Towards General Function Approximation in Nonstationary Reinforcement Learning

Songtao Feng, Ming Yin, Ruiquan Huang,
Yu-Xiang Wang, Jing Yang, Yingbin Liang

***Abstract*—Function approximation has experienced significant success in the field of reinforcement learning (RL). Despite a handful of progress on developing theory for Nonstationary RL with function approximation under structural assumptions, existing work for nonstationary RL with general function approximation is still limited. In this work, we propose a UCB-type of algorithm LSVI-Nonstationary following the popular least-square-value-iteration (LSVI) framework. LSVI-Nonstationary features the restart mechanism and a new design of bonus term to handle nonstationarity, and performs no worse than the existing confidence-set based algorithm SW-OPEA in [1], which has been shown to outperform the existing algorithms for nonstationary linear and tabular MDPs in the small variation budget setting.**

I. Introduction

Reinforcement learning (RL) focuses on the problem of maximizing the cumulative reward through interactions with an unknown environment. RL has witnessed a great success in practical applications, including robotics [2, 3], games [4–6], and autonomous driving [7]. The unknown environment in RL is commonly modeled as a Markov decision process (MDP), where the set of states $\mathcal{S}$ describes all possible status of the environment. At a state $s \in \mathcal{S}$, an agent takes an action a from an action set $\mathcal{A}$ to interact with the environment, after which the environment transits to the next state $s' \in \mathcal{S}$ drawn from some unknown transition distributions, and then the agent receives an immediate reward. The interaction between the agent and the environment takes place episodically, where each episode consists of H steps. The notion called regret has been typically employed to measure the performance of RL algorithms, which measures how much worse an agent performs following its current policy in comparison to the optimal policy in hindsight. The goal of the agent is to strategically interact with the environment to balance the exploration and exploitation tradeoff to minimize the regret.

Most existing RL studies adopt a static MDP model, in which both the reward and the transition kernel are time-invariant across episodes. However, stationary environment is insufficient to model enormous sequential decision problems such as online advertisement auctions [8, 9], traffic management [10], health care operations [11], and inventory control [12]. In contrast, nonstationary RL takes variations in rewards and transitions into consideration and is able to characterize larger classes of problems of interest [13]. In general, it is impossible to design algorithms that achieve sublinear regret for MDPs with drastically changing rewards and transitions in the worst case [14]. Therefore, one fundamental issue in the theoretical study of nonstationary RL is to investigate the maximum nonstationarity an agent can tolerate to adapt to the nonstationary dynamics of an MDP in order to achieve sublinear regret.

Without additional assumptions on the structure of the MDP, there is a line of extensive studies on nonstationary tabular MDPs [14–27]. However, the performance of nonstationary tabular MDPs suffers from large state and action spaces, which limits its applicability in scenarios with exponentially large or continuous state spaces. Therefore, function approximation has become a prominent tool to cope with this challenge. Several works have developed RL algorithms for nonstationary MDPs under structural assumptions, such as state-action set forming a metric space [28], linear MDPs [29, 30], linear mixture MDPs [31]. Although the developed algorithms are much more efficient than the algorithms designed for tabular setting, these algorithms require strong structural assumptions on the function approximation (such as a well-designed feature extractor in linear MDPs), which severely restricts the range of situations where these approaches can be employed.

Towards the direction of nonstationary MDPs under general function approximation, [1] initiates the study, and proposes a confidence-set based algorithm SW-OPEA, which relies on an computational inefficient oracle to select the optimistic state-action value function within the confidence set. To mitigate the computational inefficiency, we propose a UCB-type algorithm LSVI-Nonstationary in this work and our contribution is summarized below.

Our proposed UCB-type algorithm LSVI-Nonstationary adopts LSVI with upper confidence bound to handle the exploration and exploitation tradeoff. In order to handle nonstationarity, our algorithm features the restart mechanism, and incorporating the local variation budget in the design of the bonus term to ensure the optimism of the learned state-action value function.

We use the Eluder dimension to measure the complexity of the state-action value function class $\mathcal{F}$ for nonstataionry MDPs. We then theoretically characterize the dynamic regret of the proposed UCB-type algorithm, which depends on the eluder dimension of function class $\mathcal{F}$. Our newly proposed UCB-type algorithm matches with the performance of SW-OPEA in terms of horizon H, average variation budget in transitions L_P and average variation budget in rewards L_r, while performing slightly worse in the number of states and actions $|\mathcal{S}|, |\mathcal{A}|$ under tabular MDPs and the same in the linear feature $\widetilde{d}$ in linear MDPs. Our result suggests the benefit of UCB-type algorithm over confidence-set based algorithm.

979-8-3503-8284-6/24/$31.00 ©2024 IEEE

Our main technical development for these approaches lies in the single step optimization error for the least-square optimization in our UCB-type algorithm. We do not take the distribution drift in transitions and rewards into consideration, which may lead to non-trivial estimation error. In our analysis, we explicitly capture a non-trivial term due to the nonstationarity of the environment. We show that by compensating such a term involving local variation budget into the standard term due to concentration, the difference between the least-square predictor and the one-step backup estimate $r_h^k + P_h^k V_{h+1}^k$ is still bounded.

A. Related Work

Static Regret of Nonstationary MDPs: Static regret in nonstationary MDPs has been considered extensively in the past [14–24]. Static regret has also been studied for nonstationary MDPs with function approximation. In particular, [30] extends LSVI-UCB [32] for stationary linear MDPs and characterizes the static regret for the weighted least squares value iteration method. [33] studies the nonstationary RL setting with general function approximation, where the static regret is captured through a more general notion called decision-estimation coefficient (DEC).

Dynamic Regret of Nonstationary MDPs: Many studies in the past have been focused on the metric of dynamic regret, which quantifies the performance difference between the learning policy and the optimal policy at each step. For nonstationary tabular MDPs, value-based approaches have been proposed in [25, 27], where they respectively propose a sliding window strategy and a restart mechanism to handle nonstationarity. Further, [26] adopts a different method based on policy optimization. For nonstationary MDPs with function approximation, [29] and [31] focus on linear function approximation and linear-mixture function approximation, respectively, and [28] considers a kernel-based approach for nonstationary MDPs when state-action set forms a metric space. Further, [34] proposes a unified approach to nonstationary MDPs that relies on an oracle algorithm with optimal regret for stationary MDPs to develop a useful algorithm for nonstationary MDPs.

Notation: For a positive integer N, we use $[N]$ to denote the set of positive integers $\{1, 2, \ldots, N\}$. For positive integers m, n, define $\{\cdot\}_{[m:n]} = \varnothing$ if $m > n$. Given a dataset $\mathcal{D} = \{(x_i, a_i, q_i)\}_{i=1}^{|\mathcal{D}|} \subseteq \mathcal{S} \times \mathcal{A} \times \mathbb{R}$, for a function $f : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$, define $\|f\|_{\mathcal{D}} = \left(\sum_{i=1}^{|\mathcal{D}|} (f(x_i, a_i) - q_i)^2\right)^{\frac{1}{2}}$. For a set of state-action pairs $\mathcal{Z} \subseteq \mathcal{S} \times \mathcal{A}$, for a function $f : \mathcal{S} \times \mathcal{A}$, define $\|f\|_{\mathcal{Z}} = \left(\sum_{(x,a)\in\mathcal{Z}} (f(x_i, a_i))^2\right)^{\frac{1}{2}}$. For a set of functions $\mathcal{F} \subseteq \{f : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}\}$, we define the width function of the state-action pair as $w(\mathcal{F}; x, a) = \sup_{f, f' \in \mathcal{F}} (f - f')(x, a)$.

II. Preliminaries

A. Nonstationary MDPs

Our setting can be formulated as a nonstationary finite-horizon episodic Markov decision process, captured by a tuple $(\mathcal{S}, \mathcal{A}, H, K, P, r, x_1)$. Here, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, H is the length of each episode, K is the total number of episodes, $P = \{P_h^k\}_{(k,h)\in[K]\times[H-1]}$ where $P_h^k : \mathcal{S} \times \mathcal{A} \mapsto \triangle(\mathcal{S})$ is the transition kernel at step h in the k-th episode, $r = \{r_h^k\}_{(k,h)\in[K]\times[H]}$ where $r_h^k : \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$ is the mean reward function at step h in the k-th episode, and x_1 is the fixed initial state.

The agent interacts with the nonstationary MDP sequentially. At the beginning of k-th episode, the agent chooses a policy $\pi^k = \{\pi_h^k\}_{h\in[H]}$ where $\pi_h^k : \mathcal{S} \mapsto \triangle(\mathcal{A})$. At step h, the agent observes the state x_h^k, takes an action following $a_h^k \sim \pi_h^k(\cdot|x_h^k)$, obtains a reward $\widetilde{r}_h^k$ (we also use r_h^k if there is no ambiguity) with mean $r_h^k(x_h^k, a_h^k)$, and the MDP evolves into the next state $x_{h+1}^k \sim P_h^k(x_h^k, a_h^k)$. The process ends after receiving the last reward r_H^k. We define the state and state-action value functions of policy $\pi = \{\pi_h\}_{h\in[H]}$ recursively via the following equation

$$Q_{h;(*,k)}^{\pi}(x, a) = r_h^k(x, a) + (P_h^k V_{h+1;(*,k)}^{\pi})(x, a),$$
$$V_{h;(*,k)}^{\pi}(x) = \langle Q_{h;(*,k)}^{\pi}(x, \cdot), \pi_h^k(\cdot|x)\rangle_{\mathcal{A}}, \;\; V_{H+1;(*,k)} = 0,$$

where P_h^k is the operator defined as $(\mathbb{P}_h^k f)(x, a) := \mathbb{E}\left[f(x')|x' \sim P_h^k(x'|x, a)\right]$ for any function $f : \mathcal{S} \mapsto \mathbb{R}$. Here $\langle\cdot, \cdot\rangle_{\mathcal{A}}$ denotes the inner product over action space $\mathcal{A}$ and the subscript $\mathcal{A}$ is omitted when appropriate.

The learning objective is to find the optimal policy via interactions with the environment to minimize the dynamic regret

$$\mathrm{D-Regret}(K) := \sum_{k=1}^{K} \left(V_{1;(*,k)}^{\pi^{(*,k)}} - V_{1;(*,k)}^{\pi^k}\right)(x_1),$$

which quantifies the performance difference between the learning policy and the benchmark policy $\{\pi^{(*,k)}\}_{k\in[K]}$ where $\pi^{(*,k)} = \arg\max_{\pi} V_{1;(*,k)}^{\pi}(x_1)$.

B. Function Approximation

Consider a function class $\mathcal{F} = \mathcal{F}_1 \times \mathcal{F}_2 \times \ldots \times \mathcal{F}_H$, where $\mathcal{F}_h \subseteq \{f : \mathcal{S} \times \mathcal{A} \mapsto [0, H - h + 1]\}$ is the candidate function class to approximate $Q_{h;(*,k)}^{\pi^{(*,k)}}$. For convenience, we set $f_{H+1} = 0$.

Assumption 1 (Realizability)**.** *$Q_{h;(*,k)}^{*} \in \mathcal{F}_h$ for all $(k, h) \in [K] \times [H]$.*

Realizability assumption requires that the optimal state-action value function in each episode is contained in the function class $\mathcal{F}$ with no approximation error, i.e., $(Q_{1;(*,k)}^{*}, \cdots, Q_{H;(*,k)}^{*}) \in \mathcal{F}$ for $k \in [K]$.

Given functions $f = (f_1, f_2, \cdots, f_H)$ where $f_h \in (\mathcal{S}\times\mathcal{A} \mapsto [0, H - h + 1])$, define

$$(\mathcal{T}_h^k f_{h+1})(x, a) := r_h^k(x, a) + (P_h^k f_{h+1})(x, a),$$
$$(P_h^k f_{h+1})(x, a) = \mathbb{E}_{x'\sim P_h^k(\cdot|x,a)}[\max_{a'\in\mathcal{A}} f_{h+1}(x', a')],$$

where $\mathcal{T}_h^k$ is the Bellman operator at step h in episode k. Note that the optimal state-action value function satisfies $Q_{h;(*,k)}^{*}(x, a) = (\mathcal{T}_h^k Q_{h+1;(*,k)}^{*})(x, a)$ for all valid x, a, h. Moreover, we define $\mathcal{T}_h^k \mathcal{F}_{h+1} = \{\mathcal{T}_h^k f_{h+1} : f_{h+1} \in \mathcal{F}_{h+1}\}$.

Assumption 2 (Completeness)**.** *$\mathcal{T}_h^k \mathcal{F}_{h+1} \subseteq \mathcal{F}_h$ for all $(k, h) \in [K] \times [H]$.*

For the completeness assumption, we require that after applying the Bellman operator $\mathcal{T}_h^k$ of any episode k to a function f_{h+1} in the function class $\mathcal{F}_{h+1}$ at step $h+1$, the resulting function lies in the function class $\mathcal{F}_h$ at previous step h.

C. Complexity Measures

In this section, we introduce two complexity measures for a class of functions. One is Eluder dimension and the other one is distributional Eluder dimension.

The definition of Eluder dimension was first proposed in [35], and is based on the ϵ-independence of points, as illustrated in the following definition.

Definition 1 (Eluder Dimension)**.** *Let $\epsilon \geq 0$ and $\mathcal{Z} = \{(x_i, a_i)\}_{i=1}^n \subseteq \mathcal{S} \times \mathcal{A}$ be a sequence of state-action pairs.*

- *A state-action pair $(x, a) \in \mathcal{S} \times \mathcal{A}$ is ϵ-dependent on $\mathcal{Z}$ with respect to $\mathcal{F}$ if any $f, f' \in \mathcal{F}$ satisfying $\|f - f'\|_{\mathcal{Z}} \leq \epsilon$ also satisfies $|f(x, a) - f'(x, a)| \leq \epsilon$.*
- *An (x, a) is ϵ-independent on $\mathcal{Z}$ with respect to $\mathcal{F}$ if (x, a) is not ϵ-dependent on $\mathcal{Z}$.*
- *The Eluder dimension $\dim_E(\mathcal{F}, \epsilon)$ of a function class $\mathcal{F}$ is the length of the longest sequence of elements in $\mathcal{S} \times \mathcal{A}$ such that, for some $\epsilon' \geq \epsilon$, every element is ϵ'-independent of its predecessors.*

It has been shown in [36] that $\dim_E(\mathcal{F}, \epsilon) \leq |\mathcal{S}||\mathcal{A}|$ for tabular MDPs, and $\dim_E(\mathcal{F}, \varepsilon) \leq \widetilde{O}(\widetilde{d})$ for linear MDPs where $\widetilde{d}$ is the feature dimension.[1]

III. UCB-type Algorithm

In this section, we propose our UCB-type algorithm LSVI-Nonstationary. Our proposed algorithm falls into the popuplar LSVI framework, which uses LSVI with upper confidence bound to handle exploration and exploitation tradeoff. While designing the bonus term is simple in static tabular and linear MDPs, it becomes difficult in nonstationary MDPs with general function approximation. Our algorithm features the restart mechanism and incorporate the local variation budget in the design of the bonus term to handle nonstationarity. Moreover, it alleviates the computational inefficiency in the confidence-set based approach to select the optimistic state-action value functions in each step altogether.

We begin with the bounded complexity assumption [37] on the function class $\mathcal{F}$ and the state-action pairs $\mathcal{S} \times \mathcal{A}$.

Assumption 3. *For any $\varepsilon > 0$, the following statements hold:*

- *There exists an ε-cover $\mathcal{C}(\mathcal{F}, \varepsilon) \subseteq \mathcal{F}$ with size $|\mathcal{C}(\mathcal{F}, \varepsilon)| \leq \mathcal{N}(\mathcal{F}, \varepsilon)$, such that for any $f \in \mathcal{F}$, there exists $f' \in \mathcal{C}(\mathcal{F}, \varepsilon)$ with $\|f - f'\|_\infty \leq \varepsilon$;*
- *There exists an ε-cover $\mathcal{C}(\mathcal{S}\times\mathcal{A}, \varepsilon)$ with size $\mathcal{C}(\mathcal{S}\times\mathcal{A}, \varepsilon) \leq \mathcal{N}(\mathcal{S}\times\mathcal{A}, \varepsilon)$, such that for any $(x, a) \in \mathcal{S}\times\mathcal{A}$, there exists $(x', a') \in \mathcal{C}(\mathcal{S}\times\mathcal{A}, \varepsilon)$ with $\sup_{f\in\mathcal{F}} |f(x, a) - f(x', a')| \leq \varepsilon$.*

[1]The proofs for the nonstationary setting are essentially the same as the proof for the stationary setting therein, and we do not differentiate the two settings.

This assumption essentially requires both the function class and the state-action pairs have bounded covering numbers. It is acceptable for the covers to have exponential size since the regret bound scales logarithmically on both $\mathcal{N}(\mathcal{F}, \cdot)$ and $\mathcal{N}(\mathcal{S}\times\mathcal{A}, \cdot)$. For the tabular case when $\mathcal{S}, \mathcal{A}$ are finite, $\log \mathcal{N}(\mathcal{F}, \varepsilon) = \widetilde{O}(|\mathcal{S}||\mathcal{A}|)$ and $\log \mathcal{N}(\mathcal{S} \times \mathcal{A}, \varepsilon) = \log(|\mathcal{S}||\mathcal{A}|)$. For the linear MDPs with feature dimension $\widetilde{d}$, $\log \mathcal{N}(\mathcal{F}, \varepsilon) = \widetilde{O}(\widetilde{d})$ and $\log \mathcal{N}(\mathcal{S} \times \mathcal{A}, \varepsilon) = \log(\widetilde{d})$.

A. Algorithm LSVI-Nonstationary

In this section, we describe our proposed UCB-type algorithm LSVI-Nonstationary for nonstationary MDPs with general function approximation.

From a high level point of view, our algorithm features two key ingredients: least-square value iteration (LSVI) and a restart mechanism. Our algorithm uses LSVI with upper confidence bound to handle the exploration and exploitation tradeoff, where we incorporate the local variation budget in the design of bonus term to ensures the optimism of the learned state-action value function. Moreover, we use the epoch restart mechanism to adapt to the nonstationarity of the environment. Those ingredients make our design significantly different from the $\mathcal{F}$-LSVI algorithm [35] for static MDPs.

Algorithm 1 LSVI-Nonstationary

Input: failure probability $\delta \in (0, 1)$, number of episodes K and epoch size W.
for $j = 1, 2, \ldots, \lceil \frac{K}{W} \rceil$ **do**
  $\tau \leftarrow (j-1)W + 1$.
  for episode $k = \tau, \ldots, \min\{\tau + W, K\}$ **do**
    Receive initial state $s_1^k \sim \mu$.
    $Q_{H+1}^k(\cdot, \cdot) \leftarrow 0$ and $V_{H+1}^k(\cdot) \leftarrow 0$.
    $\mathcal{Z}_h^k = \{(x_h^\ell, a_h^\ell)\}_{\ell \in [\tau : k-1]}$.
    for $h = H, H-1, \ldots, 1$ **do**
      $\mathcal{D}_h^k = \{(x_h^\ell, a_h^\ell, \widetilde{r}_h^\ell + V_{h+1}^k(x_{h+1}^\ell))\}_{\ell \in [\tau, k-1]}$.
      $f_h^k \leftarrow \arg\min_{f \in \mathcal{F}_h} \|f\|_{\mathcal{D}_h^k}^2$.
      $b_h^k \leftarrow \text{bonus}(\mathcal{F}_h, f_h^k, \mathcal{Z}_h^k, \delta, j)$ (Algorithm 3).
      $Q_h^k(\cdot, \cdot) \leftarrow \min\{f_h^k(\cdot, \cdot) + b_h^k(\cdot, \cdot), H\}$ and $V_h^k(\cdot) = \max_{a \in \mathcal{A}} Q_h^k(\cdot, a)$.
      $\pi_h^k(\cdot) \leftarrow \arg\max_{a \in \mathcal{A}} Q_h^k(\cdot, a)$.
    end for
    for $h = 1, 2, \ldots, H$ **do**
      Take action $a_h^k \leftarrow \widetilde{\pi}_h^k(x_h^k)$ and observe $x_{h+1}^k \sim P_h^k(\cdot | x_h^k, a_h^k)$ and $\widetilde{r}_h^k \sim r_h^k(x_h^k, a_h^k)$.
    end for
  end for
end for

The pseudocode of LSVI-Nonstationary is presented in Algorithm 1. Our algorithm runs in epochs with length W. Within each episode, we follow these steps: Firstly, we estimate the state-action value function through a least-square problem using historical data from the current epoch. Next, we create an upper confidence bound for the state-action value function and select the policy that maximizes this upper confidence bound. A new trajectory is then collected by following the

greedy policy. Finally, we periodically restart the algorithm to handle the nonstationarity of the environment.

Least-square value iteration. At the beginning of each episode k, we maintain a replay buffer of existing samples $\{x_h^\ell, a_h^\ell, r_h^\ell\}_{\ell\in[\tau:k-1]}$, where τ is the first episode of the epoch containing episode k. Let $Q_{H+1}^k = 0$, and we set $Q_H^k, Q_{H-1}^k, \ldots, Q_1^k$ iteratively as follows (line 10-12). For $h = H, H-1, \ldots, 1$,

$$f_h^k(\cdot,\cdot) = \arg\min_{f\in\mathcal{F}_h} \sum_{\ell=\tau}^{k-1} \left(f(x_h^\ell, a_h^\ell) - r_h^\ell - \max_a Q_{h+1}^k(x_{h+1}^\tau, a)\right)^2,$$

$$Q_h^k(\cdot,\cdot) = \min\{f_h^k(\cdot,\cdot) + b_h^k(\cdot,\cdot), H\},$$

where $b_h^k(\cdot,\cdot)$ is the bonus function to be defined shortly. After obtaining $Q_h^k(\cdot,\cdot)$, we then use the greedy policy with respect to Q_h^k to collect data (line 13) for the kth episode. Note that the least-square problem does not take into consideration the distribution drift in transitions and rewards, which may potentially result in significant estimation errors. However, our analysis shows that these estimation errors can adapt to the nonstationarity. Specifically, we incorporate such estimation errors into the design of the bonus term to ensure the state-action value estimate is an optimistic upper bound of the optimal state-action value function.

Stable upper-confidence bonus function. With more collected data, the least-square solution is expected to provide a better approximation to the optimal state-action value function. To encourage exploration, we add additional bonus function $b_h^k(\cdot,\cdot)$ to guarantee that with high probability, $Q_{h+1}^k(\cdot,\cdot)$ is an optimistic upper bound of the optimal state-action value function. The design of bonus term b_h^k has two features: First, we leverage the importance sampling technique [35] to prioritize important data in the replay buffer so that the bonus function b_h^k is stable even when the replay buffer has large cardinality. Second, the distribution drift of the transitions and the rewards is characterized in the design of bonus term b_h^k in order to obtain the optimistic upper bound of the optimal state-action value function.

We define bonus function to be the width function $b_h^k(\cdot,\cdot) = w(\mathcal{F}_h^k;\cdot,\cdot)$, where $\mathcal{F}_h^k$ is defined as the confidence set so that the estimation error of the one-step backup $(r_h^k + P_h^k V_{h+1}^k)(\cdot,\cdot)$ lies within $\mathcal{F}_h^k$ with high probability. By the definition of width function, $b_h^k(\cdot,\cdot)$ provides an upper bound on the confidence interval of the state-action value estimate, since the width function maximizes the difference between all pairs of state-action value functions within the confidence set. Specifically, we define the confidence set as $\mathcal{F}_h^k = \{f \in \mathcal{F}_h : \|f - f_h^k\|_{\mathcal{Z}_h^k}^2 \le \beta + H\Delta_h^k\}$ where β is the confidence parameter properly selected so that $(r_h^k + P_h^k V_{h+1}^k)(\cdot,\cdot) \in \mathcal{F}_h^k$ with high probability, $\mathcal{Z}_h^k$ consists of the collected samples (line 5), and Δ_h^k is the local variation budget defined by

$$\Delta_h^k = \sum_{\ell=\tau}^{k-1} \sup_{x,a} |(r_h^k - r_h^\ell)(x,a)| + H \sum_{\ell=\tau}^{k-1} \sup_{x,a} \left\|(P_h^k - P_h^\ell)(\cdot|x,a)\right\|_1.$$

Note that the complexity of a bonus function could be high as it is defined by the dataset $\mathcal{Z}_h^k$ whose size can be as large as W. We adopt importance sampling technique in [35] to reduce the size of the dataset. Moreover, the data samples in $\mathcal{Z}_h^k$ are collected from nonstationary environment, we include an additional term of local variation budget Δ_h^k in the definition of $\mathcal{F}_h^k$. Intuitively, the local variation budget Δ_h^k captures the discrepancy between current episode k and previous episodes in the current epoch. By incorporating a term involving Δ_h^k in the design of $\mathcal{F}_h^k$, we ensure the true action-value function of the kth episode lies within the confidence set $\mathcal{F}_h^k$ with high probability. The formal definition of bonus term b_h^k and the selection of β is deferred to Appendix A.

Restart mechanism. We use epoch restart mechanism to handle the nonstationary environment. Specifically, we restart every W episodes as illustrated in the outer loop of Algorithm 1 (line 4), and the estimate of the state-action value function are calculated only by the samples collected in the current epoch, independent of all previous epochs. Note that while in general the epoch length W can vary for different epochs, we consider a fixed length and the corresponding dynamic regret upper bound in this work.

Compared to the confidence-set based algorithm SW-OPEA, which relies on an computational inefficient oracle to select the optimistic state-action value function within the confidence set. Instead, our algorithm is based on the popular UCB-based approach, which simplified the algorithm design and can be potentially implemented computationally efficiently [35].

B. Theoretical Guarantees

In this section, we provide the theoretical guarantee for Algorithm 1, and defer proofs to Appendix B.

For clarity, assume K/W is an integer throughout this section. The variation budget of an epoch $w \in [1 : K/W]$ is defined as

$$\Delta_h^{(w)} = \sum_{\ell=w(W-1)+1}^{wW} \sup_{x,a} |(r_h^k - r_h^\ell)(x,a)| + H \sum_{\ell=w(W-1)+1}^{wW} \sup_{x,a} \left\|(P_h^k - P_h^\ell)(\cdot|x,a)\right\|_1.$$

The dynamic regret of LSVI-Nonstationary is characterized in the following theorem.

Theorem 1 (Dynamic regret of LSVI-Nonstationary)**.** *Under Assumption 1, Assumption 2 and Assumption 3, with probability at least $1-\delta$, LSVI-Nonstationary achieves a dynamic regret bound of*

$$\widetilde{O}\left(\frac{4H^2 K d_m}{W} + \frac{KH^2}{\sqrt{W}}\sqrt{\iota} + \sqrt{d_m HW}\sum_{h=1}^{H}\sum_{w=1}^{K/W}\sqrt{\Delta_h^{(w)}}\right)$$

where $d_m = \sup_h \dim_E(\mathcal{F}_h, 1/W)$ and

$$\iota \le c\cdot \sup_h \log^3\frac{T}{\delta}\cdot \dim_E^2(\mathcal{F}_h, \frac{\delta}{16W^2})\cdot \ln(\frac{1}{\delta}\mathcal{N}(\mathcal{F}_h, \frac{\delta}{576W})) \cdot \ln(\mathcal{N}(\mathcal{S}\times\mathcal{A}, \frac{1}{16\sqrt{W/\delta}})W/\delta).$$

for some absolute constant $c > 0$.

Note that the last term depends on the length of the restart epoch W, and the dynamic regret upper bound can be further

optimized by setting appropriate W. We define the average variation budget in transitions L_P and rewards L_r

$$L_P = \max_{h\in[H],t<k} \frac{\sum_{s=t}^{k-1} \sup_{x,a} \|(P_h^{s+1} - P_h^s)(\cdot|x,a)\|_1}{k-t},$$
$$L_r = \max_{h\in[H],t<k} \frac{\sum_{s=t}^{k-1} \sup_{x,a} |(r_h^{s+1} - r_h^s)(x,a)|}{k-t}.$$

Clearly, we have $L_P, L_r \leq 1$ and $\Delta_P^w(k,h) \leq L_P w^2$, $\Delta_R^w(k,h) \leq L_r w^2$. L_P, L_r can be viewed as the the greatest average variation of transition kernels and rewards across adjacent episodes over any period of episodes maximized over step $h \in [H]$. Then the following corollary characterizes the dynamic regret by optimizing the window size w based on L_P and L_r.

The following corollary characterize the dynamic regret by optimizing the restart epoch length W based on the average variation budget L for both nonstationary tabular and nonstationary linear MDPs.

Corollary 1. *Consider the same condition as in Theorem 1. For tabular MDPs with $\widetilde{d} = |\mathcal{S}||\mathcal{A}|$, let $\mathcal{F}_h$ be the entire function space of $\mathcal{S} \times \mathcal{A} \mapsto [0, H-h+1]$ for $h \in [H]$. Since $\mathcal{S}, \mathcal{A}$ are finite, for $\varepsilon > 0$, we have $\dim_E(\mathcal{F}_h, \varepsilon) \leq \widetilde{d}$, $\log(\mathcal{N}(\mathcal{F}, \varepsilon)) = \widetilde{O}(\widetilde{d})$, and $\log(\mathcal{N}(\mathcal{S}\times\mathcal{A}, \varepsilon)) = O(\log(\widetilde{d}))$, and the dynamic regret is bounded by*

$$\widetilde{O}\left(H^{\frac{3}{2}} T^{\frac{1}{2}} \widetilde{d}^{\frac{3}{2}} + HT\widetilde{d} L_P^{\frac{1}{4}} + H^{\frac{3}{4}} T \widetilde{d} L_r^{\frac{1}{4}}\right).$$

For linear MDPs with feature dimension $\widetilde{d}$, $\dim_E(\mathcal{F}_h, \varepsilon) \leq \widetilde{O}(\widetilde{d})$, $\log(\mathcal{N}(\mathcal{F}, \varepsilon)) = \widetilde{O}(\widetilde{d})$, and $\log(\mathcal{N}(\mathcal{S}\times\mathcal{A}, \varepsilon)) = \widetilde{O}(\widetilde{d})$, and the dynamic regret is bounded by

$$\widetilde{O}\left(H^{\frac{3}{2}} T^{\frac{1}{2}} \widetilde{d}^{\frac{3}{2}} + HT\widetilde{d}^{\frac{5}{4}} L_P^{\frac{1}{4}} + H^{\frac{3}{4}} T \widetilde{d}^{\frac{5}{4}} L_r^{\frac{1}{4}}\right).$$

Compare to SW-OPEA: Under nonstationary MDPs with general function approximation, we compare the dynamic regret upper bound of our UCB-type algorithm to the dynamic regret bound of the confidence-set based algorithm SW-OPEA in [1]. For both nonstationary tabular and linear MDPs, SW-OPEA gives $\widetilde{O}\left(H^{\frac{3}{2}} T^{\frac{1}{2}} \widetilde{d} + HT\widetilde{d}^{\frac{3}{4}} L_P^{\frac{1}{4}} + H^{\frac{3}{4}} T \widetilde{d}^{\frac{3}{4}} L_r^{\frac{1}{4}}\right)$, where $\widetilde{d}$ equals $|\mathcal{S}||\mathcal{A}|$ for tabular MDPs and equals the feature dimension for linear MDPs. We see that the dynamic regret bound of UCB-type algorithm matches that of confidence-set based algorithm in horizon H as well as average variation budgets L_P and L_r while perform slightly worse in terms of $\widetilde{d}$. Similarly to the static MDPs [35], a more refined analysis specialized to the tabular and linear setting can potentially improve the dynamic regret bound. We would like to point out that our algorithm and analysis handles the nonstationary MDPs with general function approximation, which is much harder than and contains the nonstationary tabular and linear MDPs.

IV. Conclusion

In this paper, we propose a UCB-type algorithm LSVI-Nonstationary for nontationary MDPs with general function approximation, which follows the popular LSVI framework. To handle nonstationarity, LSVI-Nonstationary features the restart mechanism, and the novel design of the bonus term to ensure the optimism of the learned state-action value function. LSVI-Nonstationary performs no worse than the existing confidence-set based algorithm SW-OPEA in [1], while considerably simplifies the algorithm design and alleviate its computational inefficiency.

References

[1] S. Feng, M. Yin, R. Huang, Y.-X. Wang, J. Yang, and Y. Liang, "Non-stationary reinforcement learning under general function approximation," in *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[2] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[3] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *2017 IEEE international conference on robotics and automation (ICRA)*, 2017.

[4] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[5] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel *et al.*, "Mastering chess and shogi by self-play with a general reinforcement learning algorithm," *ArXiv*, 2017.

[6] ——, "A general reinforcement learning algorithm that masters chess, shogi, and go through self-play," *Science*, vol. 362, no. 6419, pp. 1140–1144, 2018.

[7] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58 443–58 469, 2019.

[8] H. Cai, K. Ren, W. Zhang, K. Malialis, J. Wang, Y. Yu, and D. Guo, "Real-time bidding by reinforcement learning in display advertising," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining (WSDM)*, 2017.

[9] J. Lu, C. Yang, X. Gao, L. Wang, C. Li, and G. Chen, "Reinforcement learning with sequential information clustering in real-time bidding," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019.

[10] C. Chen, H. Wei, N. Xu, G. Zheng, M. Yang, Y. Xiong, K. Xu, and Zhenhui, "Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control," in *AAAI Conference on Artificial Intelligence*, 2020.

[11] S. M. Shortreed, E. B. Laber, D. J. Lizotte, T. S. Stroup, J. Pineau, and S. A. Murphy, "Informing sequential

clinical decision-making through reinforcement learning: an empirical study," *Machine Learning*, vol. 84, pp. 109–136, 2010.

[12] S. Agrawal and R. Jia, "Learning in structured mdps with convex cost functions: Improved regret bounds for inventory management," in *Proceedings of the 2019 ACM Conference on Economics and Computation*, 2019.

[13] T. Cover and S. Pombra, "Gaussian feedback capacity," *IEEE Transactions on Information Theory*, vol. 35, no. 1, pp. 37–43, 1989.

[14] J. Y. Yu, S. Mannor, and N. Shimkin, "Markov decision processes with arbitrary reward processes," *Mathematics of Operations Research*, vol. 34, no. 3, pp. 737–757, 2009.

[15] P. Auer, T. Jaksch, and R. Ortner, "Near-optimal regret bounds for reinforcement learning," in *Advances in Neural Information Processing Systems*, 2008.

[16] P. Gajane, R. Ortner, and P. Auer, "A sliding-window algorithm for markov decision processes with arbitrarily changing rewards and transitions," *ArXiv*, 2018.

[17] E. Even-Dar, S. M. Kakade, and Y. Mansour, "Online markov decision processes," *Mathematics of Operations Research*, vol. 34, no. 3, pp. 726–736, 2009.

[18] J. Y. Yu and S. Mannor, "Arbitrarily modulated markov decision processes," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 2009.

[19] G. Neu, A. György, and C. Szepesvari, "The online loop-free stochastic shortest-path problem," in *Annual Conference Computational Learning Theory*, 2010.

[20] G. Neu, A. Gyorgy, and C. Szepesvari, "The adversarial stochastic shortest path problem with unknown transition probabilities," in *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, 2012.

[21] A. Zimin and G. Neu, "Online learning in episodic markovian decision processes by relative entropy policy search," in *Advances in Neural Information Processing Systems*, 2013.

[22] O. Dekel and E. Hazan, "Better rates for any adversarial deterministic MDP," in *Proceedings of the 30th International Conference on Machine Learning*, 2013.

[23] A. Rosenberg and Y. Mansour, "Online convex optimization in adversarial markov decision processes," in *International Conference on Machine Learning*, 2019.

[24] C. Jin, T. Jin, H. Luo, S. Sra, and T. Yu, "Learning adversarial markov decision processes with bandit feedback and unknown transition," in *International Conference on Machine Learning*, 2020.

[25] W. C. Cheung, D. Simchi-Levi, and R. Zhu, "Reinforcement learning for non-stationary Markov decision processes: The blessing of (More) optimism," in *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[26] Y. Fei, Z. Yang, Z. Wang, and Q. Xie, "Dynamic regret of policy optimization in non-stationary environments," in *Advances in Neural Information Processing Systems*, 2020.

[27] W. Mao, K. Zhang, R. Zhu, D. Simchi-Levi, and T. Basar, "Near-optimal model-free reinforcement learning in non-stationary episodic MDPs," in *Proceedings of the 38th International Conference on Machine Learning*, 2021.

[28] O. D. Domingues, P. M'enard, M. Pirotta, E. Kaufmann, and M. Valko, "A kernel-based approach to non-stationary reinforcement learning in metric spaces," in *International Conference on Artificial Intelligence and Statistics*, 2020.

[29] H. Zhou, J. Chen, L. R. Varshney, and A. Jagmohan, "Nonstationary reinforcement learning with linear function approximation," *Transactions on Machine Learning Research*, 2022.

[30] A. Touati and P. Vincent, "Efficient learning in non-stationary linear markov decision processes," *ArXiv*, 2020.

[31] H. Zhong, Z. Yang, Z. Wang, and C. Szepesvári, "Optimistic policy optimization is provably efficient in non-stationary MDPs," *ArXiv*, 2021.

[32] C. Jin, Z. Yang, Z. Wang, and M. I. Jordan, "Provably efficient reinforcement learning with linear function approximation," in *Proceedings of Thirty Third Conference on Learning Theory*, ser. Proceedings of Machine Learning Research. PMLR, 09–12 Jul 2020.

[33] D. J. Foster, A. Rakhlin, A. Sekhari, and K. Sridharan, "On the Complexity of Adversarial Decision Making," *ArXiv*, 2022.

[34] C.-Y. Wei and H. Luo, "Non-stationary reinforcement learning without prior knowledge: An optimal black-box approach," *ArXiv*, 2021.

[35] R. Wang, R. R. Salakhutdinov, and L. Yang, "Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension," in *Advances in Neural Information Processing Systems*, 2020.

[36] D. Russo and B. Van Roy, "Eluder dimension and the sample complexity of optimistic exploration," in *Advances in Neural Information Processing Systems*, 2013.

[37] R. Wang, R. R. Salakhutdinov, and L. Yang, "Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension," in *Advances in Neural Information Processing Systems*, 2020.

[38] M. Langberg and L. J. Schulman, "Universal - approximators for integrals," in *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, 2010.

[39] D. Feldman and M. Langberg, "A unified framework for approximating and clustering data," in *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, 2011.

[40] D. Feldman, M. Schmidt, and C. Sohler, "Turning big data into tiny data: Constant-size coresets for k-means, pca and projective clustering," in *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2013.