Kernel-Based Reinforcement Learning: A Finite-Time Analysis

Omar D. Domingues [1 2] **Pierre Ménard** [3] **Matteo Pirotta** [4] **Emilie Kaufmann** [1 5] **Michal Valko** [1 5 6]

Abstract

We consider the exploration-exploitation dilemma in finite-horizon reinforcement learning problems whose state-action space is endowed with a metric. We introduce Kernel-UCBVI, a model-based optimistic algorithm that leverages the smoothness of the MDP and a non-parametric kernel estimator of the rewards and transitions to efficiently balance exploration and exploitation. For problems with K episodes and horizon H, we provide a regret bound of $\widetilde{O}\left(H^3 K^{\frac{2d}{2d+1}}\right)$, where d is the covering dimension of the joint state-action space. This is the first regret bound for kernel-based RL using smoothing kernels, which requires very weak assumptions on the MDP and has been previously applied to a wide range of tasks. We empirically validate our approach in continuous MDPs with sparse rewards.

1. Introduction

Reinforcement learning (RL) is a learning paradigm in which an agent interacts with an environment by taking actions and receiving rewards. At each time step t, the environment is characterized by a state variable $x_t \in \mathcal{X}$, which is observed by the agent and influenced by its actions $a_t \in \mathcal{A}$. In this work, we consider the online learning problem where the agent has to learn how to act optimally by interacting with an unknown environment. To learn efficiently, the agent has to trade-off exploration to gather information about the environment and exploitation to act optimally with respect to the current knowledge. The performance of the agent is measured by the *regret*, i.e., the difference between the rewards that would be gathered by an optimal agent and the rewards obtained by the agent. This problem has been extensively studied for Markov Decision Processes (MDPs) with finite state-action space. *Optimism in the face of uncertainty* (OFU, Jaksch et al. 2010) and *Thompson Sampling* (Strens, 2000; Osband et al., 2013) principles have been used to design algorithms with sublinear regret. However, the guarantees for these approaches cannot be naturally extended to an arbitrarily large state-action space since the regret depends on the number of states and actions. When the state-action space is continuous, additional structure in MDP is required to efficiently solve the exploration-exploitation dilemma.

In this paper, we focus on the online learning problem in MDPs with large or continuous state-action spaces. We suppose that the state-action set $\mathcal{X} \times \mathcal{A}$ is equipped with a known *metric*. For instance, this is typically the case in continuous control problems in which the state space is a subset of $\mathbb{R}^d$ equipped with the Euclidean metric. As shown by Ormoneit & Sen (2002) and Barreto et al. (2016), smoothing-kernel approaches converge asymptotically to an optimal policy and perform well empirically in a wide range of continuous MDPs. In this paper, we tackle the problem of *exploration* in such approaches, by proposing an *optimistic* algorithm based on smoothing-kernel estimators of the reward and transition functions of the underlying MDP. The advantages of this approach are: (i) it requires weak assumptions on the MDP, (ii) it allows us to easily provide expert knowledge to the algorithm through kernel design, and (iii) it applies to problems with possibly infinite states without relying on any kind of discretization.

Related work Kernel-based RL (KBRL) using smoothing kernels has been initially proposed by Ormoneit & Sen (2002), who analyzed the algorithm assuming that transitions are generated from *independent* samples, and provide *asymptotic* convergence guarantees. Barreto et al. (2016) propose a stochastic factorization technique to reduce the computational complexity of KBRL. In this paper, we provide a modification of KBRL that collects data *online* and for which we prove *finite-time* regret guarantees under weak conditions on the MDP. Under stronger conditions, that use positive-define kernels defining reproducing kernel Hilbert spaces (RKHS) or Gaussian Processes, regret bounds are provided by Chowdhury & Gopalan (2019), Chowdhury & Oliveira (2020) and Yang et al. (2020).

Regret minimization in finite MDPs has been extensively studied both in model-based and model-free settings. While

[1]Inria Lille [2]Université de Lille [3]Otto von Guericke University [4]Facebook AI Research, Paris [5]CNRS [6]DeepMind Paris. Correspondence to: Omar D. Domingues <omar.darwiche-domingues@inria.fr>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

model-based algorithms (Jaksch et al., 2010; Azar et al., 2017; Zanette & Brunskill, 2019) use the estimated rewards and transitions to perform planning at each episode, model-free algorithms (Jin et al., 2018) directly build an estimate of the optimal Q-function that is updated incrementally.

For MDPs with continuous state-action space, the sample complexity (Kakade et al., 2003; Kearns & Singh, 2002; Lattimore et al., 2013; Pazis & Parr, 2013) or regret have been studied under structural assumptions. Regarding regret minimization, a standard assumption is that rewards and transitions are Lipschitz continuous. Ortner & Ryabko (2012) studied this problem in average reward problems. They combined the ideas of UCRL2 (Jaksch et al., 2010) and uniform discretization, proving a regret bound of $\widetilde{\mathcal{O}}\left(T^{\frac{2d+1}{2d+2}}\right)$ for a learning horizon T in d-dimensional state spaces. This work was later extended by Lakshmanan et al. (2015) to use a kernel density estimator instead of a frequency estimator for each region of the fixed discretization. For each *discrete* region $I(x)$, the density $p(\cdot|I(x),a)$ of the transition kernel is computed through kernel density estimation. The granularity of the discretization is selected in advance based on the properties of the MDP and the learning horizon T. As a result, they improve upon the bound of Ortner & Ryabko (2012), but require the transition kernels to have densities that are κ times differentiable.[1] However, these two algorithms rely on an intractable optimization problem for finding an optimistic MDP. Jian et al. (2019) solve this issue by providing an algorithm that uses exploration bonuses, but they still rely on a uniform discretization of the state space. Ok et al. (2018) studied the asymptotic regret in Lipschitz MDPs with *finite* state and action spaces, providing a nearly asymptotically optimal algorithm. Their algorithm leverages ideas from asymptotic optimal algorithms in structured bandits (Combes et al., 2017) and tabular RL (Burnetas & Katehakis, 1997), but does not scale to continuous state-action spaces.

Regarding exploration for finite-horizon MDP with continuous state-action space, Ni et al. (2019) present an algorithm for deterministic MDPs with Lipschitz transitions. Assuming that the Q-function is Lipschitz continuous, Song & Sun (2019) provided a model-free algorithm by combining the ideas of tabular optimistic Q-learning (Jin et al., 2018) with uniform discretization, showing a regret bound of $O(H^{\frac{5}{2}}K^{\frac{d+1}{d+2}})$ where d is the covering dimension of the state-action space. This approach was extended by Sinclair et al. (2019) and Touati et al. (2020) to use adaptive partitioning of the state-action space, achieving the same regret bound. Osband & Van Roy (2014) prove a *Bayesian* regret bound in terms of the eluder and Kolmogorov dimension, assuming access to an approximate MDP planner. In addition, there are many results for facing the exploration problem in continuous MDP with *parametric* structure, e.g., linear-quadratic systems (Abbasi-Yadkori & Szepesvári, 2011) or other linearity assumptions (Yang & Wang, 2020; Jin et al., 2020), which are outside the scope of our paper.

[1] For instance, when $d = 1$ and $\kappa \to \infty$, their bound approaches $T^{\frac{2}{3}}$, improving the previous bound of $T^{\frac{3}{4}}$.

Contributions The main contributions of this paper are the following. **(1)** We provide the first regret bound for KBRL, which applies to a wide range of RL tasks with an entirely *data-dependent* approach; **(2)** In order to derive our regret bound, we provide novel concentration inequalities for weighted sums (Lemmas 2 and 3) that permit to build confidence intervals for non-parametric kernel estimators (Propositions 1 and 2) that are of independent interest. **(3)** We show that the regret of model-based algorithms, although having a better empirical performance, seem to suffer from a worse dependence on the state-action dimension d than model-free ones. We discuss the origins of this issue by looking at the regret bounds of tabular algorithms.

2. Setting

Notation For any $j \in \mathbb{Z}_+$, we define $[j] \stackrel{\text{def}}{=} \{1,\dots,j\}$. For a measure P and any function f, let $Pf \stackrel{\text{def}}{=} \int f(y)\mathrm{d}P(y)$. If $P(\cdot|x,a)$ is a measure for all (x,a), we let $Pf(x,a) = P(\cdot|x,a)f = \int f(y)\mathrm{d}P(y|x,a)$.

Markov decision processes Let $\mathcal{X}$ and $\mathcal{A}$ be the sets of states and actions, respectively. We assume that there exists a metric $\rho : (\mathcal{X}\times\mathcal{A})^2 \to \mathbb{R}_{\geq 0}$ on the state-action space and that $(\mathcal{X},\mathcal{T}_{\mathcal{X}})$ is a measurable space with σ-algebra $\mathcal{T}_{\mathcal{X}}$. We consider an episodic Markov decision process (MDP), defined by the tuple $\mathcal{M} \stackrel{\text{def}}{=} (\mathcal{X},\mathcal{A},H,P,r)$ where $H \in \mathbb{Z}_+$ is the length of each episode, $P = \{P_h\}_{h\in[H]}$ is a set of transition kernels from $(\mathcal{X}\times\mathcal{A})\times\mathcal{T}_{\mathcal{X}}$ to $\mathbb{R}_{\geq 0}$, and $r = \{r_h\}_{h\in[H]}$ is a set of reward functions from $\mathcal{X}\times\mathcal{A}$ to $[0,1]$. A policy π is a mapping from $[H]\times\mathcal{X}$ to $\mathcal{A}$, such that $\pi(h,x)$ is the action chosen by π in state x at step h. The Q-value of a policy π for state-action (x,a) at step h is the expected sum of rewards obtained by taking action a in state x at step h and then following the policy π, that is

$$Q_h^\pi(x,a) \stackrel{\text{def}}{=} \mathbb{E}\left[\sum_{h'=h}^{H} r_{h'}(x_{h'},a_{h'})\Big| x_h = x, a_h = a\right],$$

where the expectation is under transitions in the MDP $x_{h'+1} \sim P_{h'}(\cdot|x_{h'},a_{h'})$ and $a_{h'} = \pi(h',x_{h'})$. The value function of policy π at step h is $V_h^\pi(x) = Q_h^\pi(x,\pi(h,x))$. The optimal value functions, defined by $V_h^*(x) \stackrel{\text{def}}{=} \sup_\pi V_h^\pi(x)$ for $h \in [H]$, satisfy the optimal Bellman equations (Puterman, 1994): $V_h^*(x) = \max_{a\in\mathcal{A}} Q_h^*(x,a)$,

where

$$Q_h^*(x,a) \stackrel{\text{def}}{=} r_h(x,a) + \int_{\mathcal{X}} V_{h+1}^*(y) \mathrm{d}P_h(y|x,a)$$

and, by definition, $V_{H+1}^*(x) = 0$ for all $x \in \mathcal{X}$.

Learning problem A reinforcerment learning agent interacts with $\mathcal{M}$ in a sequence of episodes $k \in [K]$ of fixed length H by playing a policy π_k in each episode, where the initial state x_1^k is chosen arbitrarily and revealed to the agent. The learning agent does not know P and r and it selects the policy π_k based on the samples observed over previous episodes. Its performance is measured by the regret $\mathcal{R}(K) \stackrel{\text{def}}{=} \sum_{k=1}^K \left(V_1^*(x_1^k) - V_1^{\pi_k}(x_1^k)\right)$.

We make the following assumptions:

Assumption 1. *The metric ρ is given to the learner. Also, there exists a metric $\rho_{\mathcal{X}}$ on $\mathcal{X}$ and a metric $\rho_{\mathcal{A}}$ on $\mathcal{A}$ such that, for all (x,x',a,a'), $\rho\left[(x,a),(x',a')\right] = \rho_{\mathcal{X}}\left(x,x'\right) + \rho_{\mathcal{A}}\left(a,a'\right)$.*

Assumption 2. *The reward functions are λ_r-Lipschitz and the transition kernels are λ_p-Lipschitz with respect to the 1-Wasserstein distance: $\forall (x,a,x',a')$ and $\forall h \in [H]$,*

$$|r_h(x,a) - r_h(x',a')| \le \lambda_r \rho\left[(x,a),(x',a')\right], \quad \textit{and}$$

$$W_1\left(P_h(\cdot|x,a), P_h(\cdot|x',a')\right) \le \lambda_p \rho\left[(x,a),(x',a')\right]$$

where, for two measures μ and ν, we have $W_1(\mu,\nu) \stackrel{\text{def}}{=} \sup_{f:\mathrm{Lip}(f)\le 1} \int_{\mathcal{X}} f(y)(\mathrm{d}\mu(y) - \mathrm{d}\nu(y))$ and where, for a function $f : \mathcal{X} \to \mathbb{R}$, $\mathrm{Lip}(f)$ denotes its Lipschitz constant w.r.t. $\rho_{\mathcal{X}}$.

To assess the relevance of these assumptions, we show below that they apply to deterministic MDPs with Lipschitz reward and transition functions (whose transition kernels are *not* Lipschitz w.r.t. the total variation distance).

Example 1 (Deterministic MDP in $\mathbb{R}^d$)**.** *Consider an MDP $\mathcal{M}$ with a finite action set, with a compact state space $\mathcal{X} \subset \mathbb{R}^d$, and deterministic transitions $y = f(x,a)$, i.e., $P_h(y|x,a) = \delta_{f(x,a)}(y)$. Let $\rho_{\mathcal{X}}$ be the Euclidean distance on $\mathbb{R}^d$ and $\rho_{\mathcal{A}}(a,a') = 0$ if $a = a'$ and ∞ otherwise. Then, if for all $a \in \mathcal{A}$, $x \mapsto r_h(x,a)$ and $x \mapsto f(x,a)$ are Lipschitz, $\mathcal{M}$ satisfies assumptions 1 and 2.*

Under our assumptions, the optimal Q functions are Lipschitz continuous:

Lemma 1. *Let $L_h \stackrel{\text{def}}{=} \sum_{h'=h}^H \lambda_r \lambda_p^{H-h'}$. Under Assumption 2, for all (x,a,x',a') and for all $h \in [H]$, we have $|Q_h^*(x,a) - Q_h^*(x',a')| \le L_h \rho\left[(x,a),(x',a')\right]$, i.e., the optimal Q-functions are Lipschitz continuous.*

Algorithm 1 Kernel-UCBVI

Input: global parameters $K, H, \delta, \lambda_r, \lambda_p, \sigma, \beta$
initialize data lists $\mathcal{D}_h = \emptyset$ for all $h \in [H]$
for episode $k = 1, \dots, K$ **do**
  get initial state x_1^k
  $Q_h^k = \texttt{optimisticQ}(k, \{\mathcal{D}_h\}_{h\in[H]})$
  for step $h = 1, \dots, H$ **do**
    execute $a_h^k = \mathrm{argmax}_a\, Q_h^k(x_h^k, a)$
    observe reward r_h^k and next state x_{h+1}^k
    add sample $(x_h^k, a_h^k, x_{h+1}^k, r_h^k)$ to $\mathcal{D}_h$
  end for
end for

3. Algorithm

In this section, we present Kernel-UCBVI, a model-based algorithm for exploration in MDPs in metric spaces that employs *kernel smoothing* to estimate the rewards and transitions, for which we derive confidence intervals. Kernel-UCBVI uses exploration bonuses based on these confidence intervals to efficiently balance exploration and exploitation. Our algorithm requires the knowledge of the metric ρ on $\mathcal{X} \times \mathcal{A}$ and of the Lipschitz constants of the rewards and transitions.[2]

3.1. Kernel Function

We leverage the knowledge of the state-action space metric to define the kernel function. Let $u, v \in \mathcal{X} \times \mathcal{A}$. For some function $g : \mathbb{R}_{\ge 0} \to [0,1]$, we define the kernel function as

$$\psi_\sigma(u,v) \stackrel{\text{def}}{=} g\left(\rho\left[u,v\right]/\sigma\right)$$

where σ is the bandwidth parameter that controls the degree of "smoothing" of the kernel. In order to be able to construct valid confidence intervals, we require certain structural properties for g.

Assumption 3. *The function $g : \mathbb{R}_{\ge 0} \to [0,1]$ is differentiable, non-increasing, $g(4) > 0$, and there exists two constants $C_1^g, C_2^g > 0$ that depend only on g such that*

$$g(z) \le C_1^g \exp(-z^2/2) \textit{ and } \sup_z |g'(z)| \le C_2^g.$$

This assumption is trivially verified by the Gaussian kernel $g(z) = \exp(-z^2/2)$. Other examples include the kernels $g(z) = \exp(-|z|^p/2)$ for $p > 2$.

[2] This assumption is standard in previous works in RL (e.g., Ortner & Ryabko, 2012; Sinclair et al., 2019). Theoretically, we could replace the Lipschitz constant L_1 by $\log k$, in each episode k, and our regret bound would have an additive term of order He^{L_1}, since Q_h^k would be optimistic for $\log k \ge L_1$ (see e.g., Reeve et al., 2018). However, this would degrade the performance of the algorithm in practice.

Algorithm 2 optimisticQ

Input: episode k, data $\{\mathcal{D}_h\}_{h\in[H]}$
Initialize $V^k_{H+1}(x) = 0$ for all x
for step $h = H, \dots, 1$ **do**
  // Compute optimistic targets
  for $m = 1, \dots, k-1$ **do**
    $\widetilde{Q}^k_h(x^m_h, a^m_h) = \sum_{s=1}^{k-1} \widetilde{w}^s_h(x^m_h, a^m_h)\left(r^s_h + V^k_{h+1}(x^s_{h+1})\right)$
    $\widetilde{Q}^k_h(x^m_h, a^m_h) = \widetilde{Q}^k_h(x^m_h, a^m_h) + \mathtt{B}^k_h(x^m_h, a^m_h)$
  end for
  // Interpolate the Q function
  $Q^k_h(x,a) = \min_{s\in[k-1]} \left(\widetilde{Q}^k_h(x^s_h, a^s_h) + L_h \rho\left[(x,a),(x^s_h,a^s_h)\right]\right)$
  for $m = 1, \dots, k-1$ **do**
    $V^k_h(x^m_h) = \min\left(H-h+1, \max_{a\in\mathcal{A}} Q^k_h(x^m_h, a)\right)$
  end for
end for
return Q^k_h

3.2. Kernel Estimators and Optimism

In each episode k, Kernel-UCBVI computes an optimistic estimate Q^k_h for all h, which is an upper confidence bound on the optimal Q function Q^*_h, and plays the associated greedy policy. Let $(x^s_h, a^s_h, x^s_{h+1}, r^s_h)$ be the random variables representing the state, the action, the next state and the reward at step h of episode s, respectively. We denote by $\mathcal{D}_h = \left\{(x^s_h, a^s_h, x^s_{h+1}, r^s_h)\right\}_{s\in[k-1]}$ for $h \in [H]$ the samples collected at step h before episode k.

For any (x,a) and $(s,h) \in [K]\times[H]$, we define the *weights* and the *normalized weights* as

$$w^s_h(x,a) \stackrel{\text{def}}{=} \psi_\sigma((x,a),(x^s_h,a^s_h)) \quad \text{and}$$
$$\widetilde{w}^s_h(x,a) \stackrel{\text{def}}{=} \frac{w^s_h(x,a)}{\beta + \sum_{l=1}^{k-1} w^l_h(x,a)},$$

where $\beta > 0$ is a regularization term. These weights are used to compute an estimate of the rewards and transitions for each state-action pair[3]:

$$\widehat{r}^k_h(x,a) \stackrel{\text{def}}{=} \sum_{s=1}^{k-1} \widetilde{w}^s_h(x,a) r^s_h,$$
$$\widehat{P}^k_h(y|x,a) \stackrel{\text{def}}{=} \sum_{s=1}^{k-1} \widetilde{w}^s_h(x,a) \delta_{x^s_{h+1}}(y).$$

As other algorithms using OFU, Kernel-UCBVI computes an optimistic Q-function $\widetilde{Q}^k_h$ through value iteration, a.k.a. backward induction:

$$\widetilde{Q}^k_h(x,a) = \widehat{r}^k_h(x,a) + \widehat{P}^k_h V^k_{h+1}(x,a) + \mathtt{B}^k_h(x,a) \quad (1)$$

where $V^k_{H+1}(x) = 0$ for all $x \in \mathcal{X}$ and $\mathtt{B}^k_h(x,a)$ is an exploration bonus described later. From Lemma 1, the true Q function Q^*_h is L_h-Lipschitz. Computing $\widetilde{Q}^k_h$ for all previously visited state action pairs (x^s_h, a^s_h) for $s \in [k-1]$ permits to define a L_h-*Lipschitz upper confidence bound* and the associated value function:

$$Q^k_h(x,a) \stackrel{\text{def}}{=} \min_{s\in[k-1]} \left(\widetilde{Q}^k_h(x^s_h, a^s_h) + L_h \rho\left[(x,a),(x^s_h,a^s_h)\right]\right)$$

and $V^k_h(x) \stackrel{\text{def}}{=} \min\left(H-h+1, \max_{a'} Q^k_h(x,a')\right)$. The policy π_k executed by Kernel-UCBVI is the greedy policy with respect to Q^k_h (see Alg. 1).

Let $\mathbf{C}^k_h(x,a) \stackrel{\text{def}}{=} \beta + \sum_{s=1}^{k-1} w^s_h(x,a)$ be the *generalized counts*, which are a proxy for the number of visits to (x,a). The exploration bonus is defined based on the uncertainties on the transition and reward estimates and takes the form

$$\mathtt{B}^k_h(x,a) \approx \frac{H}{\sqrt{\mathbf{C}^k_h(x,a)}} + \frac{\beta H}{\mathbf{C}^k_h(x,a)} + L_1\sigma$$

where we omit constants and logarithmic terms. Refer to Eq. 4 in App. B for the exact definition.

4. Theoretical Guarantees & Discussion

The theorem below gives a high probability regret bound for $Kernel-UCBVI$. It features the σ-covering number of the state-action space. The σ-covering number of a metric space, formally defined in Def. 2 (App. A), is roughly the number of σ-radius balls required to cover the entire space. The covering dimension of a space is the smallest number d such that its σ-covering number is $\mathcal{O}\left(\sigma^{-d}\right)$. For instance, the covering number of a ball in $\mathbb{R}^d$ with the Euclidean distance is $\mathcal{O}\left(\sigma^{-d}\right)$ and its covering dimension is d.

Theorem 1. *With probability at least $1-\delta$, the regret of $Kernel-UCBVI$ for a bandwidth σ satisfies*

$$\mathcal{R}(K) \le \widetilde{\mathcal{O}}\left(H^2\sqrt{|\mathcal{C}_\sigma|\,K} + L_1 K H \sigma + H^3 |\mathcal{C}_\sigma||\widetilde{\mathcal{C}}_\sigma|\right),$$

where $|\mathcal{C}_\sigma|$ and $|\widetilde{\mathcal{C}}_\sigma|$ are the σ-covering numbers of $(\mathcal{X}\times\mathcal{A}, \rho)$ and $(\mathcal{X}, \rho_\mathcal{X})$, respectively, and L_1 is the Lipschitz constant of the optimal Q-functions.

Proof. Restatement of Theorem 4 in App. D. A proof sketch is given in Section 6. □

Corollary 1. *By taking $\sigma = (1/K)^{1/(2d+1)}$, $\mathcal{R}(K) = \widetilde{\mathcal{O}}\left(H^3 K^{\max\left(\frac{1}{2}, \frac{2d}{2d+1}\right)}\right)$, where d is the covering dimension of the state-action space.*

Remark 1. *As for other model-based algorithms, the dependence on H can be improved if the transitions are stationary, i.e., do not depend on h. In this case, the regret of Kernel-UCBVI becomes $\widetilde{\mathcal{O}}\left(H^2 K^{\frac{2d}{2d+1}}\right)$ due to a gain a factor of H in the second order term (see App. E).*

[3] Here, δ_x denotes the Dirac measure with mass at x.

To the best of our knowledge, this is the first regret bound for kernel-based RL using smoothing kernels, and we present below further discussions on this result.

Comparison to lower bound for Lipschitz MDPs In terms of the number of episodes K and the dimension d, the lower bound for Lipschitz MDPs is $\Omega(K^{(d+1)/(d+2)})$, which is a consequence of the result for contextual Lipschitz bandits (Slivkins, 2014). In terms of H, the optimal dependence can be conjectured to be $H^{3/2}$, which is the case for tabular MDPs (Jin et al., 2018).[4] For $d = 1$, our bound has an optimal dependence on K, leading to a regret of order $\widetilde{\mathcal{O}}\left(H^3K^{2/3}\right)$, or $\widetilde{\mathcal{O}}\left(H^2K^{2/3}\right)$ when the transitions are stationary (see Remark 1).

Comparison to other upper bounds for Lipschitz MDPs The best available upper bound in this setting, in terms of K and d, is $\Omega\left(H^{5/2}K^{\frac{d+1}{d+2}}\right)$, which is achieved by model-free algorithms performing either uniform or adaptive discretization of the state-action space (Song & Sun, 2019; Sinclair et al., 2019; Touati et al., 2020).

Relevance of a kernel-based algorithm Although our upper bound does not match the lower bound for Lipschitz MDPs, kernel-based RL can be a very useful tool in practice to handle the bias-variance trade-off in RL. It allows us to easily provide expert knowledge to the algorithm through kernel design, which can be seen as introducing more bias to reduce the variance of the algorithm and, consequently, improve the learning speed. As shown by Kveton & Theocharous (2012) and Barreto et al. (2016), KBRL are empirically successful in medium-scale tasks ($d \approx 10$), such as control problems, HIV drug scheduling and an epilepsy suppression task. In such problems, Kernel-UCBVI can be used to enhance exploration, and the confidence intervals we derive here may also be useful in settings such as robust planning (Lim & Autef, 2019). Interestingly, Badia et al. (2020) have shown that kernel-based exploration bonuses similar to the ones derived in this paper can improve exploration in Atari games.

Regularity assumptions The regret bound we provide only requires only weak assumptions on the MDP: we assume that both the transitions and rewards are Lipschitz continuous, but we have no constraints on the behavior of the Bellman operator. As a consequence, the regret bounds suffer from the curse of dimensionality: as d goes to infinity, both the lower and upper bounds become linear in the number of episodes K. Other settings, such as low-rank MDPs (Jin et al., 2020) and RKHS approximations (Yang et al., 2020; Chowdhury & Oliveira, 2020) achieve regret bounds scaling with $\sqrt{K}$, but they require much stronger assumptions on the MDP, such as the closedness of the Bellman operator in the function class used to represent Q functions, which is a condition that is much harder to verify. Barreto et al. (2016) show that KBRL (with smoothing kernels) can be related to low-rank MDPs, and we believe that our analysis brings new elements to study this trade-off that exists between regularity assumptions and regret bounds.

Model-free vs. Model-based An interesting remark comes from the comparison between our algorithm and recent model-free approaches in continuous MDPs (Song & Sun, 2019; Sinclair et al., 2019; Touati et al., 2020). These algorithms are based on optimistic Q-learning (Jin et al., 2018), to which we refer as OptQL, and achieve a regret of order $\widetilde{\mathcal{O}}\left(H^{\frac{5}{2}}K^{\frac{d+1}{d+2}}\right)$, which has an optimal dependence on K and d. While we achieve the same $\widetilde{\mathcal{O}}\left(K^{2/3}\right)$ regret when $d = 1$, our bound is slightly worse for $d > 1$. To understand this gap, it is enlightening to look at the regret bound for tabular MDPs.

Since our algorithm is inspired by UCBVI (Azar et al., 2017) with Chernoff-Hoeffding bonus, we compare it to OptQL, which is used by (Song & Sun, 2019; Sinclair et al., 2019; Touati et al., 2020), with the same kind of exploration bonus. Consider an MDP with X states and A actions and non-stationary transitions. UCBVI has a regret bound of $\widetilde{\mathcal{O}}\left(H^2\sqrt{XAK} + H^3X^2A\right)$ while OptQL has $\widetilde{\mathcal{O}}\left(H^{5/2}\sqrt{XAK} + H^2XA\right)$. As we can see, OptQL is a $\sqrt{H}$-factor worse than UCBVI when comparing the first-order term, but it is HX times better in the second-order term. For large values of K, second-order terms can be neglected in the comparison of the algorithms in tabular MDPs, since they do not depend on K. However, they play an important role in continuous MDPs, where X and A are replaced by the σ-covering number of the state-action space, which is roughly $1/\sigma^d$. In tabular MDPs, the second-order term is constant (i.e., does not depend on K). On the other hand, in continuous MDPs, the algorithms define the granularity of the representation of the state-action space based on the number of episodes, connecting the number of states X with K. For example, in (Song & Sun, 2019) the ϵ-net used by the algorithm is tuned such that $\epsilon = (HK)^{-1/(d+2)}$ (see also Ortner & Ryabko 2012; Lakshmanan et al. 2015; Jian et al. 2019). Similarly, in our algorithm we have that $\sigma = K^{-1/(2d+1)}$. For this reason, the second-order term in UCBVI becomes the dominant term in our analysis, leading to a worse dependence on d compared to model-free algorithms, as highlighted in the proof sketch (Sec. 6). For similar reasons, Kernel-UCBVI has an additional $\sqrt{H}$ factor compared to model-free algorithms based on (Jin et al., 2018). This shows that the direction of achieving first-order optimal terms at the expense of higher second-order terms may not be justified outside the tabular case. Whether this

[4] See also (Sinclair et al., 2019, Sec. 4.4).

is a flaw in the algorithm design or in the analysis is left as an open question. However, as observed in Section 7, model-based algorithms seem to enjoy a better empirical performance.

5. Improving the Computational Complexity

Kernel-UCBVI is a non-parametric model-based algorithm and, consequently, it inherits the weaknesses of these approaches. In order to be data adaptive, it needs to store all the samples $(x_h^k, a_h^k, x_{h+1}^k, r_h^k)$ and their optimistic values $\widetilde{Q}_h^k$ and V_h^k for $(k,h) \in [K] \times [H]$, leading to a total memory complexity of $\mathcal{O}(HK)$. Like standard model-based algorithms, it needs to perform planning at each episode which gives a total runtime of $\mathcal{O}\left(HAK^3\right)$[5], where the factor A takes into account the complexity of computing the maximum over actions.[6] Kernel-UCBVI has similar time and space complexity of recent approaches for low-rank MDPs (Jin et al., 2020; Zanette et al., 2019).

To alleviate the computational burden of Kernel-UCBVI, we leverage Real-Time Dynamic Programming (RTDP), see (Barto et al., 1995), to perform incremental planning. Similarly to OptQL, RTDP-like algorithms maintain an optimistic estimate of the optimal value function that is updated incrementally by interacting with the MDP. The main difference is that the update is done by using an estimate of the MDP (i.e., model-based) rather than the observed transition sample. In episode k and step h, our algorithm, named Greedy-Kernel-UCBVI, computes an upper bound $\widetilde{Q}_h^k(x_h^k, a)$ for each action a using the kernel estimate as in Eq. (1). Then, it executes the greedy action $a_h^k = \operatorname{argmax}_{a \in \mathcal{A}} \widetilde{Q}_h^k(x_h^k, a)$. As a next step, it computes $\widetilde{V}_h^k(x_h^k) = \widetilde{Q}_h^k(x_h^k, a_h^k)$ and refines the previous L_h-Lipschitz upper confidence bound on the value function

$$V_h^{k+1}(x) = \min\left((V_h^k(x), \widetilde{V}_h^k(x_h^k) + L_h \rho_{\mathcal{X}}\left(x, x_h^k\right)\right).$$

The complete description of Greedy-Kernel-UCBVI is given in Alg. 3 in App. F. The total runtime of this efficient version is $\mathcal{O}\left(HAK^2\right)$ with total memory complexity of $\mathcal{O}(HK)$.

RTDP has been recently analyzed by (Efroni et al., 2019) in tabular MDPs. Following their analysis, we prove the following theorem, which shows that Greedy-Kernel-UCBVI achieves the same guarantees of Kernel-UCBVI with a large improvement in computational complexity.

Theorem 2. *With probability at least $1-\delta$, the regret of Greedy − Kernel − UCBVI for a bandwidth σ is of order $\mathcal{R}(K) = \widetilde{\mathcal{O}}\left(\mathcal{R}(K, Kernel - UCBVI) + H^2|\widetilde{\mathcal{C}}_\sigma|\right)$, where $|\widetilde{\mathcal{C}}_\sigma|$ is the σ-covering number of state space. This results in a regret of $\widetilde{\mathcal{O}}\left(H^3 K^{2d/(2d+1)}\right)$ when $\sigma = (1/K)^{1/(2d+1)}$.*

Proof. The complete proof is provided in App. F. The key properties for proving this regret bound are: *(i)* optimism, and *(ii)* the fact that (V_h^k) are point-wise non-increasing.

Besides RTDP, other techniques previously proposed to accelerate KBRL can also be applied, notably the use of *representative states* (Kveton & Theocharous, 2012; Barreto et al., 2016) that merge states that are close to each other to avoid a per-episode runtime that increases with k.

6. Proof sketch

We now provide a sketch of the proof of our main result, Theorem 1. The complete proof is given in the Appendix. The analysis splits into three parts: (i) deriving confidence intervals for the reward and transition kernel estimators; (ii) proving that the algorithm is optimistic, i.e., that $V_h^k(x) \geq V_h^*(x)$ for any (x,k,h) on a high probability event $\mathcal{G}$; and (iii) proving an upper bound on the regret by using the fact that $\mathcal{R}(K) = \sum_k \left(V_1^*(x_1^k) - V_1^{\pi_k}(x_1^k)\right) \leq \sum_k \left(V_1^k(x_1^k) - V_1^{\pi_k}(x_1^k)\right)$.

6.1. Concentration

The most interesting part is the concentration of the transition kernel. Since $\widehat{P}_h^k(\cdot|x,a)$ are weighted sums of Dirac measures, we cannot bound the distance between $P_h(\cdot|x,a)$ and $\widehat{P}_h^k(\cdot|x,a)$ directly. Instead, for V_{h+1}^* the optimal value function at step $h+1$, we bound the difference

$$\begin{aligned}
&\left|(\widehat{P}_h^k - P_h)V_{h+1}^*(x,a)\right| \\
&= \left|\sum_{s=1}^{k-1} \widetilde{w}_s^h(x,a) V_{h+1}^*(x_{h+1}^s) - P_h V_{h+1}^*(x,a)\right| \\
&\leq \underbrace{\left|\sum_{s=1}^{k-1} \widetilde{w}_s^h(x,a)\left(V_{h+1}^*(x_{h+1}^s) - P_h V_{h+1}^*(x_h^s, a_h^s)\right)\right|}_{(\mathbf{A})} \\
&+ \underbrace{\lambda_p L_{h+1} \sum_{s=1}^{k-1} \widetilde{w}_s^h(x,a)\rho\left[(x,a),(x_h^s,a_h^s)\right]}_{(\mathbf{B})} + \underbrace{\frac{\beta \left\|V_{h+1}^*\right\|_\infty}{\mathbf{C}_h^k(x,a)}}_{(\mathbf{C})}.
\end{aligned}$$

The term $(\mathbf{A})$ is a weighted sum of a martingale difference sequence. To control it, we propose a new Hoeffding-type inequality, Lemma 2, that applies to weighted sums with random weights. The term $(\mathbf{B})$ is a bias term that is ob-

[5] Since the runtime of an episode k is $\mathcal{O}\left(HAk^2\right)$.

[6] While in theory the algorithm works with a compact action space, the main practical issue resides in the computation of the best action (i.e., $a_h^k = \operatorname{argmax}_a Q_h^k(x_h^k, a)$). In this case, we could resort to black box optimization algorithms (e.g., Munos, 2014, Sec. 3.3), which might require the discretization of the action space. This is however less critical than the discretization of the state-space, since the possible actions must always be known in advance, unlike the set of possible states.

tained using the fact that V^*_{h+1} is L_{h+1}-Lipschitz and that the transition kernel is λ_p-Lipschitz, and can be shown to be proportional to the bandwidth σ under Assumption 3 (Lemma 7). The term $(\mathbf{C})$ is the bias introduced by the regularization parameter β. Hence, for a fixed state-action pair (x, a), we show that[7], with high-probability,

$$\left|(\widehat{P}^k_h - P_h)V^*_{h+1}(x,a)\right| \lesssim \frac{H}{\sqrt{\mathbf{C}^k_h(x,a)}} + \frac{\beta H}{\mathbf{C}^k_h(x,a)} + L_1\sigma.$$

Then, we extend this bound to all (x, a) by leveraging the continuity of all the terms involving (x, a) and a covering argument. This continuity is a consequence of kernel smoothing, and it is a key point in avoiding a discretization of $\mathcal{X} \times \mathcal{A}$ in the algorithm.

In Theorem 3, we define a favorable event $\mathcal{G}$, of probability larger than $1 - \delta/2$, in which (a more precise version of) the above inequality holds, the mean rewards belong to their confidence intervals, and we further control the deviations of $(\widehat{P}^k_h - P_h)f(x, a)$ for *any* $2L_1$-Lipschitz function f. This last part is obtained thanks to a *new Bernstein-like concentration inequality* for weighted sums (Lemma 3).

6.2. Optimism

To prove that the optimistic value function V^k_h is indeed an upper bound on V^*_h, we proceed by induction on h and we use the Q functions. When $h = H + 1$, we have $Q^k_{H+1}(x, a) = Q^*_{H+1}(x, a) = 0$ for all (x, a), by definition. Assuming that $Q^k_{h+1}(x, a) \geq Q^*_{h+1}(x, a)$ for all (x, a), we have $V^k_{h+1}(x) \geq V^*_{h+1}(x)$ for all x. Then, the bonuses are defined so that $\widetilde{Q}^k_h(x, a) \geq Q^*_h(x, a)$ for all (x, a), on the event $\mathcal{G}$.

In particular $\widetilde{Q}^k_h(x^s_h, a^s_h) - Q^*_h(x^s_h, a^s_h) \geq 0$ for all $s \in [k-1]$, which gives us

$$\begin{aligned}&\widetilde{Q}^k_h(x^s_h, a^s_h) + L_h\rho\left[(x, a), (x^s_h, a^s_h)\right]\\ &\geq Q^*_h(x^s_h, a^s_h) + L_h\rho\left[(x, a), (x^s_h, a^s_h)\right] \geq Q^*_h(x, a)\end{aligned}$$

for all $s \in [k-1]$, since Q^*_h is L_h-Lipschitz. It follows from the definition of Q^k_h that $Q^k_h(x, a) \geq Q^*_h(s, a)$, which in turn implies that, for all x, $V^k_h(x) \geq V^*_h(x)$ in $\mathcal{G}$.

6.3. Bounding the regret

To provide an upper bound on the regret in the event $\mathcal{G}$, let $\delta^k_h \overset{\text{def}}{=} V^k_h(x^k_h) - V^{\pi_k}_h(x^k_h)$. The fact that $V^k_h \geq V^*_h$ gives us $\mathcal{R}(K) \leq \sum_k \delta^k_1$. Introducing $(\tilde{x}^k_h, \tilde{a}^k_h)$, the state-action pair in the past data $\mathcal{D}_h$ that is the closest to (x^k_h, a^k_h) and letting $\square^k_h = \rho\left[(\tilde{x}^k_h, \tilde{a}^k_h), (x^k_h, a^k_h)\right]$, we bound δ^k_h using the following decomposition:

$$\begin{aligned}\delta^k_h &\leq Q^k_h(x^k_h, a^k_h) - Q^{\pi_k}_h(x^k_h, a^k_h)\\ &\leq \widetilde{Q}^k_h(\tilde{x}^k_h, \tilde{a}^k_h) - Q^{\pi_k}_h(x^k_h, a^k_h) + L_h\square^k_h\\ &\leq 2\,\mathtt{B}^k_h(\tilde{x}^k_h, \tilde{a}^k_h) + (L_h + \lambda_p L_h + \lambda_r)\square^k_h\\ &\quad + \left(\widehat{P}^k_h - P_h\right)V^*_{h+1}(\tilde{x}^k_h, \tilde{a}^k_h) \qquad (\mathbf{1})\\ &\quad + P_h\left(V^k_{h+1} - V^{\pi_k}_{h+1}\right)(x^k_h, a^k_h) \qquad (\mathbf{2})\\ &\quad + \left(\widehat{P}^k_h - P_h\right)\left(V^k_{h+1} - V^*_{h+1}\right)(\tilde{x}^k_h, \tilde{a}^k_h) \qquad (\mathbf{3}).\end{aligned}$$

The term $(\mathbf{1})$ is shown to be smaller than $\mathtt{B}^k_h(\tilde{x}^k_h, \tilde{a}^k_h)$, by definition of the bonus. The term $(\mathbf{2})$ can be rewritten as δ^k_{h+1} plus a martingale difference sequence ξ^k_{h+1}. To bound the term $(\mathbf{3})$, we use that $V^k_{h+1} - V^*_{h+1}$ is $2L_1$-Lipschitz. The uniform deviations that hold on event $\mathcal{G}$ yield

$$③ \lesssim \frac{1}{H}\left(\delta^k_{h+1} + \xi^k_{h+1}\right) + \frac{H^2|\widetilde{\mathcal{C}}_\sigma|}{\mathbf{C}^k_h(\tilde{x}^k_h, \tilde{a}^k_h)} + L_1\square^k_h + L_1\sigma\,.$$

When $\square^k_h > 2\sigma$, we bound δ^k_h by H and we verify that $H\sum_{h=1}^H\sum_{k=1}^K \mathbb{I}\left\{\square^k_h > 2\sigma\right\} \leq H^2|\mathcal{C}_\sigma|$ by a pigeonhole argument. Hence, we can focus on the case where $\square^k_h \leq 2\sigma$, and add $H^2|\mathcal{C}_\sigma|$ to the regret bound, to take into account the steps (k, h) where $\square^k_h > 2\sigma$. The sum of ξ^k_{h+1} over (k, h) is bounded by $\widetilde{\mathcal{O}}\left(H^{\frac{3}{2}}\sqrt{K}\right)$ by Hoeffding-Azuma's inequality, on some event $\mathcal{F}$ of probability larger than $1 - \delta/2$. Now, we focus on the case where $\square^k_h \leq 2\sigma$ and we omit the terms involving ξ^k_{h+1}. Using the definition of the bonus, we obtain

$$\delta^k_h \lesssim \left(1 + \frac{1}{H}\right)\delta^k_{h+1} + \frac{H}{\sqrt{\mathbf{C}^k_h(\tilde{x}^k_h, \tilde{a}^k_h)}} + \frac{H^2|\widetilde{\mathcal{C}}_\sigma|}{\mathbf{C}^k_h(\tilde{x}^k_h, \tilde{a}^k_h)} + L_1\sigma.$$

Using the fact that $(1 + 1/H)^H \leq e$, we have, on $\mathcal{G} \cap \mathcal{F}$,

$$\mathcal{R}(K) \lesssim \sum_{h,k}\left(\frac{H}{\sqrt{\mathbf{C}^k_h(\tilde{x}^k_h, \tilde{a}^k_h)}} + \frac{H^2|\widetilde{\mathcal{C}}_\sigma|}{\mathbf{C}^k_h(\tilde{x}^k_h, \tilde{a}^k_h)}\right) + L_1KH\sigma.$$

The term in $1/\mathbf{C}^k_h(\tilde{x}^k_h, \tilde{a}^k_h)$ is the *second order term* (in K).

In the tabular case, it is multiplied by the number of states. Here, it is multiplied by the covering number of the state space $|\widetilde{\mathcal{C}}_\sigma|$.

From there it remains to bound the sum of the first and second-order terms, and we specifically show that

$$\sum_{h,k}\frac{1}{\sqrt{\mathbf{C}^k_h(\tilde{x}^k_h, \tilde{a}^k_h)}} \lesssim H\sqrt{|\mathcal{C}_\sigma|K} \qquad (2)$$

$$\text{and} \quad \sum_{h,k}\frac{1}{\mathbf{C}^k_h(\tilde{x}^k_h, \tilde{a}^k_h)} \lesssim H|\mathcal{C}_\sigma|\log K, \qquad (3)$$

where we note that (3) has a worse dependence on $|\mathcal{C}_\sigma|$. As mentioned before, unlike in the tabular case the sum of

[7] Here, $\lesssim$ means smaller than or equal up to logarithmic terms.

"second-order" terms will actually be the leading term, since the choice of σ that minimizes the regret depends on K.

Finally, we obtain that on $\mathcal{G} \cap \mathcal{F}$ (of probability $\geq 1-\delta$)

$$\mathcal{R}(K) \lesssim H^2 \sqrt{|\mathcal{C}_\sigma| K} + H^3 |\mathcal{C}_\sigma| |\widetilde{\mathcal{C}}_\sigma| + L_1 K H \sigma + H^2 |\mathcal{C}_\sigma|,$$

where the extra $H^2|\mathcal{C}_\sigma|$ takes into account the episodes where $\square_h^k > 2\sigma$.

If the transitions kernels are stationary, i.e., $P_1 = \ldots = P_H$, the bounds (2) and (3) can be improved to $\sqrt{|\mathcal{C}_\sigma| KH}$ and $|\mathcal{C}_\sigma| \log(KH)$ respectively, thus improving the final scaling in H.[8] See App. E for details.

7. Experiments

To illustrate experimentally the properties of Kernel-UCBVI, we consider a Grid-World environment with continuous states. This Grid-World is composed of two rooms separated by a wall, such that $\mathcal{X} = ([0, 1-\Delta] \cup [1+\Delta, 2]) \times [0, 1])$ where $2\Delta = 0.1$ is the width of the wall, as illustrated by Figure 1. There are four actions: left, right, up, and down, each one resulting to a displacement of 0.1 in the corresponding direction. A two-dimensional Gaussian noise is added to the transitions, and, in each room, there is a single region with non-zero reward. The agent has 0.5 probability of starting in each of the rooms, and the starting position is at the room's bottom left corner.

We compare Kernel-UCBVI and Greedy-Kernel-UCBVI to the following baselines: (i) UCBVI (Azar et al., 2017) using a uniform discretization of the state-space; (ii) OptQL (Jin et al., 2018) also on a uniform discretization; (iii) Adaptive-Q-Learning (Sinclair et al., 2019) that uses an adaptive discretization of the state-space. We used the Euclidean distance and the Gaussian kernel with a fixed bandwidth $\sigma = 0.025$, matching the granularity of the uniform discretization used by some of the baselines. We also implemented a version of Kernel-UCBVI using the "expert knowledge" that the two rooms are equivalent under translation, by using a metric that is invariant with respect to the change of rooms. More details about the experimental setup are provided in Appendix I.[9]

We ran the algorithms for 5×10^4 episodes, and Figure 2 shows the sum of rewards obtained by each of them. When the curves start behaving as a straight line, it roughly means that the algorithm has converged to a policy whose value is the slope of the line. We see that Kernel-UCBVI is able to outperform the baselines, and that the use of expert knowledge in the kernel design can considerably increase

[8] This is because, in the non-stationary case, we bound the sums over k and then multiply the resulting bound by H. In the stationary case, we can directly bound the sums over (k, h).

[9] Implementations of Kernel-UCBVI are available on GitHub, and use the `rlberry` library (Domingues et al., 2021).

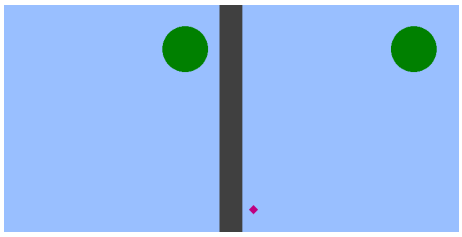

Figure 1. Continuous grid-world with two rooms separated by a wall. The circles represent the regions with non-zero rewards.

the learning speed. Asymptotically, the extra bias introduced by the kernel (especially its bandwidth) might make Kernel-UCBVI converge to a worse policy at the end: the kernel bandwidth and the discretization width are comparable, but the Gaussian kernel introduces more bias due to sample aggregation. On the other hand, we see that introducing more bias can greatly improve the learning speed in the beginning, especially when expert knowledge is used. This flexibility in handling the bias-variance trade-off is one of the strengths of kernel-based approaches.

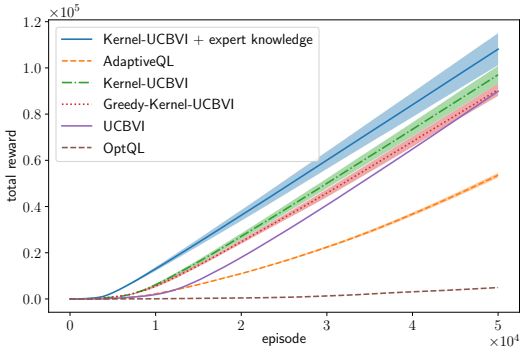

Figure 2. Total sum of rewards obtained by Kernel-UCBVI and baselines versus the number of episodes. Average over 8 runs.

8. Conclusion

In this paper, we introduced Kernel-UCBVI, a model-based algorithm for finite-horizon reinforcement learning in metric spaces which employs kernel smoothing to estimate rewards and transitions. By providing new high-probability confidence intervals for weighted sums and non-parametric kernel estimators, we generalize the techniques introduced by (Azar et al., 2017) in tabular MDPs to the continuous setting. We prove that the regret of Kernel-UCBVI is of order $H^3 K^{\max\left(\frac{1}{2}, \frac{2d}{2d+1}\right)}$, which is the first regret bound for kernel-based RL using smoothing kernels. In addition, we provide experiments illustrating the effectiveness of Kernel-UCBVI in handling the bias-variance trade-off and in the use of expert knowledge. Interesting directions for future work include the use of learned metrics (e.g., using neural networks) and the use of adaptive kernel bandwidths to better handle the bias-variance trade-off asymptotically.

Acknowledgements

At Inria and CNRS, this work was supported by the European CHIST-ERA project DELTA, French Ministry of Higher Education and Research, Nord-Pas-de-Calais Regional Council, French National Research Agency project BOLD (ANR19-CE23-0026-04), FMJH PGMO project 2018-0045. Pierre Ménard is supported by the SFI Sachsen-Anhalt for the project RE-BCI ZS/2019/10/102024 by the Investitionsbank Sachsen-Anhalt.

References

Abbasi-Yadkori, Y. and Szepesvári, C. Regret bounds for the adaptive control of linear quadratic systems. In *Proceedings of the 24th Annual Conference on Learning Theory*, pp. 1–26, 2011.

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.

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.

Badia, A. P., Sprechmann, P., Vitvitskyi, A., Guo, D., Piot, B., Kapturowski, S., Tieleman, O., Arjovsky, M., Pritzel, A., Bolt, A., et al. Never give up: Learning directed exploration strategies. *arXiv preprint arXiv:2002.06038*, 2020.

Barreto, A. M., Precup, D., and Pineau, J. Practical kernel-based reinforcement learning. *The Journal of Machine Learning Research*, 17(1):2372–2441, 2016.

Barto, A. G., Bradtke, S. J., and Singh, S. P. Learning to act using real-time dynamic programming. *Artificial intelligence*, 72(1-2):81–138, 1995.

Boucheron, S., Lugosi, G., and Massart, P. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

Burnetas, A. N. and Katehakis, M. N. Optimal adaptive policies for Markov decision processes. *Mathematics of Operations Research*, 22(1):222–255, feb 1997. ISSN 0364-765X.

Chowdhury, S. R. and Gopalan, A. Online learning in kernelized markov decision processes. In *Proceedings of Machine Learning Research*, volume 89, 2019.

Chowdhury, S. R. and Oliveira, R. No-regret reinforcement learning with value function approximation: a kernel embedding approach. *arXiv preprint arXiv:2011.07881*, 2020.

Combes, R., Magureanu, S., and Proutiere, A. Minimal exploration in structured stochastic bandits. In *Advances in Neural Information Processing Systems*, pp. 1763–1771, 2017.

Domingues, O. D., Flet-Berliac, Y., Leurent, E., Ménard, P., Shang, X., and Valko, M. rlberry - A Reinforcement Learning Library for Research and Education. https://github.com/rlberry-py/rlberry, 2021.

Efroni, Y., Merlis, N., Ghavamzadeh, M., and Mannor, S. Tight regret bounds for model-based reinforcement learning with greedy policies. In *Advances in Neural Information Processing Systems*, pp. 12203–12213, 2019.

Gottlieb, L.-A., Kontorovich, A., and Krauthgamer, R. Efficient regression in metric spaces via approximate lipschitz extension. *IEEE Transactions on Information Theory*, 63(8):4838–4849, 2017.

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

Jian, Q., Fruit, R., Pirotta, M., and Lazaric, A. Exploration bonus for regret minimization in discrete and continuous average reward mdps. In *Advances in Neural Information Processing Systems*, pp. 4891–4900, 2019.

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 (COLT)*, 2020.

Kakade, S., Kearns, M. J., and Langford, J. Exploration in metric state spaces. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pp. 306–312, 2003.

Kearns, M. and Singh, S. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2-3):209–232, 2002.

Kveton, B. and Theocharous, G. Kernel-based reinforcement learning on representative states. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

Lakshmanan, K., Ortner, R., and Ryabko, D. Improved regret bounds for undiscounted continuous reinforcement learning. In *International Conference on Machine Learning*, pp. 524–532, 2015.

Lattimore, T., Hutter, M., Sunehag, P., et al. The sample-complexity of general reinforcement learning. In *Proceedings of the 30th International Conference on Machine Learning*. Journal of Machine Learning Research, 2013.

Lim, S. H. and Autef, A. Kernel-based reinforcement learning in robust markov decision processes. In *Proceedings of the 36th International Conference on Machine Learning, (ICML)*, 2019.

Munos, R. From bandits to monte-carlo tree search: The optimistic principle applied to optimization and planning. *Found. Trends Mach. Learn.*, 7(1):1–129, 2014.

Ni, C., Yang, L. F., and Wang, M. Learning to control in metric space with optimal regret. In *2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 726–733. IEEE, 2019.

Ok, J., Proutiere, A., and Tranos, D. Exploration in structured reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 8874–8882, 2018.

Ormoneit, D. and Sen, Ś. Kernel-based reinforcement learning. *Machine learning*, 49(2-3):161–178, 2002.

Ortner, R. and Ryabko, D. Online regret bounds for undiscounted continuous reinforcement learning. In *Advances in Neural Information Processing Systems*, pp. 1763–1771, 2012.

Osband, I. and Van Roy, B. Model-based reinforcement learning and the eluder dimension. In *Advances in Neural Information Processing Systems*, pp. 1466–1474, 2014.

Osband, I., Russo, D., and Van Roy, B. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems*, pp. 3003–3011, 2013.

Pazis, J. and Parr, R. Pac optimal exploration in continuous space markov decision processes. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

Peña, V. H., Lai, T. L., and Shao, Q.-M. *Self-normalized processes: Limit theory and Statistical Applications*. Springer Science & Business Media, 2008.

Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, Inc., New York, NY, USA, 1994. ISBN 0471619779.

Reeve, H. W. J., Mellor, J., and Brown, G. The k-nearest neighbour UCB algorithm for multi-armed bandits with covariates. In *ALT*, volume 83 of *Proceedings of Machine Learning Research*, pp. 725–752. PMLR, 2018.

Sinclair, S. R., Banerjee, S., and Yu, C. L. Adaptive discretization for episodic reinforcement learning in metric spaces. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(3):1–44, 2019.

Slivkins, A. Contextual bandits with similarity information. *Journal of Machine Learning Research*, 15(1):2533–2568, 2014.

Song, Z. and Sun, W. Efficient model-free reinforcement learning in metric spaces. *arXiv preprint arXiv:1905.00475*, 2019.

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

Touati, A., Taiga, A. A., and Bellemare, M. G. Zooming for Efficient Model-Free Reinforcement Learning in Metric Spaces. *arXiv e-prints*, art. arXiv:2003.04069, March 2020.

Yang, L. F. and Wang, M. Reinforcement leaning in feature space: Matrix bandit, kernels, and regret bound. In *International Conference on Machine Learning (ICML)*, 2020.

Yang, Z., Jin, C., Wang, Z., Wang, M., and Jordan, M. Provably efficient reinforcement learning with kernel and neural function approximations. *Advances in Neural Information Processing Systems*, 33, 2020.

Zanette, A. and Brunskill, E. Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

Zanette, A., Brandfonbrener, D., Pirotta, M., and Lazaric, A. Frequentist regret bounds for randomized least-squares value iteration. *CoRR*, abs/1911.00567, 2019.