

Sample Efficient Reinforcement Learning with Partial Dynamics Knowledge

Meshal Alharbi, Mardavij Roozbehani, Munther Dahleh

Laboratory for Information and Decision Systems, Massachusetts Institute of Technology
Cambridge, MA 02139, USA
{meshal, mardavij, dahleh}@mit.edu

Abstract

The problem of sample complexity of online reinforcement learning is often studied in the literature without taking into account any partial knowledge about the system dynamics that could potentially accelerate the learning process. In this paper, we study the sample complexity of online Q-learning methods when some prior knowledge about the dynamics is available or can be learned efficiently. We focus on systems that evolve according to an additive disturbance model of the form $S_{h+1} = f(S_h, A_h) + W_h$, where f represents the underlying system dynamics, and W_h are unknown disturbances independent of states and actions. In the setting of finite episodic Markov decision processes with S states, A actions, and episode length H , we present an optimistic Q-learning algorithm that achieves $\tilde{O}(\text{POLY}(H)\sqrt{T})$ regret under perfect knowledge of f , where T is the total number of interactions with the system. This is in contrast to the typical $\tilde{O}(\text{POLY}(H)\sqrt{SAT})$ regret for existing Q-learning methods. Further, if only a noisy estimate \hat{f} of f is available, our method can learn an approximately optimal policy in a number of samples that is independent of the cardinalities of state and action spaces. The sub-optimality gap depends on the approximation error $\hat{f} - f$, as well as the Lipschitz constant of the corresponding optimal value function. Our approach does not require modeling of the transition probabilities and enjoys the same memory complexity as model-free methods.

Introduction

Motivations. Reinforcement learning (RL) algorithms have achieved tremendous success in various domains, and the literature has seen an influx of methods with provable non-asymptotic guarantees. Many of these results are problem-agnostic and the theoretical guarantees are limited by the worst-case scenarios. At the same time, one would hope that the sample efficiency of RL could be tied to the inherent measures of complexity of the problem’s dynamics, with ways to accelerate learning if some prior knowledge about the problem is available. Indeed, the following question in its full generality is still open:

How can we improve the sample efficiency of RL methods when some structure about the underlying dynamics is known or can be learned efficiently?

Copyright © 2024, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Practical methods that aim to address the previous question should carefully balance the sample complexity (number of interactions with the system) and computations (time and memory) spent in planning or improving policies. Our primary objective is to incorporate prior structural information without extensive offline computation or access to strong computational oracles.

Systems of Interest. Learning complexity in RL is multifaceted, and is influenced by uncertainty in both rewards and dynamics (see, e.g., Lu et al. (2023) for an introductory treatment on the subject). In the interest of tractability, herein, we focus our attention on a specific class of RL problems with uncertainty in the dynamics only. We consider a class \mathcal{M} of Markov Decision Processes (MDPs) where the state transition is governed by:

$$S_{h+1} = f(S_h, A_h) + W_h \quad (1)$$

where f , which may be partially or approximately known, represents the dynamics or fixed structure of the system, and W_h are unknown disturbances independent of states and actions. Further, we assume the reward r function is known; and thus, all the uncertainty is in the state transitions. Partial dynamics knowledge is modeled via the availability of an approximation \hat{f} of f that satisfies $\|\hat{f} - f\|_\infty \leq \zeta/2$. Note that this is different than *sim-to-real* literature, where it is often assumed that one has a detailed probabilistic description of an approximate model (Jiang 2018; Feng, Yin, and Yang 2019) or an approximate generative model where we can sample an arbitrary amount of trajectories without cost (Chen et al. 2022). See the next section for a detailed comparison. Furthermore, even if f is fully known, one cannot apply dynamic programming here as the disturbances W_h are not known even in distribution (in fact, this can be the main challenge in many practical problems). Finally, we shall see that restricting to linear disturbance models is beneficial for devising computationally efficient algorithms.

Many applications fit the setting described above. In operational tasks, like inventory control and demand response problems (Vázquez-Canteli and Nagy 2019), the structural function f is often known and the challenge lies in optimizing for the unknown non-stationary demand signals. In control tasks, where the dynamics rise from parametric physical models, one might only know f approximately due to parametric uncertainty (Bhattacharyya 2017; Taylor et al. 2021).

Structural Knowledge	Algorithm	Regret	Time	Space
Without	UCBVI (Azar, Osband, and Munos 2017)	$\tilde{\mathcal{O}}(\sqrt{H^2 SAT} + H^4 S^2 A)$	$\mathcal{O}(S^2 AT)$	$\mathcal{O}(S^2 AH)$
	UBEV (Dann, Lattimore, and Brunskill 2017)	$\tilde{\mathcal{O}}(\sqrt{H^3 SAT} + H^2 S^3 A^2)$	$\mathcal{O}(S^2 AT)$	$\mathcal{O}(S^2 AH)$
	UCB-H (Jin et al. 2018)	$\tilde{\mathcal{O}}(\sqrt{H^4 SAT})$	$\mathcal{O}(T)$	$\mathcal{O}(SAH)$
	Q-EarlySettled-Advantage (Li et al. 2021b)	$\tilde{\mathcal{O}}(\sqrt{H^2 SAT} + H^6 SA)$	$\mathcal{O}(T)$	$\mathcal{O}(SAH)$
With	UCB-f (this work)	$\tilde{\mathcal{O}}(\sqrt{H^6 T})$	$\mathcal{O}(SAT)$	$\mathcal{O}(SAH)$

Table 1: Summary of regret, time, and space complexities of Q-learning algorithms on finite episodic MDPs.

For all of these problems, the assumption of known reward is not restrictive as the reward function is a design parameter.

Contributions. We study the previously stated research question under the finite episodic MDPs setting. Our main findings can be summarized as follows:

- Under the perfect knowledge of f , we propose an optimistic Q-learning method that achieves $\tilde{\mathcal{O}}(\sqrt{H^6 T})^1$ regret with high probability without requiring any special assumptions on f or the distribution of the unknown disturbances W_h .
- If only a noisy estimate \hat{f} of f with $\|\hat{f} - f\|_\infty \leq \zeta/2$ is available, we show that our algorithm is able to learn (with high probability) an $\mathcal{O}(L\zeta H^2)$ -optimal policy in a number of samples that is independent of the cardinalities of state and action spaces. Here, L is the Lipschitz constant of the optimal value function, which characterizes its smoothness.
- We further show that if an asymptotically accurate online estimator that can generate a sequence of functions $\{\hat{f}_i\}_{i=1}^K$ with $\|\hat{f}_i - f\|_\infty \leq \mathcal{O}(\sqrt{d/i})$ exists, then the regret of using such estimators with our algorithm is $\tilde{\mathcal{O}}(\sqrt{H^6 T} + L\sqrt{HdT})$.

To put these results in perspective, we compare our method with different problem-agnostic Q-learning algorithms in Table 1. Our algorithm effectively incorporates the structure in a way that trades off regret and computation without requiring the S^2 space and time dependency of model-based methods.

Related Literature

The RL literature spans decades and is extensive. This section focuses on a concise review of the most relevant works to our study.

RL without Simulator. In the finite episodic setting without access to a simulator, the best regret bound $\tilde{\mathcal{O}}(\sqrt{H^2 SAT})$ is achieved by both model-based (Azar, Osband, and Munos 2017) and model-free (Li et al. 2021b) Q-learning algorithms. These results are known to be minimax-optimal, and are thus, constrained by the worst-case scenarios. See Domingues et al. (2021) for the lower bound proofs and the construction of worst-case MDPs.

¹The $\tilde{\mathcal{O}}(\cdot)$ notation is analogous to the standard $\mathcal{O}(\cdot)$ notation while ignoring $\text{POLYLOG}(S, A, H, K, 1/p)$ dependencies.

Problem-dependent bounds characterize sample complexity by additional metrics, such as the maximum per-step conditional variance (Zanette and Brunskill 2019), minimum sub-optimality gap (Yang, Yang, and Du 2021), or the gap-visitation complexity (Wagenmaker, Simchowitz, and Jamieson 2022), and show sharper bounds when these metrics are small. Still, these bounds often show the same polynomial scaling in the cardinalities of states and actions as the problem-free bounds.

RL with Approximate Model/Simulator. The use of an approximate model or an approximate simulator in the training of RL methods, sometimes called *sim-to-real transfer*, was shown to effectively improve the sample complexity both empirically (Bousmalis et al. 2018; Zhao, Queralta, and Westerlund 2020) and theoretically (Chen et al. 2022). Our work is perhaps closest to this line of work in the literature. In (Jiang 2018), the author assumes access to a probabilistic description of an approximate transition kernel, and, under some assumptions, they characterize the sample complexity by the number of incorrect state-action pairs in the approximate kernel. In an alternative approach, the work of (Ayoub et al. 2020) assumes that the actual transition kernel belongs to a known family of models and provides an algorithm with regret that scales with the Eluder dimension for this family. Both works show interesting results where the sample complexity does not scale with the number of states and actions. On the other hand, the algorithms are computationally intensive, requiring a dynamic programming (DP) solver in each iteration. Our work differs from this literature as we do not require an oracle DP solver, nor do we require a detailed probabilistic description of the approximate model.

RL with System Identification. Rather than depending on a prior approximate model, other studies in the literature alternate between learning the model (i.e., system identification) and planning with RL (Nagabandi et al. 2018; Zhang et al. 2019; Schrittwieser et al. 2021). In terms of theoretical analysis, the most prominent examples are for the Linear-quadratic regulators (LQRs), where the sample complexity of the planning under an unknown model has been thoroughly quantified by the intrinsic measures of the difficulty of learning linear systems (Dean et al. 2018; Cassel, Cohen, and Koren 2020). Algorithms for LQRs heavily rely on structural properties (e.g., the fact that the optimal policy is linear and the optimal value function is quadratic) in their design and cannot be immediately generalized to a broader class of problems.

Algorithm 1: (UCB-f) Optimistic Q-learning with \hat{f}

```

1: Initialize  $Q_h^1(s, a) \leftarrow H$  and  $V_h^1(s) \leftarrow H$  for all  $(s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]$ 
2: for episode  $k = 1, \dots, K$  do
3:   Receive  $s_1^k$   $\triangleright$  Initial state
4:   for step  $h = 1, \dots, H$  do
5:      $a_h^k \leftarrow \arg \max_{a'} Q_h^k(s_h^k, a')$   $\triangleright$  Greedy action with respect to  $Q_h^k$ 
6:     Take action  $a_h^k$  and observe next state  $s_{h+1}^k$ 
7:      $w_h^k \leftarrow s_{h+1} - \hat{f}(s_h, a_h)$   $\triangleright$  Transition randomness
8:     for all  $s \in \mathcal{S}$  do
9:       for all  $a \in \mathcal{A}$  do
10:         $s' \leftarrow \hat{f}(s, a) + w_h^k$   $\triangleright$  Simulated next state
11:         $Q_h^{k+1}(s, a) \leftarrow (1 - \alpha_k)Q_h^k(s, a) + \alpha_k [r_h(s, a) + V_{h+1}^k(s') + b_k]$   $\triangleright$  Q-learning update
12:         $V_h^{k+1}(s) \leftarrow \min \{H, \max_{a'} Q_h^{k+1}(s, a)\}$   $\triangleright$  Value function update

```

Preliminaries

We consider an episodic MDP described by the tuple $(\mathcal{S}, \mathcal{A}, H, \mathbb{P}, \mu, r)$, where $\mathcal{S} = \{1, 2, \dots, S\} \subseteq \mathbb{N}$ and $\mathcal{A} = \{1, 2, \dots, A\} \subseteq \mathbb{N}$ are the finite state and the action spaces, H is the number of steps in each episode, \mathbb{P} is the transition kernel such that $\mathbb{P}_h(\cdot | s, a)$ is the distribution over next states if action a is taken at state s and step h , μ is the distribution of initial states, and $r_h : \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$ are the deterministic reward functions. For convenience, we define $[\mathbb{P}_h V](s, a) := \mathbb{E}_{s' \sim \mathbb{P}_h(\cdot | s, a)}[V(s')]$ for any $V : \mathcal{S} \rightarrow \mathbb{R}$.

We use π to denote a stationary deterministic policy, which is a collection of H functions $\{\pi_h : \mathcal{S} \rightarrow \mathcal{A}\}_{h \in [H]}$ ². The value function $V_h^\pi : \mathcal{S} \rightarrow \mathbb{R}$ and action-value function $Q_h^\pi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ of a policy π are defined as:

$$V_h^\pi(s) := \mathbb{E} \left[\sum_{t=h}^H r_t(s_t, \pi_t(s_t)) \mid s_h = s \right] \quad (2)$$

$$Q_h^\pi(s, a) := r_h(s, a) + \mathbb{E} \left[\sum_{t=h+1}^H r_t(s_t, \pi_t(s_t)) \mid s_h = s, a_h = a \right] \quad (3)$$

Further, we adopt the convention that $V_{H+1}^\pi(s) = 0$ for any policy. Since the state space, action space, and horizon are all finite and rewards are bounded, there always exists an optimal policy π^* that achieves the optimal value $V_h^*(s) = \sup_{\pi} V_h^\pi(s)$ for all $(s, h) \in \mathcal{S} \times [H]$ (Bertsekas 2017). For all $(s, a, h) \in \mathcal{S} \times \mathcal{A} \times [H]$, the Bellman equations for a policy π and Bellman optimality equations are:

$$\begin{aligned} V_h^\pi(s) &= Q_h^\pi(s, \pi_h(s)), & Q_h^\pi(s, a) &= (r_h + \mathbb{P}_h V_{h+1}^\pi)(s, a) \\ V_h^*(s) &= \max_{a' \in \mathcal{A}} Q_h^*(s, a'), & Q_h^*(s, a) &= (r_h + \mathbb{P}_h V_{h+1}^*)(s, a) \end{aligned} \quad (4)$$

We say that a policy π is ϵ -optimal if $V_1^\pi(s) \geq V_1^*(s) - \epsilon$ for all $s \in \mathcal{S}$. Suppose an agent interacts with the MDP for a total of K episodes. The total number of steps is $T := HK$. The agent selects a policy π^k at the start of each episode $k \in [K]$, receives an initial state s_1^k drawn from the initial state distribution μ , and uses π^k to generate all actions in episode k . We define the cumulative regret for the agent as:

$$\text{Regret}(K) := \sum_{k=1}^K \left[V_1^*(s_1^k) - V_1^{\pi^k}(s_1^k) \right] \quad (5)$$

²For any integer $n \in \mathbb{Z}^+$, we define $[n] := \{1, 2, \dots, n\}$.

We restrict the transition kernel \mathbb{P} such that the state transitions are governed by $S_{h+1} = f(S_h, A_h) + W_h$, where $f : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is a deterministic function and W_h are random variables with bounded support $\{0, 1, \dots, W\}$ (i.e., $|W_h| \leq W$) generated independently of states and actions. An approximation \hat{f} of f is a function $\mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ that satisfies $\|\hat{f} - f\|_\infty = \sup_{(s, a)} |\hat{f}(s, a) - f(s, a)| \leq \zeta/2$.

Main Results

In this section, we present our main algorithm and its formal sample complexity guarantees. As we will see shortly, our algorithm is able to achieve \sqrt{T} regret without dependency in S and A when f is known ($\zeta = 0$). In the presence of persistent approximation errors, the algorithm accumulates additional regret that is upper-bounded by a term of the form $\mathcal{O}(L\zeta HT)$. In the case that approximation errors are nonzero but asymptotically diminishing, the linear term can be suppressed and \sqrt{T} scaling of regret is restored.

Algorithm

We begin by providing the intuition behind our algorithm. The state transitions in Equation 1 implies that for all $s', s_1, s_2 \in \mathcal{S}, a_1, a_2 \in \mathcal{A}$, and $h \in [H]$, we have:

$$\mathbb{P}_h(s' | s_1, a_1) = \mathbb{P}_h(s' | s_2, a_2) \text{ if } f(s_1, a_1) = f(s_2, a_2) \quad (6)$$

Thus, if one observes a transition from (s_h, a_h) to (s_{h+1}) , the structure of the transition probabilities suggests that the Q-learning update can be extended to all $(s, a) \in \mathcal{S} \times \mathcal{A}$ pairs according to:

$$\begin{aligned} \hat{Q}_h(s, a) &\leftarrow (1 - \alpha)\hat{Q}_h(s, a) + \\ &\alpha \left[r_h(s, a) + \hat{V}_{h+1}(f(s, a) + w_h) \right] \end{aligned} \quad (7)$$

where $w_h = s_{h+1} - f(s_h, a_h)$ and α is a learning rate. One interpretation of the above scheme is that f is used to simulate transitions for state-action pairs not vided in the real trajectories. We contrast this with the typical asynchronous Q-learning that *only* update (s_h, a_h) :

$$\hat{Q}_h(s, a) \leftarrow \begin{cases} (1 - \alpha)\hat{Q}_h(s, a) + \\ \alpha [r_h(s, a) + \hat{V}_{h+1}(s_{h+1})], & \text{if } (s, a) = (s_h, a_h) \\ \hat{Q}_h(s, a), & \text{otherwise} \end{cases} \quad (8)$$

In Algorithm 1, we present an optimistic implementation of the previous scheme, where bonuses are added to the Q-update to incentivize exploration. The exact bonus values will be given in the formal results as they depend on the context. Moreover, there is a specific choice for the learning rate, and it should be set to $\alpha_t = \frac{H+1}{H+t}$. We will discuss the importance of this choice in the next section as well. We also note that all the value functions in Algorithm 1 can be updated in place, and the dependency on k (i.e., $Q^{k+1} \leftarrow Q^k$) is only for clarity of the analysis.

While the updates carried by Algorithm 1 might at first seem similar to synchronous Q-learning (Li et al. 2021a), there are fundamental differences. In synchronous Q-learning, the algorithm would receive independent samples for every state-action pair, and no cost (i.e., regret) is incurred during this process. On the other hand, our algorithm only receives samples from the trajectory of its greedy policy and accumulates regret while it explores. Finally, our algorithm can be implemented in a model-based fashion, with explicit parameterization of transition probabilities. This could enable more complex transition structures than the one in Equation 6 to be encoded in the Q-updates, but it is unnecessary in our case, so we choose the model-free implementation for its computational advantages.

Sample Complexity

In general, we need to assume some regularities in the value functions for learning with a noisy estimator to be feasible. In our analysis, we assume Lipschitz continuity. That is, for all $s_1, s_2 \in \mathcal{S}$ and $h \in [H]$, we have³:

$$|V_h^*(s_1) - V_h^*(s_2)| \leq L|s_1 - s_2|. \quad (9)$$

In the following we assume that Algorithm 1 is run with a noisy estimate \hat{f} that satisfies:

$$\|\hat{f} - f\|_\infty \leq \zeta/2 \quad (10)$$

where ζ is known. With these, we are ready to state our formal regret guarantee.

Theorem 1. *For any $p \in (0, 1)$, let the bonus term applied for all Q-learning updates during the k -th episode be $b_k = c\sqrt{\frac{H^3 \log(SAH/p)}{k}} + L\zeta$. Then with probability $1-p$, the total regret of Algorithm 1 when applied to the class of MDPs in Equation 1, is at most $\mathcal{O}(\sqrt{H^6 T \iota} + L\zeta HT)$, where $c > 0$ is an absolute constant and $\iota = \log(SAH/p) \log K$.*

A few comments are in order. The theorem implies that when $\zeta = 0$, we are able to achieve \sqrt{T} regret with only a logarithmic dependency on the cardinalities of states and actions. When $\zeta > 0$, a linear ζT term in the regret is unavoidable due to the persisting bias from using \hat{f} instead of the true function f (see, e.g., Jin et al. (2020) for similar results under model mismatch). A key observation is that our

³Note that $L = H$ holds trivially for any MDP in our setting; nonetheless, one would hope that $L \ll \mathcal{O}(H)$. An upper bound on L can be derived from Lipschitz constants of \hat{f} and r without access to the optimal value functions (Hinderer 2005).

algorithm does not amplify this term by any factor that depends on the number of states and actions. Besides regret, our algorithm also enjoys direct guarantees on its policies.

Theorem 2. *Under the same conditions of Theorem 1, Algorithm 1 will generate a policy that is $(\epsilon + \mathcal{O}(L\zeta H^2))$ -optimal with probability $1-p$, when the number of episodes K is at least $\mathcal{O}\left(\frac{H^7 \iota}{\epsilon^2}\right)$, where $\iota = \log(SAH/p)$.*

Similar to Theorem 1, the policy converges to the optimal policy when $\zeta = 0$, and to a sub-optimal policy when $\zeta > 0$. Further, we believe that the dependency on H is not sharp and can be improved (see, the Discussion section). It is worth noting that Theorem 2 does not rely on heuristic translations of regret to best policy identification. These translations typically have suboptimal $1/p^2$ dependency and only hold with constant probability (Ménard et al. 2021). Whereas, Theorem 2 has the optimal $\log(1/p)$ scaling and holds with high probability.

Finally, we state the results for the case where the estimates \hat{f} can be improved over time in an online fashion.

Theorem 3. *Suppose that there exists an online estimator that generates a sequence of functions $\{\hat{f}_i\}_{i=1}^K$ such that $\|\hat{f}_i - f\|_\infty \leq \mathcal{O}(\sqrt{d/i})$ holds uniformly with high probability, then the regret of using such estimators with Algorithm 1 is at most $\tilde{\mathcal{O}}(\sqrt{H^6 T} + L\sqrt{HdT})$.*

In general, without any assumptions on f , the parameter d could scale with $\mathcal{O}(SA)$, and the results in Theorem 3 would match the worst-case bounds in terms of the scaling with the number of states and actions. On the other hand, if f is parameterized with few parameters or belongs to a family of functions where efficient online estimators exist, the theorem implies that the sample complexity is tied directly to the complexity of learning f . An important point to mention is that the online estimator needs to work from the data generated by the greedy policies $\{\pi^i\}_{i=1}^K$. If the $\mathcal{O}(K)$ switching of policies is too fast for the online estimator, one could exponentially reduce the switching of policies without any large impact on the regret (see, e.g., the work of Bai et al. (2019) for a $\mathcal{O}(\log K)$ switching and the work of Qiao et al. (2022) for a $\mathcal{O}(\log \log K)$ switching).

Proofs Sketch

In this section, we provide the intuition behind the proof of our results. For clarity, we will focus on the noiseless case $\zeta = 0$. The full proofs for the general case $\zeta > 0$ are provided in the archival version of this paper. We use (s_h^k, a_h^k, w_h^k) to denote the actual state, action, and randomness observed or chosen at step h of episode k . In addition, we use π_h^k to denote the greedy policy with respect to Q_h^k . Notice that $a_h^k = \pi_h^k(s_h^k)$. Further, we define an empirical transition operator at step h of episode k as $[\hat{\mathbb{P}}_h^k V](s, a) := V(f(s, a) + w_h^k)$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$. Moreover, we use $c > 0$ to denote an absolute constant and $\iota := \log(SAH/p)$ to denote a log factor.

$(Q_h^k - Q_h^*)$ Decomposition

To start, we define useful quantities for the learning rate that will help us expand the updates carried out by Algorithm 1:

$$\begin{aligned} \alpha_t^0 &:= (1 - \alpha_1) \cdots (1 - \alpha_t) = \prod_{j=1}^t (1 - \alpha_j) \\ \alpha_t^i &:= \alpha_i (1 - \alpha_{i+1}) \cdots (1 - \alpha_t) = \alpha_i \prod_{j=i+1}^t (1 - \alpha_j) \end{aligned} \quad (11)$$

α_t^0 can be understood as the weight of the Q-value initialization in the current estimate of Q-values after t total updates, while α_t^i as the weight of the i th update. With these definitions, we have for $t = k - 1 \geq 1$:

$$\begin{aligned} (Q_h^k - Q_h^*)(s, a) &= \sum_{i=1}^t \alpha_t^i \left[(V_{h+1}^i - V_{h+1}^*)(f(s, a) + w_h^i) \right. \\ &\quad \left. + (\hat{\mathbb{P}}_h^i - \mathbb{P}_h) V_{h+1}^*(s, a) + b_i \right] \end{aligned} \quad (12)$$

There are three sources of errors in the previous equation: The bootstrapping with V_{h+1}^i instead of V_{h+1}^* , using samples $\hat{\mathbb{P}}_h^i$ to approximate the true expectation \mathbb{P}_h , and finally the bias due to adding optimism bonuses. Notice that, from concentration (Dubhashi and Panconesi 2009), we know that the variance of $(\hat{\mathbb{P}}_h^i - \mathbb{P}_h)$ will scale with $\mathcal{O}(1/\sqrt{i})$ and the bonuses should be designed to have the same scaling. Further, $(V_{h+1}^i - V_{h+1}^*)$ is related to $(Q_{h+1}^i - Q_{h+1}^*)$, which would allow us to accumulate the errors over h by recursion.

A key challenge in model-free Q-learning is controlling the propagation of errors over the H steps. To illustrate this point, consider if we used a typical learning rate of $\eta_t = 1/t$. In this case, we have $\eta_t^i := (1 - \eta_1) \cdots (1 - \eta_t) = 1/t$ and $\sum_{i=1}^t \frac{\eta_t^i}{\sqrt{i}} \approx \sqrt{\frac{\log t}{t}}$. If we then do a recursion over h we would end up with $(Q_1^k - Q_1^*) \approx \frac{(\log k)^{H/2}}{\sqrt{k}}$, which is exponential in H . The following lemma, which is a fine-grained version of the one shown in (Jin et al. 2018), shows that we can avoid this exponential scaling with the rescaled linear rate $\alpha_t = \frac{H+1}{H+t}$.

Lemma 1. *The following holds for α_t^i :*

$$\frac{1}{t^a} \leq \sum_{i=1}^t \frac{\alpha_t^i}{i^a} \leq \frac{1 + \frac{1}{H}}{t^a} \quad \text{for every } t \geq 1 \text{ and } a \in \left[\frac{1}{2}, 1\right]$$

Convergence of On-Policy Errors

Then, using Azuma-Hoeffding inequality, we choose b_t such that the following holds simultaneously for all $(s, a, h, k) \in \mathcal{S} \times \mathcal{A} \times [H] \times \{2, \dots, K\}$ with high probability:

$$0 \leq \sum_{i=1}^t \alpha_t^i \left[(\hat{\mathbb{P}}_h^i - \mathbb{P}_h) V_{h+1}^*(s, a) + b_i \right] \leq 2c \sqrt{\frac{H^3 \iota}{t}} \quad (13)$$

Now, starting from Equation 12 and equipped with Lemma 1 and the upper bound in Equation 13, we construct a careful recursive argument that shows a convergence of the Q-value on-policy (i.e., for the actions chosen by π^k) errors:

Lemma 2. *Let $\{P_h\}_{h=1}^H$ be a sequence that satisfy the following recursive relationship:*

$$P_h = 4 \left(1 + \frac{1}{H}\right) c \sqrt{H^3 \iota} + \left(1 + \frac{1}{H}\right) P_{h+1}$$

where $P_{H+1} := 0$. Then, for any $p \in (0, 1)$, letting $b_t = c \sqrt{H^3 \iota / t}$, the following holds simultaneously for all $(s, h, k) \in \mathcal{S} \times [H] \times \{2, \dots, K\}$ with at least $1 - p$ probability:

$$0 \leq (V_h^k - V_h^*)(s) \leq (Q_h^k - Q_h^*)(s, \pi_h^k(s)) \leq \frac{P_h}{\sqrt{t}}$$

where $t = k - 1$.

Lemma 2 states that the value function estimates are optimistic and converge with a $1/\sqrt{k}$ rate. Importantly, the statement holds uniformly for all $s \in \mathcal{S}$. Note that the error in Q_h^1 is determined by the initialization, and thus, is excluded from the lemma. Finally, the elements $\{P_h\}_{h=1}^H$ are upper bounded by a function that is polynomial⁴ in H and and log-polynomial in S and A :

$$P_h \leq 4(1 + H - h) c \sqrt{H^3 \iota} \left(1 + \frac{1}{H}\right)^{1+H-h} \quad (14)$$

We note that the argument presented here is different than the one used in previous literature (e.g., in Jin et al. (2018) or Li et al. (2021b)), where the propagation of errors over the H steps is controlled after summing over K and using different property of α_t^i . We cannot apply the same argument here as it requires that the Q-values are updated only for a single trajectory per episode, while in our case all the Q-values are updates per episode. Our argument exploits the uniformity of the number of updates for every (s, a) pair to show convergence before summing over K .

Regret and Best Policy Identification

Now, we bound the suboptimality gap $(V_h^* - V_h^{\pi^k})$. Following the standard techniques for optimum-based methods, we start by showing:

$$\begin{aligned} (V_h^* - V_h^{\pi^k})(s) &\leq (V_h^k - V_h^{\pi^k})(s) \\ &\leq (Q_h^k - Q_h^*)(s, \pi_h^k(s)) \\ &\quad + \mathbb{P}_h(V_{h+1}^* - V_{h+1}^{\pi^k})(s, \pi_h^k(s)) \end{aligned} \quad (15)$$

The first equality follows as $V_h^k \geq V_h^*$ and the second from Bellman Equations 4 after adding and subtracting Q_h^* . Notice that the first term in Equation 15 is upper bounded by Lemma 2 and the second term is an expectation over suboptimality gap in the next step $h + 1$. Thus, we can establish by recursion that the following holds with high probability:

$$(V_1^* - V_1^{\pi^k})(s) \leq \sum_{h=1}^H \frac{P_h}{\sqrt{k-1}} \leq \mathcal{O} \left(\sqrt{\frac{H^7 \iota}{k-1}} \right) \quad (16)$$

This is sufficient to show convergence to optimal policies, and regret follows immediately by summing over K and applying Jensen's inequality. The proofs for $\zeta > 0$ follow the same recipe presented here but are more elaborate as the bonus design needs to guarantee that the estimates Q_h^k are optimistic even with the errors in \hat{f} .

⁴The elementary fact $(1 + \frac{1}{x})^x \leq e$ is helpful here.

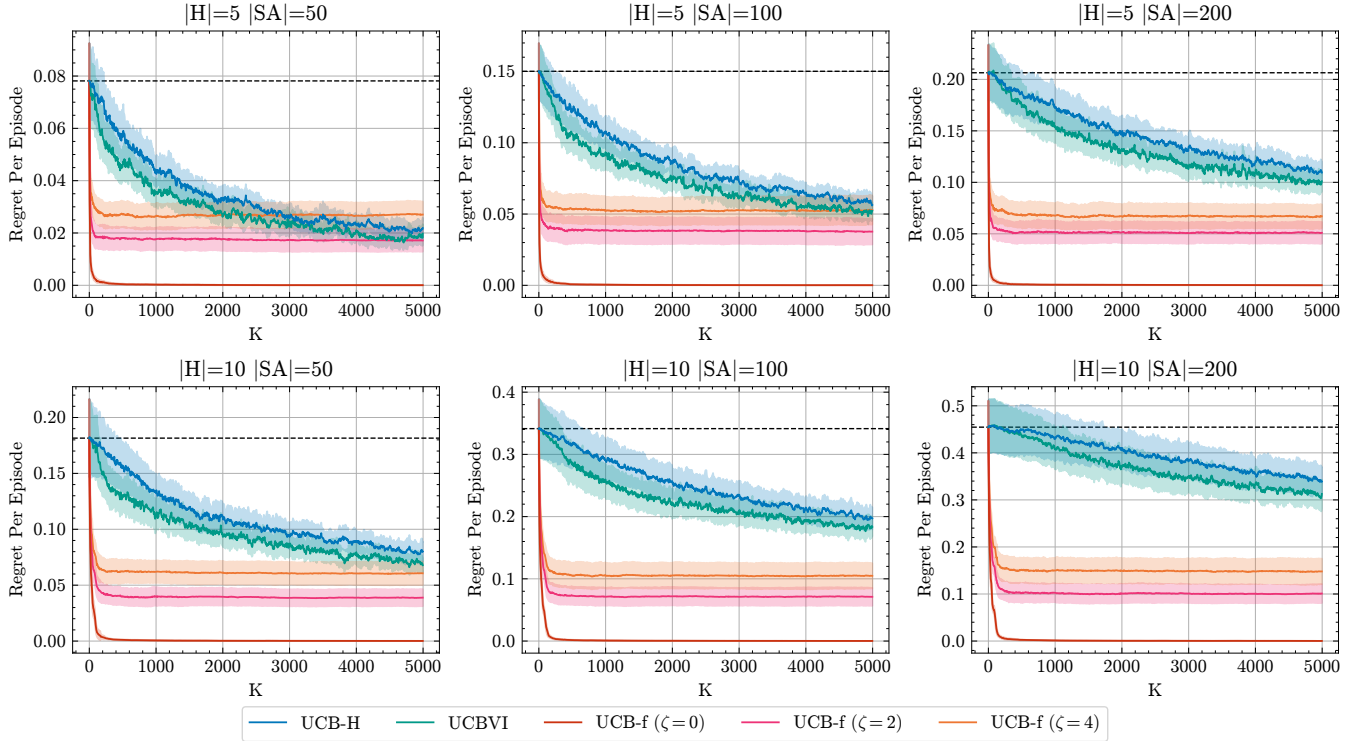


Figure 1: $(V_1^* - V_1^k)$ for different Q-learning algorithms on randomly generated MDPs. Each curve is an average of 50 simulations. The dashed lines are the performance of the greedy policy with respect to the rewards.

Experiment

In this section, we empirically compare our algorithm with multiple problem-agnostic Q-learning algorithms from the literature. In particular, we compare with the model-free UCB-H method of Jin et al. (2018) and the model-based UCBVI method of Azar, Osband, and Munos (2017). When we test Algorithm 1, we either provide it the true function f (to test $\zeta = 0$) or we corrupt f with random noise (to test $\zeta > 0$) before we give it to the algorithm. Inside a single run, the approximate function \hat{f} is kept fixed across the time T .

Environments. We compare the algorithms on randomly generated MDPs with varying cardinalities. Further, as our method and UCBVI require the rewards to be known, we choose to make $r_h(s, a) = r_h(s)$. This ensures that there is no utility in being greedy with respect to the rewards, and the optimal policy is the one that controls the system toward a desired region in the state space.

Bonuses. As multiplicative constants in the bonuses can drastically change empirical performances, we unify the bonus design across the different methods. Specifically, we use the following bonuses in the empirical simulations:

$$\beta_1 = c\sqrt{\frac{H^2}{N_h^k(s, a)}}, \quad \beta_2 = c\sqrt{\frac{H^2}{k}} + c\zeta L \quad (17)$$

UCB-H and UCBVI use β_1 , and our method uses β_2 . $N_h^k(s, a)$ is the state-action visitation counter used by the

problem-agnostic methods. Further, as our goal is to study qualitative behavior, we choose to optimize c by setting it $c = 0.05$. This would accelerate the convergence of all algorithms, especially UCB-H and UCBVI as their visitation counter is distributed over the state-action pairs.

Results. In Figure 1, we plot the regret per episode (i.e., the suboptimality gap) for $S = 25$, $A = \{2, 4, 8\}$, $H = \{5, 10\}$, $W = 5$, $\zeta = \{0, 2, 4\}$, $L = 0.25$, and $K \in [5000]$. Each curve is an average of 50 simulations, and the shadings represent a width of one standard deviation. Moreover, the dashed lines represent the performance of the greedy policy with respect to the rewards. Because of our choice for r_h , the rewards-greedy policy is no better than the random policies that the algorithms start with.

As we can see from Figure 1, Algorithm 1 always converge to the optimal policy (when $\zeta = 0$) or to a suboptimal policy (when $\zeta > 0$). Moreover, the time until convergence is not affected by increasing the cardinalities of states and actions. This is in contrast to UCB-H and UCBVI, where it is evident from the figure that their convergence is affected by S and A , as they take longer to outperform the case of $\zeta > 0$ when the cardinalities increase. Finally, we note that the variance in the figure when $\zeta > 0$ is not from within run variations but is from the randomness of generating a new MDP in each simulation. The code to reproduce these results is publicly available⁵.

⁵<https://github.com/meshal-h/ucb-f>

Discussion

In this section, we elaborate on some aspects of our algorithm and methodology and discuss potential extensions of our study.

Dependency on the Horizon. We believe that the scaling with the horizon in our results is not optimal, and there is an extra H factor that is an artifact of the analysis. Further, sharpening the horizon dependency by a \sqrt{H} factor is possible with a more elaborate bonus design that relies on Bernstein inequality. Moreover, variance reduction techniques were shown to yield another \sqrt{H} factor improvement (Li et al. 2021b). We note that using these techniques usually adds lower-order terms to the regret (see, Table 1). As our focus in this paper is on the dependency on S and A , we leave applying these ideas to our algorithm as future directions.

Value Function Assumptions. As shown by Jiang (2018), without further assumptions, no algorithm can always achieve sample complexity independent of the number of states and actions, even if it has access to a transition kernel that is only corrupted at one state-action pair. We note that our assumptions in Equation 9 only concern the optimal value functions $V_h^*(s)$, and no smoothness is required for the value function generated by non-optimal policies. For future analysis, we could consider more local regularity assumptions on the optimal value function or adapt some of the metrics used by the problem-dependent bounds discussed in the literature review.

Noise in the Model. Not all noise structures in \hat{f} are equivalent. In fact, some can be completely benign; The current analysis suggests that if the approximate function differs from the true function by a constant (i.e., $\hat{f} = f + c$ for all state-action pairs), then there is no linear term in the regret when we use \hat{f} (owing to the cancellation of c that would happen between Line 7 and 10 in Algorithm 1). This would suggest that the use of $\|\hat{f} - f\|_\infty$ in the analysis can be refined, which will also provide relaxations to the conditions required for the online estimator in Theorem 3.

Systems that Fit the Additive Disturbance Model. Perhaps the simplest example is inventory control $S_{h+1} = S_h + A_h - W_h$, where S_h is the inventory level, A_h is the amount of goods ordered, and W_h is the demand observed after taking action A_h . In this model, the state can be negative to represent backlogged demand. We note that although we restricted the states, actions, and disturbances to be positive, there is no loss of generality and our analysis extends to the case where negative values are allowed. The additive disturbance model in our study is flexible and captures systems such as inventory control and many other similar problems that arise in operations research. Further, as the additive disturbance model is prevalent in control problems with continuous spaces, we discuss the implications of our results on these systems in the following subsection.

Continuous Spaces. With the appropriate topological assumptions, Algorithm 1 can be applied to continuous prob-

lems after discretization and inheriting the metric (to measure $f - \hat{f}$) from the continuous space (see, e.g., Shah and Xie (2018) for an example of Q-learning with discretization). Our results would suggest that we will be able to learn an optimal policy (up to Δ discretization errors) when $\zeta = 0$ in a number of samples that do not scale with state and action cardinalities, which is $\mathcal{O}(\frac{1}{\Delta^2})$ in this case. A caveat here is that computation will also scale with $\mathcal{O}(\frac{1}{\Delta^2})$ (cf. Table 1). Still, one might be able to modify Algorithm 1 to make it more practical under sufficient smoothness conditions. A more interesting line of future work is to see if our methodology can be extended to the value function approximation setting, which is more appropriate in continuous spaces.

Conclusion

In this paper, we studied the sample complexity of online RL for systems that evolve according to an additive disturbance model of the form $S_{h+1} = f(S_h, A_h) + W_h$. We developed an optimistic Q-learning algorithm that achieves \sqrt{T} regret without dependency in S and A when f is known. Further, if an asymptotically accurate online estimator for f exists, we also achieve \sqrt{T} regret that depends only on the complexity of learning f . Our methodology accomplishes all of that without explicitly modeling transition probabilities, and our algorithm enjoys the same memory complexity as model-free methods. Future research directions include extending our algorithm and analysis to infinite horizon MDPs and problems involving value function approximations.

Acknowledgments

Meshal Alharbi is supported by a scholarship from King Abdulaziz City for Science and Technology (KACST).

References

- Ayoub, A.; Jia, Z.; Szepesvari, C.; Wang, M.; and Yang, L. 2020. Model-Based Reinforcement Learning with Value-Targeted Regression. In *International Conference on Machine Learning*, 463–474. PMLR.
- Azar, M. G.; Osband, I.; and Munos, R. 2017. Minimax Regret Bounds for Reinforcement Learning. In *International Conference on Machine Learning*, 263–272. PMLR.
- Bai, Y.; Xie, T.; Jiang, N.; and Wang, Y.-X. 2019. Provably efficient q-learning with low switching cost. *Advances in Neural Information Processing Systems*, 32.
- Bertsekas, D. 2017. *Dynamic Programming and Optimal Control*, volume 4. Athena scientific.
- Bhattacharyya, S. 2017. Robust Control Under Parametric Uncertainty: An Overview and Recent Results. *Annual Reviews in Control*, 44: 45–77.
- Bousmalis, K.; Irpan, A.; Wohlhart, P.; Bai, Y.; Kelcey, M.; Kalakrishnan, M.; Downs, L.; Ibarz, J.; Pastor, P.; Konolige, K.; et al. 2018. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE international conference on robotics and automation (ICRA)*, 4243–4250. IEEE.

- Cassel, A.; Cohen, A.; and Koren, T. 2020. Logarithmic regret for learning linear quadratic regulators efficiently. In *International Conference on Machine Learning*, 1328–1337. PMLR.
- Chen, X.; Hu, J.; Jin, C.; Li, L.; and Wang, L. 2022. Understanding Domain Randomization for Sim-to-Real Transfer. In *International Conference on Learning Representations*.
- Dann, C.; Lattimore, T.; and Brunskill, E. 2017. Unifying PAC and Regret: Uniform PAC Bounds for Episodic Reinforcement Learning. *Advances in Neural Information Processing Systems*, 30.
- Dean, S.; Mania, H.; Matni, N.; Recht, B.; and Tu, S. 2018. Regret bounds for robust adaptive control of the linear quadratic regulator. *Advances in Neural Information Processing Systems*, 31.
- Domingues, O. D.; Ménard, P.; Kaufmann, E.; and Valko, M. 2021. Episodic Reinforcement Learning in Finite MDPs: Minimax Lower Bounds Revisited. In *Algorithmic Learning Theory*, 578–598. PMLR.
- Dubhashi, D. P.; and Panconesi, A. 2009. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press.
- Feng, F.; Yin, W.; and Yang, L. F. 2019. How Does an Approximate Model Help in Reinforcement Learning? *arXiv preprint arXiv:1912.02986*.
- Hinderer, K. 2005. Lipschitz Continuity of Value Functions in Markovian Decision Processes. *Mathematical Methods of Operations Research*, 62: 3–22.
- Jiang, N. 2018. PAC Reinforcement Learning with an Imperfect Model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.
- Jin, C.; Allen-Zhu, Z.; Bubeck, S.; and Jordan, M. I. 2018. Is Q-learning Provably Efficient? *Advances in neural information processing systems*, 31.
- Jin, C.; Yang, Z.; Wang, Z.; and Jordan, M. I. 2020. Provably efficient reinforcement learning with linear function approximation. In *Conference on Learning Theory*, 2137–2143. PMLR.
- Li, G.; Cai, C.; Chen, Y.; Gu, Y.; Wei, Y.; and Chi, Y. 2021a. Tightening the dependence on horizon in the sample complexity of Q-learning. In *International Conference on Machine Learning*, 6296–6306. PMLR.
- Li, G.; Shi, L.; Chen, Y.; Gu, Y.; and Chi, Y. 2021b. Breaking the Sample Complexity Barrier to Regret-Optimal Model-Free Reinforcement Learning. *Advances in Neural Information Processing Systems*, 34: 17762–17776.
- Lu, X.; Van Roy, B.; Dwaracherla, V.; Ibrahimi, M.; Osband, I.; Wen, Z.; et al. 2023. Reinforcement Learning, Bit by Bit. *Foundations and Trends® in Machine Learning*, 16(6): 733–865.
- Ménard, P.; Domingues, O. D.; Jonsson, A.; Kaufmann, E.; Leurent, E.; and Valko, M. 2021. Fast Active Learning for Pure Exploration in Reinforcement Learning. In *International Conference on Machine Learning*, 7599–7608. PMLR.
- Nagabandi, A.; Kahn, G.; Fearing, R. S.; and Levine, S. 2018. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *2018 IEEE international conference on robotics and automation (ICRA)*, 7559–7566. IEEE.
- Qiao, D.; Yin, M.; Min, M.; and Wang, Y.-X. 2022. Sample-efficient reinforcement learning with loglog (t) switching cost. In *International Conference on Machine Learning*, 18031–18061. PMLR.
- Schrittwieser, J.; Hubert, T.; Mandhane, A.; Barekatin, M.; Antonoglou, I.; and Silver, D. 2021. Online and offline reinforcement learning by planning with a learned model. *Advances in Neural Information Processing Systems*, 34: 27580–27591.
- Shah, D.; and Xie, Q. 2018. Q-learning with nearest neighbors. *Advances in Neural Information Processing Systems*, 31.
- Taylor, A. J.; Dorobantu, V. D.; Dean, S.; Recht, B.; Yue, Y.; and Ames, A. D. 2021. Towards robust data-driven control synthesis for nonlinear systems with actuation uncertainty. In *2021 60th IEEE Conference on Decision and Control (CDC)*, 6469–6476. IEEE.
- Vázquez-Canteli, J. R.; and Nagy, Z. 2019. Reinforcement Learning for Demand Response: A review of Algorithms and Modeling Techniques. *Applied energy*, 235: 1072–1089.
- Wagenmaker, A. J.; Simchowitz, M.; and Jamieson, K. 2022. Beyond No Regret: Instance-Dependent PAC Reinforcement Learning. In *Conference on Learning Theory*, 358–418. PMLR.
- Yang, K.; Yang, L.; and Du, S. 2021. Q-Learning with Logarithmic Regret. In *International Conference on Artificial Intelligence and Statistics*, 1576–1584. PMLR.
- Zanette, A.; and Brunskill, E. 2019. Tighter Problem-Dependent Regret Bounds in Reinforcement Learning without Domain Knowledge using Value Function Bounds. In *International Conference on Machine Learning*, 7304–7312. PMLR.
- Zhang, M.; Vikram, S.; Smith, L.; Abbeel, P.; Johnson, M.; and Levine, S. 2019. Solar: Deep structured representations for model-based reinforcement learning. In *International conference on machine learning*, 7444–7453. PMLR.
- Zhao, W.; Queralta, J. P.; and Westerlund, T. 2020. Sim-to-Real Transfer in Deep Reinforcement Learning for Robotics: A Survey. In *2020 IEEE symposium series on computational intelligence (SSCI)*, 737–744. IEEE.