Article

Analysis of Performance Measure in Q Learning with UCB Exploration

Weicheng Ye ^{1,*} and Dangxing Chen ^{2,*}¹ Credit Suisse Securities, New York, NY 10010-3698, USA² Zu Chongzhi Center for Mathematics and Computational Sciences, Duke Kunshan University, Kunshan 215316, China

* Correspondence: weicheny@alumni.cmu.edu (W.Y.); dangxing.chen@dukekunshan.edu.cn (D.C.)

† Most work was done while at Carnegie Mellon University. Opinions expressed in this paper are of the author, and do not reflect the view of Credit Suisse.

Abstract: Compared to model-based Reinforcement Learning (RL) approaches, model-free RL algorithms, such as Q -learning, require less space and are more expressive, since specifying value functions or policies is more flexible than specifying the model for the environment. This makes model-free algorithms more prevalent in modern deep RL. However, model-based methods can more efficiently extract the information from available data. The Upper Confidence Bound (UCB) bandit can improve the exploration bonuses, and hence increase the data efficiency in the Q -learning framework. The cumulative regret of the Q -learning algorithm with an UCB exploration policy in the episodic Markov Decision Process has recently been explored in the underlying environment of finite state-action space. In this paper, we study the regret bound of the Q -learning algorithm with UCB exploration in the scenario of compact state-action metric space. We present an algorithm that adaptively discretizes the continuous state-action space and iteratively updates Q -values. The algorithm is able to efficiently optimize rewards and minimize cumulative regret.

Keywords: reinforcement learning; Q -learning; multi-armed bandit; theory of machine learning

Citation: Ye, W.; Chen, D. Analysis of Performance Measure in Q Learning with UCB Exploration. *Mathematics* **2022**, *10*, 575. <https://doi.org/10.3390/math10040575>

Academic Editor: Anatoliy Swishchuk

Received: 3 January 2022

Accepted: 5 February 2022

Published: 12 February 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Background

1.1. Markov Decision Process

The Markov Decision Process (MDP) is a discrete-time sequential decision-making problem. In MDP, there is a finite set of states $\mathcal{S}$ and a finite set of actions $\mathcal{A}$. At time-step t , the agent is at some state $s \in \mathcal{S}$, and the agent takes some action $a \in \mathcal{A}$. By the state transition probability function $\mathcal{P}$, the agent moves to some new state $s' \in \mathcal{S}$ at time $t + 1$, where $s' \sim \mathcal{P}(\cdot | s, a)$. The agent will then receive some immediate reward $r(s', a)$, the reward after transitioning to state s' , due to action a . In MDP, for the certain pair (s, a) , the state transition process satisfies the Markov property. That means the process is conditionally independent of all of the past states and actions. For example, if there exists only one action in each state (fixed transition probability in each state) and all the rewards are zeros, then MDP reduces to the Markov Chain. Policy π is a map from state to action ($\pi : \mathcal{S} \rightarrow \mathcal{A}$). If the environment has finite time-steps, then we call it a finite time horizon. Otherwise, the environment is an infinite time horizon. The core problem in MDP is to find an optimal policy that can maximize the cumulative function of random rewards over a finite/infinite time horizon. For example, the typical cumulative function of random rewards in the finite time horizon is a sum of rewards. For example, for the specific policy π with a finite time horizon H , it is $\mathbb{E}_{s_{t+1} \sim \mathcal{P}(\cdot | s_t, \pi(s_t))} \left[\sum_{t=0}^H r(s_{t+1}, \pi(s_t)) \right]$.

MDP hence contains two important functions: the reward function r and state transition function $\mathcal{P}$. If we know both functions, that is, we can predict which reward will be

received and which states will be in the next transition, then MDP can be solved by Dynamic Programming (DP). DP requires the full description of MDP, with known transition probabilities and reward functions.

1.2. Model-Based and Model-Free Reinforcement Learning

However, for most cases, reward function r and state transition function $\mathcal{P}$ are not precisely known. The purpose of reinforcement learning (RL) is to solve MDP when the reward and transition functions are unknown. One approach in RL is to learn a model of how the environment works from its observation and then find a solution using that model. For example, with a new state-action pair and the observation of the corresponding reward, we can update the estimate of r and $\mathcal{P}$. With the simulation of different actions in many states, we can have a learned transition function and learned reward function with high confidence. If we adequately model the environment, then we can use the learned model to find policy. This approach is called model-based RL. However, model-based RL requires too many trials to learn a particular task independent of the choice of policy iteration or value iteration. Hence, model-based RL methods are often inapplicable to large-scale systems. In contrast, we do not need to learn a model, but can directly find a policy. For example, the Q -learning algorithm is one approach that learns MDP and solves it to find the optimal policy at the same time. Q -learning directly estimates the optimal Q -values for each action in each state, and the policy could be derived by choosing the action with the highest Q -value in the given state. Since we do not learn a model of the environment, we call it model-free RL. The core idea of Q -learning is to balance the exploration and exploitation trade-off: trying random actions to explore the underlying MDP, while following the so far optimal policy to maximize the long-term reward.

2. Introduction

Reinforcement Learning (RL), as a learning problem to maximize the numerical reward, learns how to map situations to actions in unknown underlying system dynamics [1]. While model-based RL can approach high-quality and effective decisions, the stability issue about scaling up to the approximate setting with high-dimensional state and action spaces has been demonstrated [2,3]. Besides, due to the assumption of the model, model-based RL suffers from model bias, especially when the samples are few and informative prior knowledge is limited [4–6]. On the contrary, model-free RL algorithms can directly update the value and policy functions without any assumption on the model environment [1]. As the most classical model-free RL algorithm, Q -learning was introduced by Watkins in his Ph.D. dissertation. Watkins designed the action-reply MDP algorithm, which achieves convergence to the optimal Q function with a finite state and action space setting [7,8]. This approach is considered as a simpler way of learning optimality in a controlled Markov domain. From the classical Q -learning algorithm to modern Deep Q -Networks [9], Advantage Actor-Critic [10], and Trust Region Policy Optimization [11], most state-of-the-art RL are in the model-free approach. Model-free RL algorithms require less space and are more expressive since specifying value functions or policies are more flexible compared to specifying the model of the environment. The advantages of model-free methods underlie their success in deep RL applications [12].

However, model-based methods are generally more promising for efficient extraction of the information from available data than the model-free approaches, such as Q -learning [6,11]. The requirement of data efficiency in model-free RL is hence increasing. The key to achieve good data efficiency generally lies in the tradeoff between exploration and exploitation. Using the bandit to explore the uncertain environment while maximizing the reward has been imported to obtain optimal data efficiency [12–14]. Q -learning with a common ϵ -greedy exploration strategy exponentially takes many episodes to learn [12]. The Upper Confidence Bound (UCB) algorithm, as a multi-armed bandit algorithm, assigns exploration bonuses in many stochastic settings [15–18]. The UCB exploration policy is hence considered as a tool for improving the exploration bonuses in Q -learning [12].

This approach can improve data efficiency while leveraging the advantages of model-free methods. However, the Q-learning algorithm with UCB exploration is restricted to the environment of finite state and action space [7,12,19]. Due to the natural settings of RL applications, RL over underlying MDP with the environment of continuous state and action space has substantially been an intractable and challenging problem [20,21]. For example, most robotic applications of RL require continuous state spaces defined by utilizing continuous variables, such as position, velocity, torque, and so forth. Although Q-learning is commonly applied to problems with discrete states and actions, it has been extended to continuous space under different scenarios [22,23]. This motivates us to extend the state-of-the-art Q-learning algorithm equipped with an UCB exploration policy to a continuous state-action space environment. In modern RL, the algorithm that can leverage the advantages of model-free methods and data efficiency in the continuous state-action space is demanding.

Our contributions are: (1) Development of a state-of-the-art Q-learning algorithm with an UCB exploration policy to compact state and action metric space; and (2) Achievement of a tight bound for the cumulative regret for the underlying algorithm using the covering dimension techniques. We provide a rigorous analysis of bounding the regret with the concentration of measure analysis and combinatorial approaches.

3. Notation and Settings

3.1. Q-Learning

We consider a setting of an episodic Markov decision process MDP $(\mathcal{S}, \mathcal{A}, H, \mathcal{P}, r)$, where state space is a compact metric space $(\mathcal{S}, d_{\mathcal{S}})$, and action space is also a compact metric space $(\mathcal{A}, d_{\mathcal{A}})$. $\mathcal{S} \times \mathcal{A}$ is a product compact metric space with metric $d_{\mathcal{S} \times \mathcal{A}}$. H is the number of steps in each episode. $\mathcal{P}$ is the state transition function where the transitional probability from the current state-action pair in $\mathcal{S} \times \mathcal{A}$ to the next state is $x' \sim \mathcal{P}(\cdot|x, a)$. The reward function at step $h \in [H]$, is $r_h : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$. Since we discuss the finite time horizon Q-learning here, there is no discount factor included in the value function. Let the policy function be $\pi : \mathcal{S} \rightarrow \mathcal{A}$. Denote by $V_h^\pi : \mathcal{S} \rightarrow \mathbb{R}$ the value function at step h under policy π . V_h^π stands for the expected sum of remaining rewards collected from $x_h = x$ to the end of episode given by policy π . Mathematically,

$$V_h^\pi(x) := \mathbb{E} \left[\sum_{h'=h}^H r_{h'}(x_{h'}, \pi_{h'}(x_{h'})) | x_h = x \right].$$

Accordingly, we can define $Q_h^\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ as the Q-value function at step h under policy π . This gives the expected sum of the remaining rewards, starting from $x_h = x, a_h = a$ to the end of episode. Mathematically,

$$Q_h^\pi(x, a) := r_h(x, a) + \mathbb{E} \left[\sum_{h'=h+1}^H r_{h'}(x_{h'}, \pi_{h'}(x_{h'})) | x_h = x, a_h = a \right].$$

For simplicity, we define the expected future value

$$\mathcal{P}_h V_h(x, a) := \mathbb{E}_{x' \sim \mathcal{P}(\cdot|x, a)} V_{h+1}(x'),$$

where V_{h+1} is the proxy of the value function. The empirical counterpart of episode k , say x_{h+1}^k the next observed state from (x, a) , is

$$\hat{\mathcal{P}}_h^k V_h(x, a) := V_{h+1}(x_{h+1}^k). \quad (1)$$

The above notations follow from [12]. Define the optimal policy π^* by the optimal value $V_h^*(x) = \sup_{\pi} V_h^{\pi}(x)$. The Bellman equation for any (deterministic) policy π is

$$\begin{cases} V_h^{\pi}(x) = Q_h^{\pi}(x, \pi(x)) \\ Q_h^{\pi}(x, a) = r_h(x, a) + \mathcal{P}_h V_h^{\pi}(x, a). \end{cases}$$

In addition, V^* and Q^* are the value and Q -value functions associated with the optimal policy:

$$\begin{cases} V_h^*(x) = \sup_{a \in \mathcal{A}} Q_h^*(x, a) \\ Q_h^*(x, a) = r_h(x, a) + \mathcal{P}_h V_h^*(x, a). \end{cases} \quad (2)$$

3.2. Lipschitz Assumption

In this work, we assume V^* is Lipschitz-continuous with Lipschitz constant D . Assume the next step from the current state-action pair is Lipschitz-continuous. That is, let x_{h+1} be the next step from (x_h, a_h) and y_{h+1} be the next step from (y_h, b_h) . Then $d_S(x_{h+1}, y_{h+1}) \leq M \cdot d_{S \times \mathcal{A}}((x_h, a_h), (y_h, b_h))$ with probability 1 for some constant M . This implies that $\hat{\mathcal{P}}V^*$ is also Lipschitz-continuous:

$$\begin{aligned} \|\hat{\mathcal{P}}_h V_h^*(x_h, a_h) - \hat{\mathcal{P}}_h V_h^*(y_h, b_h)\| &= \|V_{h+1}^*(x_{h+1}) - V_{h+1}^*(y_{h+1})\| \\ &\leq D \cdot d_S(x_{h+1}, y_{h+1}) \leq D \cdot M \cdot d_{S \times \mathcal{A}}((x_h, a_h), (y_h, b_h)) \\ &= L \cdot d_{S \times \mathcal{A}}((x_h, a_h), (y_h, b_h)) \end{aligned} \quad (3)$$

for $L = D \cdot M$. The Lipschitz constant is L for $\hat{\mathcal{P}}V^*$.

3.3. Performance Measure

Our main interest is to provide a practical algorithm that can bound the regret. Assume K is the total number of episodes. At the beginning of each episode k , the adversary picks a starting state x_1^k . The agent will choose a corresponding greedy policy π_k before the start of episode k . Recall that $V_h^{\pi_k}$ is the value function of π_k in episode k . The cumulative regret (performance measure) of K episodes is

$$\text{Regret}(K) := \sum_{k=1}^K \left(V_1^*(x_1^k) - V_1^{\pi_k}(x_1^k) \right). \quad (4)$$

3.4. Iterative Updating Rule

Since $\mathcal{S} \times \mathcal{A}$ is a compact metric space, running the algorithm will partition the space into a discrete framework, for example, balls. In the algorithm, for $(x, a) \in \mathcal{S} \times \mathcal{A}$ at episode k , we look at all the past state-action pairs that are in the same ball of (x, a) , $B(x, a)$, from the episode up to $k - 1$. Assume at the past episode, $k_i \in \{1, \dots, k - 1\}$ and step h , $(x_h^{k_i}, a_h^{k_i}) \in B(x, a)$ was selected. We call $\{(x_h^{k_i}, a_h^{k_i})\}$ active points. Denote by $n_h^k(x, a)$ the total number of state-action pairs $\{(x_h^{k_i}, a_h^{k_i})\}$ in $B(x, a)$ at step h up to episode k . Additionally, its observed state at step $h + 1$ is $x_{h+1}^{k_i}$ for episode k_i . We denote Q_h^k, V_h^k, n_h^k as Q_h, V_h , and n_h at the beginning of episode k . We also have a bandit term b_t , of which more details will be given later.

Definition 1 (Iterative updating rule of Q_h^k).

$$Q_h^{k+1}(x, a) = \begin{cases} (1 - \alpha_t) Q_h^k(x, a) + \alpha_t [r_h(x, a) + V_{h+1}^k(x_{h+1}^k) + b_t] & \text{if } (x_h^k, a_h^k) \text{ in } B(x, a); \\ Q_h^k(x, a) & \text{otherwise.} \end{cases} \quad (5)$$

Here, t is the counter for the times the algorithm has visited the $B(x, a)$ at step h , so $t = n_h^k(x, a)$. Accordingly, the iterative updating rule for V_h^k is

$$V_h^k(x) \leftarrow \min \left\{ H, \sup_{a' \in \mathcal{A}} Q_h^k(x, a') \right\}, \forall x \in \mathcal{S}. \quad (6)$$

For the choice of learning rate α_t in the updating rule, we choose $\alpha_t := \frac{1}{t}$. Then, we define the following related quantities:

$$\alpha_t^0 = \prod_{j=1}^t (1 - \alpha_j), \quad \alpha_t^i = \alpha_i \cdot \prod_{j=i+1}^t (1 - \alpha_j). \quad (7)$$

The above definition yields the following properties that are useful in our analysis.

Lemma 1. *The following properties hold for α_t and α_t^i :*

$$1. \sum_{t=1}^{\infty} \alpha_t = \infty \text{ and } \sum_{t=1}^{\infty} \alpha_t^2 < \infty, \quad 2. \frac{1}{\sqrt{t}} \leq \sum_{i=1}^t \frac{\alpha_i^i}{\sqrt{i}} \leq \frac{2}{\sqrt{t}}.$$

Proof. The first property is trivial. We will prove the second property by induction on t .

Let us check the right-hand side. Clearly the statement is true for $t = 1$. Assume the result holds for some $t - 1$, i.e., $\sum_{i=1}^{t-1} \frac{\alpha_i^i}{\sqrt{i}} \leq \frac{2}{\sqrt{t-1}}$. Note that $\alpha_t^i = \alpha_t = \frac{1}{t}$. By using the induction hypothesis we can write

$$\begin{aligned} \sum_{i=1}^t \frac{\alpha_i^i}{\sqrt{i}} &= \frac{\alpha_t^t}{\sqrt{t}} + (1 - \alpha_t) \sum_{i=1}^{t-1} \frac{\alpha_i^i}{\sqrt{i}} \\ &= \frac{\alpha_t}{\sqrt{t}} + (1 - \alpha_t) \sum_{i=1}^{t-1} \frac{\alpha_i^i}{\sqrt{i}} \\ &\leq \frac{\alpha_t}{\sqrt{t}} + (1 - \alpha_t) \frac{2}{\sqrt{t-1}} \end{aligned}$$

All the remaining is to prove that $\frac{\alpha_t}{\sqrt{t}} + (1 - \alpha_t) \frac{2}{\sqrt{t-1}} \leq \frac{2}{\sqrt{t}}$, i.e., $\frac{2(\sqrt{t}-\sqrt{t-1})}{2\sqrt{t}-\sqrt{t-1}} \leq \alpha_t = \frac{1}{t}$. Clearly, we have

$$\begin{aligned} \frac{2\sqrt{t}-\sqrt{t-1}}{2(\sqrt{t}-\sqrt{t-1})} &= \frac{1}{2} (\sqrt{t} + \sqrt{t-1}) (2\sqrt{t} - \sqrt{t-1}) \\ &= \frac{1}{2} (t + 1 + \sqrt{t(t-1)}) \\ &\geq \frac{1}{2} (t + 1 + t - 1) \\ &= t. \end{aligned}$$

We have hence proven the right-hand side (upper bound). Followed by the same induction process, the statement on the left-hand side (lower bound) is equivalent to showing $\frac{\alpha_t}{\sqrt{t}} + (1 - \alpha_t) \frac{1}{\sqrt{t-1}} \geq \frac{1}{\sqrt{t}}$. This is trivial, since $\alpha_t = \frac{1}{t}$. $\square$

By the updated rule for Q in Definition 1, and the definition of the learning rate (7), we have

$$Q_h^k(x, a) = \alpha_t^0 \cdot H + \sum_{i=1}^t \alpha_t^i [r_h(x, a) + V_{h+1}^{k_i}(x_{h+1}^{k_i}) + b_i]. \quad (8)$$

4. Performance Measure Algorithms and Theorems

Next, we present the main algorithm and the main theorem.

Theorem 1. For any $p \in (0, 1)$, if we choose bandit $b_t = c\sqrt{\frac{H^3 \iota}{t}}$ where $\iota := \log\left(\frac{K}{p}\right)$ and some absolute constant c (shown later in the analysis), then under certain conditions (shown later in this paper), the Q -learning with UCB exploration algorithm (in Algorithm 1) has the performance measure $\text{Regret}(K) = \sum_{k=1}^K (V_1^*(x_1^k) - V_1^{\pi_k}(x_1^k))$ bounded above by $\tilde{O}(K^{\frac{d+1}{d+2}})$ with probability $1 - p$, where d is the covering dimension of the underlying $\mathcal{S} \times \mathcal{A}$.

We can propose the corresponding main algorithm in the following:

Algorithm 1 Q -learning with UCB bandit in metric space.

```

1: initialize a covering  $\mathcal{B}$  of  $\mathcal{S} \times \mathcal{A}$  with ball of  $r = 1$ . Initialize  $Q_h(x, a) \leftarrow H$  and  $n_h(x, a) \leftarrow 0$  for all  $(x, a)$ , and  $V_h(x) \leftarrow H$  for all  $x$ . Choose  $p \in (0, 1)$ .
2: for  $k = 1, 2, \dots, K$  do
3:   receive  $x_1$  from randomly sampling.
4:   for  $h = 1, 2, \dots, H$  do
5:     if  $r \leq \frac{c}{8L} \sqrt{\frac{H^3 \iota}{1+n_h(x_h, a_h)}}$  is violated then
6:       for the violated ball  $B(x_h, a_h)$ ,  $r \leftarrow \frac{r}{2}$ , while keeping the same center.
7:       while there exists some region in  $\mathcal{S} \times \mathcal{A}$  not covered by  $\mathcal{B}$  do
8:          $\mathcal{B} \leftarrow \mathcal{B} \cup B_r(x, a)$ 
9:       update  $n_h(x_h, a_h) \leftarrow n_h(x_h, a_h) + 1$ .
10:      take action  $a_h \leftarrow \arg \max_{a' \in \mathcal{A}} Q_h(x_h, a')$  and observe  $x_{h+1}$ .
11:      iteratively update  $Q_h(x, a)$  and  $V_h(x)$  if  $(x, a)$  in  $B(x_h, a_h)$ :
          
$$Q_h(x_h, a_h) \leftarrow (1 - \alpha_t)Q_h(x_h, a_h) + \alpha_t[r_h(x_h, a_h) + V_{h+1}(x_{h+1}) + b_t].$$

12:
```

$$V_h(x_h) \leftarrow \min \left\{ H, \sup_{a' \in \mathcal{A}} Q_h(x_h, a') \right\}.$$

The main theorem is a bound for the cumulative regret in (4).

In the following sections, we prove some very useful techniques to bound the performance measure. We provide the novel and rigorous analysis that can be valuable in the analysis of the Q -learning problem in general.

5. Main Analysis

5.1. Concentration of Measure

First, we use the following lemmas to bound $(Q^k - Q^*)$ with high concentration.

Recall that the chosen bandit $b_t = c\sqrt{\frac{H^3 \iota}{t}}$. Let $\beta_t := \sum_{i=1}^t \alpha_i^j b_i$. The next-step empirical observation for (x, a) at step h in episode k is $\hat{\mathcal{P}}_h^k V_h(x, a) := V_{h+1}(x_{h+1}^k)$.

Lemma 2. For any $(x, a, h, k) \in \mathcal{S} \times \mathcal{A} \times [H] \times [K]$, $t = n_h^k(x, a)$, and suppose $B(x, a)$ was selected previously at step h of episodes $\{k_i\}_{i=1}^t$. We have

$$\begin{aligned} (Q_h^k - Q_h^*)(x, a) &= \alpha_t^0 (H - Q_h^*(x, a)) \\ &+ \sum_{i=1}^t \alpha_t^i \left[\left(V_{h+1}^{k_i} - V_{h+1}^* \right) \right. \\ &\quad \left. (x_{h+1}^{k_i}) + \left(\hat{\mathcal{P}}_h^{k_i} - \mathcal{P}_h \right) \right. \\ &\quad \left. V_h^*(x, a) + \left(\hat{\mathcal{P}}_h^{k_i} V_h^*(x_{h+1}^{k_i}, a_{h+1}^{k_i}) - \hat{\mathcal{P}}_h^{k_i} V_h^*(x, a) \right) + b_i \right]. \end{aligned} \quad (9)$$

Proof. From the Bellman optimality in Equation (2), $Q_h^*(x, a) = (r_h + \mathcal{P}_h V_h^*)(x, a)$. Using the fact that $\sum_{i=0}^t \alpha_t^i = 1$ from the definition of learning rate, $Q_h^*(x, a) = \alpha_t^0 Q_h^*(x, a) + \sum_{i=1}^t \alpha_t^i \cdot Q_h^*(x, a) = \alpha_t^0 Q_h^*(x, a) + \sum_{i=1}^t \alpha_t^i [r_h(x, a) + \mathcal{P}_h V_h^*(x, a)]$. From (8), $(Q_h^k - Q_h^*)(x, a) = \alpha_t^0 (H - Q_h^*(x, a)) + \sum_{i=1}^t \alpha_t^i \left[V_{h+1}^{k_i} (x_{h+1}^{k_i}) - \mathcal{P}_h V_h^*(x, a) + b_i \right]$,

$$\begin{aligned} (Q_h^k - Q_h^*)(x, a) &= \alpha_t^0 (H - Q_h^*(x, a)) + \sum_{i=1}^t \alpha_t^i \left[\left(V_{h+1}^{k_i} - V_{h+1}^* \right) (x_{h+1}^{k_i}) + \left(\hat{\mathcal{P}}_h^{k_i} - \mathcal{P}_h \right) V_h^*(x, a) + b_i \right] \\ &+ \sum_{i=1}^t \alpha_t^i \left[V_{h+1}^* (x_{h+1}^{k_i}) - \hat{\mathcal{P}}_h^{k_i} V_h^*(x, a) \right]. \end{aligned} \quad (10)$$

Note that $x_{h+1}^{k_i}$ is the next step from $(x_h^{k_i}, a_h^{k_i})$ where $(x_h^{k_i}, a_h^{k_i}) \in B(x, a)$ by the Definition 1. Additionally, $V_{h+1}^*(x_{h+1}^{k_i}) - \hat{\mathcal{P}}_h^{k_i} V_h^*(x, a)$ satisfies

$$\begin{aligned} &V_{h+1}^*(x_{h+1}^{k_i}) - \hat{\mathcal{P}}_h^{k_i} V_h^*(x, a) \\ &= \left(V_{h+1}^*(x_{h+1}^{k_i}) - \hat{\mathcal{P}}_h^{k_i} V_h^*(x_h^{k_i}, a_h^{k_i}) \right) + \left(\hat{\mathcal{P}}_h^{k_i} V_h^*(x_h^{k_i}, a_h^{k_i}) - \hat{\mathcal{P}}_h^{k_i} V_h^*(x, a) \right) \\ &= \hat{\mathcal{P}}_h^{k_i} V_h^*(x_h^{k_i}, a_h^{k_i}) - \hat{\mathcal{P}}_h^{k_i} V_h^*(x, a), \end{aligned} \quad (11)$$

where $V_{h+1}^*(x_{h+1}^{k_i}) = \hat{\mathcal{P}}_h^{k_i} V_h^*(x_h^{k_i}, a_h^{k_i})$ by (1). Combining (10) and (11), we have the statement of Lemma 2.

□

To complete the concentration of $Q^k - Q^*$, we bound each term in (9).

Lemma 3. $\left| \sum_{i=1}^t \alpha_t^i \left(\hat{\mathcal{P}}_h^{k_i} V_h^*(x, a) - \hat{\mathcal{P}}_h^{k_i} V_h^*(x_h^{k_i}, a_h^{k_i}) \right) \right| \leq \frac{\beta_t}{4}$.

Proof. Note that $\beta_t = \sum_{i=1}^t \alpha_t^i b_i = \sum_{i=1}^t \alpha_t^i c \sqrt{\frac{H^3 l}{i}} = c \sqrt{H^3 l} \sum_{i=1}^t \frac{\alpha_t^i}{\sqrt{i}} \geq c \sqrt{\frac{H^3 l}{t}}$. Since $\left\{ (x_h^{k_i}, a_h^{k_i}) \right\}_{i=1}^t \in B(x, a)$, the distance between $(x_h^{k_i}, a_h^{k_i})$ and (x, a) is bounded by the radius of the ball. As Algorithm 1 goes on, it keeps refining balls, and hence, the bound will be smaller. By the Lipschitz assumption of $\hat{\mathcal{P}} V^*$ in (3), $\left\| \hat{\mathcal{P}}_h^{k_i} V_h^*(x, a) - \hat{\mathcal{P}}_h^{k_i} V_h^*(x', a') \right\| \leq L \cdot d_{\mathcal{S} \times \mathcal{A}}((x, a), (x', a'))$.

Following from Algorithm 1 and Lemma 1,

$$\left| \sum_{i=1}^t \alpha_t^i \left(\hat{\mathcal{P}}_h^{k_i} V_h^*(x, a) - \hat{\mathcal{P}}_h^{k_i} V_h^*(x_h^{k_i}, a_h^{k_i}) \right) \right| \leq \sum_{i=1}^t \alpha_t^i \left(\frac{c}{8} \sqrt{\frac{H^3 l}{1 + n_h^k(x, a)}} \right) \leq \frac{\beta_t}{4}.$$

□

Lemma 4. For any $p \in (0, 1)$, with a probability of at least $1 - p$, we have

$$0 \leq (Q_h^k - Q_h^*)(x, a) \leq \alpha_t^0 H + \sum_{i=1}^t \alpha_t^i (V_{h+1}^{k_i} - V_{h+1}^*) (x_{h+1}^{k_i}) + \frac{3}{2} \beta_t.$$

The proof of Lemma 4 is similar to Lemma 4.3 in [12].

Proof. Based on the definition in (7),

$$\beta_t = \sum_{i=1}^t \alpha_t^i b_i \leq \sum_{i=1}^t \alpha_i b_i = \sum_{i=1}^t \frac{1}{i} c \sqrt{\frac{H^3 t}{i}} < 4c \sqrt{\frac{H^3 t}{t}}.$$

Recall $t = n_h^k(x, a)$. Let $k_i = \min(\{k \in [K] | k > k_{i-1} \wedge (x_h^k, a_h^k) \in B(x, a)\} \cup \{K+1\})$. That is, k_i is the episode in which $B(x, a)$ was selected at step h for the i th time (or $k_i = K+1$ if it is selected for fewer than i times). The random variable k_i is a stopping time in this case. Let $\mathcal{F}_i$ be the σ -field generated by all the random variables until episode k_i at step h . Then $(\mathbb{1}_{k_i \leq K} [(\hat{\mathcal{P}}_h^{k_i} - \mathcal{P}_h) V_h^*](x, a))_{i=1}^t$ is a martingale difference sequence with filtration $\{\mathcal{F}_i\}_{i \geq 0}$. By the sequence bounded difference, for any i , $\alpha_t^i [(\hat{\mathcal{P}}_h^{k_i} - \mathcal{P}_h) V_h^*](x, a) \leq \alpha_t^i \cdot H$. Applying the Azuma–Hoeffding inequality, for any $s > 0$, we have

$$\mathcal{P} \left(\left| \sum_{i=1}^t \alpha_t^i [(\hat{\mathcal{P}}_h^{k_i} - \mathcal{P}_h) V_h^*](x, a) \right| \leq s \right) \geq 1 - 2 \exp \left\{ \frac{-s^2}{2H^2 \sum_{i=1}^t (\alpha_t^i)^2} \right\}. \quad (12)$$

There exists some absolute constant c , such that $\sqrt{2} \cdot H \sqrt{\sum_{i=1}^t (\alpha_t^i)^2} \sqrt{\ln \left(\frac{2H}{p} \right)} \leq s \leq \frac{c}{4} \sqrt{\frac{H^3 t}{t}}$. This is true since (1)

$$\beta_t = \sum_{i=1}^t \alpha_t^i b_i = \sum_{i=1}^t \alpha_t^i c \sqrt{\frac{H^3 t}{i}} = c \sqrt{H^3 t} \sum_{i=1}^t \frac{\alpha_t^i}{\sqrt{i}} \geq c \sqrt{\frac{H^3 t}{t}} \geq 4s, \quad (13)$$

and (2). since $s \geq \sqrt{2} \cdot H \sqrt{\sum_{i=1}^t (\alpha_t^i)^2} \sqrt{\ln \left(\frac{2H}{p} \right)}$. Then

$$1 - 2 \exp \left\{ \frac{-s^2}{2H^2 \sum_{i=1}^t (\alpha_t^i)^2} \right\} \geq 1 - \frac{p}{H} > 1 - p. \quad (14)$$

From (12) and (13), and for any fixed step h , we have

$$\mathcal{P} \left(\left| \sum_{i=1}^t \alpha_t^i [(\hat{\mathcal{P}}_h^{k_i} - \mathcal{P}_h) V_h^*](x, a) \right| \leq \frac{\beta_t}{4} \right) \geq 1 - p. \quad (15)$$

From (15), Lemmas 2 and 3, we have the right-hand side of Lemma 4.

Now for the left-hand side, we use the backward induction on $h = H, H-1, \dots, 1$. Observe the initial case $(Q_H^k - Q_H^*)(x, a) \geq 0$. Assume $(Q_h^k - Q_h^*)(x, a) \geq 0$. Since $V_h^k(x) = \sup_{a' \in \mathcal{A}} Q_h^k(x, a') \geq \sup_{a' \in \mathcal{A}} Q_h^*(x, a') = V_h^*(x)$. From (9) and (14), this implies that $(Q_{h-1}^k - Q_{h-1}^*)(x, a) \geq 0$ with a probability of at least $1 - \frac{p}{H}$ for one step backward. Now using the union bound on $h = H, H-1, \dots, 1$, we have the left-hand side. We hence have for all h with a probability of at least $1 - p$, $0 \leq (Q_h^k - Q_h^*)(x, a) \leq \alpha_t^0 H + \sum_{i=1}^t \alpha_t^i (V_{h+1}^{k_i} - V_{h+1}^*) (x_{h+1}^{k_i}) + \frac{3}{2} \beta_t$. $\square$

5.2. Assumptions for Concentration

We need two mild and reasonable assumptions shown below. From Lemma 4,

$$\sum_{k=1}^K (Q_h^k - Q_h^*) (x, a) \leq \sum_{k=1}^K \alpha_t^0 H + \sum_{k=1}^K \sum_{i=1}^t \alpha_t^i (V_{h+1}^{k_i} - V_{h+1}^*) (x_{h+1}^{k_i}) + \frac{3}{2} \sum_{k=1}^K \beta_t, \quad (16)$$

for any $i = 1, \dots, n_h^k$ where $n_h^k = t$, each k_i is the actual selection. Here we make the first mild assumption: $V_{h+1}^{k_i} - V_{h+1}^* \leq R \cdot \sqrt{\frac{H^3 t}{t}}$ for some large R . This assumption is very mild due to the iterative convergence. Then, $V_{h+1}^{k_i} - V_{h+1}^* \leq \frac{R}{c} \cdot b_i$. We hence have

$$\sum_{i=1}^t \alpha_t^i (V_{h+1}^{k_i} - V_{h+1}^*) (x_{h+1}^{k_i}) \leq \frac{R}{c} \sum_{i=1}^t \alpha_t^i b_i = \frac{R}{c} \cdot \beta_t. \quad (17)$$

In (16), we have a term $\sum_{k=1}^K \alpha_t^0 \cdot H$. Recall from (7), $\alpha_j = \frac{1}{j}$. Since

$$\alpha_t^0 = \prod_{j=1}^{n_k} (1 - \alpha_j) = \begin{cases} 1 & \text{if } n_k = 0; \\ 0 & \text{if } n_k \geq 1, \end{cases} \quad (18)$$

then $\sum_{k=1}^K \alpha_t^0 \cdot H = H \sum_{k=1}^K \prod_{j=1}^{n_k} (1 - \alpha_j) = H \sum_{k=1}^K \mathbb{1}_{\{n_k=0\}}$. Intuitively, $\mathbb{1}_{\{n_k=0\}}$ stands for the partitioned balls from the algorithm with no active state-action pair contained. Here, we make the second mild assumption: $\sum_{k=1}^K \mathbb{1}_{\{n_k=0\}} \leq \Theta(K^{\frac{d+1}{d+2}})$. This assumption is again mild, since we will soon show the cumulative number of partitions $\mathbf{P}_K$ up to K is bounded above by $\Theta(K^{\frac{d}{d+2}})$ in Lemma 7. Note that here, d is the covering dimension in $\mathcal{S} \times \mathcal{A}$ that will be introduced in the next section.

From the second mild assumption, (16), (17) and Lemma 4, for any $p \in (0, 1)$, with a probability of at least $1 - p$, we have

$$\begin{aligned} \sum_{k=1}^K (Q_h^k - Q_h^*) (x, a) &\leq \sum_{k=1}^K \left(\alpha_t^0 \cdot H + \sum_{i=1}^t \alpha_t^i (V_{h+1}^{k_i} - V_{h+1}^*) (x_{h+1}^{k_i}) + \frac{3}{2} \beta_t \right) \\ &\leq \Theta(K^{\frac{d+1}{d+2}}) + \mu \sum_{k=1}^K \beta_t, \end{aligned} \quad (19)$$

where $\mu = \frac{R}{c} + \frac{3}{2}$. To bound $(Q_h^k - Q_h^*)$, the remaining is to bound β_t .

Note that we also provide some additional approaches that can be valuable in bounding Q-value function in the Appendix A. These approaches can be well applicable given some special conditions satisfied.

5.3. Upper Bound of Cumulative Regret

Now we return to our main discussion about the $\text{Regret}(K)$. The following lemma shows the relationship between $\text{Regret}(K)$ and $(Q_h^k - Q_h^*)$. Note that the following approach provides a very general trajectory analysis technique that can be easily applicable to reinforcement learning problems with the continuous state and action space framework.

Lemma 5. For any $p \in (0, 1)$, with a probability of at least $1 - p$, the cumulative regret up to K has the following upper bound:

$$\begin{aligned} \sum_{k=1}^K V_1^k(x_1^k) - V_1^{\pi_k}(x_1^k) &\leq \Theta(K^{\frac{d+1}{d+2}}) \\ &+ \sum_{j=0}^{H-1} \int_{\text{traj}(x_h^k, a_h^k) \in \text{TRAJ}(h \rightarrow h+j)} \mathcal{P}(\text{traj}(x_h^k, a_h^k)) \\ &\cdot \left(\sum_{k=1}^K (Q_{h+j}^k - Q_{h+j}^*)(s, a^s) \right) d\mathcal{P}. \end{aligned}$$

Here, the sequence of actions $\{a^s\}$ is corresponding to the sequence of states $\{s\}$, which is the same as maximum selection in (21) and $\text{traj}(x_h^k, a_h^k)$ is the trajectory starting from (x_h^k, a_h^k) . $\text{TRAJ}(h \rightarrow h+j)$ is a trajectory space that contains all the possible trajectories from step h up to step $h+j$, where d is the covering dimension of $\mathcal{S} \times \mathcal{A}$.

Proof. The proof follows the idea from the Ref. [24]. Note that for some action a_h^k ,

$$V_h^k(x_h^k) - V_h^{\pi_k}(x_h^k) \leq (Q_h^k(x_h^k, a_h^k) - Q_h^*(x_h^k, a_h^k)) + (Q_h^*(x_h^k, a_h^k) - Q_h^{\pi_k}(x_h^k, a_h^k)) + \frac{k^{\frac{-1}{d+2}}}{H}. \quad (20)$$

The inequality holds because

$$V_h^k(x_h^k) \leq \sup_{a' \in \mathcal{A}} Q_h^k(x_h^k, a') \leq Q_h^k(x_h^k, a_h^k) + \frac{k^{\frac{-1}{d+2}}}{H} \quad (21)$$

for some a_h^k from (6). We name it maximum action selection. Additionally,

$$\begin{aligned} Q_h^*(x_h^k, a_h^k) - Q_h^{\pi_k}(x_h^k, a_h^k) &= (\mathcal{P}_h V_h^*(x_h^k, a_h^k) - \mathcal{P}_h V_h^{\pi_k}(x_h^k, a_h^k)) \\ &= \int_{s \in \mathcal{S}} \mathcal{P}(s | x_h^k, a_h^k) (V_{h+1}^* - V_{h+1}^{\pi_k})(s) d\mathcal{P}(s), \end{aligned} \quad (22)$$

where we take all the states at step $h+1$ into account, $s \sim \mathcal{P}(\cdot | x_h^k, a_h^k)$. Note that from Lemma 4, for any p , we have $Q_h^k \geq Q_h^*$ with a probability of at least $1 - p$ for all steps h . Then with a probability of at least $1 - p$, $V_h^k = \sup_{a \in \mathcal{A}} Q_h^k \geq \sup_{a \in \mathcal{A}} Q_h^* = V_h^*$. Combining this inequality, (20) and (22), we have

$$V_h^k(x_h^k) - V_h^{\pi_k}(x_h^k) \leq \frac{k^{\frac{-1}{d+2}}}{H} + (Q_h^k(x_h^k, a_h^k) - Q_h^*(x_h^k, a_h^k)) + \int_{s \in \mathcal{S}} \mathcal{P}(s | x_h^k, a_h^k) (V_{h+1}^k - V_{h+1}^{\pi_k})(s) d\mathcal{P}(s) \quad (23)$$

We observe that (23), $(V_h^k - V_h^{\pi_k})(x_h^k)$ is on the left-hand side and $(V_{h+1}^k - V_{h+1}^{\pi_k})(s)$ is in the integral on the right-hand side. Recall that the space that contains all the possible

trajectories from step h up to step $h + j$ is $\text{TRAJ}(h \rightarrow h + j)$. We have the following recursive relationship:

$$\begin{aligned}
 V_h^k(x_h^k) - V_h^{\pi_k}(x_h^k) &\leq \frac{2k^{\frac{-1}{d+2}}}{H} + \left(Q_h^k(x_h^k, a_h^k) - Q_h^*(x_h^k, a_h^k) \right) \\
 &+ \int_{s \in \mathcal{S}} \mathcal{P}(s | x_h^k, a_h^k) \\
 &\left((Q_{h+1}^k(s, a^s) - Q_{h+1}^*(s, a^s)) + \int_{s' \in \mathcal{S}} \mathcal{P}(s' | s, a) \right. \\
 &\left. (V_{h+2}^k - V_{h+2}^{\pi_k})(s') d\mathcal{P}(s') \right) \\
 &d\mathcal{P}(s) \dots \\
 &\leq k^{\frac{-1}{d+2}} + \sum_{j=0}^{H-h} \int_{\text{traj}(x_h^k, a_h^k) \in \text{TRAJ}(h \rightarrow h+j)} (\mathcal{P}(\text{traj}(x_h^k, a_h^k))) \\
 &\cdot (Q_{h+j}^k - Q_{h+j}^*)(s, a^s) d\mathcal{P}, \tag{24}
 \end{aligned}$$

where sequence $\{a^s\}$ corresponding to each $\{s\}$ is the same as our selection in (21), and $\text{traj}(x_h^k, a_h^k)$ is the trajectory starting from (x_h^k, a_h^k) .

For simplicity, we use $(Q_{h+j}^k - Q_{h+j}^*)$ to stand for $(Q_{h+j}^k - Q_{h+j}^*)(s, a^s)$ and traj to stand for $\text{traj}(x_h^k, a_h^k)$. Then,

$$\sum_{k=1}^K V_1^k(x_1^k) - V_1^{\pi_k}(x_1^k) \leq \sum_{k=1}^K k^{\frac{-1}{d+2}} + \sum_{k=1}^K \sum_{j=0}^{H-1} \int_{\text{traj} \in \text{TRAJ}(h \rightarrow h+j)} \mathcal{P}(\text{traj}) \cdot (Q_{1+j}^k - Q_{1+j}^*) d\mathcal{P}. \tag{25}$$

If we apply Fubini's theorem [25], then (25) is equivalent to

$$\sum_{k=1}^K V_1^k(x_1^k) - V_1^{\pi_k}(x_1^k) \leq \Theta(K^{\frac{d+1}{d+2}}) + \sum_{j=0}^{H-1} \int_{\text{traj} \in \text{TRAJ}(h \rightarrow h+j)} \mathcal{P}(\text{traj}) \cdot \left(\sum_{k=1}^K (Q_{1+j}^k - Q_{1+j}^*) \right) d\mathcal{P}.$$

□

6. Regret Bound in Covering Dimension

We have mentioned the covering dimension in the earlier section. For bounding the cumulative regret (performance measure), a covering dimension on the regret bound has been suggested [15,16,26]. The covering dimension is a useful tool when we need to reduce the upper bound of the regret in multi-armed bandit problems. In this section, we will present the analysis of a tight upper bound in the covering dimension. Note that this proof technique is very general and can be easily extended to works where a covering dimension is applicable. We will follow the same definition as in the Refs. [15,16,26].

Definition 2. Consider a metric space $(\mathcal{X}, l)$. Denote by $N(r)$ the minimum number of balls with diameter r to cover $\mathcal{X}$. The covering dimension of $\mathcal{X}$ is defined as $\text{COVD}(\mathcal{X}) = \inf\{d : \exists c \forall r > 0 \ N(r) \leq c \cdot r^{-d}\}$.

Lemma 6. Let $\mathcal{E}$ be a bounded space that can be embedded into an n -dimensional Euclidean space. Then, $\text{COVD}(\mathcal{E}) \leq n$.

Proof. W.L.O.G. we assume $\mathcal{E} \subseteq [0, 1]^n$. Now, let $r > 0$ be any real number, then we can find $\left(\frac{\sqrt{n}}{r}\right)^n$ many n -dimensional hypercubes with side-length $\frac{r}{\sqrt{n}}$ covering $[0, 1]^n$. Now for every such hypercube, there exists a ball with diameter r such that the hypercube is

contained inside the ball. Hence, we can have $c \cdot r^{-n}$ many balls with diameter r cover $\mathcal{E}$, where the constant $c = (\sqrt{n})^n$ in this case. Since $\text{COV}(\mathcal{E})$ takes the infimum, then $\text{COV}(\mathcal{E}) \leq n$. $\square$

Because of Lemma 6, to obtain a tighter upper bound, we use the covering dimension of $\mathcal{S} \times \mathcal{A}$ instead of the regular Euclidean dimension. In this section, we use the covering dimension to bound $\sum_{k=1}^K \beta_t$. Assume $\mathbf{P}_K$ stands for the number of balls (partitions) created up to episode K in the Algorithm 1.

Lemma 7. We have the tight bound $\mathbf{P}_K \leq \Theta\left(K^{\frac{d}{d+2}}\right)$.

Proof. Assume d is the covering dimension of $\mathcal{S} \times \mathcal{A}$. Then, the number of balls with diameter m to cover $\mathcal{S} \times \mathcal{A}$ will be bounded by $c \cdot m^{-d}$ for some c . Note that Algorithm 1 produces many balls during adaptive partition. The center of the balls (past active points) of radius $\frac{m}{2}$ will always be in different balls from the covering of $\mathcal{S} \times \mathcal{A}$ because each pair of centers has a distance of at least m between them. For example, say (s_i, a_i) and (s_j, a_j) are two past active centers. Consider the covering of $\mathcal{S} \times \mathcal{A}$. Then, $(s_i, a_i) \in B_i$ and $(s_j, a_j) \in B_j$ for B_i, B_j in the covering, and they are disjoint. Thus, the number of the balls with diameter m from Algorithm 1 is bounded by $c \cdot m^{-d}$, the covering number.

Note that for each ball B of radius r with the center inside $\mathcal{S} \times \mathcal{A}$, we can use at most $c' \cdot 2^n$ balls of radius $\frac{r}{2}$ to cover $B \cap (\mathcal{S} \times \mathcal{A})$, where n is the Euclidean dimension of $\mathcal{S} \times \mathcal{A}$ and c' is some constant. From the algorithm, each ball will split only if $r \leq c\sqrt{\frac{H^3 l}{1+n_k}}$ is violated. That is, for any ball of radius $\frac{r}{2}$ split from the original ball of radius r (condition violated), it must be filled by $c^2 \cdot \frac{H^3 l}{r^2}$ past active points. Since there are K points in total, then the number of balls of radius r violated is at most $\frac{K}{c^2 \cdot \frac{H^3 l}{r^2}} = \frac{1}{c^2 H^3 l} \cdot r^2 K$. Thus, after splitting (violation), the number of balls of radius $\frac{r}{2}$ is at most $(c' \cdot 2^n) \cdot \left(\frac{1}{c^2 H^3 l} \cdot r^2 K\right) = c'' \cdot r^2 K$, where $c'' = c' \cdot 2^n \cdot \frac{1}{c^2 H^3 l}$.

Along the algorithm, as radius r becomes $\frac{r}{2}$, the number of partitioned balls is bounded by the covering number, which is $c \cdot r^{-d}$, and also bounded by $c'' \cdot r^2 K$ for any $r > 0$ from the above analysis. Consider $r = 1, \frac{1}{2}, \frac{1}{4}, \dots, 2^{-i} \dots$, where we have $\mathbf{P}_K = \sum_{i=1}^{\infty} \min\left\{c(2^{-i})^{-d}, c''(2^{-i})^2 K\right\}$.

We split this term into two cases. (1). For a large radius, use $c \cdot r^{-d}$ and (2). For a very small radius, use the bound $c'' \cdot r^2 K$. To maximize the upper bound, we find some optimal J where $c(2^{-J})^{-d} = c \cdot 2^{J \cdot d} = c'' \cdot 2^{-2J} \cdot K$. Thus, $2^{J \cdot d} = \Theta(K^{\frac{d}{d+2}})$, where $D = \frac{c''}{c}$. This implies that $\mathbf{P}_K \leq \sum_{i=1}^J c(2^{-i})^{-d} + \sum_{i=J+1}^{\infty} c''(2^{-i})^2 K \leq 2c \cdot 2^{J \cdot d} + 8c'' \cdot 2^{-2(J+1)} K = \Theta\left(K^{\frac{d}{d+2}}\right)$. $\square$

Lemma 8. We have the following tight upper bound $\sum_{k=1}^K \beta_t \leq \tilde{\mathcal{O}}\left(K^{\frac{d+1}{d+2}}\right)$, where d is the covering dimension of $\mathcal{S} \times \mathcal{A}$.

Proof. Recall that $\beta_t = \sum_{i=1}^t \alpha_t^i b_i \leq \sum_{i=1}^t \alpha_i b_i = \sum_{i=1}^t \frac{1}{i} c \sqrt{\frac{H^3 l}{i}} < 4c \sqrt{\frac{H^3 l}{t}}$. Up to episode K , we create $\mathbf{P}_K$ many balls in total. Assume there are b_i past active state-action pairs in the i^{th} ball. Clearly, $\sum_{i=1}^{\mathbf{P}_K} b_i = K$. Thus,

$$\begin{aligned}
\sum_{k=1}^K \beta_t &\leq \sum_{k=1}^K 4c \sqrt{\frac{H^3 t}{t}} \leq \sum_{i=1}^{\mathbf{P}_K} \sum_{j=1}^{b_i} 4c \sqrt{\frac{H^3 t}{j}} \\
&\leq \sum_{i=1}^{\mathbf{P}_K} 4c \sqrt{H^3 t} \cdot 2\sqrt{b_i} = \sum_{i=1}^{\mathbf{P}_K} 8c \sqrt{H^3 t} \cdot \sqrt{b_i} \\
&\leq 8c \sqrt{H^3 t} \cdot \mathbf{P}_K \sqrt{\sum_{i=1}^{\mathbf{P}_K} \frac{1}{\mathbf{P}_K} \cdot b_i} \quad (26)
\end{aligned}$$

$$= 8c \sqrt{H^3 t} \cdot \mathbf{P}_K \sqrt{\frac{K}{\mathbf{P}_K}} \quad (27)$$

$$= 8c \sqrt{H^3 t} \cdot \sqrt{K} \sqrt{\mathbf{P}_K}, \quad (28)$$

where (26) is from Jensen's inequality on the concave function $\sqrt{x}$, and (27) is from $\sum_{i=1}^{\mathbf{P}_K} b_i = K$. With (28), we have $\sum_{k=1}^K \beta_t \leq \tilde{\mathcal{O}}\left(K^{\frac{d+1}{d+2}}\right)$.

□

7. Conclusions of Main Theorem

7.1. Proof of Main Theorem

We can combine the above results to complete the proof of the main theorem of this paper, Theorem 1. Recall that from Lemma 5, the cumulative regret up to episode K is given by

$$\sum_{k=1}^K V_1^k(x_1^k) - V_1^{\pi_k}(x_1^k) \leq \Theta(K^{\frac{d+1}{d+2}}) + \sum_{j=0}^{H-1} \int_{\text{traj} \in \text{TRAJ}(h \rightarrow h+j)} \mathcal{P}(\text{traj}) \cdot \sum_{k=1}^K (Q_{1+j}^k - Q_{1+j}^*) d\mathcal{P}. \quad (29)$$

In addition, from (19), $\sum_{k=1}^K (Q_h^k - Q_h^*)(x, a) \leq \Theta\left(K^{\frac{d+1}{d+2}}\right) + \mu \sum_{k=1}^K \beta_t$. Combined with Lemma 8, we have $\sum_{k=1}^K (Q_h^k - Q_h^*) \leq \tilde{\mathcal{O}}\left(K^{\frac{d+1}{d+2}}\right)$, where d is the covering dimension of $\mathcal{S} \times \mathcal{A}$. Thus, $\max_{h \leq H} \sum_{k=1}^K (Q_h^k - Q_h^*) \leq \tilde{\mathcal{O}}\left(K^{\frac{d+1}{d+2}}\right)$. Inequality (29) becomes

$$\begin{aligned}
&\sum_{k=1}^K V_1^k(x_1) - V_1^{\pi_k}(x_1) \\
&\leq \Theta(K^{\frac{d+1}{d+2}}) + \sum_{j=0}^{H-1} \int_{\text{traj} \in \text{TRAJ}(h \rightarrow h+j)} \mathcal{P}(\text{traj}) \\
&\quad \cdot \left(\max_j \sum_{k=1}^K (Q_{1+j}^k - Q_{1+j}^*) \right) d\mathcal{P} \\
&\leq \Theta(K^{\frac{d+1}{d+2}}) + \sum_{j=0}^{H-1} \int_{\text{traj} \in \text{TRAJ}(h \rightarrow h+j)} \mathcal{P}(\text{traj}) \\
&\quad \cdot \tilde{\mathcal{O}}\left(K^{\frac{d+1}{d+2}}\right) d\mathcal{P} \\
&= \tilde{\mathcal{O}}\left(K^{\frac{d+1}{d+2}}\right).
\end{aligned}$$

This completes Theorem 1, which is the main scope of this paper.

7.2. Discussion and Conclusions

Some applications in discrete space include board games like poker, bridge, and chess. For example, DeepMind's powerful computer program AlphaGo Zero has its self-learning

algorithm based on a discrete space set up in Go [27]. AlphaGo demonstrated super-human capabilities in navigating through an exhaustive discrete action search space and achieved a breakthrough of AI in challenging gaming. On the other hand, most robotics and automated vehicle applications of reinforcement learning require continuous state spaces which utilize continuous variables, such as position, velocity, torque, and so forth. For example, motion estimation and depth estimation in the perception tasks of autonomous driving are in the continuous environment [28]. If reinforcement learning algorithms with a continuous state and action space environment can be efficiently implemented, then these applications, such as robotics and automated vehicles, can potentially be improved.

In this paper, to accommodate the needs of the continuous environment, we proposed a state-of-the-art Q -learning algorithm with an UCB exploration policy in the compact state-action metric space. Our algorithm adaptively partitioned the state-action space and flexibly selected the future actions to maximize Q -values with bandit exploration bonuses. By some mild assumptions for concentration, we found a tight upper bound with respect to covering dimension for the performance measure in a general state-action metric space. Our work contained rigorous techniques, including novel combinatorial and probabilistic analysis. These approaches are not limited to the analysis of performance measures in Q -learning, but are also valuable to general reinforcement learning analysis. This work suggests that the bandit strategy is an effective way to increase data efficiency for a general model-free reinforcement learning problem. Our analysis is easily applicable to reinforcement learning problems with the continuous state and action space framework. For future steps, we will apply our algorithms to various practices, including robotics and automated vehicles.

Author Contributions: Conceptualization, D.C.; Formal analysis, W.Y.; methodology, W.Y.; writing—original draft, W.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank Debsoumya Chakraborti, Shlomo Ta'asan and Tianyu Wang for their insight and comments.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Here we provide some approaches that can be valuable in bounding Q -value function.

Appendix A.1. Sublinearity

Lemma A1 (Sublinearity). *If Q_h^k converges to Q_h^* in k and fixed h , then $\sum_{k=1}^K |Q_h^k - Q_h^*| < \mathcal{O}(K)$.*

Proof. We prove by contradiction. Assume a_i converges to a . Then there exists $m > 0$ such that we have the contradiction: $\sum_{i=1}^K |a - a_i| \geq mK$ for any large K . This is equivalent to

$$\sum_{i=1}^K |a - a_i| - mK = \sum_{i=1}^K (|a - a_i| - m) \geq 0$$

Since a_i converges to a , there exists N_1 such that for any $i > N_1$, $|a - a_i| < \frac{m}{2}$, then $(|a - a_i| - m) < -\frac{m}{2}$. Hence for N_2 large, $\left| \sum_{i=N_1+1}^{N_2} (|a - a_i| - m) \right| > \left| \sum_{i=1}^{N_1} (|a - a_i| - m) \right|$ and $\sum_{i=N_1+1}^{N_2} (|a - a_i| - m) < 0$.

This implies that $\sum_{i=1}^{N_2} (|a - a_i| - m) < 0$. This contradicts our assumption and completes the proof of contradiction. For Q_h^k converges to Q_h^* in k , we have

$$\sum_{k=1}^K |Q_h^k - Q_h^*| < \mathcal{O}(K)$$

□

From Lemma 4 and the above Lemma A1, if one can have Q_h^k converges to Q_h^* in k , then one can show sublinear performance measure.

Appendix A.2. Differential Equation Approach

Here we use Grönwall's inequality. First we recall Grönwall's inequality. Given I is the interval of real line in a format of $[a, +\infty)$. If we have

$$u'(t) \leq \beta(t)u(t)$$

then for all t , u is bounded by the corresponding differential equation $v'(t) = \beta(t)v(t)$,

$$u(t) \leq u(a) \exp\left(\int_1^t \beta(s)ds\right)$$

Lemma A2 (Differential Equation Approach). For some τ where $0 < \tau < 1$, if

$$\frac{d}{dk}\theta_k \leq (\tau - 1) \cdot \frac{1}{k}\theta_k \quad (\text{A1})$$

Then $\sum_{k=1}^K |Q^k - Q^*| \leq \mathcal{O}(K^\tau)$.

Assume $\theta_k = Q_h^k - Q_h^*$, which stands for the difference between Q functions. By using Grönwall's inequality and the analysis of differential equations, we have

Proof. One potential approach is to use Grönwall's inequality [29]. Notice that $e^{\left(\int_1^k ((\tau-1) \cdot \frac{1}{s})ds\right)} = k^{(\tau-1)}$. By Gronwall's inequality, for constant $C = \theta_k$, if

$$\frac{d}{dk}\theta_k \leq (\tau - 1) \cdot \frac{1}{k}\theta_k \quad (\text{A2})$$

Then

$$\theta_k \leq C \cdot k^{\tau-1} \quad (\text{A3})$$

Thus, $\sum_{k=1}^K |Q^k - Q^*| = \sum_{k=1}^K \theta_k \leq \mathcal{O}(K^\tau)$. □

From Lemma 4 and the above Lemma A2, if the difference θ_k satisfies the given differential equation, then one can also show sublinear performance measure with specific convergence rate.

The above approaches are helpful to bound the performance measure in Q -learning.

References

1. Sutton, R.; Barto, A. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 1998.
2. Asadi, K.; Misra, D.; Litterman, M. Lipschitz continuity in model-based Reinforcement Learning. In Proceedings of the International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018; pp. 264–273.
3. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep Reinforcement Learning: A brief survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [[CrossRef](#)]
4. Schaal, S. Learning from demonstration. In *Advances in Neural Information Processing System*; Mozer, M.C., Jordan, M., Petsche, T., Eds.; MIT Press: Cambridge, MA, USA, 1997; Volume 9.
5. Schneider, J.G. Exploiting model uncertainty estimates for safe dynamic control learning. In *Advances in Neural Information Processing System*; Mozer, M.C., Jordan, M., Petsche, T., Eds.; MIT Press: Cambridge, MA, USA, 1997; Volume 9.

6. Deisenroth, M.P.; Rasmussen, C.E. PILCO: A model-based and data-efficient approach to policy search. In Proceedings of the International Conference on Machine Learning, Bellevue, WA, USA, 28 June–2 July 2011.
7. Watkins, C.J. Learning from Delayed Rewards. Ph.D. Thesis, King's College, University of Cambridge, Cambridge, UK, 1989.
8. Watkins, C.J.; Dayan, P. Technical Note: Q-Learning. *Mach. Learn.* **1992**, *8*, 279–292. [[CrossRef](#)]
9. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with deep Reinforcement Learning. *arXiv* **2013**, arXiv:1312.5602.
10. Mnih, V.; Badia, A.P.; Mehdi, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep Reinforcement Learning. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1928–1937.
11. Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; Moritz, P. Trust region policy optimization. In Proceedings of the International Conference on Machine Learning (ICML), Lille, France, 6–11 July 2015; pp. 1889–1897.
12. Jin, C.; Zhu, A.Z.; Yan, X.; Bubeck, S.; Jordan, M. Is Q-learning provably efficient? In Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montreal, QC, Canada, 3–8 December 2018.
13. Agrawal, S.; Jia, R. Optimistic posterior sampling for Reinforcement Learning: Worst-case regret bounds. In Proceedings of the 31st Conference on Neural Information Processing Systems (NeurIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
14. Azar, M.G.; Osband, I.; Munos, R. Minimax regret bounds for Reinforcement Learning. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017.
15. Kleinberg, R.; Slivkins, A.; Upfal, E. Multi-armed bandits in metric spaces. In Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, Victoria, BC, Canada, 17–20 May 2008; pp. 681–690.
16. Bubeck, S.; Munos, R.; Stoltz, G.; Szepesvári, C. X-armed bandits. *J. Mach. Learn. Res.* **2011**, *12*, 1655–1695.
17. Auer, P.; Ortner, R. UCB Revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Period. Math. Hung.* **2010**, *61*, 55–65. [[CrossRef](#)]
18. Wang, T.; Ye, W.; Geng, D.; Rudin, C. Towards practical Lipschitz bandits. In Proceedings of the 2020 ACM-IMS on Foundations of Data Science Conference, San Francisco, CA, USA, 19–20 October 2020; pp. 129–138.
19. Strehl, A.L.; Li, L.; Wiewiora, E.; Langford, J.; Litterman, M.L. PAC model-free reinforcement learning. In Proceedings of the International Conference on Machine Learning (ICML), Pittsburgh, PA, USA, 25–29 June 2006; pp. 881–888.
20. Szepesvari, C.; Smart, W.D. Interpolation-based Q-learning. In Proceedings of the International Conference on Machine Learning (ICML), Banff, AB, Canada, 4–8 July 2004.
21. Tsitsiklis, J.N.; van Roy, B. Feature-based methods for large scale dynamic programming. *Mach. Learn.* **1996**, *22*, 59–94. [[CrossRef](#)]
22. Gaskett, C.; Wettergreen, D.; Zelinsky, A. Q-Learning in continuous state and action spaces. In *Lecture Notes in Computer Science, Proceedings of the Advanced Topics in Artificial Intelligence, Sydney, Australia, 6–10 December 1999*; Springer: Berlin/Heidelberg, Germany, 1999.
23. Irpan, A. *Q-Learning in Continuous State Action Spaces*; Technical report; University of California: Berkeley, CA, USA, 2015.
24. Dong, K.; Wang, Y.; Chen, X.; Wang, L. Q-learning with UCB exploration is sample efficient for infinite-horizon MDP. *arXiv* **2019**, arXiv:1901.09311v1.
25. Folland, G.B. *Real Analysis: Modern Techniques and Their Applications*; Wiley: Hoboken, NJ, USA, 1999; Volume 40, pp. 67–68.
26. Slivkins, A. Contextual bandits with similarity information. *J. Mach. Learn. Res.* **2014**, *15*, 2533–2568.
27. Liu, Y.C.; Tsuruoka, Y. Asymmetric Move Selection Strategies in Monte-Carlo Tree Search: Minimizing the Simple Regret at Max Nodes. *arXiv* **2016**, arXiv:1605.02321.
28. Kiran, B.R.; Sobh, I.; Talpaert, V.; Mannion, P.; Al Sallab, A.A.; Yogamani, S.; Pérez, P. Deep reinforcement learning for autonomous driving: A survey. *IEEE Trans. Intell. Transp. Syst.* **2021**, early access. [[CrossRef](#)]
29. Grönwall, T.H. Note on the derivatives with respect to a parameter of the solutions of a system of differential equations. *Ann. Math.* **1919**, *20*, 292–296. [[CrossRef](#)]