcal{B}} \big\langle \boldsymbol{\theta}, \boldsymbol{\phi}_{V^{(u-1)}}(\cdot,\cdot)\big\rangle \\ &= r(\cdot,\cdot) + \gamma \max_{\widetilde{M}\in\mathcal{M}} \big[\widetilde{\mathbb{P}}V^{(u-1)}\big](\cdot,\cdot). \end{aligned}$$

EVI returns the last action-value function as its output and sets $Q_k = Q^{(U)}$.

Execution phase (Line 7 to 12) Execution phase is used to execute the policy in each epoch, collect observations, and update parameters. At round t, Algorithm 1 follows the greedy policy π_t induced by Q_k to take the action $\pi_t(s_t)$ and observes the new state s_{t+1}. Algorithm 1 then computes vector $\boldsymbol{\phi}_{V_k}(s_t, a_t)$ according to Definition 3.1 and the value function at s_{t+1}, i.e., $V_k(s_{t+1})$. Next, Algorithm 1 updates parameters $\mathbf{\Sigma}_t$ and $\mathbf{b}_t$ by $\boldsymbol{\phi}_{V_k}(s_t, a_t)$. The loop repeats until det$(\mathbf{\Sigma}_t) > 2$det$(\mathbf{\Sigma}_{t_k})$. This is the same as the stopping criterion used by Rarely Switching OFUL in Abbasi-Yadkori et al. (2011).

Implementation of Algorithms 1 and 2 There are two main implementation issues in Algorithms 1 and 2. First, we need to compute the integration $\boldsymbol{\phi}_V(s,a)$ efficiently. Second, for Algorithm 2, we need to compute $Q(\cdot,\cdot)$ from EVI efficiently. Both of them can be efficiently achieved by Monte Carlo integration in some special cases, and we deferred the details to the appendix. Finally, it is worth noting that UCLK is an online reinforcement learning algorithm as it does not need to store all the past observations. UCLK only needs to maintain a vector $\mathbf{b}_t$ and a matrix $\mathbf{\Sigma}_t$, which costs $O(d^2)$ space complexity.

5. Main Theory

In this section, we provide the theoretical analysis of Algorithm 1. We introduce a shorthand notation $\text{Regret}(T)$ for $\text{Regret}(\text{UCLK}, M_{\boldsymbol{\theta}^*}, T)$, when there is no confusion.

We present our main theorem, which gives an upper bound of the regret for Algorithm 1.

Theorem 5.1. Let $M_{\boldsymbol{\theta}^*}$ be the underlying linear kernel MDP. If we set β and U in Algorithm 1 as follows:

$$\begin{aligned}\beta &= \frac{1}{1-\gamma}\sqrt{d\log\frac{\lambda(1-\gamma)^2+Td}{\delta\lambda(1-\gamma)^2}}+\sqrt{\lambda d},\\ U &= \left\lceil\frac{\log(T/(1-\gamma))}{1-\gamma}\right\rceil, \end{aligned} \tag{5.1}$$

then with probability at least $1-2\delta$, we have

$$\begin{aligned}\text{Regret}(T) &\leq \frac{6\beta}{1-\gamma}\sqrt{dT\log\frac{\lambda+T/(1-\gamma)^2}{\lambda}}+\frac{5}{(1-\gamma)^2}\\ &\quad+\frac{3\sqrt{T\log 1/\delta}}{(1-\gamma)^2}+\frac{3d}{(1-\gamma)^2}\log\frac{2\lambda+Td}{\lambda(1-\gamma)^2}.\end{aligned} \tag{5.2}$$

Theorem 5.1 suggests that the regret of Algorithm 1 is in the order of $\widetilde{O}(d\sqrt{T}/(1-\gamma)^2)$.

Remark 5.2. Several aspects of Theorem 5.1 are worth to comment. Thanks to the feature mapping $\boldsymbol{\phi}$ and the multi-epoch nature of Algorithm 1, the regret bound (5.2) in Theorem 5.1 is independent of $|\mathcal{S}|$ and $|\mathcal{A}|$, which suggests that UCLK is sample efficient even for MDPs with large state and action spaces. This is in sharp contrast to the tabular RL algorithms, whose regret bound or sample complexity depends on $|\mathcal{S}|$ and $|\mathcal{A}|$ polynomially. Moreover, the exploration parameter β and the number of extended value iteration rounds U depend on T logarithmically. For the case where T is unknown, we can use the "doubling trick" (Besson & Kaufmann, 2018) to learn T adaptively, and it will only increase the regret (5.2) by a constant factor.

Remark 5.3. For the tabular MDPs, UCLK uses the feature mapping in Example 3.2 with a $|\mathcal{S}|^2|\mathcal{A}|$-dimension feature mapping. In that case, UCLK has a $|\mathcal{S}|^2|\mathcal{A}|\sqrt{T}/(1-\gamma)^2$ regret according to Theorem 5.1, which is worse than that of Liu & Su (2020) considering the dependence of $|\mathcal{S}|$ and $|\mathcal{A}|$. There is no contradiction here, as in this paper, we aim to deliver a generic RL algorithm for linear kernel MDPs, which is a strictly larger class of MDPs than tabular MDPs. In fact, the regret bound in Theorem 5.1 can be improved by providing a tighter confidence set $\mathcal{C}_k$ specialized to the tabular MDP case. This is beyond the focus of this paper, and we leave it in the future work.

In addition to the upper bound result, we also prove the lower bound result. The following theorem shows a lower bound for any algorithm to learn a linear kernel MDP.

Theorem 5.4. Suppose $\gamma \geq 2/3, d \geq 2$ and $T \geq \max\{d^2/225, 5\gamma\}/(1-\gamma)$. Then for any policy π, there exists a linear kernel MDP $M_{\widetilde{\boldsymbol{\theta}}}$ such that

$$\mathbb{E}\big[\text{Regret}(\pi, M_{\widetilde{\boldsymbol{\theta}}}, T)\big] \geq \frac{\gamma d\sqrt{T}}{1600c(1-\gamma)^{1.5}}-\frac{\gamma}{(1-\gamma)^2}. \tag{5.3}$$

Remark 5.5. Theorem 5.4 suggests that when T is large enough, the lower bound of regret (5.3) is $\Omega(d\sqrt{T}/(1-\gamma)^{1.5})$. Compared with the upper regret bound $\widetilde{O}(d\sqrt{T}/(1-\gamma)^2)$, we can conclude that UCLK has an optimal dependence on the feature mapping dimension d and the time horizon T, and the dependence on the discount factor is only worse than the lower bound by a $(1-\gamma)^{-0.5}$ factor.

6. Proof Sketch of the Main Theory

In this section, we provide the proof sketches of the upper and lower bounds on the regret. The complete proofs are deferred to the appendix.

6.1. Proof Sketch of Theorem 5.1

In this section we prove Theorem 5.1. Let $K(T)-1$ be the number of epochs when Algorithm 1 executes $t=T$ rounds, and $t_{K(T)}=T+1$. We have the following technical lemmas.

Lemma 6.1. Let β be defined in (5.1). Then with probability at least $1-\delta$, for all $0 \leq k \leq K(T)-1$, we have $\mathcal{C}_k \cap \mathcal{B}$ is non-empty and $\boldsymbol{\theta}^* \in \mathcal{C}_k \cap \mathcal{B}$.

Lemma 6.1 suggests that in every epoch of Algorithm 1, $\boldsymbol{\theta}^*$ is contained in the confidence sets $\{\mathcal{C}_k \cap \mathcal{B}\}_{k=0}^{K(T)-1}$ with a high probability.

Lemma 6.2. Let the event in Lemma 6.1 hold. Then for all $0 \leq k \leq K(T)-1$, we have $1/(1-\gamma) \geq Q_k(s,a) \geq Q^*(s,a)$ for any $(s,a) \in \mathcal{S}\times\mathcal{A}$.

Lemma 6.2 suggests that in every epoch of Algorithm 1, $Q_k(s,a)$ found by EVI is an upper bound for the optimal action-value function $Q^*(s,a)$.

Recall that the goal of EVI is to find the action-value function Q_k corresponding to the optimal MDP in $\mathcal{M}_k$, which should satisfy the following optimality condition

$$Q_k(s_t,a_t) = r(s_t,a_t)+\gamma\max_{\boldsymbol{\theta}\in\mathcal{C}_k\cap\mathcal{B}}\big\langle\boldsymbol{\theta}, \boldsymbol{\phi}_{V_k}(s_t,a_t)\big\rangle.$$

However, it is impossible to find the exactly optimal value function since EVI only performs finite number of iterations. The following lemma characterizes the error of EVI after U iterations.

Lemma 6.3. Let the event in Lemma 6.1 hold. Then for any $0 \le k \le K(T) - 1$ and $t_k \le t \le t_{k+1} - 1$, there exists a $\boldsymbol{\theta}_t \in \mathcal{C}_k \cap \mathcal{B}$ such that $Q_k(s_t, a_t) \le r(s_t, a_t) + \gamma \big\langle \boldsymbol{\theta}_t, \boldsymbol{\phi}_{V_k}(s_t, a_t) \big\rangle + 2\gamma^U$.

Lemma 6.3 suggests that for any $\epsilon > 0$, EVI in Algorithm 2 only needs to perform $\log(1/\epsilon)$ iterations to achieve an ϵ-suboptimal action-value function.

Lemma 6.4. We have $K(T) \le 2d \log[(\lambda + dT)/(\lambda(1-\gamma)^2)]$.

Lemma 6.4 suggests that Algorithm 1 only needs to update its policy for $K(T) = \widetilde{O}(d)$ times, which is almost independent of the time horizon T. In sharp contrast, RL algorithms with feature mapping in the finite-horizon setting need to update their policy every H steps (Jin et al., 2019; Modi et al., 2019), which leads to $O(T/H)$ number of updates.

Proof sketch of Theorem 5.1. The regret can be decomposed as follows:

$$
\begin{aligned}
\text{Regret}(T) &= \sum_{k=0}^{K(T)-1} \sum_{t=t_k}^{t_{k+1}-1} \big[V^*(s_t) - V_t^\pi(s_t)\big] \\
&\le \sum_{k=0}^{K(T)-1} \underbrace{\sum_{t=t_k}^{t_{k+1}-1} \big[V_k(s_t) - V_t^\pi(s_t)\big]}_{E_k}, \qquad (6.1)
\end{aligned}
$$

where the inequality holds due to Lemma 6.2. E_k can be further bounded as follows by Bellman equation and Lemma 6.3.

$$
\begin{aligned}
E_k &\le 2/(1-\gamma)^2 + 2\gamma^U(t_{k+1} - t_k)/(1-\gamma) \\
&\quad + \sum_{t=t_k}^{t_{k+1}-1} \langle \boldsymbol{\theta}_t - \boldsymbol{\theta}^*, \boldsymbol{\phi}_{V_k}(s_t, a_t)\rangle/(1-\gamma) + \Xi_t,
\end{aligned} \qquad (6.2)
$$

where $\Xi_t = \big[\big[\mathbb{P}(V_k - V_{t+1}^\pi)\big](s_t, a_t) - \big(V_k(s_{t+1}) - V_{t+1}^\pi(s_{t+1})\big)\big]/(1-\gamma)$. Taking summation of (6.2) from $k = 0$ to $K(T) - 1$ and rearranging it, we obtain that $\sum_{k=0}^{K(T)-1} E_k$ is upper bounded as follows

$$
\begin{aligned}
\sum_{k=0}^{K(T)-1} E_k &\le \frac{2K(T}{(1-\gamma)^2} + \frac{2\gamma^U T}{1-\gamma} + \sum_{k=0}^{K(T)-1} \sum_{t=t_k}^{t_{k+1}-1} \Xi_t \\
&\quad + \sum_{k=0}^{K(T)-1} \sum_{t=t_k}^{t_{k+1}-1} \frac{\langle \boldsymbol{\theta}_t - \boldsymbol{\theta}^*, \boldsymbol{\phi}_{V_k}(s_t, a_t)\rangle}{1-\gamma},
\end{aligned}
$$

where the first term on the R.H.S. can be further bounded by $\widetilde{O}(d/(1-\gamma)^2)$ by Lemma 6.4, the second term can be bounded by 1 with the choice of U, the third term can be bounded by $\widetilde{O}(d\sqrt{T}/(1-\gamma)^2)$ by Lemma 6.1, and the last term can be bounded by $\widetilde{O}(\sqrt{T}/(1-\gamma)^2)$ by Azuma-Hoeffding inequality. □

6.2. Proof Sketch of Theorem 5.4

At the core of the proof of Theorem 5.4 is to construct a class of hard-to-learn MDP instances. We show the construction of these instances here and defer the detailed proof to Appendix C. Let $M(\mathcal{S}, \mathcal{A}, \gamma, r, \mathbb{P}_{\boldsymbol{\theta}})$ denote these hard MDPs. The state space $\mathcal{S}$ consists of two states x_0, x_1. The action space $\mathcal{A}$ consists of 2^{d-1} vectors $\mathbf{a} \in \{-1, 1\}^{d-1}$. The reward function r satisfies that $r(x_0, \mathbf{a}) = 0$ and $r(x_1, \mathbf{a}) = 1$ for any $\mathbf{a} \in \mathcal{A}$. The probability transition function $\mathbb{P}_{\boldsymbol{\theta}}$ is parameterized by a $(d-1)$-dimensional vector $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, $\boldsymbol{\Theta} = \{-\Delta/(d-1), \Delta/(d-1)\}^{d-1}$, which is defined as $\mathbb{P}_{\boldsymbol{\theta}}(x_0|x_0, \mathbf{a}) = 1 - \delta - \langle \mathbf{a}, \boldsymbol{\theta}\rangle$, $\mathbb{P}_{\boldsymbol{\theta}}(x_1|x_0, \mathbf{a}) = \delta + \langle \mathbf{a}, \boldsymbol{\theta}\rangle$, $\mathbb{P}_{\boldsymbol{\theta}}(x_0|x_1, \mathbf{a}) = \delta$, $\mathbb{P}_{\boldsymbol{\theta}}(x_1|x_1, \mathbf{a}) = 1 - \delta$, where δ and Δ are positive parameters that need to be determined in later proof. It can be verified that M is indeed a linear kernel MDP with the vector $\widetilde{\boldsymbol{\theta}} = (\boldsymbol{\theta}^\top, 1)^\top \in \mathbb{R}^d$ while $\Delta \le d - 1$ and the feature mapping $\boldsymbol{\phi}(s'|s, a)$ defined as follows:

$$
\begin{aligned}
\boldsymbol{\phi}(x_0|x_0, \mathbf{a}) &= \begin{pmatrix} -\mathbf{a} \\ 1 - \delta \end{pmatrix}, \boldsymbol{\phi}(x_1|x_0, \mathbf{a}) = \begin{pmatrix} \mathbf{a} \\ \delta \end{pmatrix}, \\
\boldsymbol{\phi}(x_0|x_1, \mathbf{a}) &= \begin{pmatrix} \mathbf{0} \\ \delta \end{pmatrix}, \boldsymbol{\phi}(x_1|x_1, \mathbf{a}) = \begin{pmatrix} \mathbf{0} \\ 1 - \delta \end{pmatrix}.
\end{aligned}
$$

Remark 6.5. The class of hard-to-learn linear kernel MDPs can be regarded as an extension of the hard instance in linear bandits literature (Dani et al., 2008; Lattimore & Szepesvári, 2018) to MDPs. Our constructed MDPs are similar to those in Jaksch et al. (2010); Osband & Van Roy (2016) for the average-reward MDPs and Lattimore & Hutter (2012) for the discounted MDPs. By Example 3.2, we know that tabular MDPs can be regarded as specialized linear kernel MDPs with a $|\mathcal{S}|^2|\mathcal{A}|$-dimensional feature mapping. However, simply applying the MDPs in Jaksch et al. (2010); Osband & Van Roy (2016); Lattimore & Hutter (2012) to our setting would yield a $\Omega(\sqrt{|\mathcal{S}||\mathcal{A}|T}/(1-\gamma)^{1.5})$ lower bound for regret, which is looser than our result because $\sqrt{|\mathcal{S}||\mathcal{A}|} \le |\mathcal{S}|^2|\mathcal{A}| = d$.

From now on, we set $\delta = 1 - \gamma$, $\Delta = d\sqrt{1-\gamma}/(90\sqrt{2T})$ and only consider the case where π is a deterministic policy, since the regret result of the case where π is stochastic is lower bounded by that of the deterministic one. Let N_0 denote the total visit number to state x_0. Similiarly, let N_1 denote the total visit number to state x_1, $N_0^{\mathbf{a}}$ denote the total visit number to state x_0 followed by action $\mathbf{a}$ and $N_0^{\widetilde{\mathcal{A}}}$ denote the total visit number to state x_0 followed by actions

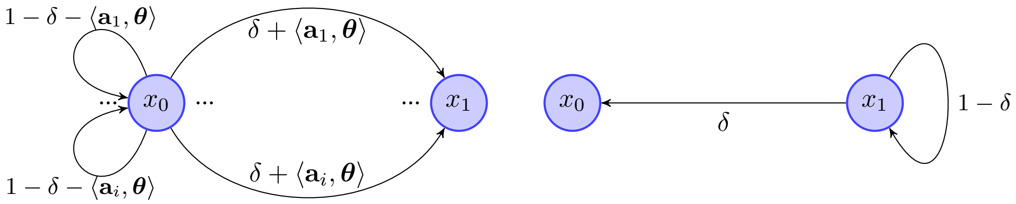
$1-\delta-\langle\mathbf{a}_1,\boldsymbol{\theta}\rangle$
$\delta+\langle\mathbf{a}_1,\boldsymbol{\theta}\rangle$
x_0
x_1
$1-\delta-\langle\mathbf{a}_i,\boldsymbol{\theta}\rangle$
$\delta+\langle\mathbf{a}_i,\boldsymbol{\theta}\rangle$
x_0
δ
x_1
$1-\delta$

Figure 1. Class of hard-to-learn linear kernel MDPs considered in Section 6.2. The left figure demonstrates the state transition probability starting from x_0 with different action $\mathbf{a}_i$. The right figure demonstrates the state transition probability starting from x_1 with any action.

in subset $\widetilde{\mathcal{A}} \subseteq \mathcal{A}$. Let $\mathcal{P}_{\boldsymbol{\theta}}(\cdot)$ denote the distribution over $\mathcal{S}^T$, where $s_1 = x_0$, $s_{t+1} \sim \mathbb{P}_{\boldsymbol{\theta}}(\cdot|s_t, a_t)$, a_t is decided by π_t. Let $\mathbb{E}_{\boldsymbol{\theta}}$ denote the expectation w.r.t. distribution $\mathcal{P}_{\boldsymbol{\theta}}$. Suppose we have an MDP $M(\mathcal{S}, \mathcal{A}, \gamma, r, \mathbb{P}_{\boldsymbol{\theta}})$. During this proof the starting state s_1 is set to be x_0. For simplicity, let Regret$(\boldsymbol{\theta})$ denote Regret$(\pi, M(\mathcal{S}, \mathcal{A}, \gamma, r, \mathbb{P}_{\boldsymbol{\theta}}), T)$ without confusion. We need the following lemmas. The first lemma shows that to bound Regret$(\boldsymbol{\theta})$, we only need to bound the summation of rewards over s_t, a_t.

Lemma 6.6. The regret Regret$(\boldsymbol{\theta})$ satisfies that

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{\theta}}\text{Regret}(\boldsymbol{\theta})\\ &\geq \mathbb{E}_{\boldsymbol{\theta}}\bigg[\sum_{t=1}^T \Big[V^*(s_t) - \frac{1}{1-\gamma}r(s_t, a_t)\Big] - \frac{\gamma}{(1-\gamma)^2}\bigg].\end{aligned}$$

Next lemma gives the relation between $\mathbb{E}_{\boldsymbol{\theta}}N_1$, $\mathbb{E}_{\boldsymbol{\theta}}N_0^{\mathbf{a}}$ and $\mathbb{E}_{\boldsymbol{\theta}}N_0$, which is useful to our proof.

Lemma 6.7. Suppose $2\Delta < \delta$ and $(1-\delta)/\delta < T/5$, then for $\mathbb{E}_{\boldsymbol{\theta}}N_1$ and $\mathbb{E}_{\boldsymbol{\theta}}N_0$, we have

$$\mathbb{E}_{\boldsymbol{\theta}}N_1 \leq \frac{T}{2} + \frac{1}{2\delta}\sum_{\mathbf{a}}\langle\mathbf{a}, \boldsymbol{\theta}\rangle\mathbb{E}_{\boldsymbol{\theta}}N_0^{\mathbf{a}}, \text{ and } \quad \mathbb{E}_{\boldsymbol{\theta}}N_0 \leq 4T/5.$$

Next lemma gives the bound for KL divergence.

Lemma 6.8. Suppose that $\boldsymbol{\theta}$ and $\boldsymbol{\theta}'$ only differs from j-th coordinate, $2\Delta < \delta \leq 1/3$. Then we have the following bound for the KL divergence between $\mathcal{P}_{\boldsymbol{\theta}}$ and $\mathcal{P}_{\boldsymbol{\theta}'}$:

$$\text{KL}(\mathcal{P}_{\boldsymbol{\theta}'}\|\mathcal{P}_{\boldsymbol{\theta}}) \leq \frac{16\Delta^2}{(d-1)^2\delta}\mathbb{E}_{\boldsymbol{\theta}}N_0.$$

Proof Sketch of Theorem 5.4. By Lemma 6.6, we only need to lower bound the difference between V^* and $r(s_t, a_t)$. We can calculate V^* through the definition of our MDP as

$$\begin{aligned}V^*(x_0) &= \frac{\gamma(\Delta+\delta)}{(1-\gamma)(\gamma(2\delta+\Delta-1)+1)},\\ V^*(x_1) &= \frac{\gamma(\Delta+\delta)+1-\gamma}{(1-\gamma)(\gamma(2\delta+\Delta-1)+1)}.\end{aligned}$$

Since $r(x_0, \mathbf{a}) = 0$ and $r(x_1, \mathbf{a}) = 1$, then the lower bound can be fully characterized by $\mathbb{E}_{\boldsymbol{\theta}}N_1$. Furthermore, we can derive that

$$\begin{aligned}\frac{1}{|\boldsymbol{\Theta}|}\sum_{\boldsymbol{\theta}}\mathbb{E}_{\boldsymbol{\theta}}N_1 \leq \frac{T}{2} + \frac{1}{4\delta}\frac{\Delta}{(d-1)|\boldsymbol{\Theta}|}\sum_{j=1}^{d-1}\sum_{\boldsymbol{\theta}}\\ \Big[\mathbb{E}_{\boldsymbol{\theta}'}N_0 + \frac{cT}{8}\sqrt{\text{KL}(\mathcal{P}_{\boldsymbol{\theta}'}\|\mathcal{P}_{\boldsymbol{\theta}})}\Big],\end{aligned} \tag{6.3}$$

where $\boldsymbol{\theta}'$ only differs from $\boldsymbol{\theta}$ at j-th coordinate. By Lemma 6.7 and 6.8 we can obtain an upper bound of (6.3) in terms of δ and Δ. Selecting $\delta = 1-\gamma$, $\Delta = d\sqrt{1-\gamma}/(90\sqrt{2T})$ gives the final result. □

7. Conclusion

We proposed a novel algorithm for solving linear kernel MDPs called UCLK. We prove that the regret of UCLK can be upper bounded by $\widetilde{O}(d\sqrt{T}/(1-\gamma)^2)$, which is the first result of its kind for learning discounted MDPs without accessing the generative model or making strong assumptions like uniform ergodicity. We also proved a lower bound $\Omega(d\sqrt{T}/(1-\gamma)^{1.5})$ which holds for any algorithm. There still exists a gap of $(1-\gamma)^{-0.5}$ between the upper and lower bounds, and we leave it as an open problem for future work.

Acknowledgement

We would like to thank the anonymous reviewers for their helpful comments. We would also like to thank Lin Yang for pointing out the related work (Jia et al., 2020; Ayoub et al., 2020), and Csaba Szepesvári for helpful comments. DZ, JH and QG are partially supported by the National Science Foundation CAREER Award 1906169, IIS-1904183 and AWS Machine Learning Research Award. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

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.

Ayoub, A., Jia, Z., Szepesvari, C., Wang, M., and Yang, L. F. Model-based reinforcement learning with value-targeted regression. *arXiv preprint arXiv:2006.01107*, 2020.

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. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 263–272. JMLR. org, 2017.

Bertsekas, D. P. Feature-based aggregation and deep reinforcement learning: A survey and some new implementations. *IEEE/CAA Journal of Automatica Sinica*, 6(1): 1–31, 2018.

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

Bhandari, J., Russo, D., and Singal, R. A finite time analysis of temporal difference learning with linear function approximation. *arXiv preprint arXiv:1806.02450*, 2018.

Boyd, S., Boyd, S. P., and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

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

Dong, K., Wang, Y., Chen, X., and Wang, L. Q-learning with ucb exploration is sample efficient for infinite-horizon mdp. *arXiv preprint arXiv:1901.09311*, 2019.

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*, 2019.

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

Jia, Z., Yang, L., Szepesvari, C., and Wang, M. Model-based reinforcement learning with value-targeted regression. 2020.

Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J., and Schapire, R. E. Contextual decision processes with low bellman rank are pac-learnable. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1704–1713. JMLR. org, 2017.

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. et al. *On the sample complexity of reinforcement learning*. PhD thesis, 2003.

Kearns, M. J. and Singh, S. P. Finite-sample convergence rates for q-learning and indirect algorithms. In *Advances in neural information processing systems*, pp. 996–1002, 1999.

Lattimore, T. and Hutter, M. Pac bounds for discounted mdps. In *International Conference on Algorithmic Learning Theory*, pp. 320–334. Springer, 2012.

Lattimore, T. and Szepesvári, C. Bandit algorithms. *preprint*, pp. 28, 2018.

Lattimore, T., Szepesvari, C., and Weisz, G. Learning with good feature representations in bandits and in rl with a generative model. In *International Conference on Machine Learning*, pp. 5662–5670. PMLR, 2020.

Liu, S. and Su, H. Regret bounds for discounted mdps, 2020.

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): 529–533, 2015.

Modi, A., Jiang, N., Tewari, A., and Singh, S. Sample complexity of reinforcement learning using linearly combined model ensembles. *arXiv preprint arXiv:1910.10597*, 2019.

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

Russo, D. and Van Roy, B. Eluder dimension and the sample complexity of optimistic exploration. In *NIPS*, pp. 2256–2264. Citeseer, 2013.

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*, 2018a.

Sidford, A., Wang, M., Wu, X., and Ye, Y. Variance reduced value iteration and faster algorithms for solving markov decision processes. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 770–787. SIAM, 2018b.

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.

Szita, I. and Szepesvári, C. Model-based reinforcement learning with nearly tight exploration complexity bounds. 2010.

Van Roy, B. and Dong, S. Comments on the du-kakade-wang-yang lower bounds. *arXiv preprint arXiv:1911.07910*, 2019.

Wang, Y., Wang, R., Du, S. S., and Krishnamurthy, A. Optimism in reinforcement learning with generalized linear function approximation. *arXiv preprint arXiv:1912.04136*, 2019.

Watkins, C. J. C. H. Learning from delayed rewards. 1989.

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

Yang, L. F. and Wang, M. Reinforcement leaning in feature space: Matrix bandit, kernels, and regret bound. *arXiv preprint arXiv:1905.10389*, 2019b.

Zanette, A., Lazaric, A., Kochenderfer, M., and Brunskill, E. Learning near optimal policies with low inherent bellman error. *arXiv preprint arXiv:2003.00153*, 2020.

Zheng, G., Zhang, F., Zheng, Z., Xiang, Y., Yuan, N. J., Xie, X., and Li, Z. Drn: A deep reinforcement learning framework for news recommendation. In *Proceedings of the 2018 World Wide Web Conference*, pp. 167–176, 2018.

Zou, S., Xu, T., and Liang, Y. Finite-sample analysis for sarsa with linear function approximation. In *Advances in Neural Information Processing Systems*, pp. 8665–8675, 2019.