Near-Optimal Model-Free Reinforcement Learning in Non-Stationary Episodic MDPs

Weichao Mao¹  Kaiqing Zhang¹  Ruihao Zhu²  David Simchi-Levi²  Tamer Başar¹

Abstract

We consider model-free reinforcement learning (RL) in non-stationary Markov decision processes. Both the reward functions and the state transition functions are allowed to vary arbitrarily over time as long as their cumulative variations do not exceed certain variation budgets. We propose Restarted Q-Learning with Upper Confidence Bounds (RestartQ-UCB), the first model-free algorithm for non-stationary RL, and show that it outperforms existing solutions in terms of dynamic regret. Specifically, RestartQ-UCB with Freedman-type bonus terms achieves a dynamic regret bound of $\tilde{O}(S^3A^2\Delta^{\frac{1}{2}}HT^{\frac{3}{2}})$, where $S$ and $A$ are the numbers of states and actions, respectively, $\Delta > 0$ is the variation budget, $H$ is the number of time steps per episode, and $T$ is the total number of time steps. We further show that our algorithm is nearly optimal by establishing an information-theoretical lower bound of $\Omega(S^3A^2\Delta^{\frac{1}{2}}HT^{\frac{3}{2}})$, the first lower bound in non-stationary RL. Numerical experiments validate the advantages of RestartQ-UCB in terms of both cumulative rewards and computational efficiency. We further demonstrate the power of our results in the context of multi-agent RL, where non-stationarity is a key challenge.

1. Introduction

Reinforcement learning (RL) focuses on the class of problems where an agent maximizes its cumulative reward through sequential interactions with an initially unknown but fixed environment, usually modeled by a Markov Decision Process (MDP). In classical RL problems, the state transition functions and the reward functions are assumed to be time-invariant, i.e., stationary. However, stationary models cannot capture the time-varying environments in a wide range of sequential decision-making problems, such as online advertisement auctions (Cai et al., 2017; Lu et al., 2019), dynamic pricing (Chawla et al., 2016; Mao et al., 2018), traffic management (Chen et al., 2020), healthcare operations (Shortreed et al., 2011), and inventory control (Agrawal & Jia, 2019).

Among others, we want to highlight the connection between non-stationary RL and multi-agent RL (Littman, 1994). In multi-agent RL, a set of agents collaborate or compete by taking actions in a shared environment. Consequently, each agent is facing a non-stationary environment, especially when the agents learn and update policies simultaneously, as the actions of the other agents can alter the environment. We discuss this connection in greater details in Section 7 and also provide more applications of non-stationary RL to other important problems, such as sequential transfer and multi-task RL, in Appendix A.

RL in a non-stationary MDP is highly non-trivial due to the following challenges. First, similar to stationary RL, the agent faces the exploration vs. exploitation dilemma: it needs to explore the uncertain environment efficiently while maximizing its rewards along the way. In Jaksch et al. (2010), the authors proposed to leverage the "optimism in the face of uncertain" principle to guide exploration. Another challenge, which is unique to non-stationary RL, is the trade-off between remembering and forgetting. On the one hand, since the underlying MDP varies over time, data samples collected in prior interactions can become obsolete. In fact, it has been shown that a standard stationary RL algorithm might incur a linear regret if the non-stationarity is not handled properly (Ortner et al., 2019). On the other hand, the agent needs to extract a sufficient amount of information from historical data to inform future decision-making.

To resolve the aforementioned challenges, Ortner et al. (2019) and Cheung et al. (2020) have proposed algorithms to guide learning in non-stationary MDPs. Although both
model-based and model-free algorithms have been proposed for stationary RL, existing solutions for non-stationary RL are often built upon model-based methods. Nevertheless, it has been observed that model-based solutions often suffer from the following shortcomings:

- **Time- and space-inefficiency:** Model-based methods are in general more time- and space-consuming, and are less compatible with the design of modern deep RL architectures (Jin et al., 2018; Zhang et al., 2020).

- **Inefficient exploration:** In Cheung et al. (2020), an example was given to show that under non-stationarity, the estimated model can incorrectly indicate that transitioning between states is very unlikely. This suggests that model-based methods, which try to estimate the latent model, might suffer “The Perils of Drift” (Cheung et al., 2020).

- **Limited applicability:** In an important application of nonstationary RL — decentralized multi-agent RL, the agents cannot observe the actions taken by the other agents. This information structure precludes model-based methods, as the explicit estimation of the state transition functions is hardly possible without observing all the agents’ actions.

These observations have thus motivated us to turn our attention to model-free methods, which, instead of maintaining estimates of the unknown underlying model, directly learn the Q-values.

**Main Contributions.** In this paper, we focus on the problem of designing model-free algorithms with nearly-optimal performances for non-stationary RL. Our contributions can be summarized as follows:

1. We introduce an algorithm named Restart Q-Learning with Upper Confidence Bounds (RestartQ-UCB), which is the first model-free algorithm in the general setting of non-stationary RL. Our algorithm adopts a simple but effective restarting strategy (Jaksch et al., 2010; Besbes et al., 2014) that resets the memory of the agent according to a calculated schedule. The restarting strategy ensures that our algorithm only refers to the most up-to-date experience for decision-making. RestartQ-UCB also utilizes an extra optimism term (in addition to the standard Hoeffding/Freedman-based bonus) for exploration to counteract the non-stationarity of the MDP. This additional bonus term, depending on the local variations (i.e., the environmental variation in each restarting interval), guarantees that our optimistic Q-value is still an upper bound of the optimal Q*-value even when the environment changes. We further show that our algorithm can easily remove the dependence on local variations, an assumption commonly made in the literature (Ortner et al., 2019; Zhou et al., 2020). Our analysis shows that RestartQ-UCB achieves the lowest dynamic regret bound when compared to existing works in the literature;

2. We conduct simulations showing that RestartQ-UCB achieves highly competitive cumulative rewards against a state-of-the-art solution (Zhou et al., 2020), while only taking 0.18% of its computation time;

3. We establish the first lower bounds in non-stationary RL, which suggest that our algorithm is optimal in all parameter dependences except for an \( H^{\frac{1}{2}} \) factor, where \( H \) is the episode length;

4. To further showcase the flexibility and potential of non-stationary RL, we illustrate how it can be utilized to address the non-stationarity issue inherent in multi-agent RL. Specifically, we show that RestartQ-UCB can be readily applied to a multi-agent RL example against a slowly-changing opponent (Radanovic et al., 2019; Lee et al., 2014).

### Table 1: Dynamic regret comparisons for RL in non-stationary MDPs.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Algorithm</th>
<th>Regret</th>
<th>Model-free?</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Undiscounted</td>
<td>Jaksch et al. (2010)</td>
<td>( \tilde{O}(S A^\frac{1}{2} L^\frac{1}{2} D T^\frac{3}{2}) )</td>
<td>( \times )</td>
<td>only abrupt changes</td>
</tr>
<tr>
<td></td>
<td>Gajane et al. (2018)</td>
<td>( \tilde{O}(S^2 A^\frac{3}{2} L^\frac{1}{2} D^\frac{3}{2} T^\frac{3}{2}) )</td>
<td>( \times )</td>
<td>only abrupt changes</td>
</tr>
<tr>
<td></td>
<td>Ortner et al. (2019)</td>
<td>( \tilde{O}(S^2 A^\frac{1}{2} + L^\frac{1}{2} D^\frac{3}{2} T^\frac{3}{2}) )</td>
<td>( \times )</td>
<td>requires local variations</td>
</tr>
<tr>
<td></td>
<td>Cheung et al. (2020)</td>
<td>( \Omega(S^2 A^\frac{1}{2} D^\frac{3}{2} T^\frac{3}{2}) )</td>
<td>( \times )</td>
<td>does not require ( \Delta )</td>
</tr>
<tr>
<td>Lower bound</td>
<td>Domingues et al. (2020)</td>
<td>( \tilde{O}(S^2 A^\frac{1}{2} D^\frac{1}{2} H^\frac{1}{2} T^\frac{3}{2}) )</td>
<td>( \times )</td>
<td>also metric spaces</td>
</tr>
<tr>
<td></td>
<td>RestartQ-UCB</td>
<td>( \tilde{O}(S^2 A^\frac{1}{2} \Delta^\frac{3}{2} H^\frac{1}{2} T^\frac{3}{2}) )</td>
<td>( \checkmark )</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lower bound</td>
<td>( \Omega(S^2 A^\frac{1}{2} \Delta^\frac{3}{2} H^\frac{1}{2} T^\frac{3}{2}) )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Another related line of research studies online/adversarial when the state-action set forms a metric space. Their results (MAB) settings, various methods have been proposed, in-research on non-stationary RL. A more recent line of work based strategies (Auer et al., 2002; Besbes et al., 2014; Alle-

Non-stationarity has also been considered in bandit prob-

2. Preliminaries

Model: We consider an episodic RL setting where an agent interacts with a non-stationary MDP for \( M \) episodes, with each episode containing \( H \) steps. We use a pair of integers \((m, h)\) as a time index to denote the \( h \)-th step of the \( m \)-th episode. The environment can be denoted by a tuple \((S, A, H, P, r)\), where \( S \) is the finite set of states with \(|S| = S, A \) is the finite set of actions with \(|A| = A, H \) is the number of steps in one episode, \( P = \{P_m\}_{m \in [M], h \in [H]} \) is the set of transition kernels, and \( r = \{r_m\}_{m \in [M], h \in [H]} \) is the set of mean reward functions. Specifically, when the agent takes action \( a_h^m \in A \) in state \( s_h^m \in S \) at the time \((m, h)\), it will receive a random reward \( R_h^m(s_h^m, a_h^m) \in [0, 1] \) with expected value \( r_h^m(s_h^m, a_h^m) \), and the environment transitions to a next state \( s_{h+1}^m \) following the distribution \( P_h^m(\cdot | s_h^m, a_h^m) \). It is worth emphasizing that the transition kernel and the mean reward function depend both on \( m \) and \( h \), and hence the environment is non-stationary over time. The episode ends when \( s_{H+1}^m \) is reached. We further denote \( T = MH \) as the total number of steps.

A deterministic policy \( \pi : [M] \times [H] \times S \rightarrow A \) is a mapping from the time index and state space to the action space, and we let \( \pi_h^m(s) \) denote the action chosen in state \( s \) at time \((m, h)\). Define \( V_h^{m, \pi} : S \rightarrow \mathbb{R} \) to be the value function under policy \( \pi \) at time \((m, h)\), i.e.,

\[
V_h^{m, \pi}(s) = \mathbb{E}\left[ \sum_{h'=h}^H r_{h'}^m(s_{h'}^m, \pi_h^m(s_{h'}^m)) \mid s_h = s \right],
\]

where \( s_{h'+1}^m \sim P_{h'}^m(\cdot \mid s_{h'}^m, a_{h'}^m) \). Accordingly, the state-

Non-stationarity has also been considered in bandit prob-

Within different non-stationary multi-armed bandit (MAB) settings, various methods have been proposed, including decaying memory and sliding windows (Garivier & Moulines, 2011; Keskin & Zeevi, 2017), as well as restart-based strategies (Auer et al., 2002; Besbes et al., 2014; Alle-

These methods largely inspired later research on non-stationary RL. A more recent line of work
action value function \( Q^{m,\pi}_h : S \times A \rightarrow \mathbb{R} \) is defined as:

\[
Q^{m,\pi}_h(s, a) \overset{\text{def}}{=} r^{m}_h(s, a) + \mathbb{E}_h \left[ \sum_{t=1}^{H} r^{m}_t(s'_{h}, \pi^m_h(s'_{h})) \mid s_h = s, a_h = a \right]
\]

For simplicity of notation, we let \( P^{m}_{h} V_{h+1}(s, a) \overset{\text{def}}{=} \mathbb{E}_{s' \sim P^{m}_{h}}(s, a) [V_{h+1}(s')] \). Then, the Bellman equation gives \( V^{m,\pi}_h(s) = Q^{m,\pi}_h(s, \pi^m_h(s)) \) and \( Q^{m,\pi}_h(s, a) = (r^{m}_h + P^{m}_{h} V^{m,\pi}_{h+1})(s, a) \), and we also have \( V^{m,\pi}_{h+1}(s) = 0, \forall s \in S \) by definition. Since the state space, the action space, and the length of each episode are all finite, there always exists an optimal policy \( \pi^* \) that gives the optimal value \( V^{m,\pi^*}_h(s) \) defined \( V^{m,\pi^*}_h(s) = \sup_{\pi} V^{m,\pi}_h(s), \forall s \in S, m \in [M], h \in [H] \). From the Bellman optimality equation, we have \( V^{m,\pi^*}_h(s) = \max_{\alpha \in A} Q^{m,\pi^*}_h(s, a) \), where \( Q^{m,\pi^*}_h(s, a) \overset{\text{def}}{=} (r^{m}_h + P^{m}_{h} V^{m,\pi^*}_{h+1})(s, a) \), and \( V^{m,\pi^*}_{h+1}(s) = 0, \forall s \in S \).

**Dynamic Regret:** The agent aims to maximize the cumulative expected reward over the entire \( M \) episodes, by adopting some policy \( \pi \). We measure the optimality of the policy \( \pi \) in terms of its dynamic regret (Cheung et al., 2020; Domingues et al., 2020), which compares the agent’s policy with the optimal policy of each individual episode in hindsight:

\[
\mathcal{R}(\pi, M) \overset{\text{def}}{=} \sum_{m=1}^{M} \left( V^{1,\pi^*}_m(s^m_1) - V^{1,\pi}_m(s^m_1) \right),
\]

where the initial state \( s^m_0 \) of each episode is chosen by an oblivious adversary (Zhang et al., 2020). Dynamic regret is a stronger measure than the standard (static) regret, which only considers the single policy that is optimal over all episodes combined.

**Variation:** We measure the non-stationarity of the MDP in terms of its variation in the mean reward function and transition kernels:

\[
\Delta_r \overset{\text{def}}{=} \sum_{m=1}^{M-1} \sum_{h=1}^{H} \sup_{s,a} |r^{m}_h(s, a) - r^{m+1}_h(s, a)|,
\]

\[
\Delta_p \overset{\text{def}}{=} \sum_{m=1}^{M-1} \sum_{h=1}^{H} \sup_{s,a} \| P^{m}_h(\cdot \mid s, a) - P^{m+1}_h(\cdot \mid s, a) \|_1,
\]

where \( \| \cdot \|_1 \) is the \( L^1 \)-norm. Note that our definition of variation only imposes restrictions on the summation of non-stationarity across two different episodes, and does not put any restriction on the difference between two consecutive steps in the same episode; that is, \( P^{m}_h(\cdot \mid s, a) \) and \( P^{m+1}_h(\cdot \mid s, a) \) are allowed to be arbitrarily different. We further let \( \Delta = \Delta_r + \Delta_p \), and assume \( \Delta > 0 \).

**3. Algorithm: RestartQ-UCB**

We present our algorithm Restarted Q-Learning with Hoeffding Upper Confidence Bounds (RestartQ-UCB Hoeffding) in Algorithm 1. Replacing the Hoeffding term with a Freedman-style one will lead to a tighter regret bound, but the analysis is more involved. For clarity of presentation, we defer the exposition of the Freedman-based algorithm to Appendix E.

RestartQ-UCB breaks the \( M \) episodes into \( D \) epochs, with each epoch containing \( K = \lceil \frac{M}{D} \rceil \) episodes (except for the last epoch which possibly has less than \( K \) episodes). The optimal value of \( D \) (and hence \( K \)) will be specified later in our analysis. RestartQ-UCB periodically restarts a Q-learning algorithm with UCB exploration at the beginning of each epoch, thereby addressing the non-stationarity of the environment. For each \( d \in [D] \), define \( \Delta_r(d) \) to be the local variation of the mean reward function within epoch \( d \). By definition, we have \( \sum_{d=1}^{D} \Delta_r(d) \leq \Delta_r \). Define the local variation of transitions \( \Delta_p(d) \) analogously.

Since our algorithm essentially invokes the same procedure for every epoch, in the following, we focus our analysis on what happens inside one epoch only (and without loss of generality, we focus on epoch 1, which contains episodes \( 1, 2, \ldots, K \)). At the end of our analysis, we will merge the results across all epochs.

For each triple \( (s, a, h) \in S \times A \times [H] \), we divide the visitations (within epoch 1) to the triple into multiple stages, where the length of the stages increases exponentially at a rate of \( (1 + \frac{1}{H}) \). Specifically, let \( \varepsilon_1 = H \), and \( \varepsilon_{i+1} = [(1 + \frac{1}{H})\varepsilon_i] \), \( i \geq 1 \) denote the lengths of the stages. Further, let the partial sums \( \mathcal{L} \overset{\text{def}}{=} \left( \sum_{j=1}^{i} \varepsilon_j \mid j = 1, 2, 3, \ldots \right) \) denote the set of the ending times of the stages. We remark that the stages are defined for each individual triple \( (s, a, h) \), and for different triples the starting and ending times of their stages do not necessarily align in time. Such a definition of stages is mostly motivated by the design of the learning rate \( \alpha_i = \frac{H + \varepsilon_{i-1}}{H + 1} \) in Jin et al. (2018). It ensures that only the last \( O(1/H) \) fraction of samples is given non-negligible weights when used to estimate the optimistic \( Q_h(s, a) \) values, while the first \( 1 - O(1/H) \) fraction is forgotten (Zhang et al., 2020). We set \( t \overset{\text{def}}{=} \log \left( \frac{\delta}{H} \right) \), where \( \delta \) is the failure probability.

Recall that the time index \( (k, h) \) represents the \( h \)-th step of the \( k \)-th episode. At each step \( (k, h) \), we take the optimal action with respect to the optimistic \( Q_h(s, a) \) value (Line 6 in Algorithm 1), which is designed as an optimistic estimate of the optimal \( Q^*_h(s, a) \) value of the corresponding episode. For each triple \( (s, a, h) \), we update the optimistic \( Q_h(s, a) \) value at the end of each stage, using samples only from this latest stage that is about to end (Line 12 in Algorithm 1). The optimism in \( Q_h(s, a) \) comes from two bonus terms \( b^k_h \).
Algorithm 1: RestartQ-UCB (Hoeffding)

for epoch $d \leftarrow 1$ to $H$

1. Initialize: $V_h(s) \leftarrow H - h + 1$, $Q_h(s,a) \leftarrow H - h + 1$, $N_h(s,a) \leftarrow 0$, $\hat{N}_h(s,a) \leftarrow 0$, $\bar{N}_h(s,a) \leftarrow 0$, for all $(s,a,h) \in \mathcal{S} \times \mathcal{A} \times [H]$;

2. for episode $k \leftarrow (d-1)K + 1$ to $\min\{dK, M\}$ do

3. observe $s_h^k$;

4. for step $h \leftarrow 1$ to $H$

5. Take action $a_h^k \leftarrow \arg \max_a Q_h(s_h^k,a_h^k)$, receive $R_h^k(s_h^k,a_h^k)$, and observe $s_{h+1}^k$;

6. $\hat{r}_h(s_h^k,a_h^k) \leftarrow r_h(s_h^k,a_h^k) + 1$, $\bar{r}_h(s_h^k,a_h^k) \leftarrow 0$, $\bar{v}_h(s_h^k,a_h^k) \leftarrow 0$, $\bar{v}_h(s_h^k,a_h^k) \leftarrow 0$, $\hat{N}_h(s_h^k,a_h^k) \leftarrow 0$, $\bar{N}_h(s_h^k,a_h^k) \leftarrow 0$;

7. $N_h(s_h^k,a_h^k) \leftarrow N_h(s_h^k,a_h^k) + 1$,

8. $\hat{N}_h(s_h^k,a_h^k) \leftarrow \hat{N}_h(s_h^k,a_h^k) + 1$;

9. $V_h(s_h^k,a_h^k) \leftarrow \max_a Q_h(s_h^k,a_h^k)$;

10. $\bar{N}_h(s_h^k,a_h^k) \leftarrow \bar{N}_h(s_h^k,a_h^k)$;

11. $\hat{r}_h(s_h^k,a_h^k) \leftarrow \hat{r}_h(s_h^k,a_h^k)$;

12. $\bar{v}_h(s_h^k,a_h^k) \leftarrow \bar{v}_h(s_h^k,a_h^k)$;

end

$H$ Reaching the end of the stage

$b_h^k \leftarrow \sqrt{\frac{\text{H}^2}{N_h(s_h^k,a_h^k)}} + \sqrt{\frac{1}{N_h(s_h^k,a_h^k)}}$;

$\Delta \leftarrow \Delta^{(d)} + H \Delta^{(d)}$;

$Q_h(s_h^k,a_h^k) \leftarrow \max_a Q_h(s_h^k,a_h^k)$;

13. $V_h(s_h^k,a_h^k) \leftarrow \max_a Q_h(s_h^k,a_h^k)$;

14. $\bar{N}_h(s_h^k,a_h^k) \leftarrow 0$, $\bar{v}_h(s_h^k,a_h^k) \leftarrow 0$, $\bar{v}_h(s_h^k,a_h^k) \leftarrow 0$;

end

and $b_\Delta$, where $b_h^k$ is a standard Hoeffding-based optimism that is commonly used in upper confidence bounds (Jin et al., 2018; Zhang et al., 2020), and $b_\Delta$ is the extra optimism that we need to take into account the non-stationarity of the environment. The definition of $b_\Delta$ requires knowledge of the local variation budget in each epoch, which is a rather strong assumption in practice. However, we can further show (later in Theorem 2) that if we simply replace Equation (1*) in Algorithm 1 with the following update rule:

$$Q_h(s_h^k,a_h^k) \leftarrow \min \left\{ Q_h(s_h^k,a_h^k), \frac{\hat{r}_h(s_h^k,a_h^k)}{N_h(s_h^k,a_h^k)} + \frac{\bar{v}_h(s_h^k,a_h^k)}{N_h(s_h^k,a_h^k)} + b_h^k, Q_h(s_h^k,a_h^k) \right\},$$  \hspace{1cm} (1)

then our algorithm can achieve the same regret bound without the assumption on the local variation budget.

4. Analysis

In this section, we present our main result—a dynamic regret analysis of the RestartQ-UCB algorithm. Our first result on RestartQ-UCB with Hoeffding-style bonus terms is summarized in the following theorem. Complete proofs of its supporting lemmas are given in Appendix B.

Theorem 1. (Hoeffding) For $T = \Omega(SA\Delta H^2)$, and for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the dynamic regret of RestartQ-UCB with Hoeffding bonuses (Algorithm 1) is bounded by $\mathcal{O}(S^2 A^2 \Delta^3 H^2 T^2)$, where $\mathcal{O}(\cdot)$ hides poly-logarithmic factors of $T$ and $1/\delta$.

Our proof relies on the following technical lemma, stating that for any triple $(s,a,h)$, the difference of their optimal $Q$-values at two different episodes $1 \leq k_1 < k_2 \leq K$ is bounded by the variation of this epoch.

Lemma 1. For any triple $(s,a,h)$ and any $1 \leq k_1 < k_2 \leq K$, it holds that $|Q_{h}^{k_1}(s,a) - Q_{h}^{k_2}(s,a)| \leq \Delta_{r}^{(1)} + H \Delta_{p}^{(1)}$.

Let $Q_{h}^{k}(s,a)$ denote the value of $Q_{h}(s,a)$ at the beginning of the $k$-th episode in Algorithm 1. The following lemma states that the optimistic $Q$-value $Q_{h}^{k}(s,a)$ is an upper bound of the optimal $Q$-value $Q_{h}^{k,*}(s,a)$ with high probability. Note that we only need to show that the event holds with probability $1 - \text{poly}(K,H)/\delta$, because we can replace $\delta$ with $\delta/\text{poly}(K,H)$ in the end to get the desired high probability bound without affecting the polynomial part of the regret bound.

Lemma 2. (Hoeffding) For $\delta \in (0, 1)$, with probability at least $1 - 2KH/\delta$, it holds that $Q_{h, \Delta}^{k}(s,a) \leq Q_{h, \Delta}^{k+1}(s,a) \leq Q_{h, \Delta}^{k}(s,a)$, $\mathbf{v}(s,a,h,k) \in S \times A \times [H] \times [K]$.

Building upon Lemmas 1 and 2, a complete proof of Theorem 1 is given in Appendix C. We remark that Algorithm 1 relies on the assumption that the local variations $b_\Delta$ are known a priori, which is a strong but commonly made assumption in the literature on non-stationary RL (Ortner et al., 2019; Zhou et al., 2020). To the best of our knowledge, existing restart-based solutions either crucially rely on this local variation assumption (Ortner et al., 2019), or suffer a severe regret degeneration after removing this assumption (Zhou et al., 2020). Interestingly, in the following theorem, we show that this assumption can be safely removed in our approach without affecting the regret bound. The only modification to the algorithm is to replace the $Q$-value update rule in Equation (1*) of Algorithm 1 with the new update...
rule in Equation (1).

**Theorem 2.** (Hoeffding, no local budgets) For $T = \Omega(SA\Delta^2H^2)$, and for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the dynamic regret of `RestartQ-UCB with Freedman bonuses (presented in Algorithm 2)` is upper bounded by $O(S^2 \Delta^2 \hat{H}^2 T^2 \delta)$, where $O(\cdot)$ hides poly-logarithmic factors.

To understand why this simple modification works, notice that in (*) we are adding exactly the same value $2b_\Delta$ to the upper confidence bounds of all $(s, a)$ pairs in the same epoch. Subtracting the same value from all optimistic $Q$-values simultaneously should not change the choice of actions in future steps. The only difference is that the new “optimistic” $Q^k_h(s, a)$ values would no longer be strictly upper bounds of the optimal $Q^{k,*}_h(s, a)$ anymore, but only an “upper bound” subject to some error term of the order $b_\Delta$.

**Remark 1.** The easy removal of the local budget assumption is non-trivial in the design of the algorithm, and to the best of our knowledge is absent in the non-stationary RL literature with restarts. In fact, it has been shown in a concurrent work (Zhou et al., 2020) that removing this assumption could lead to a much worse regret bound (cf. Corollary 2 and Corollary 3 therein).

Replacing the Hoeffding-based upper confidence bound with a Freedman-style one will lead to a tighter regret bound, summarized in Theorem 3 below. The proof of the theorem follows a similar procedure as in the proof of Theorem 1, and is given in Appendix F. It relies on a reference-advantage decomposition technique for variance reduction as coined in Zhang et al. (2020).

**Theorem 3.** (Freedman) For $T$ greater than some polynomial of $S, A, \Delta$ and $H$, and for any $\delta \in (0, 1)$, with probability at least $1 - \delta$, the dynamic regret of `RestartQ-UCB with Freedman bonuses (presented in Algorithm 2)` is upper bounded by $O(S^2 \Delta^2 \hat{H}^2 T^2 \delta)$, where $O(\cdot)$ hides poly-logarithmic factors.

### 5. Lower Bounds

In this section, we provide information-theoretical lower bounds of the dynamic regret to characterize the fundamental limits of any algorithm in non-stationary RL.

**Theorem 4.** For any algorithm, there exists an episodic non-stationary MDP such that the dynamic regret of the algorithm is at least $\Omega(S^2 \Delta^2 \hat{H}^2 T^2 \delta)$.

**Proof sketch.** The proof of our lower bound relies on the construction of a “hard instance” of non-stationary MDPs. The instance we construct is essentially an MDP with piece-wise constant dynamics on each segment of the horizon, and its dynamics experience an abrupt change at the beginning of each new segment. Specifically, we divide the horizon $T$ into $L$ segments, where each segment has $T_0 \triangleq \lfloor \frac{T}{L} \rfloor$ steps and contains $M_0 \triangleq \lfloor \frac{\hat{H}}{L} \rfloor$ episodes. Within each segment, the system dynamics of the MDP do not vary, and we construct the dynamics for each segment in a way such that the instance is a hard instance of stationary MDPs on its own. The MDP within each segment is essentially similar to the hard instances constructed in Osband & Van Roy (2016); Jin et al. (2018). Between two consecutive segments, the dynamics of the MDP change abruptly, and we let the dynamics vary in a way such that no information learned from previous interactions with the MDP can be used in the new segment. In this sense, the agent needs to learn a new hard stationary MDP in each segment. Finally, optimizing the value of $L$ and the variation magnitude between consecutive segments (subject to the constraints of the total variation budget) leads to our lower bound.

A useful side result of our proof is the following lower bound for non-stationary RL in the un-discounted setting, which is the same setting as studied in Gajane et al. (2018), Ortner et al. (2019) and Cheung et al. (2020).

**Proposition 1.** Consider a reinforcement learning problem in un-discounted non-stationary MDPs with horizon length $T$, total variation budget $\Delta$, and maximum MDP diameter $D$ (Cheung et al., 2020). For any learning algorithm, there exists a non-stationary MDP such that the dynamic regret of the algorithm is at least $\Omega(S^2 \Delta^2 \Delta^2 D^2 T^2 \delta)$.

---

1. The definition of segments is irrelevant to, and should not be confused with, the notion of epochs we previously defined.


6. Simulations

In this section, we empirically evaluate RestartQ-UCB on reinforcement learning tasks with various types of non-stationarity. We compare RestartQ-UCB with three baseline algorithms: LSVI-UCB-Restart (Zhou et al., 2020), Q-Learning UCB, and Epsilon-Greedy (Watkins, 1989). LSVI-UCB-Restart is a state-of-the-art non-stationary RL algorithm that combines optimistic least-squares value iteration with periodic restarts. Q-Learning UCB is simply our RestartQ-UCB algorithm with no restart. It is a Q-learning based algorithm that uses upper confidence bounds to guide the exploration. Epsilon-Greedy is a restart-based algorithm that uses an epsilon-greedy strategy for action selection.

We evaluate the cumulative rewards of the four algorithms on a variant of a reinforcement learning task named Bidirectional Diabolical Combination Lock (Agarwal et al., 2020; Misra et al., 2020). This task is designed to be particularly difficult for exploration. We introduce two types of non-stationarity to the task, namely abrupt variations and gradual variations. A detailed discussion on the task settings as well as the configuration of the hyper-parameters is deferred to Appendix I. The cumulative rewards of the four algorithms in the abruptly-changing and gradually-changing environments are shown in Figures 1(a) and 1(b), respectively. All results are averaged over 30 runs.

As we can see, RestartQ-UCB outperforms Q-Learning UCB and Epsilon-Greedy under both types of environment variations. For the abruptly-changing environment as an example, RestartQ-UCB achieves 1.36 and 2.52 times of the cumulative rewards of Q-Learning UCB and Epsilon-Greedy, respectively. This demonstrates the importance of both addressing the environment variations (using restarts) and actively exploring the environment (using UCB-based bonus terms) in non-stationary RL. LSVI-UCB-Restart nearly matches the performance of RestartQ-UCB, which is unsurprising because both of them use the restarting strategy and optimistic exploration. Nevertheless, LSVI-UCB-Restart requires a higher time and space complexity. It needs to store all the history information in one epoch and solve a regularized least-squares minimization problem at every time step. This is indeed evidenced by our simulation results (shown in Figure 1(c)) that RestartQ-UCB only takes 0.18% of the computation time of LSVI-UCB-Restart.

Remark 2. The heavy computation in LSVI-UCB-Restart mostly comes from the usage of a high-dimensional feature. In our simulations, we followed Example 2.1 in Jin et al. (2019) to convert a linear MDP algorithm to a tabular one, which results in a feature dimension of $d = S \times A$. This is essentially the most efficient feature encoding when no special structure is imposed on the tabular MDP. We believe that designing low-dimensional features for specific MDP instances can possibly reduce the computations for LSVI-UCB-Restart by a large amount, and is an interesting future direction for learning in linear MDPs per se.

7. Application to Multi-Agent RL

In this section, we discuss the application of our non-stationary RL method to multi-agent RL in episodic stochastic games (Shapley, 1953), which by nature leads to a non-stationary RL problem from one-agent’s perspective.

7.1. Problem Setup

In general, an $N$-player episodic stochastic game is defined by a tuple $(\mathcal{N}, H, S, \{A^i\}_{i=1}^N, \{r^i\}_{i=1}^N, P)$, where (1) $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of agents; (2) $H \in \mathbb{N}_+$ is the number of time steps in each episode; (3) $S$ is the finite state space; (4) $A^i$ is the finite action space for agent $i \in \mathcal{N}$; (5) $r^i_h : S \times A \rightarrow [0, 1]$ is the reward function at step $h \in [H]$ for agent $i \in \mathcal{N}$, where $A = \times_{i=1}^N A^i$; and (6) $P_h : S \times A \rightarrow \Delta(S)$ is the transition kernel at step $h \in [H]$, where the next state depends on the current state and the joint actions of all the agents. The game lasts for $M$ episodes, and we let $T = MH$ be the total number of time steps. At each time step $(m, h)$, the agents observe the state $s^m_h \in S$, and take actions $a^m_i \in A^i, i \in \mathcal{N}$ simultaneously. We let $a^m_i = (a^m_{1i}, \ldots, a^m_{Ni})$. Agent $i$ receives a reward with an expected value of $r^i_h(s^m_h, a^m_i)$, and the environment transitions to the next state $s^{m+1}_h \sim P_h(\cdot|s^m_h, a^m_i)$. For each agent $i$, a policy is a mapping from the time index and state space to (possibly a distribution over) the action space. We denote the set of policies for agent $i$ by $\Pi^i = \{\pi^i : [M] \times [H] \times S \rightarrow \Delta(A^i)\}$. The set of joint policies are denoted by $\Pi = \times_{i=1}^N \Pi^i$. Each agent seeks to find a policy that maximizes its own reward.

For notational convenience, we consider two-player games, i.e., $N = 2$. We consider the problem where we can control the policy of agent 1, while agent 2 is an opponent that is adapting its own policy in an unknown way. Achieving sublinear regret in the face of an arbitrarily changing opponent is known to be computationally hard (Radanovic et al., 2019). Therefore, existing works (Radanovic et al., 2019; Lee et al., 2020) often focus on a setting where the opponent is only “slowly changing” its policy over time. One such example is when the opponent is using a relatively stable learning algorithm. We also focus on the decentralized setting, where each agent cannot observe the actions and rewards of the other agent. This is generally considered

---

2This setting has been studied under various names in the literature, including individual learning (Leslie & Collins, 2005), decentralized learning (Arslan & Yüksel, 2016), online agnostic learning (Tian et al., 2020), and independent learning (Daskalakis et al., 2020). It is also related to the broader category of teams and games with decentralized information structure (Ho, 1980; Nayyar et al., 2013a,b).
to be a more practical multi-agent RL paradigm, and also more challenging than those that we will compare with in the literature (Radanovic et al., 2019; Lee et al., 2020).

A joint policy induces a probability measure on the sequence of states and joint actions. For a joint policy \( \pi = (\pi^1, \pi^2) \in \Pi \), and for each time step \( (m, h) \in [M] \times [H] \), state \( s \in S \), we define the state value function for agent 1 as follows:

\[
V^m_\pi(s) = E \left[ \sum_{h=m}^{H} r^1_{h'} \left( s_{h'}, \pi^1_{h'}(s_{h'}), \pi^2_{h'}(s_{h'}) \right) | s_h = s \right].
\]

For a joint policy \( (\pi^1, \pi^2) \), we again evaluate the optimality of agent 1’s policy \( \pi^1 \) in terms of its dynamic regret, which compares the agent's policy with the optimal policy of each individual episode in hindsight:

\[
\mathcal{R}^2(\pi^1, M) \defeq \sum_{m=1}^{M} \left( \sup_{\pi^2} V^m_1(\pi^1, \pi^2) - V^m_1(\pi^1, \pi^2)(s^m) \right).
\]

The initial state of each episode \( s^m_0 \) is again chosen by an oblivious adversary.

7.2. Regret Against a Slowly-Changing Opponent

We model the slowly-changing behavior of agent 2 by requiring it to have a low switching cost (Bai et al., 2019; Gao et al., 2021). This is a standard notion in the literature to measure the changing behavior of an RL algorithm. We consider the following definition of the (local) switching cost from Bai et al. (2019).

**Definition 1.** The switching cost between any pair of policies \( (\pi, \pi') \) is the number of \( (h, s) \) pairs on which \( \pi \) and \( \pi' \) act differently:

\[
n_{\text{switch}}(\pi, \pi') \defeq | \{(h, s) \in [H] \times S : \pi_h(s) \neq \pi'_h(s)\} |.
\]

For a policy trajectory \( (\pi^1, \ldots, \pi^M) \) across \( M \) episodes, its switching cost is defined as

\[
N_{\text{switch}} \defeq \sum_{m=1}^{M} n_{\text{switch}}(\pi^m, \pi^{m+1}).
\]

Bai et al. (2019) develops a learning algorithm that achieves a switching cost of \( O(SAH^3 \log T) \), while Zhang et al. (2020) improves the switching cost to \( O(SAH^2 \log T) \). For the sake of generality, we characterize the behavior of agent 2 by assuming that the switching cost of its policy trajectory is upper bounded by \( O(T^\beta) \) for some \( 0 < \beta < 1 \). Clearly, the two state-of-the-art RL algorithms mentioned above satisfy this upper bound. A direct application of RestartQ-UCB leads to the following result for agent 1:

**Theorem 5.** Suppose that the switching cost of agent 2 satisfies \( N_{\text{switch}} = O(T^\beta) \) for \( 0 < \beta < 1 \). Let agent 1 run the RestartQ-UCB (Hoeffding/Freudman) algorithm. For \( T \) large enough, the dynamic regret of agent 1 is upper bounded by \( O(T^{\beta+\gamma}) \).

7.3. Learning Team-Optimality

Theorem 5 can be readily applied to learning team-optimal policies in “smooth games”, which is the setting considered in Radanovic et al. (2019). This corresponds to the setting where a team of agents learn to collaborate. Before we present our results, a few definitions are in order.

**Definition 2.** A stochastic game is called a stochastic team (or simply a team) if there exists a reward function \( r_h : S \times A \rightarrow [0, 1] \) such that \( r_h = r_h, \forall i \in N, h \in [H] \).

**Definition 3.** A joint policy \( \pi^* = (\pi^1*, \pi^2*) \in \Pi \) is called team-optimal if

\[
V_h^{(\pi^1*, \pi^2*)}(s) = \sup_{\pi^1, \pi^2} V_h^{(\pi^1, \pi^2)}(s), \forall s \in S, h \in [H],
\]

where \( V_h^{(\pi^1, \pi^2)}(s) \defeq E \left[ \sum_{h'=h}^{H} r_{h'}(s_{h'}, \pi^1_{h'}(s_{h'}), \pi^2_{h'}(s_{h'})) | s_h = s \right] \) is the value function.

In a stochastic team, the agents share the same objective, and aim to maximize the team accumulated reward. Team
optimality is achieved when the joint policy of the agents induces the highest possible accumulated reward.

Since we cannot control the behavior of agent 2, its behavior might be sub-optimal and drive us away from team-optimality. To avoid such scenarios, we impose a structural assumption that allows us to quantify the distance from optimality. In particular, we assume that the team is \((\lambda, \mu)\)-smooth, following the definition in Radanovic et al. (2019).

**Definition 4.** (Adapted from Definition 1 in Radanovic et al. (2019)) A two-player stochastic team is \((\lambda, \mu)\)-smooth if there exists a pair of policies \((\pi^1, \pi^2)\) such that for every policy pair \((\pi^1, \pi^2)\) and every \(h \in [H], s \in S:\)

\[
V_h^{(\pi^1, \pi^2)}(s) \geq V_h(\pi^1, \pi^2)(s),
\]

\[
V_h(\pi^1, \pi^2)(s) \geq \lambda \cdot V_h^{(\pi^1, \pi^2)}(s) - \mu \cdot V(\pi^1, \pi^2)(s).
\]

The \((\lambda, \mu)\)-smoothness ensures that agent 2’s sub-optimal behavior only has a bounded negative impact on the joint value. Our definition of smoothness is adapted from Radanovic et al. (2019), where the infinite-horizon average-reward setting is considered. We adapt it to the finite-horizon case. This notion of smoothness is motivated by the definition of smooth games in Roughgarden (2009); Syrgkanis et al. (2015), as stated in Radanovic et al. (2019).

Applying our RestartQ-UCB algorithm would lead to the following theorem, which implies that the time-average return of the agents converges to a \(\frac{1}{T}\) factor of the team-optimal value as \(T\) grows. This is the same factor as has been achieved in Radanovic et al. (2019).

**Theorem 6.** Let \(\pi^2\) denote the policy of agent 2, and suppose that the switching cost of agent 2 satisfies \(N_{\text{switch}} = O(T^\beta)\) for \(0 < \beta < 1\). Assume that the team problem is \((\lambda, \mu)\)-smooth. Let agent 1 run the RestartQ-UCB algorithm, and let \(\pi^1\) denote its induced policy. For \(T\) large enough, the return of the algorithm is lower bounded by:

\[
\sum_{m=1}^{M} V_1(\pi^1, \pi^2)(s^m_1) \geq \frac{\lambda}{1+\mu} \left[ \sum_{m=1}^{M} V_1(\pi^1, \pi^2)(s^m_1) - \tilde{O}(T^{\frac{\beta+2}{1+\beta}}) \right].
\]

**Remark 3.** (Comparison with Radanovic et al. (2019) and Lee et al. (2020).) It might first appear to the reader that our regret guarantee is weaker than the bounds of \(O(\frac{T^{\max(1-\frac{2}{\beta}, \frac{1}{1+\beta})}}{1+\beta})\) and \(O(\frac{T^{\max(1-\frac{2}{\alpha}, \frac{1}{1+\beta})}}{\alpha})\) given in Radanovic et al. (2019) and Lee et al. (2020), respectively, where \(\alpha\) can be essentially translated\(^4\) to \(1 - \beta\). However, we would like to emphasize that our setting significantly generalizes the other two works and is inherently more challenging due to the following facts: First, we are considering a learning problem where the transition and reward functions are unknown; the other two works essentially consider planning with a known MDP model. Second, we are using the more challenging dynamic regret as a measure of optimality, while the other two use the static regret. Third, we study decentralized learning, where each agent cannot observe the actions and rewards of the other agent; the algorithms proposed in the other two works critically rely on the observation of the other agent’s policies.

**Remark 4.** (Significance of model-freeness.) Decentralized multi-agent RL is generally only possible with model-free approaches (see, e.g., Arslan & Yüksel (2016); Tian et al. (2020); Daskalakis et al. (2020)); model-based methods proceed by explicitly estimating the transition and reward functions, which crucially relies on observing the other agents’ actions. This further demonstrates the flexibility and significance of model-free methods, when one addresses the non-stationarity issues in multi-agent RL through the lens of non-stationary RL.

8. Concluding Remarks

In this paper, we have considered model-free reinforcement learning in non-stationary episodic MDPs. We have proposed an algorithm named RestartQ-UCB that adopts a simple restarting strategy. RestartQ-UCB with Freedman-type bonus terms achieves a dynamic regret of \(\tilde{O}(S^2 A^2 \Delta^2 HT^{-\frac{2}{3}})\), which nearly matches the information-theoretical lower bound \(\Omega(S^2 A^2 \Delta^2 HT^{-\frac{1}{2}})\). Numerical experiments have validated the advantages of RestartQ-UCB in terms of both cumulative rewards and computational efficiency. A multi-agent RL example has been considered as an application to illustrate the power of our method. An interesting future direction would be to close the \(\tilde{O}(H^{-\frac{1}{2}})\) factor gap between the upper and lower bounds that we have established for the non-stationary RL problem. It would also be interesting to explore if non-stationary RL can be helpful in other multi-agent RL scenarios.

Acknowledgements

We thank Zihan Zhang for helpful discussions. Research of W.M, K.Z, and T.B. was supported in part by the US Army Research Laboratory (ARL) Cooperative Agreement W911NF-17-2-0196, in part by the Office of Naval Research (ONR) MURI Grant N00014-16-1-2710, and in part by the Air Force Office of Scientific Research (AFOSR) Grant FA9550-19-1-0353.

References

ing, pp. 20–29, 2018.


