Randomized Exploration for Reinforcement Learning with General Value Function Approximation

Haque Ishfaq [*] [1] [2] Qiwen Cui [*] [3] Viet Nguyen [1] [2] Alex Ayoub [4] Zhuoran Yang [5] Zhaoran Wang [6]
Doina Precup [1] [2] [7] Lin F. Yang [8]

Abstract

We propose a model-free reinforcement learning algorithm inspired by the popular randomized least squares value iteration (RLSVI) algorithm as well as the optimism principle. Unlike existing upper-confidence-bound (UCB) based approaches, which are often computationally intractable, our algorithm drives exploration by simply perturbing the training data with judiciously chosen i.i.d. scalar noises. To attain optimistic value function estimation without resorting to a UCB-style bonus, we introduce an optimistic reward sampling procedure. When the value functions can be represented by a function class $\mathcal{F}$, our algorithm achieves a worst-case regret bound of $\widetilde{O}(\mathrm{poly}(d_E H)\sqrt{T})$ where T is the time elapsed, H is the planning horizon and d_E is the *eluder dimension* of $\mathcal{F}$. In the linear setting, our algorithm reduces to LSVI-PHE, a variant of RLSVI, that enjoys an $\widetilde{O}(\sqrt{d^3 H^3 T})$ regret. We complement the theory with an empirical evaluation across known difficult exploration tasks.

1. Introduction

The exploration-exploitation trade-off is a core problem in reinforcement learning (RL): an agent may need to sacrifice short-term rewards to achieve better long-term returns. A good RL algorithm should explore efficiently and find a near-optimal policy as quickly and robustly as possible. A big open problem is the design of provably efficient exploration when general function approximation is used to estimate the value function, i.e., the expectation of long-term return. In this work, we propose an exploration strategy inspired by the popular Randomized Least Squares Value Iteration (RLSVI) algorithm (Osband et al., 2016b; Russo, 2019; Zanette et al., 2020a) as well as by the optimism principle (Brafman & Tennenholtz, 2001; Jaksch et al., 2010; Jin et al., 2018; 2020; Wang et al., 2020), which is efficient in both statistical and computational sense, and can be easily plugged into common RL algorithms, including UCB-VI (Azar et al., 2017), UCB-Q (Jin et al., 2018) and OPPO (Cai et al., 2019).

The main exploration idea is the well-known "optimism in the face of uncertainty (OFU)" principle, which leads to numerous upper confidence bound (UCB)-type algorithms. These algorithms compute statistical confidence regions for the model or the value function, given the observed history, and perform the greedy policy with respect to these regions, or upper confidence bounds. However, it is costly or even intractable to compute the upper confidence bound explicitly, especially for structured MDPs or general function approximations. For instance, in Wang et al. (2020), computing the confidence bonus requires sophisticated sensitivity sampling and a width function oracle. The computational cost hinders the practical application of these UCB-type algorithms.

Another recently rediscovered exploration idea is Thompson sampling (TS) (Thompson, 1933; Osband et al., 2013). It is motivated by the Bayesian perspective on RL, in which we have a prior distribution over the model or the value function; then we draw a sample from this distribution and compute a policy based on this sample. Theoretical guarantees exist for both Bayesian regret (Russo & Van Roy, 2013) and worst-case regret (Agrawal & Jia, 2017) for this approach. Although TS is conceptually simple, in many cases the posterior is intractable to compute and the prior may not exist at all. Recently, approximate TS, also known as randomized least squares value iteration (RLSVI) or following the perturbed leader (Kveton et al., 2019), has received significant attention due to its good empirical performance. It has been proven that RLSVI enjoys sublinear worst-case or frequentist regret in tabular RL, by simply adding Gaus-

[*] Equal contribution [1] Mila [2] School of Computer Science, McGill University [3] School of Mathematical Science, Peking University [4] Amii and Department of Computing Science, University of Alberta [5] Department of Operations Research and Financial Engineering, Princeton University [6] Industrial Engineering Management Sciences, Northwestern University [7] DeepMind, Montreal [8] Department of Electrical and Computer Engineering, University of California, Los Angeles. Correspondence to: Haque Ishfaq <haque.ishfaq@mail.mcgill.ca>.

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

sian noise on the reward (Russo, 2019; Agrawal et al., 2020) However, in the improved bound for tabular MDP (Agrawal et al., 2020) and linear MDP (Zanette et al., 2020a), the uncertainty of the estimates still needs to be computed in order to perform optimistic exploration; it is unknown whether this can be removed. Moreover, this computation is difficult to do in the general function approximation setting.

In this work, we propose a novel exploration idea called optimistic reward sampling, which combines OFU and TS organically. The algorithm is surprisingly simple: we perturb the reward several times and act greedily with respect to the maximum of the estimated state-action values. The intuition is that after the perturbation, the estimate has a constant probability of being optimistic, and sampling multiple times guarantees that the maximum of these sampled estimates is optimistic with high probability. Thus, our algorithm utilizes approximate TS to achieve optimism.

Similar algorithms have been shown to work empirically, including SUNRISE (Lee et al., 2020), NoisyNet (Fortunato et al., 2017) and bootstrapped DQN (Osband et al., 2016a). However, the theoretical analysis of perturbation-based exploration is still missing. We prove that it enjoys near optimal regret $\widetilde{O}(\sqrt{H^3d^3T})$ for linear MDP and the sampling time is only $M = \widetilde{O}(d)$. We also prove similar bounds for the general function approximation case, by using the notion of eluder dimension (Russo & Van Roy, 2013; Wang et al., 2020). In addition, this algorithm is computationally efficient, as we no longer need to compute the upper confidence bound. In the experiments, we find that a small sampling time M is sufficient to achieve good performance, which suggests that the theoretical choice of $M = \widetilde{O}(d)$ is too conservative in practice.

Optimistic reward sampling can be directly plugged into most RL algorithms, improving the sample complexity without harming the computational cost. The algorithm only needs to perform perturbed regression. To our best knowledge, this is the first online RL algorithm that is both computationally and statistically efficient with linear function approximation and general function approximation. We hope optimistic reward sampling can be a large step towards bridging the gap between algorithms with strong theoretical guarantees and those with good computational performance.

2. Preliminaries

We begin by introducing some necessary notations. For any positive integer n, we denote the set $\{1, 2, \dots, n\}$ by $[n]$. For any set A, $\langle\cdot,\cdot\rangle_A$ denotes the inner product over set A. For a positive definite matrix $A \in \mathbb{R}^{d\times d}$ and a vector $x \in \mathbb{R}^d$, we denote the norm of x with respect to matrix A by $\|x\|_A = \sqrt{x^T A x}$. We denote the cumulative distribution function of the standard Gaussian by $\Phi(\cdot)$. For function growth, we use $\widetilde{O}(\cdot)$, ignoring poly-logarithmic factors.

We consider episodic MDPs of the form $(\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r)$, where $\mathcal{S}$ is the (possibly uncountable) state space, $\mathcal{A}$ is the (possibly uncountable) action space, H is the number of steps in each episode, $\mathbb{P} = \{\mathbb{P}_h\}_{h=1}^H$ are the state transition probability distributions, and $r = \{r_h\}_{h=1}^H$ are the reward functions. For each $h \in [H]$, $\mathbb{P}_h(\cdot \mid s, a)$ is the transition kernel over the next states if action a is taken at state s during the h-th time step of the episode. Also, $r_h : \mathcal{S}\times\mathcal{A} \to [0, 1]$ is the deterministic reward function at step h.[1]

A policy π is a collection of H functions $\{\pi_h : \mathcal{S} \to \Delta(\mathcal{A})\}_{h\in[H]}$ where $\Delta(\mathcal{A})$ denotes probability simplex over action space $\mathcal{A}$. We denote by $\pi(\cdot \mid s)$ the action distribution of policy π for state s, and by π^* the optimal policy, which maximizes the value function defined below.

The value function $V_h^\pi : \mathcal{S} \to \mathbb{R}$ at step $h \in [H]$ is the expected sum of remaining rewards until the end of the episode, received under π when starting from $s_h = s$,

$$V_h^\pi(s) = \mathbb{E}_\pi\Big[\sum_{h'=h}^{H} r_{h'}(s_{h'}, a_{h'}) \,|\, s_h = s\Big].$$

The action-value function $Q_h^\pi : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$ is defined as the expected sum of rewards given the current state and action when the agent follows policy π afterwards,

$$Q_h^\pi(s, a) = \mathbb{E}_\pi\Big[\sum_{h'=h}^{H} r_{h'}(s_{h'}, a_{h'}) \,|\, s_h = s, a_h = a\Big].$$

We denote $V_h^*(s) = V_h^{\pi^*}(s)$ and $Q_h^*(s, a) = Q_h^{\pi^*}(s, a)$. Moreover, to simplify notation, we denote $[\mathbb{P}_h V_{h+1}](s, a) = \mathbb{E}_{s'\sim\mathbb{P}_h(\cdot\,|\,s,a)} V_{h+1}(s')$.

Recall that value functions obey the Bellman equations:

$$\begin{aligned} Q_h^\pi(s, a) &= (r_h + \mathbb{P}_h V_{h+1}^\pi)(s, a), \\ V_h^\pi(s) &= \langle Q_h^\pi(s, \cdot), \pi_h(\cdot \mid s)\rangle_{\mathcal{A}}, \\ V_{H+1}^\pi(s) &= 0. \end{aligned} \tag{1}$$

The aim of the agent is to learn the optimal policy by acting in the environment for K episodes. Before starting each episode $k \in [K]$, the agent chooses a policy π^k and an adversary chooses the initial state s_1^k. Then, at each time step $h \in [H]$, the agent observes $s_h^k \in \mathcal{S}$, picks an action $a_h^k \in \mathcal{A}$, receives a reward $r_h(s_h^k, a_h^k)$ and the environment transitions to the next state $s_{h+1}^k \sim \mathbb{P}_h(\cdot \mid s_h^k, a_h^k)$. The episode ends after the agent collects the H-th reward and reaches the state s_{H+1}^k. The suboptimality of an agent can be measured by its regret, the cumulative difference of optimal

[1] We assume the reward function is deterministic for notational convenience. Our results can be straightforwardly generalized to the case when rewards are stochastic.

Algorithm 1 $\mathcal{F}$-LSVI-PHE

```
Set $M$ to be a fixed integer.
For episode $k = 1, 2, \dots, K$ do
  Receive the initial state $s_1^k$.
  Set $V_{H+1}^k(s) = 0$ for all $s \in \mathcal{S}$.
  For step $h = H, H-1, \dots, 1$ do
    For $m = 1, 2, \dots, M$ do
      Sample i.i.d. Gaussian noise $\xi_{h,k}^{\tau,m} \sim N(0, \sigma_{h,k}^2)$.
      Perturbed dataset: $\widetilde{\mathcal{D}}_h^{k,m} \leftarrow \{(s_h^\tau, a_h^\tau, r_h^\tau + \xi_{h,k}^{\tau,m}$
      $+V_{h+1}^k(s_{h+1}^\tau))\}_{\tau \in [k-1]}$.
      Set $\widetilde{f}_h^{k,m} \leftarrow \arg\min_{f \in \mathcal{F}} L(f \mid \widetilde{\mathcal{D}}_h^{k,m})^2 + \lambda \widetilde{R}(f)$.
      Set $Q_h^{k,m}(\cdot,\cdot) \leftarrow \widetilde{f}_h^{k,m}(\cdot,\cdot)$.
    Set $Q_h^k(\cdot,\cdot) \leftarrow \min\{\max_{m \in [M]}\{Q_h^{k,m}(\cdot,\cdot)\}$,
      $H - h + 1\}$.
    Set $V_h^k(\cdot) \leftarrow \max_{a \in \mathcal{A}} Q_h^k(\cdot, a)$ and
      $\pi_h^k(\cdot) \leftarrow \arg\max_{a \in \mathcal{A}} Q_h^k(\cdot, a)$.
  For step $h = 1, 2, \dots, H$ do
    Take action $a_h^k \leftarrow \arg\max_{a \in \mathcal{A}} Q_h^k(s_h^k, a)$.
    Observe reward $r_h^k(s_h^k, a_h^k)$, get next state $s_{h+1}^k$.
```

and achieved return, which after K episodes is

$$\text{Regret}(K) = \sum_{k=1}^{K} \Big[V_1^*(s_1^k) - V_1^{\pi^k}(s_1^k) \Big]. \tag{2}$$

Additional notations. The performance of function f on dataset $\mathcal{D} = \{(x_t, y_t)\}_{t \in [|\mathcal{D}|]}$ is defined by $L(f \mid \mathcal{D}) = \Big(\sum_{t=1}^{|\mathcal{D}|} (f(x_t) - y_t)^2\Big)^{1/2}$. The empirical ℓ_2 norm of function f on input set $\mathcal{Z} = \{x_t\}_{t \in [|\mathcal{Z}|]}$ is defined by $\|f\|_{\mathcal{Z}} = \Big(\sum_{t=1}^{|\mathcal{Z}|} f(x_t)^2\Big)^{1/2}$. Given a function class $\mathcal{F} \subseteq \{f : X \to \mathbb{R}\}$, we define the width function given some input x as $w(\mathcal{F}, x) = \max_{f, f' \in \mathcal{F}} f(x) - f'(x)$.

3. Algorithm: LSVI-PHE

In this section, we lay out our algorithm (Algorithm 1), an optimistic modification of RLSVI, where the optimism is realized by, what we will call, optimistic reward sampling. To describe our algorithm and facilitate its analysis in Section 4, we first define the perturbed least squares regression. We add noises on the regression target and the regularizer to achieve enough randomness in all directions of the regressor.

Definition 3.1 (Perturbed Least Squares)**.** Consider a function class $\mathcal{F} : X \to \mathbb{R}$. For an arbitrary dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$, a regularizer $R(f) = \sum_{j=1}^{D} p_j(f)^2$ where $p_j(\cdot)$ are functionals, and positive constant σ, the perturbed dataset and perturbed regularizer are defined as

$$\widetilde{\mathcal{D}}_\sigma = \{(x_i, y_i + \xi_i)\}_{i=1}^n, \quad \widetilde{R}_\sigma(f) = \sum_{j=1}^{D} [p_j(f) + \xi'_j]^2,$$

where ξ_i and ξ'_j are i.i.d. zero-mean Gaussian noises with variance σ^2. For a loss function L, the corresponding perturbed least squares regression solution is

$$\widetilde{f}_\sigma = \arg\min_{f \in \mathcal{F}} L(f \mid \widetilde{\mathcal{D}}_\sigma)^2 + \lambda \widetilde{R}_\sigma(f).$$

Within each episode $k \in [K]$, at each time-step h, we perturb the dataset by adding zero mean random Gaussian noise to the reward in the replay buffer $\{(s_h^\tau, a_h^\tau, r_h^\tau)\}_{\tau \in [k-1]}$ and the regularizer before we solve the perturbed regularized least-squares regression. At each time step h, we repeat the process for M (to be specified in Section 4) times and use the maximum of the regressor as the optimistic estimate of the state-action value function. Concretely, we set $V_{H+1}^k = 0$ and calculate $Q_H^k, Q_{H-1}^k, \dots, Q_1^k$ iteratively as follows. For each $h \in [H]$ and $m \in [M]$, we solve the following perturbed regression problem,

$$\widetilde{f}_h^{k,m} \leftarrow \arg\min_{f \in \mathcal{F}} L(f \mid \widetilde{\mathcal{D}}_h^{k,m})^2 + \lambda \widetilde{R}(f). \tag{3}$$

We set $Q_h^{k,m}(\cdot,\cdot) = \widetilde{f}_h^{k,m}(\cdot,\cdot)$ and define

$$Q_h^k(\cdot,\cdot) = \min\{\max_{m \in [M]} \{Q_h^{k,m}(\cdot,\cdot)\}, H - h + 1\}. \tag{4}$$

We then choose the greedy policy with respect to Q_h^k and collect a trajectory data for the k-th episode. We repeat the procedure until all the K episodes are completed.

3.1. LSVI-PHE with Linear Function Class

We now present LSVI-PHE when we consider linear function class (see Algorithm 2). In this case, the following proposition shows that, adding scalar Gaussian noise to the reward is equivalent to perturbing the least-squares estimate using d-dimensional multivariate Gaussian noise.

Proposition 3.2. *In line 9 of Algorithm 2, conditioned on all the randomness except $\{\epsilon_h^{k,i,j}\}_{(i,j) \in [k-1] \times [M]}$ and $\{\xi_h^{k,j}\}_{j \in [M]}$, the estimated parameter $\widetilde{\theta}_h^{k,j}$ satisfies*

$$\widetilde{\theta}_h^{k,j} - \widehat{\theta}_h^{k,j} \sim N(0, \sigma^2 (\Lambda_h^k)^{-1}),$$

where $\widehat{\theta}_h^{k,j} = (\Lambda_h^k)^{-1} (\sum_{\tau=1}^{k-1} [r_h^\tau + V_{h+1}^k(s_{h+1}^\tau)] \phi(s_h^\tau, a_h^\tau))$ is the unperturbed regressor.

Intuitively, adding a zero-mean multivariate Gaussian noise on the parameter $\widehat{\theta}_h^k$ can guarantee that $\widetilde{Q}_h^k$ is optimistic with constant probability. By repeating this procedure multiple times, this constant probability can be amplified to arbitrary high probability.

4. Theoretical Analysis

For the analysis we will need the concept of the *eluder dimension* due to (Russo & Van Roy, 2013). Let $\mathcal{F}$ be

Algorithm 2 LSVI-PHE with Linear function class

```
Set $M$ to be a fixed integer.
For episode $k = 1, 2, \dots, K$ do
  Receive the initial state $s_1^k$.
  For step $h = H, H-1, \dots, 1$ do
    $\Lambda_h^k \leftarrow \sum_{\tau=1}^{k-1} \phi(s_h^\tau, a_h^\tau)\phi(s_h^\tau, a_h^\tau)^\top + \lambda I$.
    Sample i.i.d. $\{\epsilon_h^{k,\tau,j}\}_{(\tau,j)\in[k-1]\times[M]} \sim N(0, \sigma^2)$.
    Sample i.i.d. $\{\xi_h^{k,j}\}_{j\in[M]} \sim N(0, \sigma^2 \lambda I_d)$.
    $\rho_h^{k,j} \leftarrow \sum_{\tau=1}^{k-1} \Big( [r_h^\tau + V_{h+1}^k(s_{h+1}^\tau) + \epsilon_h^{k,\tau,j}]\phi(s_h^\tau, a_h^\tau) \Big)$.
    $\widetilde{\theta}_h^{k,j} \leftarrow (\Lambda_h^k)^{-1}(\rho_h^k + \xi_h^{k,j})$.
    $\widetilde{Q}_h^{k,j}(\cdot,\cdot) \leftarrow \phi(\cdot,\cdot)^\top \widetilde{\theta}_h^{k,j}$ for $j \in [M]$.
    $Q_h^k(\cdot,\cdot) \leftarrow \min\{\max_{j\in[M]} \widetilde{Q}_h^{k,j}(\cdot,\cdot), H-h+1\}^+$
    $V_h^k(\cdot) \leftarrow \max_{a\in\mathcal{A}} Q_h^k(\cdot, a)$.
  For step $h = 1, 2, \dots, H$ do
    Take action $a_h^k \leftarrow \arg\max_{a\in\mathcal{A}} Q_h^k(s_h^k, a)$.
    Observe reward $r_h^k(s_h^k, a_h^k)$, get next state $s_{h+1}^k$.
```

a set of real-valued functions with domain $\mathcal{X}$. For $f \in \mathcal{F}, x_1, ..., x_t \in \mathcal{X}$, introduce the notation $f|_{(x_1,...,x_t)} = (f(x_1), ..., f(x_t))$. We say that $x \in \mathcal{X}$ is ϵ-independent of $x_1, ..., x_t \in \mathcal{X}$ given $\mathcal{F}$ if there exists $f, f' \in \mathcal{F}$ such that $||(f - f')|_{(x_1,...,x_t)}||_2 \le \epsilon$ while $f(x) - f'(x) > \epsilon$.

Definition 4.1 (Eluder dimension, (Russo & Van Roy, 2013))**.** The eluder dimension $\dim_{\mathcal{E}}(\mathcal{F}, \epsilon)$ of $\mathcal{F}$ at scale ϵ is the length of the longest sequence $(x_1, ..., x_n)$ in $\mathcal{X}$ such that for some $\epsilon' \ge \epsilon$, for any $2 \le t \le n$, x_t is ϵ'-independent of $(x_1, ..., x_{t-1})$ given $\mathcal{F}$.

For a more detailed introduction of eluder dimension, readers can refer to (Russo & Van Roy, 2013; Osband & Van Roy, 2014; Wang et al., 2020; Ayoub et al., 2020).

4.1. Assumptions for General Function Approximation

For our general function approximation analysis, we make a few assumptions first. To emphasize the generality of our assumptions, in Section 4.1.1, we show that our assumptions are satisfied by linear function class.

Our algorithm (Algorithm 1) receives a function class $\mathcal{F} \subseteq \{f : \mathcal{S} \times \mathcal{A} \to [0, H]\}$ as input and furthermore, similar to (Wang et al., 2020; Ayoub et al., 2020), we assume that for any $V : \mathcal{S} \to [0, H]$, upon applying the Bellman backup operator, the output function lies in the function class $\mathcal{F}$. Concretely, we have the following assumption.

Assumption A. For any $V : \mathcal{S} \to [0, H]$ and for any $h \in [H]$, $r_h + P_h V \in \mathcal{F}$, i.e. there exists a function $f_V \in \mathcal{F}$ such that for all $(s, a) \in \mathcal{S} \times \mathcal{A}$ it satisfies

$$f_V(s, a) = r_h(s, a) + P_h V(s, a). \qquad (5)$$

We emphasize that many standard assumptions in the RL theory literature such as tabular MDPs (Jaksch et al., 2010; Jin et al., 2018) and Linear MDPs (Yang & Wang, 2019; Jin et al., 2020) are special cases of Assumption A. In the appendix, we consider a misspecified setting and show that even when (5) holds approximately, Algorithm 1 achieves provable regret bounds.

We further assume that our function class has bounded covering number.

Assumption B. For any $\varepsilon > 0$, there exists an ε-cover $\mathcal{C}(\mathcal{F}, \varepsilon)$ with bounded covering number $\mathcal{N}(\mathcal{F}, \varepsilon)$.

Next we define anti-concentration width, which is a function of the function class $\mathcal{F}$, dataset $\mathcal{D}$ and noise variance σ^2.

Definition 4.2 (Anti-concentration Width Function)**.** For a loss function $L(\cdot\,|\,\cdot)$ and dataset $\mathcal{D}$, let $\widehat{f} = \arg\min_{f\in\mathcal{F}} L(f\,|\,\mathcal{D})^2 + \lambda R(f)$ be the regularized least squares solution and $\widetilde{f}_\sigma = \arg\min_{f\in\mathcal{F}} L(f\,|\,\widetilde{\mathcal{D}}_\sigma)^2 + \lambda \widetilde{R}_\sigma(f)$ be the perturbed regularized least-squares solution. For a fixed $v \in (0, 1)$, let $g_\sigma : X \to \mathbb{R}$ be a function such that for any input x:

$$g_\sigma(x) = \sup_{g\in\mathbb{R}} \mathbb{P}\left(\widetilde{f}_\sigma(x) \ge \widehat{f}(x) + g\right) \ge v.$$

We call $g_\sigma(\cdot)$ the anti-concentration width function.

In words, $g_\sigma(\cdot)$ is the largest value some $g \in \mathbb{R}$ can take such that the probability that $\widetilde{f}_\sigma$ is greater than $\widehat{f} + g$ is at least v.

We assume that for a concentrated function class, there exists a σ such that the anti-concentration width is larger than the function class width.

Assumption C (Anti-concentration)**.** Given the input $X = \{x_i\}_{i=1}^n$ of dataset $\mathcal{D}$ and some arbitrary positive constant β, we define a function class $\mathcal{F}_{X,\beta} = \{f : \|f - \widehat{f}\|_X^2 + \lambda R(f - \widehat{f}) \le \beta\}$. We assume that there exists a σ such that

$$g_{\sigma'}(x) \ge w(\mathcal{F}_{X,\beta}, x),$$

for all inputs x and $\sigma' \ge \sigma$.

This assumption guarantees that the randomized perturbation over the regression target has large enough probability of being optimistic. This assumption is satisfied by the linear function class. For more details, see Section 4.1.1.

Our next assumption is on the regularizer function $R(\cdot)$.

Assumption D (Regularization)**.** We assume that our regularizer $R(\cdot)$ has several basic properties.

- $R(f) + R(f') \ge cR(f + f')$ for some positive constant $c > 0$, for all $f, f' \in \mathcal{F}$.
- $R(f) = R(-f) \ge 0$, for all $f, f' \in \mathcal{F}$.

- For any $V : \mathcal{S} \to [0, H]$, $R(r + PV) \le B$ for some constant $B \in \mathbb{R}$.

Here, the first property is nothing but a variation of triangle inequality. The second property is a symmetry property which is natural for norms. Both these properties are satisfied by commonly used regularizers such as ℓ_0, ℓ_1 or ℓ_2 norms. The last property is a boundedness assumption. For the case of ℓ_0 norm B takes the value of the dimension of the space. Moreover, along with the most commonly used (weighted) ℓ_2 regularizer, many other regularizers also satisfy this property.

Our final assumption is regarding the boundedness of the eluder dimension of the function class.

Assumption E (Bounded Function Class)**.** For any $V : \mathcal{S} \to [0, H]$ and any $\mathcal{Z} \in (\mathcal{S} \times \mathcal{A})^{\mathbb{N}}$, let $\mathcal{F}'$ be a subset of function class $\mathcal{F}$, consisting of all $f \in \mathcal{F}$ such that

$$\|f - v\|_{\mathcal{Z}}^2 + \lambda R(f - v) \le \beta,$$

where $v = r + PV$. We assume that $\mathcal{F}'$ has bounded eluder dimension.

Note that in Wang et al. (2020), they assume that the eluder dimension of the whole function class $\mathcal{F}$ is bounded. In contrast, ours is a weaker assumption since we only assume a subset $\mathcal{F}'$ to have a bounded eluder dimension.

In the following section, we show that the linear function class and ridge regularizer satisfy all the above assumptions.

4.1.1. Linear Function Class

First, we recall the standard linear MDP definition which was introduced in (Yang & Wang, 2019; Jin et al., 2020).

Definition 4.3 (Linear MDP, (Yang & Wang, 2019; Jin et al., 2020))**.** We consider a linear Markov decision process, $\mathrm{MDP}(\mathcal{S}, \mathcal{A}, H, \mathbb{P}, r)$ with a feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}^d$, where for any $(h, k) \in [H] \times [K]$, there exist d unknown (signed) measures $\mu_h = (\mu_h^{(1)}, \cdots, \mu_h^{(d)})$ over $\mathcal{S}$ and an unknown vector $w_h \in \mathbb{R}^d$, such that for any $(s, a) \in \mathcal{S} \times \mathcal{A}$, the following holds:

$$\mathbb{P}_h(s'|s, a) = \langle \phi(s, a), \mu_h(s') \rangle, \quad r_h(s, a) = \langle \phi(s, a), w_h \rangle.$$

Without loss of generality, we assume, for all $(s, a) \in \mathcal{S} \times \mathcal{A}$, $\|\phi(s, a)\| \le 1$, and for all $h \in [H]$, $\|w_h\| \le \sqrt{d}$ and $\|\mu_h(\mathcal{S})\| \le \sqrt{d}$.

Consider a fixed episode k and step h. We define $\mathcal{F} = \{f_\theta : f_\theta(s, a) = \phi(s, a)^\top \theta\}$ where $\theta \in \mathbb{R}^d$, $\mathcal{D} = \{(s_h^\tau, a_h^\tau, r_h^\tau)\}_{\tau \in [k-1]}$, and $R(f_\theta) = \|\theta\|^2 = \sum_{j=1}^d p_j(f_\theta)^2$ where $p_j(f_\theta) = e_j^\top \theta$ with e_j being the j-th standard basis vector. It is well known that linear function class satisfies Assumption A in linear MDP (Yang & Wang, 2019; Jin et al., 2020). We set $\widehat{f} = \arg\min_{f \in \mathcal{F}} L(f \,|\, \mathcal{D})^2 + \lambda R(f)$ to be $f_{\widehat{\theta}}$. Then we have

$$\begin{aligned}\widehat{\theta} &= \arg\min_{\theta} \sum_{\tau=1}^{k-1} (\phi(s_h^\tau, a_h^\tau)^\top \theta - r_h^\tau)^2 + \lambda \|\theta\|^2 \\ &= (\Lambda_h^k)^{-1} \sum_{\tau=1}^{k-1} r_h^\tau \phi(s_h^\tau, a_h^\tau),\end{aligned}$$

where $\Lambda_h^k = \sum_{\tau=1}^{k-1} \phi(s_h^\tau, a_h^\tau)\phi(s_h^\tau, a_h^\tau)^\top + \lambda I$. Similarly we set $f_{\widetilde{\theta}} = \widetilde{f}_\sigma = \arg\min_{f \in \mathcal{F}} L(f \,|\, \widetilde{\mathcal{D}}_\sigma)^2 + \lambda \widetilde{R}_\sigma(f)$. Then we have

$$\begin{aligned}\widetilde{\theta} &= (\Lambda_h^k)^{-1} \sum_{\tau=1}^{k-1} (r_h^\tau + \xi_\tau)\phi(s_h^\tau, a_h^\tau) + (\Lambda_h^k)^{-1} \sum_{j=1}^{d} \xi_j' e_j \\ &\sim N(\widehat{\theta}, \sigma^2 (\Lambda_h^k)^{-1}).\end{aligned}$$

For Definition 4.2, we set $v = \Phi(-1)$. Using the anti-concentration property of Gaussian distribution, it is straightforward to show that for any $(s, a) \in \mathcal{S} \times \mathcal{A}$:

$$\mathbb{P}\left(f_{\widetilde{\theta}}(s, a) \ge f_{\widehat{\theta}}(s, a) + \sigma \|\phi(s, a)\|_{(\Lambda_h^k)^{-1}}\right) = v.$$

So we have $g_\sigma(s, a) \ge \sigma \|\phi(s, a)\|_{(\Lambda_h^k)^{-1}}$ from Definition 4.2.

For Assumption C, the function class $\mathcal{F}_{\mathcal{D},\beta} = \{f : L(f - \widehat{f} \,|\, \mathcal{D})^2 + \lambda R(f - \widehat{f}) \le \beta\}$ is equivalent to $\Theta_{\mathcal{D},\beta} = \{\theta : (\theta - \widehat{\theta})^\top \Lambda_h^k (\theta - \widehat{\theta}) \le \beta\}$. So the width on the state-action pair (s, a) is $2\sqrt{\beta}\|\phi(s, a)\|_{(\Lambda_h^k)^{-1}}$. If we set $\sigma = 2\sqrt{\beta}$, we have

$$g_\sigma(s, a) \ge w(\mathcal{F}_{\mathcal{D},\beta}, s, a).$$

For Assumption D, as $R(f_\theta) = \|\theta\|^2$ is a ℓ_2 norm function, the first two properties are direct to show with constant $c = 1/2$. For the third property, we have that

$$g(s, a) = r(s, a) + P(s, a)V = \phi(s, a)(w + \sum_{s'} V(s')\mu(s')).$$

So we have $g = g_\theta$ where $\theta = w + \sum_{s'} V(s')\mu(s')$ and $\|\theta\|^2 \le 2Hd$.

For Assumption E, we set $\theta_f : f = f_{\theta_f}$, $\theta_v : v = f_{\theta_v}$ and $\Theta_{\mathcal{F}'} = \{\theta : f_\theta \in \mathcal{F}'\}$ to be the parameterization. From Assumption D, we have $\|\theta_v\|^2 \le 2Hd$. In addition, we have $\lambda R(f - v) = \lambda \|\theta_f - \theta_v\|^2 \le \beta$. Then we have

$$\begin{aligned}\Theta_{\mathcal{F}'} &\subseteq \{\theta_f : \|\theta_f - \theta_v\|^2 \le \beta/\lambda, \|\theta_v\|^2 \le 2Hd\} \\ &= \{\theta_f : \|\theta_f\|^2 \le 2\beta/\lambda + 4Hd\}.\end{aligned}$$

As shown in (Russo & Van Roy, 2013), this $\mathcal{F}'$ has eluder dimension $\widetilde{O}(d)$.

4.2. Regret bound for General Function Approximation

First, we specify our choice of the noise variance σ^2 in the algorithm. We prove certain concentration properties of the regularized regressor $\widehat{f}_h^k$ so that the condition in Assumption C holds. Thus we can choose an appropriate σ such that the Assumption C is satisfied. A more detailed description is provided in the appendix.

Our first lemma is about the concentration of the regressor. A similar argument appears in (Wang et al., 2020) but their result does not include regularization, which is essential in our randomized algorithm to ensure exploration in all directions.

Lemma 4.4 (Informal Lemma on Concentration)**.** *Under Assumptions A, B, C, D, and E, let* $\mathcal{F}_h^{k,m} = \{f \in \mathcal{F} | \|f - \widetilde{f}_h^{k,m}\|^2_{\mathcal{Z}_h^k} + \lambda R(f - \widetilde{f}_h^{k,m}) \le \beta(\mathcal{F}, \delta)\}$, *where* $\mathcal{Z}_h^k = \{(s_h^\tau, a_h^\tau)\}_{\tau \in [k-1]}$, *and*

$$\beta(\mathcal{F}, \delta) = \widetilde{O}\left((H+\sigma)^2 \log \mathcal{N}(\mathcal{F}, 1/T)\right).$$

With high probability, for all $(k, h, m) \in [K] \times [H] \times [M]$, *we have*

$$r_h(\cdot,\cdot) + P_h V_{h+1}^k(\cdot,\cdot) \in \mathcal{F}_h^{k,m}.$$

This lemma shows that the perturbed regularized regression still enjoys concentration.

Our next lemma shows that LSVI-PHE is optimistic with high probability.

Lemma 4.5 (Informal Lemma on Optimism)**.** *Let*

$$M = \ln\left(\frac{T|\mathcal{S}||\mathcal{A}|}{\delta}\right) / \ln\left(\frac{1}{1-v}\right).$$

With probability at least $1-\delta$, *for all* $(s, a, h, k) \in \mathcal{S} \times \mathcal{A} \times [H] \times [K]$, *we have*

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

With optimism, the regret is known to be bounded by the sum of confidence width (Wang et al., 2020). As Assumption E assumes that all the confidence region is in a bounded function class in the measure of eluder dimension, we can adapt proof techniques from (Wang et al., 2020) and prove our final result.

Theorem 4.6 (Informal Theorem)**.** *Under Assumptions A, B, C, D, and E, with high probability, Algorithm 1 achieves a regret bound of*

$$\mathrm{Regret}(K) \le \widetilde{O}\left(\sqrt{\dim_{\mathcal{E}}(\mathcal{F}, 1/T)\beta(\mathcal{F},\delta)HT}\right),$$

where

$$\beta(\mathcal{F}, \delta) = \widetilde{O}\left((H+\sigma)^2 \log \mathcal{N}(\mathcal{F}, 1/T)\right).$$

The theorem shows that our algorithm enjoys sublinear regret and have polynomial dependence on the horizon H, noise variance σ^2 and eluder dimension $\dim_{\mathcal{E}}(\mathcal{F}, 1/T)$, and have logarithmic dependence on the covering number of the function class $\mathcal{N}(\mathcal{F}, 1/T)$.

4.3. Regret bound for linear function class

Now we present the regret bound for Algorithm 2 under the assumption of linear MDP setting. In the appendix, we provide a simple yet elegant proof of the regret bound.

Theorem 4.7. *Let* $M = d\log(\delta/9)/\log\Phi(1)$, $\sigma = \widetilde{O}(H\sqrt{d})$, *and* $\delta \in (0,1]$. *Under linear MDP assumption from Definition 4.3, the regret of Algorithm 2 satisfies*

$$\mathit{Regret}(T) \le \widetilde{O}(d^{3/2}H^{3/2}\sqrt{T}),$$

with probability at least $1-\delta$.

Remark 4.8. *Under linear MDP assumption, this regret bound is at the same order as the LSVI-UCB algorithm from (Jin et al., 2020) and* $\sqrt{dH}$ *better than the state-of-the-art TS-type algorithm (Zanette et al., 2020a). The only work that enjoys a* $\sqrt{d}$ *better regret is (Zanette et al., 2020b), which requires solving an intractable optimization problem.*

Remark 4.9. *Along with being a competitive algorithm in statistical efficiency, we want to emphasize that our algorithm has good computational efficiency. LSVI-PHE with linear function class only involves linear programming to find the greedy policy while LSVI-UCB (Jin et al., 2020) requires solving a quadratic programming. The optimization problem in OPT-RLSVI (Zanette et al., 2020a) is hard too because the Q-function there is a piecewise continuous function and in one piece, it includes the product of the square root of a quadratic term and a linear term.*

5. Numerical Experiments

We run our experiments on RiverSwim (Strehl & Littman, 2008), DeepSea (Osband et al., 2016b) and sparse MountainCar (Brockman et al., 2016) environments as these are considered to be hard exploration problems where ε-greedy is known to have poor performance. For both RiverSwim and DeepSea experiments, we make use of linear features. The objective here is to compare an exploration method that randomizes the targets in the history (LSVI-PHE) with an exploration method that computes upper confidence bounds given the history (LSVI-UCB) (Jin et al., 2020; Cai et al., 2019). For the continous control MountainCar environment, we use neural-network as function approximator to implement LSVI-PHE. The objective here is to compare deep RL variant of LSVI-PHE against other popular deep RL algorithms specifically designed to tackle exploration task.

5.1. Measurements

We plot the per episode return of each algorithm to benchmark their performance. As the agent begins to act optimally the per episode return begins to converge to the optimal, or baseline, return. The per episodes returns are the sum of all the rewards obtained in an episode. We also report the performance of LSVI-PHE when σ^2 is fixed and M varies.

5.2. Results for RiverSwim

A diagram of the RiverSwim environment is shown in the Appendix. RiverSwim consists of $\mathcal{S}$ states lined up in a chain. The agent begins in the leftmost state s_1 and has the choice of swimming to the left or to the right at each state. The agent's goal is to maximize its return by trying to reach the rightmost state which has the highest reward. Swimming to the left, with the current, transitions the agent to the left deterministically. Swimming to the right, against the current, stochastically transitions the agent and has relatively high probability of moving right toward the goal state. However, because the current is strong there is a high chance the agent will stay in the current state and a low chance the agent will get swept up in the current and transition to the left. Thus, smart exploration is required to learn the optimal policy in this environment. We experiment with the variant of RiverSwim where $\mathcal{S} = 12$ and $H = 40$. For this experiment, we swept over the exploration parameters in both LSVI-UCB (Jin et al., 2020) and LSVI-PHE and report the best performing run on a 12 state RiverSwim. LSVI-UCB computes confidence widths of the following form $\beta\|\phi(s,a)\|_{\Sigma^{-1}}$ where $\phi(s,a) \in \mathbb{R}^d$ are the features for a given state-action pair and $\Sigma \in \mathbb{R}^{d\times d}$ is the empirical covariance matrix. We sweep over β for LSVI-UCB and σ^2 for LSVI-PHE, where M is chosen according to our theory (Theorem 4.7). We sweep over these parameters to speed up learning as choosing the theoretically optimal choices for β and σ^2 often leads to a more conservative exploration policy which is slow to learn. As shown in Figure 1, the best performing LSVI-PHE achieves similar performance to the best performing LSVI-UCB on the 12 state RiverSwim environment.

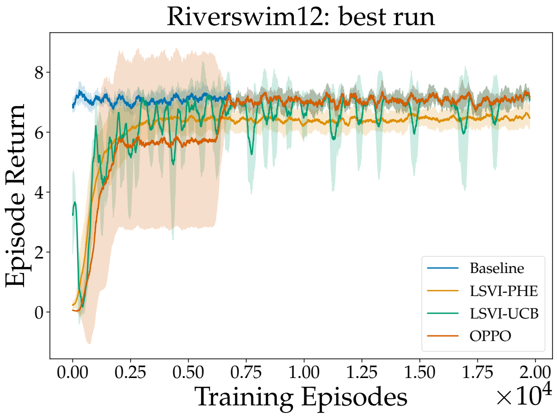

Figure 1: The results are averaged over 10 independent runs and error bars are reported for the regret plots. For this plot, $\beta = 5.0$ for LSVI-UCB and $\sigma^2 = 2 \times 10^{-1}$ for LSVI-PHE.

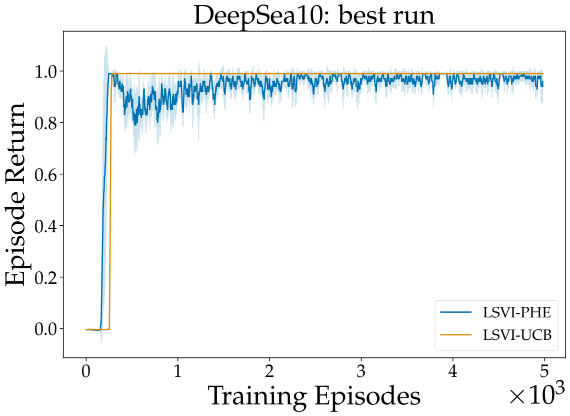

Figure 2: The results are averaged over 5 independent runs and error bars are reported for the return per episode plots. For this plot, $\beta = 5 \times 10^{-3}$ for LSVI-UCB and $\sigma^2 = 5 \times 10^{-5}$ for LSVI-PHE.

5.3. Results for DeepSea

DeepSea (Osband et al., 2016b) consists of $\mathcal{S} = N \times N$ states arranged in a grid, where N is the depth of the sea. The agent begins at the top leftmost state in the grid s_1 and has the choice of moving down and left or down and right at each state. Once the agent reaches the bottom of the sea it transitions back to state s_1. The agent's goal is to maximize its return by reaching the bottom right most state. The agent gets a small negative reward for transitioning to the right while no reward is given if the agent transitions to the left. Thus, smart exploration is required; otherwise the agent will rarely go right the necessary amount of time to reach the goal state. We run our experiments on a 10×10 DeepSea environment. As shown in Figure 2, the best performing LSVI-PHE achieves similar performance to the best performing LSVI-UCB on DeepSea. We also vary M given a fixed $\sigma^2 = 5 \times 10^{-4}$. As shown in Figure 3, as we increase M, the performance of LSVI-PHE increases.

These experiments on hard exploration problems highlight that we are able to simulate optimistic exploration, as in UCB, by perturbing the targets multiple times and taking the max over the perturbations to boost the probability of an optimistic estimate. If we are willing to sweep over M, the number of times we perturb the history, and σ^2, we can then get a faster algorithm that still performs well in practice. If we let $M = 1$ and $\sigma^2 = 1$ then LSVI-PHE

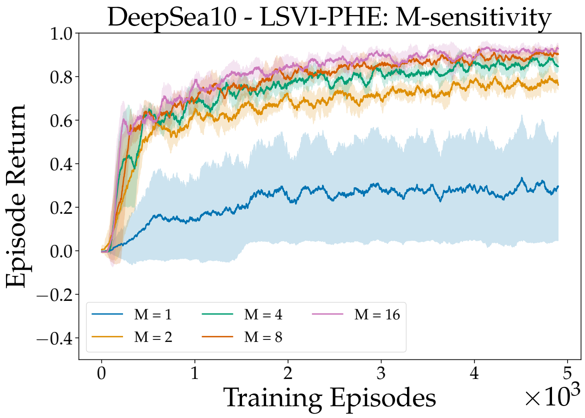

Figure 3: The results are averaged over 5 runs and error bars are reported for the return per episode plots. For this plot we fix $\sigma^2 = 5 \times 10^{-4}$.

reduces to RLSVI and we would get the same performance as in (Osband et al., 2016b).

5.4. Results for MountainCar

We further evaluated LSVI-PHE on a continuous control task which requires exploration: sparse reward variant of continuous control MountainCar from OpenAI Gym (Brockman et al., 2016). This environment consists of a 2-dimensional continuous state space and a 1-dimensional continuous action space $[-1, 1]$. The agent only receives a reward of $+1$ if it reaches the top of the hill and everywhere else it receives a reward of 0. We set the length of the horizon to be 1000 and discount factor $\gamma = 0.99$.

For this setting, we compare four algorithms: LSVI-PHE, DQN with epsilon-greedy exploration, Noisy-Net DQN (Fortunato et al., 2017) and Bootstrapped DQN (Osband et al., 2016a). Our experiments are based on the baseline implementations of (Lan, 2019). As neural network, we used a multi-layer perceptron with hidden layers fixed to $[32, 32]$. The size of the replay buffer was $10,000$. The weights of neural networks were optimized by Adam (Kingma & Ba, 2014) with gradient clip 5. We used a batch size of 32. The target network was updated every 100 steps. The best learning rate was chosen from $[10^{-3}, 5 \times 10^{-4}, 10^{-4}]$. For LSVI-PHE, we set $M = 8$ and we chose the best value of σ from $[10^{-4}, 10^{-3}, 10^{-2}]$. Results are shown in Figure 4.

6. Related Works

RL with Function Approximation. Many recent works have studied RL with function approximation, especially in the linear case (Jin et al., 2020; Cai et al., 2019; Zanette et al., 2020a;b; Wang et al., 2020; Ayoub et al., 2020; Foster et al., 2020; Jiang et al., 2017; Sun et al., 2019). Under the assumption that the agent has access to a well-designed feature extractor, these works design provably efficient algorithms for linear MDPs and linear kernel MDPs. LSVI-UCB (Jin et al., 2020), the first work with both polynomial runtime and polynomial sample complexity with linear function approximation, has a regret of $\widetilde{\mathcal{O}}(\sqrt{d^3 H^3 T})$. The state-of-the-art regret bound is $\widetilde{\mathcal{O}}(Hd\sqrt{T})$, achieved by ELEANOR (Zanette et al., 2020b). However, ELEANOR needs to solve an optimization problem in each episode, which is computationally intractable. Wang et al. (2019) introduces a new expressivity condition named *optimistic closure* for generalized linear function approximation under which they propose a variant of optimistic LSVI with regret bound $\widetilde{\mathcal{O}}(\sqrt{d^3 H^3 T})$. Wang et al. (2020); Ayoub et al. (2020) focus on online RL with general function approximation and their analysis is based on the eluder dimension (Russo & Van Roy, 2013). Other complexity measures of general function classes include disagreement coefficient (Foster et al., 2020), Bellman rank (Jiang et al., 2017) and Witness rank (Sun et al., 2019).

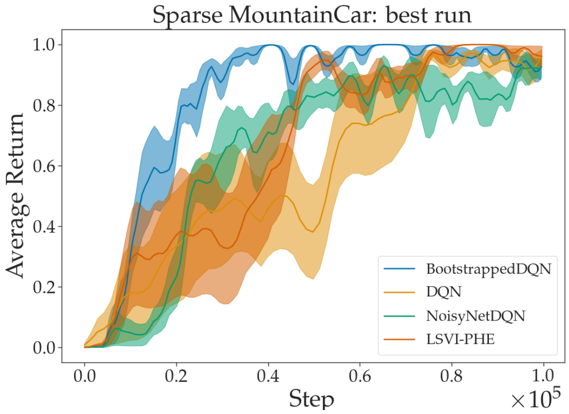

Figure 4: : Comparison of four algorithms on sparse MountainCar. The results are averaged over 5 independent runs and error bars are reported for the return per episode plots.

Thompson Sampling. Thompson Sampling (Thompson, 1933) was proposed almost a century ago and rediscovered several times. Strens (2000) was the first work to apply TS to RL. Osband et al. (2013) provides a Bayesian regret bound and Agrawal et al. (2016); Ouyang et al. (2017) provide worst case regret bounds for TS.

Randomized least-squares value iteration (RLSVI), proposed in Osband et al. (2019), uses random perturbations to approximate the posterior. Recently, several works focussed on the theoretical analysis of RLSVI (Russo, 2019; Zanette et al., 2020b; Agrawal et al., 2020). Russo (2019) provides the first worst-case regret $\widetilde{O}(H^{5/2}S^{3/2}\sqrt{AT})$ for tabular MDP and Agrawal et al. (2020) improves it to $\widetilde{O}(H^2 S\sqrt{AT})$. Zanette et al. (2020a) proves $\widetilde{O}(H^2 d^2\sqrt{T})$ regret bound for linear MDP. However, Agrawal et al. (2020); Zanette et al. (2020a) both need to compute the confidence width as a warm-up stage, which is complicated and computationally costly.

7. Conclusion

In this work, we propose an algorithm LSVI-PHE for online RL with function approximation based on optimistic sampling. We prove the theoretical guarantees of LSVI-PHE and through experiments also demonstrate that it performs competitively against previous algorithms. We believe optimistic sampling provides a new provably efficient exploration paradigm in RL and it is practical in complicated real-world applications. We hope our work can be one step towards filling the gap between theory and application.

Acknowledgements

The authors would like to thank Csaba Szepesvári for helpful discussions. HI thanks Qingfeng Lan for many insightful discussions regarding the experiments. The work was initiated when HI, ZY, ZW and LY was visiting the Simons Institute for the Theory of Computing at UC-Berkeley (Theory of Reinforcement Learning Program).

References

Abbasi-Yadkori, Y., Pál, D., and Szepesvári, C. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pp. 2312–2320, 2011.

Agrawal, P., Chen, J., and Jiang, N. Improved worst-case regret bounds for randomized least-squares value iteration. *arXiv preprint arXiv:2010.12163*, 2020.

Agrawal, S. and Jia, R. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. In *Advances in Neural Information Processing Systems*, pp. 1184–1194, 2017.

Agrawal, S., Avadhanula, V., Goyal, V., and Zeevi, A. A near-optimal exploration-exploitation approach for assortment selection. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pp. 599–600, 2016.

Ayoub, A., Jia, Z., Szepesvari, C., Wang, M., and Yang, L. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pp. 463–474. PMLR, 2020.

Azar, M. G., Osband, I., and Munos, R. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 263–272. JMLR. org, 2017.

Brafman, R. I. and Tennenholtz, M. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *JOURNAL OF MACHINE LEARNING RESEARCH*, 3:213–231, 2001.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Cai, Q., Yang, Z., Jin, C., and Wang, Z. Provably efficient exploration in policy optimization. *arXiv preprint arXiv:1912.05830*, 2019.

Fortunato, M., Azar, M. G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., et al. Noisy networks for exploration. *arXiv preprint arXiv:1706.10295*, 2017.

Foster, D. J., Rakhlin, A., Simchi-Levi, D., and Xu, Y. Instance-dependent complexity of contextual bandits and reinforcement learning: A disagreement-based perspective. *arXiv preprint arXiv:2010.03104*, 2020.

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

Jiang, N., Krishnamurthy, A., Agarwal, A., Langford, J., and Schapire, R. E. Contextual decision processes with low bellman rank are pac-learnable. In *International Conference on Machine Learning*, pp. 1704–1713. PMLR, 2017.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. Is q-learning provably efficient? In *Advances in Neural Information Processing Systems*, pp. 4863–4873, 2018.

Jin, C., Yang, Z., Wang, Z., and Jordan, M. I. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, pp. 2137–2143. PMLR, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kveton, B., Zaheer, M., Szepesvari, C., Li, L., Ghavamzadeh, M., and Boutilier, C. Randomized exploration in generalized linear bandits, 2019.

Lan, Q. A pytorch reinforcement learning framework for exploring new ideas. `https://github.com/qlan3/Explorer`, 2019.

Lee, K., Laskin, M., Srinivas, A., and Abbeel, P. Sunrise: A simple unified framework for ensemble learning in deep reinforcement learning. *arXiv preprint arXiv:2007.04938*, 2020.

Osband, I. and Van Roy, B. Model-based reinforcement learning and the eluder dimension. *arXiv preprint arXiv:1406.1853*, 2014.

Osband, I., Russo, D., and Van Roy, B. (more) efficient reinforcement learning via posterior sampling. *arXiv preprint arXiv:1306.0940*, 2013.

Osband, I., Blundell, C., Pritzel, A., and Van Roy, B. Deep exploration via bootstrapped dqn. *arXiv preprint arXiv:1602.04621*, 2016a.

Osband, I., Van Roy, B., and Wen, Z. Generalization and exploration via randomized value functions. In *International Conference on Machine Learning*, pp. 2377–2386. PMLR, 2016b.

Osband, I., Van Roy, B., Russo, D. J., and Wen, Z. Deep exploration via randomized value functions. *Journal of Machine Learning Research*, 20(124):1–62, 2019.

Ouyang, Y., Gagrani, M., Nayyar, A., and Jain, R. Learning unknown markov decision processes: A thompson sampling approach. In *Advances in Neural Information Processing Systems*, pp. 1333–1342, 2017.

Russo, D. Worst-case regret bounds for exploration via randomized value functions. In *Advances in Neural Information Processing Systems*, pp. 14410–14420, 2019.

Russo, D. and Van Roy, B. Eluder dimension and the sample complexity of optimistic exploration. In *Advances in Neural Information Processing Systems*, pp. 2256–2264, 2013.

Strehl, A. L. and Littman, M. L. An analysis of model-based interval estimation for markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.

Strens, M. A bayesian framework for reinforcement learning. In *ICML*, volume 2000, pp. 943–950, 2000.

Sun, W., Jiang, N., Krishnamurthy, A., Agarwal, A., and Langford, J. Model-based rl in contextual decision processes: Pac bounds and exponential improvements over model-free approaches. In *Conference on Learning Theory*, pp. 2898–2933. PMLR, 2019.

Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Vershynin, R. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

Wang, R., Salakhutdinov, R. R., and Yang, L. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. *Advances in Neural Information Processing Systems*, 33, 2020.

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

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

Zanette, A., Brandfonbrener, D., Brunskill, E., Pirotta, M., and Lazaric, A. Frequentist regret bounds for randomized least-squares value iteration. In *International Conference on Artificial Intelligence and Statistics*, pp. 1954–1964. PMLR, 2020a.

Zanette, A., Lazaric, A., Kochenderfer, M., and Brunskill, E. Learning near optimal policies with low inherent bellman error. In *International Conference on Machine Learning*, pp. 10978–10989. PMLR, 2020b.