Logarithmic Regret for Reinforcement Learning with Linear Function Approximation

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

Abstract

Reinforcement learning (RL) with linear function approximation has received increasing attention recently. However, existing work has focused on obtaining $\sqrt{T}$-type regret bound, where T is the number of interactions with the MDP. In this paper, we show that logarithmic regret is attainable under two recently proposed linear MDP assumptions provided that there exists a positive sub-optimality gap for the optimal action-value function. More specifically, under the linear MDP assumption (Jin et al., 2020), the LSVI-UCB algorithm can achieve $\widetilde{O}(d^3H^5/\text{gap}_{\text{min}} \cdot \log(T))$regret; and under the linear mixture MDP assumption (Ayoub et al., 2020), the UCRL-VTR algorithm can achieve $\widetilde{O}(d^2H^5/\text{gap}_{\text{min}} \cdot \log^3(T))$ regret, where d is the dimension of feature mapping, H is the length of episode, gap_{min} is the minimal sub-optimality gap, and $\widetilde{O}$ hides all logarithmic terms except $\log(T)$. To the best of our knowledge, these are the first logarithmic regret bounds for RL with linear function approximation. We also establish gap-dependent lower bounds for the two linear MDP models.

1. Introduction

Designing efficient algorithms that learn and plan in sequential decision-making tasks with large state and action spaces has become a central task of modern reinforcement learning (RL) in recent years. RL often assumes the environment as a Markov Decision Process (MDP), described by a tuple of state space, action space, reward function, and transition probability function. Due to a large number of possible states and actions, traditional tabular reinforcement learning methods such as Q-learning (Watkins, 1989), which directly access each state-action pair, are computationally intractable. A common approach to cope with high-dimensional state and action spaces is to utilize function approximation such as linear functions or neural networks to map states and actions to a low-dimensional space.

Recently, a large body of literature has been devoted to provide regret bounds for online RL with linear function approximation. These works can be divided into two main categories. The first category is *model-free* algorithms, which directly parameterize the action-value function as a linear function of some given feature mapping. For instance, Jin et al. (2020) studied the episodic MDPs with *linear MDP* assumption, which assumes that both transition probability function and reward function can be represented as a linear function of a given feature mapping. Under this assumption, Jin et al. (2020) showed that the action-value function is a linear function of the feature mapping and proposed a model-free LSVI-UCB algorithm to obtain an $\widetilde{O}(\sqrt{d^3H^3T})$ regret, where d is the dimension of the feature mapping, H is the length of the episode, and T is the number of interactions with the MDP. The second category is *model-based* algorithms, which parameterize the underlying transition probability function as a linear function of a given feature mapping. For example, Ayoub et al. (2020) studied a class of MDPs where the underlying transition probability kernel is a linear mixture model. Ayoub et al. (2020) proposed a model-based UCRL-VTR algorithm with an $\widetilde{O}(d\sqrt{H^3T})$ regret. Zhou et al. (2020b) studied the linear kernel MDP[1] in the infinite horizon discounted setting and proposed a algorithm with $\sqrt{T}$-type regret. Although these $\sqrt{T}$-type regrets are standard and easy to interpret, they do not consider any additional *problem-dependent structure* of the underlying MDPs. This motivates us to seek a tighter and instance-dependent regret analysis for RL.

There is a large body of literature on bandits, which study the instance-dependent regret bounds (See Bubeck and Cesa-Bianchi (2012); Slivkins et al. (2019); Lattimore and Szepesvári (2018) and references therein). Note that bandits can be seen as a special instance of RL problems. Sub-optimality gap has been playing a central role in many gap-dependent bounds for bandits, which is defined as gap

[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).

[1] Linear kernel MDPs are essentially the same as linear mixture MDPs.

between the optimal action and the rest ones. For general RL, previous works (Simchowitz and Jamieson, 2019; Yang et al., 2020) have considered the tabular MDP with sub-optimality gap and proved gap-dependent regret bounds. However, as far as we know, there does not exist such gap-dependent regret results for RL with linear function approximation. Therefore, a natural question arises:

Can we derive instance/gap-dependent regret bounds for RL with linear function approximation?

We answer the above question affirmatively in this paper. In detail, following Simchowitz and Jamieson (2019); Yang et al. (2020), we consider an instance-dependent quantity called *gap*$_{min}$, which is the minimal sub-optimality gap for the optimal action-value function. Under the assumption that $\text{gap}_{\min}$ is strictly positive, we show that LSVI-UCB proposed in Jin et al. (2020) achieves a $\widetilde{O}(d^3H^5/\text{gap}_{\min}\cdot\log(T))$ regret, and UCRL-VTR proposed by Ayoub et al. (2020) achieves a regret of order $\widetilde{O}(d^2H^5/\text{gap}_{\min}\cdot\log^3(T))$. Furthermore, we prove an $\Omega(dH/\text{gap}_{\min})$ lower bound on the regret for both linear MDPs and linear mixture MDPs. To the best of our knowledge, this is the first instance-dependent $\log T$-type regret achieved by RL with linear function approximation. Our results suggest that the dependence on T in regrets can be drastically decreased from $\sqrt{T}$ to $\log T$ when considering the problem structure for both model-free and model-based RL algorithms with linear function approximation.

Notation We use lower case letters to denote scalars, and use lower and upper case bold face letters to denote vectors and matrices respectively. For any positive integer n, we denote by $[n]$ the set $\{1,\ldots,n\}$. For a vector $\mathbf{x}\in\mathbb{R}^d$, we denote by $\|\mathbf{x}\|_1$ the Manhattan norm and denote by $\|\mathbf{x}\|_2$ the Euclidean norm. For a vector $\mathbf{x}\in\mathbb{R}^d$ and matrix $\mathbf{\Sigma}\in\mathbb{R}^{d\times d}$, we define $\|\mathbf{x}\|_{\mathbf{\Sigma}}=\sqrt{\mathbf{x}^\top\mathbf{\Sigma}\mathbf{x}}$. For two sequences $\{a_n\}$ and $\{b_n\}$, we write $a_n=O(b_n)$ if there exists an absolute constant C such that $a_n\le Cb_n$. We use $\widetilde{O}(\cdot)$ to further hide the logarithmic factors. For logarithmic regret, we use $\widetilde{O}(\cdot)$ to hide all logarithmic terms except $\log T$.

2. Related Work

Logarithmic regret bound for RL. A line of works focus on proving $\log T$-style regret bounds for RL algorithms based on problem-dependent quantities. It has been shown that such a $\log T$ dependence is unavoidable according to the lower bound results shown in Ok et al. (2018). For the upper bounds, Auer and Ortner (2007) showed that the UCRL algorithm achieves logarithmic regret in the average reward setting, while the regret bound depends on both the hitting time and the policy sub-optimal gap. Tewari and Bartlett (2008) proposed an OLP algorithm for average-reward MDP and showed that OLP achieves logarithmic regret $O\big(C(P)\log T\big)$ where $C(P)$ is an explicit MDP-dependent constant. Both results in Auer and Ortner (2007) and Tewari and Bartlett (2008) are asymptotic, which required the number of steps T is large enough. For non-asymptotic bounds, Jaksch et al. (2010) proposed a UCRL2 algorithm for average-reward MDP with regret $O\big(D^2S^2A\log(T)/\Delta\big)$, where D is the diameter of the MDP and Δ is the policy sub-optimal gap. For episodic MDPs, Simchowitz and Jamieson (2019) proposed a model-based StrongEuler algorithm with a logarithmic regret, and proved a regret lower bound for tabular MDPs that depends on the minimal sub-optimality gap. Recently, Yang et al. (2020) showed that the model-free algorithm optimistic Q-learning achieves $O\big(SAH^6\log(SAT)/\text{gap}_{\min}\big)$ regret. However, all the above results are limited to tabular MDPs.

Linear function approximation. Recently, there has emerged a large body of literature on learning MDPs with linear function approximation. These results can be categorized based on their assumptions on the MDPs. The first category of works consider linear MDPs (Yang and Wang, 2019; Jin et al., 2020). Jin et al. (2020) proposed LSVI-UCB algorithm with $\widetilde{O}(\sqrt{d^3H^3T})$ regret. Wang et al. (2019b) proposed USVI-UCB algorithm in a weaker assumption called "optimistic closure" and achieved $\widetilde{O}(H\sqrt{d^3T})$ regret. Zanette et al. (2020) proposed a weaker assumption which is called low inherent Bellman error, and improved the regret to $\widetilde{O}(dH\sqrt{T})$ by considering a global planning oracle. Jiang et al. (2017) studied a larger class of MDPs with low Bellman rank and proposed an OLIVE algorithm with polynomial sample complexity. The second line of works consider linear mixture MDPs (Jia et al., 2020; Ayoub et al., 2020). Jia et al. (2020) and Ayoub et al. (2020) proposed UCLR-VTR algorithm for episodic MDPs which achieves $\widetilde{O}(d\sqrt{H^3T})$ regret. Cai et al. (2019) proposed policy optimization algorithm OPPO which achieves $\widetilde{O}(\sqrt{d^2H^3T})$ regret. Zhou et al. (2020b) focused on infinite-horizon discounted setting and proposed a UCLK algorithm, which achieves $\widetilde{O}(d\sqrt{T}/(1-\gamma)^2)$ regret.

3. Preliminaries

We consider episodic Markov Decision Processes (MDP) which can be denoted by a tuple $\mathcal{M}(\mathcal{S},\mathcal{A},H,\{r_h\}_{h=1}^H,\{\mathbb{P}_h\}_{h=1}^H)$. Here, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the finite action space, H is the length of each episode, $r_h:\mathcal{S}\times\mathcal{A}\to[0,1]$ is the reward function at step h and $\mathbb{P}_h(s'|s,a)$ is the transition probability function at step h which denotes the probability for state s to transfer to state s' with action a at step h.

A policy $\pi:\mathcal{S}\times[H]\to\mathcal{A}$ is a function which maps a state s and the step number h to an action a. For any policy π and step $h\in[H]$, we denote the action-value function $Q_h^\pi(s,a)$

and value function $V_h^\pi(s)$ as follows

$$Q_h^\pi(s,a) = r_h(s,a) + \mathbb{E}\bigg[\sum_{h'=h+1}^{\infty} r_{h'}\big(s_{h'}, \pi(s_{h'}, h')\big)|s_h = s, a_h = a\bigg],$$
$$V_h^\pi(s) = Q_h^\pi(s, \pi(s,h)),$$

where $s_{h'+1} \sim \mathbb{P}_h(\cdot|s_{h'}, a_{h'})$. We define the optimal value function V_h^* and the optimal action-value function Q_h^* as $V_h^*(s) = \sup_\pi V_h^\pi(s)$ and $Q^*(s,a) = \sup_\pi Q^\pi(s,a)$. By definition, the value function $V_h^\pi(s)$ and action-value function $Q_h^\pi(s,a)$ are bounded in $[0,H]$. For simplicity, for any function $V : \mathcal{S} \to \mathbb{R}$, we denote $[\mathbb{P}_h V](s,a) = \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s,a)} V(s')$. Therefore, for each $h \in [H]$ and policy π, we have the following Bellman equation, as well as the Bellman optimality equation:

$$Q_h^\pi(s,a) = r_h(s,a) + [\mathbb{P}_h V_{h+1}^\pi](s,a),$$
$$Q_h^*(s,a) = r_h(s,a) + [\mathbb{P}_h V_{h+1}^*](s,a), \qquad (3.1)$$

where $V_{H+1}^\pi = V_{H+1}^* = 0$. In the *online learning setting*, for eack episode $k \geq 1$, at the beginning of the episode k, the agent determine a policy π_k to be followed in this episode. At each step $h \in [H]$, the agent observe the state s_h^k, choose an action following the policy π_k and observe the next state with $s_{h+1}^k \sim \mathbb{P}_h(\cdot|s_h^k, a_h^k)$. Furthermore, we define the total regret in the first K episodes as follows.

Definition 3.1. For any algorithm, we define its regret on MDP $M(\mathcal{S}, \mathcal{A}, H, r, \mathbb{P})$ in the first K episodes as the sum of the suboptimality for epsiode $k = 1, \ldots, K$, i.e.,

$$\text{Regret}(K) = \sum_{k=1}^K V_1^*(s_1^k) - V_1^{\pi_k}(s_1^k),$$

where π_k is the policy in the episodes k.

In this paper, we focus on the minimal sub-optimality gap condition (Simchowitz and Jamieson, 2019; Du et al., 2019; 2020; Yang et al., 2020; Mou et al., 2020) and linear function approximation (Jin et al., 2020; Ayoub et al., 2020; Jia et al., 2020; Zhou et al., 2020b).

Definition 3.2 (Minimal sub-optimality gap)**.** For each $s \in \mathcal{S}, a \in \mathcal{A}$ and step $h \in [H]$, the sub-optimality gap $\text{gap}_h(s,a)$ is defined as

$$\text{gap}_h(s,a) = V_h^*(s) - Q_h^*(s,a),$$

and the minimal sub-optimality gap is defined as

$$\text{gap}_{\min} = \min_{h,s,a} \big\{\text{gap}_h(s,a) : \text{gap}_h(s,a) \neq 0\big\}. \qquad (3.2)$$

In this paper, we assume the minimal sub-optimality gap is strictly positive.

Assumption 3.3. The minimal sub-optimality gap is strictly positive, i.e., $\text{gap}_{\min} > 0$.

4. Model-free RL

In this section, we focus on model-free RL algorithms with linear function approximation. We make the following linear MDP assumption (Jin et al., 2020; Yang and Wang, 2019) where the probability transition kernels and the reward functions are assumed to be linear with respect to a given feature mapping $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$.

Assumption 4.1. MDP $\mathcal{M}(\mathcal{S}, \mathcal{A}, H, \{r_h\}_{h=1}^H, \{\mathbb{P}_h\}_{h=1}^H)$ is a linear MDP such that for any step $h \in [H]$, there exists an unknown vector $\boldsymbol{\mu}_h$, unknown measures $\boldsymbol{\theta}_h = \big(\boldsymbol{\theta}_h^{(1)}, .., \boldsymbol{\theta}_h^{(d)}\big)$ and a known feature mapping $\boldsymbol{\phi} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, where for each $(s,a) \in \mathcal{S} \times \mathcal{A}$ and $s' \in \mathcal{S}$,

$$\mathbb{P}_h(s'|s,a) = \big\langle \boldsymbol{\phi}(s,a), \boldsymbol{\theta}_h(s') \big\rangle, r_h(s,a) = \big\langle \boldsymbol{\phi}(s,a), \boldsymbol{\mu}_h \big\rangle.$$

For simplicity, we assume that the unknown vector $\boldsymbol{\mu}_h$ and feature $\boldsymbol{\phi}(s,a)$ satisfy $\|\boldsymbol{\phi}(s,a)\|_2 \leq 1$, $\|\boldsymbol{\mu}_h\|_2 \leq \sqrt{d}$ and $\big\|\boldsymbol{\theta}_h(\mathcal{S})\big\| \leq \sqrt{d}$.

Remark 4.2. Under Assumption 4.1, by the Bellman equation (3.1), it can be shown that for any policy π, the action-value function $Q_h^\pi(s,a)$ is a linear function $\langle \boldsymbol{\phi}(s,a), \boldsymbol{\theta}_h^\pi \rangle$ with respect to the feature mapping $\boldsymbol{\phi}$, where $\boldsymbol{\theta}_h^\pi$ is a vector decided by the policy π. This suggests to estimate the unknown optimal action-value function Q_h^*, we only need to estimate its corresponding parameter $\boldsymbol{\theta}_h^*$.

Remark 4.3. Though the probability transition kernel and the reward function are linear with respect to $\boldsymbol{\phi}(s,a)$, the degree of freedom of measure $\boldsymbol{\theta}_h$ is $|\mathcal{S}| \times d$. Therefore, when $\mathcal{S}$ is large, it is computationally intractable to directly estimate the probability transition kernel $\mathbb{P}_h$.

4.1. Algorithm

We analyze the LSVI-UCB algorithm proposed in Jin et al. (2020), which is showed in Algorithm 1. At a high level, Algorithm 1 treats the optimal action-value function Q_h^* as a linear function of the feature $\boldsymbol{\phi}$ and an unknown parameter $\boldsymbol{\theta}_h^*$. The goal of Algorithm 1 is to estimate $\boldsymbol{\theta}_h^*$. Algorithm 1 directly estimates the action-value function, and that is why it is “model-free”. Algorithm 1 uses the least-square value iteration to estimate the $\boldsymbol{\theta}_h^*$ for each h with additional exploration bonuses. In Line 5, Algorithm 1 computes $\mathbf{w}_h^k$, the estimate of $\boldsymbol{\theta}_h^*$, by solving a regularized least-square problem:

$$\mathbf{w}_h^k \leftarrow \mathop{\text{argmin}}_{\mathbf{w} \in \mathbb{R}^d} \lambda\|\mathbf{w}\|_2^2 + \sum_{i=1}^{k-1} \big(\boldsymbol{\phi}(s_h^i, a_h^i)^\top \mathbf{w} - r_h(s_h^i, a_h^i) - \max_a Q_{h+1}^k(s_{h+1}^i, a)\big)^2.$$

In Line 6, Algorithm 1 computes the action-value function $Q_h^k(s,a)$ by $\mathbf{w}_h^k$ and adds a UCB bonus to make sure the estimate of action-value function $Q_h^k(s,a)$ is an upper bound

Algorithm 1 Least Square Value-iteration with UCB (LSVI-UCB) (Jin et al., 2020)

```
1:  for episodes $k = 1, \dots, K$ do
2:    Received the initial state $s_1^k$.
3:    for step $h = H, \dots, 1$ do
4:      $\Lambda_h^k = \sum_{i=1}^{k-1} \boldsymbol{\phi}(s_h^i, a_h^i)\boldsymbol{\phi}(s_h^i, a_h^i)^\top + \lambda \cdot \mathbf{I}$
5:      $\mathbf{w}_h^k = (\Lambda_h^k)^{-1} \sum_{i=1}^{k-1} \boldsymbol{\phi}(s_h^i, a_h^i)\big[r_h(s_h^i, a_h^i) + \max_a Q_{h+1}^k(s_{h+1}^i, a)\big]$
6:      $Q_h^k(s,a) = \min\big\{\beta\sqrt{\boldsymbol{\phi}(s,a)^\top(\Lambda_h^k)^{-1}\boldsymbol{\phi}(s,a)} + \mathbf{w}_h^\top\boldsymbol{\phi}(s,a), H\big\}$
7:    end for
8:    for step $h = 1, \dots, H$ do
9:      Take action $a_h^k \leftarrow \mathrm{argmax}_a\, Q_h^k(s_h^k, a)$ and receive next state $s_{h+1}^k$
10:   end for
11: end for
```

of the optimal action-value function $Q_h^*(s,a)$. In Line 9, a greedy policy with respect to estimated action-value function $Q_h^k(s,a)$ is used to choose action and transit to the next state.

4.2. Regret analysis

In this subsection, we present our regret analysis for LSVI-UCB. For simplicity, we denote $T = KH$, which is the total number of steps.

Theorem 4.4. Under Assumptions 3.3 and 4.1, there exists a constant C such that, if we set $\lambda = 1$, $\beta = 78dH\sqrt{\log(2dT/\delta)}$ in Algorithm 1, then with probability at least $1 - 2(K+1)H\log(H/\mathrm{gap}_{\min})\delta - \log T\delta$, the regret for Algorithm 1 in first T steps is upper bounded by

$$\mathrm{Regret}(K) \le \frac{9Cd^3H^5\log(2dT/\delta)}{\mathrm{gap}_{\min}}\iota + \frac{16H^2\log\delta}{3},$$

where ι is defined as follows:

$$\iota = \log\bigg(\frac{Cd^3H^4\log(2dT/\delta)}{\mathrm{gap}_{\min}^2}\bigg).$$

Remark 4.5. If we set the δ in Theorem 4.4 as $\delta = 1/(2K(K+1)H^3)$ and define the high probability event Ω as: $\{$Theorem 4.4 holds$\}$. Then, for the expected regret, we have

$$\begin{aligned}
&\mathbb{E}\big[\mathrm{Regret}(K)\big] \\
&\le \mathbb{E}\big[\mathrm{Regret}(K)|\Omega\big]\Pr[\Omega] + T\Pr[\bar{\Omega}] \\
&\le \frac{9Cd^3H^5\log(2dT/\delta)}{\mathrm{gap}_{\min}}\iota + \frac{16H^2\log\delta}{3} + T\Pr[\bar{\Omega}] \\
&= \widetilde{O}(d^3H^5/\mathrm{gap}_{\min}\log T).
\end{aligned}$$

The regret bound in Theorem 4.4 is independent of the size of the state space $\mathcal{S}$, action space $\mathcal{A}$, and is only logarithmic in the number of steps T, which suggests that Algorithm 1 is sample efficient for MDPs with large state and action spaces. To our knowledge, this is the first theoretical result that achieves logarithmic regret for model-free RL with linear function approximation. Besides, the UCB bonus parameter β depends on T logarithmically. When the number of steps T is unknown at the beginning, we can use the "doubling trick" (Besson and Kaufmann, 2018) to learn T adaptively, and the regret will only be increased by a constant factor.

The following theorem gives a lower bound of the regret for any algorithm learning linear MDPs.

Theorem 4.6. Suppose $\mathrm{gap}_{\min} \le 1/(3dH), H \ge 3$, then for any algorithm, there exist a linear MDP such that expected regret is lower bounded by

$$\mathbb{E}\big[\mathrm{Regret(K)}\big] \ge \Omega\bigg(\frac{Hd}{\mathrm{gap}_{\min}}\bigg).$$

5. Model-based RL

In this section we focus on model-based RL with linear function approximation. We make the following linear mixture MDP assumption (Jia et al., 2020; Ayoub et al., 2020; Zhou et al., 2020b), which assumes that the unknown transition probability function is an aggregation of several known basis models.

Assumption 5.1. MDP $\mathcal{M}(\mathcal{S}, \mathcal{A}, H, \{r\}_{h=1}^H, \{\mathbb{P}_h\}_{h=1}^H)$ is called a linear mixture MDP if there exists an unknown vector $\boldsymbol{\theta}_h^* \in \mathbb{R}^d$ with $\|\boldsymbol{\theta}_h^*\|_2 \le C_{\boldsymbol{\theta}}$ and a known feature mapping $\boldsymbol{\phi}(s'|s,a) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}^d$, such that

- For any state-action-next-state triplet $(s,a,s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}$, we have $\mathbb{P}_h(s'|s,a) = \langle\boldsymbol{\phi}(s'|s,a), \boldsymbol{\theta}_h^*\rangle$; Moreover, the reward function r is *deterministic and known*.
- For any bounded function $V : \mathcal{S} \to [0,1]$ and any tuple $(s,a) \in \mathcal{S} \times \mathcal{A}$, we have $\|\boldsymbol{\phi}_V(s,a)\|_2 \le 1$, where $\boldsymbol{\phi}_V(s,a) = \sum_{s' \in \mathcal{S}} \boldsymbol{\phi}(s'|s,a)V(s') \in \mathbb{R}^d$.

5.1. Algorithm

In this subsection, we analyze the model-based UCRL with the Value-Targeted Model Estimation (UCRL-VTR) algorithm proposed in Jia et al. (2020); Ayoub et al. (2020), which is shown in Algorithm 2. It is worth noting that the original UCRL-VTR algorithm is designed for the time-homogeneous MDP, where the transition probability functions $\mathbb{P}_h$ are identical across different step h. In this paper, we consider the time-inhomogeneous MDP and therefore propose the following time-inhomogeneous version of UCRL-VTR algorithm, which is slightly different from

Algorithm 2 UCRL with Value-Targeted Model Estimation (UCRL-VTR) (Jia et al., 2020; Ayoub et al., 2020)

```
1: Set $\mathbf{\Sigma}_h^1 = \lambda\mathbf{I}$, $\mathbf{b}_h^1 = \mathbf{0}$
2: for episodes $k = 1, \dots, K$ do
3:     Compute $\boldsymbol{\theta}_{k,h} \leftarrow (\mathbf{\Sigma}_h^k)^{-1}\mathbf{b}_h^k$
4:     for step $h = H, \dots, 1$ do
5:         $Q_h^k(s,a) = r(s,a) + \boldsymbol{\phi}_{V_{h+1}^k}(s,a)^\top \boldsymbol{\theta}_{k,h} + \beta_k\sqrt{\big(\boldsymbol{\phi}_{V_{h+1}^k}(s,a)\big)^\top (\mathbf{\Sigma}_h^k)^{-1}\boldsymbol{\phi}_{V_{h+1}^k}(s,a)}$
6:     end for
7:     Received the initial state $s_1^k$
8:     for step $h = 1, \dots, H$ do
9:         Take action $a_h^k \leftarrow \mathrm{argmax}_a\, Q_h^k(s_h^k, a)$ and receive next state $s_{h+1}^k$
10:        Update value matrix $\mathbf{\Sigma}$ and vector $\mathbf{b}$:
11:        $\mathbf{\Sigma}_h^{k+1} \leftarrow \mathbf{\Sigma}_h^k + \boldsymbol{\phi}_{V_{h+1}^k}(s_h^k, a_h^k)\big(\boldsymbol{\phi}_{V_{h+1}^k}(s_h^k, a_h^k)\big)^\top$
12:        $\mathbf{b}_h^{k+1} = \mathbf{b}_h^k + V_{h+1}^k(s_{h+1}^k) \cdot \boldsymbol{\phi}_{V_{h+1}^k}(s_h^k, a_h^k)$
13:    end for
14: end for
```

the original algorithm. At a high level, unlike Algorithm 1 which treats the action-value function as a linear function, Algorithm 2 treats the transition probability function as a linear function of the feature mapping $\boldsymbol{\phi}(\cdot|\cdot,\cdot)$ and an unknown parameter $\boldsymbol{\theta}^*$. The goal of Algorithm 2 is to estimate $\boldsymbol{\theta}^*$, which makes Algorithm 2 a model-based algorithm since it directly estimates the underlying transition model. To estimate $\boldsymbol{\theta}^*$, Algorithm 2 computes the estimate $\boldsymbol{\theta}_{k+1}$ by solving the following regularized least-square problem in Line 3:

$$\boldsymbol{\theta}_{k+1,h} \leftarrow \underset{\boldsymbol{\theta}\in\mathbb{R}^d}{\mathrm{argmin}}\, \lambda\|\boldsymbol{\theta}\|_2^2 + \sum_{i=1}^{k}\big(\boldsymbol{\phi}_{V_{h+1}^i}(s_h^i, a_h^i)^\top\boldsymbol{\theta} - V_{h+1}^i(s_{h+1}^i)\big)^2,$$

where for any value function $V : \mathcal{S} \to \mathbb{R}$, we denote $\boldsymbol{\phi}_V(s,a) = \sum_{s'\in\mathcal{S}}\boldsymbol{\phi}(s'|s,a)V(s') \in \mathbb{R}^d$. The close-form solution to $\boldsymbol{\theta}_{k+1}$ can be computed by considering the accumulated covariance matrix $\mathbf{\Sigma}_1^{k+1}$ in Line 11 and 12. To guarantee exploration, in Line 5, Algorithm 2 computes the action-value function $Q_h^{k+1}(s,a)$ by $\boldsymbol{\theta}_{k+1}$ and adds a UCB bonus to make sure the estimate of action-value function $Q_h^{k+1}(s,a)$ is an upper bound of the optimal action-value function $Q_h^*(s,a)$. Algorithm 2 then follows the greedy policy induced by the estimated action-value function $Q_h^k(s,a)$ in Line 5.

5.2. Regret Analysis

In this subsection, we propose our regret analysis for UCRL-VTR. For simplicity, we denote $T = KH$, which is the total number of steps.

Theorem 5.2. Suppose Assumption 3.3 and Assumption 5.1 hold. If we set $\lambda = H^2 d$ and $\beta_k = 4C_\theta H\sqrt{d\log(1+Hk)\log^2\big((k+1)^2H/\delta\big)}$ in Algorithm 2, then with probability at least $1 - 2(K+1)H\log(H/\mathrm{gap}_{\min})\delta - \log T\delta$, the regret for Algorithm 2 in first T steps is upper bounded by

$$\mathrm{Regret}(K) \le \frac{4097C_{\boldsymbol{\theta}}^2 d^2 H^5 \log^3(2dT/\delta)}{\mathrm{gap}_{\min}}\iota + \frac{16H^2\log\delta}{3},$$

where ι is defined as follows:

$$\iota = \log\bigg(\frac{512C_{\boldsymbol{\theta}}^2 d^2 H^4 \log^3(2dT/\delta)}{\mathrm{gap}_{\min}^2}\bigg).$$

Remark 5.3. If we set the δ in Theorem 4.4 as $\delta = 1/(2K(K+1)H^3)$ and define the high probability event Ω as: $\{$Theorem 4.4 holds$\}$. Then, for the expected regret, we have

$$\begin{aligned}
&\mathbb{E}\big[\mathrm{Regret}(K)\big] \\
&\le \mathbb{E}\big[\mathrm{Regret}(K)|\Omega\big]\Pr[\Omega] + T\Pr[\bar{\Omega}] \\
&\le \frac{4097C_{\boldsymbol{\theta}}^2 d^2 H^5 \log^3(2dT/\delta)}{\mathrm{gap}_{\min}}\iota + \frac{16H^2\log\delta}{3} + T\Pr[\bar{\Omega}] \\
&= \widetilde{O}(d^2H^5/\mathrm{gap}_{\min}\log T).
\end{aligned}$$

The regret bound in Theorem 5.2 depends on $\mathrm{gap}_{\min}$ inversely. It is independent of the size of the state, action space $\mathcal{S}, \mathcal{A}$, and is logarithmic in the number of steps T, similar to that of Theorem 4.4. This suggests that model-based RL with linear function approximation also enjoys a $\log T$-type regret considering the problem structure.

Similar to the model-free setting, the following theorem gives a lower bound of the regret for any algorithm learning linear mixture MDPs.

Theorem 5.4. Suppose $\mathrm{gap}_{\min} \le 1/(3dH), H \ge 3$, then for any algorithm, there exist a linear mixture MDP such that $C_\theta = 2$ and the lower bounded of the expected regret is bounded by

$$\mathbb{E}\big[\mathrm{Regret(K)}\big] \ge \Omega\bigg(\frac{Hd}{\mathrm{gap}_{\min}}\bigg).$$

6. Proof of the Main Results

In this section, we give a proof outline of Theorem 4.4, along with the proofs of the key technical lemmas.

6.1. Proof outline of Theorem 4.4

The proof can be divided into three main steps.

Step 1: Regret decomposition

Our goal is to upper bound the total regret $\text{Regret}(K)$. Following the regret decomposition procedure proposed in Simchowitz and Jamieson (2019); Yang et al. (2020), for a given policy π, we rewrite the sub-optimality $V_h^*(s_h) - V_h^{\pi_k}(s_h)$ as follows:

$$\begin{aligned}
&V_h^*(s_h) - V_h^{\pi}(s_h) \\
&= \big(V_h^*(s_h) - Q_h^*(s_h, a_h)\big) + \big(Q_h^*(s_h, a_h) - V_h^{\pi_k}(s_h)\big) \\
&= \text{gap}_h(s_h, a_h) + \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot|s_h, a_h)}\big[V_{h+1}^*(s') - V_{h+1}^{\pi_k}(s')\big],
\end{aligned} \tag{6.1}$$

where $a_h = \pi(s_h, h)$ and $\text{gap}_h(s, a) = V_h^*(s) - Q_h^*(s, a)$. Taking expectation on both sides of (6.1) with respect to the randomness of state-transition and taking summation over all $h \in [H]$, for any policy π and initial state s_1^k, we have

$$V_1^*(s_1^k) - V_1^{\pi}(s_1^k) = \mathbb{E}\bigg[\sum_{h=1}^H \text{gap}_h(s_h, a_h)\bigg], \tag{6.2}$$

where $s_1 = s_1^k$ and for each $h \in [H]$, $a_h = \pi(s_h, h)$,$s_{h+1} \sim \mathbb{P}_h(\cdot|s_h, a_h)$. Taking summation of (6.2) over all $k \in [K]$ with $\pi = \pi_k$, we have

$$\mathbb{E}\big[\text{Regret}(K)\big] = \mathbb{E}\bigg[\sum_{k=1}^K \sum_{h=1}^H \text{gap}_h(s_h^k, a_h^k)\bigg]. \tag{6.3}$$

Furthermore, we have

Lemma 6.1. For each MDP $\mathcal{M}(\mathcal{S}, \mathcal{A}, H, r_h, \mathbb{P}_h)$ and any $\tau > 0$, with probability at least $1 - me^{-\tau}$, we have

$$\text{Regret}(K) \le 2\sum_{k=1}^K \sum_{h=1}^H \text{gap}_h(s_h^k, a_h^k) + \frac{16H^2\tau}{3} + 2.$$

where $m = \lceil \log T \rceil$.

Lemma 6.1 and (6.2) suggest that the total (expected) regret can be represented as a summation of $\text{gap}_h(s_h^k, a_h^k)$ over time step h and episode k. Therefore, to bound the total regret, it suffices to bound each $\text{gap}_h(s_h^k, a_h^k)$ separately, which leads to our next proof step.

Step 2: Bound the number of sub-optimalities

Recall the range of sub-optimality gap $\text{gap}_h(s_h^k, a_h^k)$ is $[\text{gap}_{\min}, H]$. Therefore, to bound the summation of $\text{gap}_h(s_h^k, a_h^k)$, it suffices to divide the range $[\text{gap}_{\min}, H]$ into several intervals and count the number of $\text{gap}_h(s_h^k, a_h^k)$ falling into each interval. Such a division is also used in Yang et al. (2020) which is similar to the "peeling technique" widely used in local Rademacher complexity analysis (Bartlett et al., 2005). Formally speaking, we divide the interval $[\text{gap}_{\min}, H]$ to $N = \big\lceil \log(H/\text{gap}_{\min})\big\rceil$ intervals $\big[2^{i-1}\text{gap}_{\min}, 2^i\text{gap}_{\min}\big)\big(i \in [N]\big)$. Therefore, for each $\text{gap}_h(s_h^k, a_h^k)$ falling into $\big[2^{i-1}\text{gap}_{\min}, 2^i\text{gap}_{\min}\big)\big(i \in [N]\big)$, it can be upper bounded by $2^i\text{gap}_{\min}$. Meanwhile, we have the following inequality by considering $V_h^*(s_h^k) - Q_h^{\pi_k}(s_h^k, a_h^k)$, which is the upper bound of $\text{gap}_h(s_h^k, a_h^k)$:

$$V_h^*(s_h^k) - Q_h^{\pi_k}(s_h^k, a_h^k) \ge \text{gap}_h(s_h^k, a_h^k) \ge 2^{i-1}\text{gap}_{\min},$$

which suggests that to count how many $\text{gap}_h(s_h^k, a_h^k)$ belong to the interval $\big[2^{i-1}\text{gap}_{\min}, 2^i\text{gap}_{\min}\big)\big(i \in [N]\big)$, we only need to count the number of sub-optimalities $V_h^*(s_h^k) - Q_h^{\pi_k}(s_h^k, a_h^k)$ belonging to the interval. The following lemma is our main technical lemma. It is inspired by Jin et al. (2020), and it shows that the number of sub-optimalities can indeed be upper bounded.

Lemma 6.2. There exist a constant C such that, for any $h \in [H]$, $n \in N$, with probability at least $1 - (K+1)\delta$, we have

$$\begin{aligned}
&\sum_{k=1}^K \mathbb{1}\left[V_h^*(s_h^k) - Q_h^{\pi_k}(s_h^k, a_h^k) \ge 2^n\text{gap}_{\min}\right] \\
&\le \frac{Cd^3H^4\log(2dT/\delta)}{4^n\text{gap}_{\min}^2}\log\left(\frac{Cd^3H^4\log(2dT/\delta)}{4^n\text{gap}_{\min}^2}\right).
\end{aligned}$$

Step 3: Summation of total error

Lemma 6.2 gives an upper bound on the number of $\text{gap}_h(s_h^k, a_h^k)$ in each interval $\big[2^{i-1}\text{gap}_{\min}, 2^i\text{gap}_{\min}\big)$. We further give the following upper bound for the $\text{gap}_h(s_h^k, a_h^k)$ within each interval:

$$\begin{aligned}
&\sum_{\text{gap}_h(s_h^k, a_h^k) \in \left[2^{i-1}\text{gap}_{\min}, 2^i\text{gap}_{\min}\right)} \text{gap}_h(s_h^k, a_h^k) \\
&\le \sum_{k=1}^K 2^i\text{gap}_{\min}\, \mathbb{1}\left[\text{gap}_h(s_h^k, a_h^k) \in \left[2^{i-1}\text{gap}_{\min}, 2^i\text{gap}_{\min}\right)\right] \\
&\le \sum_{k=1}^K 2^i\text{gap}_{\min}\, \mathbb{1}\left[V_h^*(s_h^k) - Q_h^{\pi_k}(s_h^k, a_h^k) \ge 2^{i-1}\text{gap}_{\min}\right].
\end{aligned}$$

Thus, by using the upper bound on the number of $\text{gap}_h(s_h^k, a_h^k)$ in Lemma 6.2, we have the following lemma:

Lemma 6.3. There exist a constant C such that, for $h \in [H]$, with probability at least $1 - 2(K+1)\log(H/\text{gap}_{\min})\delta$, we have

$$\begin{aligned}
&\sum_{k=1}^K \big(V_h^*(s_h^k) - Q_h^*(s_h^k, a_h^k)\big) \\
&\le \frac{4Cd^3H^4\log(2dT/\delta)}{\text{gap}_{\min}}\log\left(\frac{Cd^3H^4\log(2dT/\delta)}{\text{gap}_{\min}^2}\right).
\end{aligned}$$

Lemma 6.3 suggests that with high probability, the summation of $\text{gap}_h(s_h^k, a_h^k)$ over episode k at step h is logarithmic in the number of steps $T = KH$ and its dependency in $\text{gap}_{\min}$ is $1/\text{gap}_{\min}$. This leads to our final proof of our main theorem.

Proof of Theorem 4.4. We define the high probability event Ω as follows.

$$\Omega = \{\text{Lemma 6.3 holds for all } h \in [H], \\ \text{and Lemma 6.1 holds on for } \tau = \lceil \log(1/\delta) \rceil\}.$$

According to Lemma 6.3 and Lemma 6.1, we have $\Pr[\Omega] \geq 1 - 2(K+1)H\log(H/\text{gap}_{\text{min}})\delta - \delta\log T$. Given the event Ω, we have

$$\begin{aligned}
&\text{Regret}(K) \\
&\leq 2\sum_{k=1}^{K}\sum_{h=1}^{H}\text{gap}_h(s_h^k, a_h^k) + \frac{16H^2\log\delta}{3} + 2 \\
&= 2\sum_{k=1}^{K}\sum_{h=1}^{H}V_h^*(s_h^k) - Q_h^*(s_h^k, a_h^k) + \frac{16H^2\log\delta}{3} + 2 \\
&\leq \frac{9Cd^3H^5\log(2dHK/\delta)}{\text{gap}_{\text{min}}}\log\bigg(\frac{Cd^3H^4\log(2dHK/\delta)}{\text{gap}_{\text{min}}^2}\bigg) \\
&\quad + \frac{16H^2\log\delta}{3},
\end{aligned}$$

where the first inequality holds due to Lemma 6.1 and the last inequality holds due to Lemma 6.3. Thus, we complete the proof. □

6.2. Proof of the Key Technical Lemma

In this subsection, we propose the proof to the main technical lemma, Lemma 6.2. Our proof follows the idea of error decomposition proposed in Wang et al. (2019a); Yang et al. (2020), that is, to upper bound the summation of sub-optamalities by considering their summation of the exploration bonuses. The key difference between our proof and that of Wang et al. (2019a); Yang et al. (2020) is the choice of exploration bonus. Wang et al. (2019a); Yang et al. (2020) considered the tabular MDP setting and adapted a $1/\sqrt{n}$-type bonus term, while we consider the linear function approximation setting and adapt a linear bandit-style exploration bonus (Dani et al., 2008; Abbasi-Yadkori et al., 2011; Li et al., 2010) as suggested in Line 6. The following lemmas guarantee that our constructed Q_h^k is indeed the UCB of the optimal action-value function:

Lemma 6.4 (Lemma B.4 in Jin et al. 2020)**.** With probability at least $1-\delta$, for any policy π and all $s \in \mathcal{S}, a \in \mathcal{A}, h \in [H], k \in [K]$, we have

$$\langle \boldsymbol{\phi}(s,a), \mathbf{w}_h^k \rangle - Q_h^\pi(s,a) = \big[\mathbb{P}_h(V_{h+1}^k - V_{h+1}^\pi)\big](s,a) + \Delta,$$

where $|\Delta| \leq \beta\sqrt{\boldsymbol{\phi}(s,a)^\top(\Lambda_h^k)^{-1}\boldsymbol{\phi}(s,a)}$

Lemma 6.5 (Lemma B.5 in Jin et al. 2020)**.** With probability at least $1-\delta$, for all $s \in \mathcal{S}, a \in \mathcal{A}, h \in [H], k \in [K]$, we have

$$Q_h^k(s,a) \geq Q_h^*(s,a).$$

We also need the following technical lemma, which gives us a slightly stronger upper bound for the summation of exploration bonuses:

Lemma 6.6. For any subset $C = \{c_1, .., c_k\} \subseteq [K]$ and any $h \in [H]$, we have

$$\sum_{i=1}^{k}(\boldsymbol{\phi}_h^{c_i})^\top(\Lambda_h^{c_i})^{-1}\boldsymbol{\phi}_h^{c_i} \leq 2d\log\bigg(\frac{\lambda+k}{\lambda}\bigg),$$

where $\boldsymbol{\phi}_h^{c_i}$ is the abbreviation of $\boldsymbol{\phi}_h^{c_i}(s_h^{c_i}, a_h^{c_i})$.

With the lemmas above, we begin to prove Lemma 6.2.

Proof of Lemma 6.2. We fix h in this proof. Let $k_0 = 0$, and for $i \in [N]$, we denote k_i as the minimum index of the episode where the sub-optimality at step h is no less than $2^n\text{gap}_{\text{min}}$:

$$\begin{aligned}
k_i = \min\big\{&k : k > k_{i-1}, \\
&V_h^*(s_h^k) - Q_h^{\pi_k}(s_h^k, a_h^k) \geq 2^n\text{gap}_{\text{min}}\big\}.
\end{aligned} \tag{6.4}$$

For simplicity, we denote by K' the number of episodes such that the sub-optimality of this episode at step h is no less than $2^n\text{gap}_{\text{min}}$. Formally speaking, we have

$$K' = \sum_{k=1}^{K}\mathbb{1}\big[V_h^*(s_h^k) - Q_h^{\pi_k}(s_h^k, a_h^k) \geq 2^n\text{gap}_{\text{min}}\big].$$

From now we only consider the episodes whose sub-optimality is no less than $2^n\text{gap}_{\text{min}}$. We first lower bound the summation of difference between the estimated action-value function $Q_h^{k_i}$ and the action-value function induced by the policy π_{k_i}, which can be represented as follows:

$$\begin{aligned}
&\sum_{i=1}^{K'}\big(Q_h^{k_i}(s_h^{k_i}, a_h^{k_i}) - Q_h^{\pi_{k_i}}(s_h^{k_i}, a_h^{k_i})\big) \\
&\quad\geq \sum_{i=1}^{K'}\Big(Q_h^{k_i}\big(s_h^{k_i}, \pi_h^*(s_h^{k_i}, h)\big) - Q_h^{\pi_{k_i}}(s_h^{k_i}, a_h^{k_i})\Big) \\
&\quad\geq \sum_{i=1}^{K'}\Big(Q_h^*\big(s_h^{k_i}, \pi_h^*(s_h^{k_i}, h)\big) - Q_h^{\pi_{k_i}}(s_h^{k_i}, a_h^{k_i})\Big) \\
&\quad= \sum_{i=1}^{K'}\big(V_h^*(s_h^{k_i}) - Q_h^{\pi_{k_i}}(s_h^{k_i}, a_h^{k_i})\big) \\
&\quad\geq 2^n\text{gap}_{\text{min}}K',
\end{aligned} \tag{6.5}$$

where the first inequality holds due to the definition of policy π_{k_i}, the second inequality holds due to Lemma 6.5 and the last inequality holds due to the definition of k_i in (6.4). On the other hand, we upper bound $\sum_{i=1}^{K'}\big(Q_h^{k_i}(s_h^{k_i}, a_h^{k_i}) -$

$Q_h^{\pi_{k_i}}(s_h^{k_i}, a_h^{k_i})\big)$ as follows. For any $h' \in [H], k \in [K]$, we have

$$
\begin{aligned}
&Q_{h'}^k(s_{h'}^k, a_{h'}^k) - Q_{h'}^{\pi_k}(s_{h'}^k, a_{h'}^k) \\
&\quad = \langle \boldsymbol{\phi}(s_{h'}^k, a_{h'}^k), \mathbf{w}_{h'}^k \rangle - Q_{h'}^{\pi_k}(s_{h'}^k, a_{h'}^k) \\
&\qquad + \beta\sqrt{\boldsymbol{\phi}(s_{h'}^k, a_{h'}^k)^\top (\Lambda_{h'}^k)^{-1} \boldsymbol{\phi}(s_{h'}^k, a_{h'}^k)} \\
&\quad \le \big[\mathbb{P}_h(V_{h'+1}^k - V_{h'+1}^{\pi_k})\big](s_{h'}^k, a_{h'}^k) \\
&\qquad + 2\beta\sqrt{\boldsymbol{\phi}(s_{h'}^k, a_{h'}^k)^\top (\Lambda_{h'}^k)^{-1} \boldsymbol{\phi}(s_{h'}^k, a_{h'}^k)} \\
&\quad = V_{h'+1}^k(s_{h'+1}^k) - V_{h'+1}^{\pi_k}(s_{h'+1}^k) + \epsilon_{h'}^k \\
&\qquad + 2\beta\sqrt{\boldsymbol{\phi}(s_{h'}^k, a_{h'}^k)^\top (\Lambda_{h'}^k)^{-1} \boldsymbol{\phi}(s_{h'}^k, a_{h'}^k)} \\
&\quad = Q_{h'+1}^k(s_{h'+1}^k, a_{h'+1}^k) - Q_{h'+1}^{\pi_k}(s_{h'+1}^k, a_{h'+1}^k) + \epsilon_{h'}^k \\
&\qquad + 2\beta\sqrt{\boldsymbol{\phi}(s_{h'}^k, a_{h'}^k)^\top (\Lambda_{h'}^k)^{-1} \boldsymbol{\phi}(s_{h'}^k, a_{h'}^k)}, \qquad (6.6)
\end{aligned}
$$

where

$$
\begin{aligned}
\epsilon_{h'}^k = &\big[\mathbb{P}_h(V_{h'+1}^k - V_{h'+1}^{\pi_k})\big](s_{h'}^k, a_{h'}^k) \\
&- \big(V_{h'+1}^k(s_{h'+1}^k) - V_{h'+1}^{\pi_k}(s_{h'+1}^k)\big),
\end{aligned}
$$

and the inequality holds due to Lemma 6.4. Taking summation for (6.6) over all k_i and $h \le h' \le H$, we have

$$
\begin{aligned}
&\sum_{i=1}^{K'} \big(Q_h^{k_i}(s_h^{k_i}, a_h^{k_i}) - Q_h^{\pi_{k_i}}(s_h^{k_i}, a_h^{k_i})\big) - \underbrace{\sum_{i=1}^{K'}\sum_{h'=h}^{H} \epsilon_{h'}^{k_i}}_{I_1} \\
&\le \underbrace{\sum_{i=1}^{K'}\sum_{h'=h}^{H} 2\beta\sqrt{\boldsymbol{\phi}(s_{h'}^{k_i}, a_{h'}^{k_i})^\top (\Lambda_{h'}^{k_i})^{-1} \boldsymbol{\phi}(s_{h'}^{k_i}, a_{h'}^{k_i})}}_{I_2}. \qquad (6.7)
\end{aligned}
$$

It therefore suffices to bound I_1 and I_2 separately. For I_1, by Lemma E.1, for each episode $k \in [K]$, with probability at least $1 - \delta$, we have

$$
\begin{aligned}
&\sum_{i=1}^{k}\sum_{j=h}^{H} \Big(\big[\mathbb{P}_j(V_{j+1}^{k_i} - V_{j+1}^{\pi_{k_i}})\big](s_j^{k_i}, a_j^{k_i}) - \\
&\qquad \big(V_{j+1}^{k_i}(s_{j+1}^{k_i}) - V_{j+1}^{\pi_{k_i}}(s_{j+1}^{k_i})\big)\Big) \le \sqrt{2kH^3\log(1/\delta)},
\end{aligned}
$$

where we use the fact that $\big[\mathbb{P}_j(V_{j+1}^{k_i} - V_{j+1}^{\pi_{k_i}})\big](s_j^{k_i}, a_j^{k_i}) - \big(V_{j+1}^{k_i}(s_{j+1}^{k_i}) - V_{j+1}^{\pi_{k_i}}(s_{j+1}^{k_i})\big)$ forms a martingale difference sequence. Taking a union bound for all $k \in [K]$ gives that, with probability at least $1 - K\delta$,

$$
\begin{aligned}
&\sum_{i=1}^{K'}\sum_{j=h}^{H} \big[\mathbb{P}_j(V_{j+1}^{k_i} - V_{j+1}^{\pi_{k_i}})\big](s_j^{k_i}, a_j^{k_i}) \\
&\qquad - \sum_{i=1}^{k}\sum_{j=h}^{H} \big(V_{j+1}^{k_i}(s_{j+1}^{k_i}) - V_{j+1}^{\pi_{k_i}}(s_{j+1}^{k_i})\big) \\
&\qquad \le \sqrt{2K'H^3\log(1/\delta)}, \qquad (6.8)
\end{aligned}
$$

For I_2, we have

$$
\begin{aligned}
I_1 &= \sum_{i=1}^{K'}\sum_{h'=h}^{H} 2\beta\sqrt{\boldsymbol{\phi}(s_{h'}^{k_i}, a_{h'}^{k_i})^\top (\Lambda_{h'}^{k_i})^{-1} \boldsymbol{\phi}(s_{h'}^{k_i}, a_{h'}^{k_i})} \\
&\le 2\beta\sqrt{K'}\sum_{h'=h}^{H}\sqrt{\sum_{i=1}^{K'} \boldsymbol{\phi}(s_{h'}^{k_i}, a_{h'}^{k_i})^\top (\Lambda_{h'}^{k_i})^{-1} \boldsymbol{\phi}(s_{h'}^{k_i}, a_{h'}^{k_i})} \\
&\le 2H\beta\sqrt{K'}\sqrt{2d\log(K'+1)}, \qquad (6.9)
\end{aligned}
$$

where the first inequality holds due to Cauchy-Schwarz inequality and the second inequality holds due to Lemma 6.6.

Substituting (6.9) and (6.8) into (6.7), we obtain that with probability at least $1 - (K+1)\delta$,

$$
\begin{aligned}
&\sum_{i=1}^{K'} \big(Q_h^{k_i}(s_h^{k_i}, a_h^{k_i}) - Q_h^{\pi_{k_i}}(s_h^{k_i}, a_h^{k_i})\big) \\
&\le \sqrt{2K'H^3\log(1/\delta)} + 2H\beta\sqrt{K'}\sqrt{2d\log(K'+1)}. \qquad (6.10)
\end{aligned}
$$

By now, we have obtained both the lower and upper bounds for $\sum_{i=1}^{K'}\big(Q_h^{k_i}(s_h^{k_i}, a_h^{k_i}) - Q_h^{\pi_{k_i}}(s_h^{k_i}, a_h^{k_i})\big)$ from (6.5) and (6.10). Finally, combining (6.5) and (6.10), we can derive the following constraint on K':

$$
\begin{aligned}
2^n \mathrm{gap}_{\min} K' \le &\sqrt{2K'H^3\log(1/\delta)} \\
&+ 2H\beta\sqrt{2K'd\log(K'+1)}. \qquad (6.11)
\end{aligned}
$$

Solving out K' from (6.11), we conclude that there exists a constant C such that

$$
\begin{aligned}
K' \le &\frac{Cd^3H^4\log(2dHK/\delta)}{4^n \mathrm{gap}_{\min}^2} \\
&\times \log\left(\frac{Cd^3H^4\log(2dHK/\delta)}{4^n \mathrm{gap}_{\min}^2}\right),
\end{aligned}
$$

which ends our proof. □

7. Conclusion

In this paper, we analyze the RL algorithms with function approximation by considering a specific problem-dependent quantity $\mathrm{gap}_{\min}$. We show that two existing algorithms LSVI-UCB and UCRL-VTR attain $\log T$-type regret instead of $\sqrt{T}$-type regret under their corresponding linear function approximation assumptions. It remains unknown whether the dependence of the length of the episode H and dimension d is optimal or not, and we leave it as future work.

Acknowledgement

We thank the anonymous reviewers for their helpful comments. JH, DZ 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. (2011). Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*.

AUER, P. and ORTNER, R. (2007). Logarithmic online regret bounds for undiscounted reinforcement learning. In *Advances in Neural Information Processing Systems*.

AYOUB, A., JIA, Z., SZEPESVARI, C., WANG, M. and YANG, L. F. (2020). Model-based reinforcement learning with value-targeted regression. *arXiv preprint arXiv:2006.01107* .

AZAR, M. G., OSBAND, I. and MUNOS, R. (2017). Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org.

BARTLETT, P. L., BOUSQUET, O., MENDELSON, S. ET AL. (2005). Local rademacher complexities. *The Annals of Statistics* **33** 1497–1537.

BESSON, L. and KAUFMANN, E. (2018). What doubling tricks can and can't do for multi-armed bandits. *arXiv preprint arXiv:1803.06971* .

BUBECK, S. and CESA-BIANCHI, N. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Machine Learning* **5** 1–122.

CAI, Q., YANG, Z., JIN, C. and WANG, Z. (2019). Provably efficient exploration in policy optimization. *arXiv preprint arXiv:1912.05830* .

CESA-BIANCHI, N. and LUGOSI, G. (2006). *Prediction, learning, and games*. Cambridge university press.

DANI, V., HAYES, T. P. and KAKADE, S. M. (2008). Stochastic linear optimization under bandit feedback. In *Conference on Learning Theory*.

DANN, C. and BRUNSKILL, E. (2015). Sample complexity of episodic fixed-horizon reinforcement learning. In *Advances in Neural Information Processing Systems*.

DU, S. S., LEE, J. D., MAHAJAN, G. and WANG, R. (2020). Agnostic q-learning with function approximation in deterministic systems: Tight bounds on approximation error and sample complexity. *arXiv preprint arXiv:2002.07125* .

DU, S. S., LUO, Y., WANG, R. and ZHANG, H. (2019). Provably efficient q-learning with function approximation via distribution shift error checking oracle. In *Advances in Neural Information Processing Systems*.

JAKSCH, T., ORTNER, R. and AUER, P. (2010). Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research* **11** 1563–1600.

JIA, Z., YANG, L., SZEPESVARI, C. and WANG, M. (2020). Model-based reinforcement learning with value-targeted regression .

JIANG, N., KRISHNAMURTHY, A., AGARWAL, A., LANGFORD, J. and SCHAPIRE, R. E. (2017). Contextual decision processes with low bellman rank are pac-learnable. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org.

JIN, C., ALLEN-ZHU, Z., BUBECK, S. and JORDAN, M. I. (2018). Is q-learning provably efficient? In *Advances in Neural Information Processing Systems*.

JIN, C., YANG, Z., WANG, Z. and JORDAN, M. I. (2020). Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*.

LATTIMORE, T. and SZEPESVÁRI, C. (2018). Bandit algorithms. *preprint* 28.

LI, L., CHU, W., LANGFORD, J. and SCHAPIRE, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*.

MOU, W., WEN, Z. and CHEN, X. (2020). On the sample complexity of reinforcement learning with policy space generalization. *arXiv preprint arXiv:2008.07353* .

OK, J., PROUTIERE, A. and TRANOS, D. (2018). Exploration in structured reinforcement learning. In *Advances in Neural Information Processing Systems*.

OSBAND, I. and VAN ROY, B. (2016). On lower bounds for regret in reinforcement learning. *arXiv preprint arXiv:1608.02732* .

SIMCHOWITZ, M. and JAMIESON, K. G. (2019). Non-asymptotic gap-dependent regret bounds for tabular mdps. In *Advances in Neural Information Processing Systems*.

SLIVKINS, A. ET AL. (2019). Introduction to multi-armed bandits. *Foundations and Trends® in Machine Learning* **12** 1–286.

STREHL, A. L., LI, L., WIEWIORA, E., LANGFORD, J. and LITTMAN, M. L. (2006). Pac model-free reinforcement learning. In *Proceedings of the 23rd international conference on Machine learning*. ACM.

TEWARI, A. and BARTLETT, P. L. (2008). Optimistic linear programming gives logarithmic regret for irreducible mdps. In *Advances in Neural Information Processing Systems*.

WANG, Y., DONG, K., CHEN, X. and WANG, L. (2019a). Q-learning with ucb exploration is sample efficient for infinite-horizon mdp. In *International Conference on Learning Representations*.

WANG, Y., WANG, R., DU, S. S. and KRISHNAMURTHY, A. (2019b). Optimism in reinforcement learning with generalized linear function approximation. *arXiv preprint arXiv:1912.04136* .

WATKINS, C. J. C. H. (1989). *Learning from delayed rewards*. Ph.D. thesis, University of Cambridge.

YANG, K., YANG, L. F. and DU, S. S. (2020). q-learning with logarithmic regret. *arXiv preprint arXiv:2006.09118* .

YANG, L. and WANG, M. (2019). Sample-optimal parametric q-learning using linearly additive features. In *International Conference on Machine Learning*.

ZANETTE, A. and BRUNSKILL, E. (2019). Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. In *International Conference on Machine Learning*.

ZANETTE, A., LAZARIC, A., KOCHENDERFER, M. and BRUNSKILL, E. (2020). Learning near optimal policies with low inherent bellman error. *arXiv preprint arXiv:2003.00153* .

ZHANG, Z., ZHOU, Y. and JI, X. (2020). Almost optimal model-free reinforcement learning via reference-advantage decomposition. *arXiv preprint arXiv:2004.10019* .

ZHOU, D., GU, Q. and SZEPESVARI, C. (2020a). Nearly minimax optimal reinforcement learning for linear mixture markov decision processes. *arXiv preprint arXiv:2012.08507* .

ZHOU, D., HE, J. and GU, Q. (2020b). Provably efficient reinforcement learning for discounted mdps with feature mapping. *arXiv preprint arXiv:2006.13165* .