Abstract

The upper confidence reinforcement learning (UCRL2) strategy introduced in (Jaksch et al., 2010) is a popular method to perform regret minimization in unknown discrete Markov Decision Processes under the average-reward criterion. Despite its nice and generic theoretical regret guarantees, this strategy and its variants have remained until now mostly theoretical as numerical experiments on simple environments exhibit long burn-in phases before the learning takes place. In pursuit of practical efficiency, we present UCRL3, following the lines of UCRL2, but with two key modifications: First, it uses state-of-the-art time-uniform concentration inequalities, to compute confidence sets on the reward and (component-wise) transition distributions for each state-action pair. Further, to tighten exploration, it uses an adaptive computation of the support of each transition distributions, which in turn enables us to revisit the extended value iteration procedure to optimize over distributions with reduced support by disregarding low probability transitions, while still ensuring near-optimism. We demonstrate, through numerical experiments on standard environments, that reducing exploration this way yields a substantial numerical improvement compared to UCRL2 and its variants. On the theoretical side, these key modifications enable us to derive a regret bound for UCRL3 improving on UCRL2, that for the first time makes appear notions of local diameter and effective support, thanks to variance-aware concentration bounds.

1. Introduction

In this paper, we consider Reinforcement Learning (RL) in an unknown and discrete Markov Decision Process (MDP) under the average-reward criterion, when the learner interacts with the system in a single, infinite stream of observations, starting from an initial state without any reset. More formally, let \( \mathcal{M} = (\mathcal{S}, \mathcal{A}, p, \nu) \) be an undiscounted MDP, where \( \mathcal{S} \) denotes the discrete state-space with cardinality \( S \), and \( \mathcal{A} \) denotes the discrete action-space with cardinality \( A \). \( p \) is the transition kernel such that \( p(s'|s,a) \) denotes the probability of transiting to state \( s' \), starting from state \( s \) and executing action \( a \). We denote by \( \mathcal{K}_{s,a} \) the set of successor states of the state-action pair \((s,a)\), that is \( \mathcal{K}_{s,a} := \{ x \in \mathcal{S} : p(x|s,a) > 0 \} \), and further define \( K_{s,a} := |\mathcal{K}_{s,a}| \). Finally, \( \nu \) is a reward distribution function on \([0,1]\) with mean function denoted by \( \mu \). The interaction between the learner and the environment proceeds as follows. The learner starts in some state \( s_1 \in \mathcal{S} \) at time \( t = 1 \). At each time step \( t \in \mathbb{N} \), where the learner is in state \( s_t \), she chooses an action \( a_t \in \mathcal{A} \) based on \( s_t \) as well as her past decisions and observations. When executing action \( a_t \) in state \( s_t \), the learner receives a random reward \( r_t := r_t(s_t, a_t) \) drawn independently from distribution \( \nu(s_t, a_t) \), and whose mean is \( \mu(s_t, a_t) \). The state then transits to a next state \( s_{t+1} \sim p(\cdot|s_t, a_t) \), and a new decision step begins. For background material on MDPs and RL, we refer to standard textbooks (Sutton & Barto, 1998; Puterman, 2014).

The goal of the learner is to maximize the cumulative reward gathered in the course of her interaction with the environment. The transition kernel \( p \) and reward function \( \nu \) are initially unknown, and so the learner has to learn them by trying different actions and recording the realized rewards and state transitions. The performance of the learner can be assessed through the notion of regret, which compares the cumulative reward gathered by an oracle, being aware of \( p \) and \( \nu \), to that gathered by the learner. Following (Jaksch et al., 2010), we define the regret of a learning algorithm \( \mathcal{A} \) after \( T \) steps as \( \Theta(\mathcal{A}, T) := Tg^* - \sum_{t=1}^{T} r_t(s_t, a_t) \), where \( g^* \) denotes the average-reward (or gain) attained by an optimal policy. Alternatively, the objective of the learner is to minimize the regret, which calls for balancing between exploration and exploitation.

To date, several algorithms have been proposed in order to minimize the regret based on the optimism in the face of
uncertainty principle, originated from the seminal work (Lai & Robbins, 1985) on stochastic multi-armed bandits. Algorithms designed based on this principle typically maintain confidence bounds on the unknown reward and transition distributions, and choose an optimistic model that leads to the highest average long-term reward. A popular algorithm for the presented RL setup is UCRL2, which was introduced in the seminal work (Jaksch et al., 2010). UCRL2 achieves a non-asymptotic regret upper bound scaling as $O(DS\sqrt{T})$ with high probability, in any communicating MDP with $S$ states, $A$ actions, and diameter $D$. (Jaksch et al., 2010) also report a regret lower bound scaling as $\Omega(\sqrt{DSAT})$, indicating that the above regret bound for UCRL2 is rate-optimal, i.e., it has a tight dependence on $T$, and can only be improved by a factor of, at most, $\sqrt{DS}$.

Since the advent of UCRL2, several of its variants have been presented in the literature; see, e.g., (Filippi et al., 2010; Maillard et al., 2014; Fruit et al., 2018b; Talebi & Maillard, 2018). These variants mainly strive to improve the regret guarantee and/or empirical performance of UCRL2 by using improved confidence bounds. Although these algorithms enjoy delicate and strong theoretical regret guarantees, their numerical assessment has shown that they typically achieve a bad performance even for state-spaces of moderate size. In particular, they suffer from a long burn-in phase before the learning takes place, rendering them impractical for state-spaces of moderate size. It is natural to ask whether such a bad empirical performance is due to the main principle of UCRL2 strategies, such as the optimistic principle, or to a not careful enough application of this principle. For instance in a different, episodic and Bayesian framework, PSRL has been reported to significantly improve on UCRL2. In this paper, we answer this question by showing, perhaps surprisingly, that a simple but crucial modification of UCRL2 that we call UCRL3 significantly outperforms other variants, while preserving (an improving) theoretical guarantees. Though our result does not imply that optimistic strategies are the best, it shows that they can be much stronger competitors than the vanilla UCRL2.

Contributions. We introduce UCRL3, a refined variant of UCRL2, whose design combines the following key elements: First, it uses tighter confidence bounds on components of transition kernel (similarly to Dann et al. (2017)) that are uniform in time, a property of independent interest for algorithm design in other RL setups; we refer to Section 3.1 for a detailed presentation. More specifically, for each component of a next-state transition distribution, it uses one time-uniform concentration inequality for $[0, 1]$-bounded observations and one for Bernoulli distributions with a Bernstein flavor.

The second key design of the algorithm is an novel procedure, which we call NOSS3, which adaptively computes an estimate of the support of transition probabilities of various state-action pairs. Such estimates are in turn used to compute a near-optimistic value and policy (Section 3.2). Combining NOSS with the Extended Value Iteration (EVI) procedure used for planning in UCRL2, allows us to devise EVI-NOSS, which is a refined variant of EVI. This step is non-trivial as it requires to consider a near-optimistic, as opposed to fully optimistc, for planning. Our adaptive procedure enables us to control this additional error. Further, this enables us to make appear in the regret analysis a notion of local diameter as well as a local effective support (Section 3.3), which in turn reduces the regret bounds. We define the local diameter below.

Definition 1 (Local Diameter of State $s$) Consider state $s \in S$. For $s_1, s_2 \in \cup_{a \in A} K_{s,a}$ with $s_1 \neq s_2$, let $T^\pi(s_1, s_2)$ denote the number of steps it takes to get to $s_2$ starting from $s_1$ and following policy $\pi$. Then, the local diameter of MDP $M$ for $s$, denoted by $D^\text{local}_s := D^\text{local}(M)$, is defined as

$$D^\text{local}_s := \max_{s_1, s_2 \in \cup_{a \in A} K_{s,a}} \min_{\pi} \mathbb{E}[T^\pi(s_1, s_2)].$$

On the theoretical side, we show in Theorem 1 that UCRL3 enjoys a regret bound scaling similarly to that established for the best variant of UCRL2 in the literature as in, e.g., (Fruit et al., 2018b). For better comparison with other works, we make sure to have an explicit bound including small constants for the leading terms. Thanks to a refined and careful analysis that we detail in the appendix, we also improve on the lower-order terms of the regret that we show should not be overlooked in practice. We provide in Section 4 a detailed comparison of the leading terms involved in several state-of-the-art algorithms to help better understand the behavior of these bounds. We also demonstrate through numerical experiments on standard environments that combining these refined, state-of-the-art confidence intervals together with this adaptive support estimation procedure yield a substantial improvement over UCRL2 and its variants. In particular, UCRL3 admits a burn-in phase, which is smaller than that of UCRL2 by an order of magnitude.

Related work. RL under the average-reward criterion dates back to the seminal papers (Graves & Lai, 1997) and (Burnetas & Katehakis, 1997), followed by (Tewari & Bartlett, 2008). Among these studies, for the case of ergodic MDPs, (Burnetas & Katehakis, 1997) derives an asymptotic MDP-dependent lower bound on the regret, and provides an asymptotically optimal algorithm. Algorithms with finite-time regret guarantees and for a wider class of MDPs are

---

1The notation $O(\cdot)$ hides terms that are poly-logarithmic in $T$.

2Given a MDP $M$, the diameter $D := D(M)$ is defined as $D(M) := \max_{s \in S} \min_{s'} \mathbb{E}[T^\pi(s, s')]$, where $T^\pi(s, s')$ denotes the number of steps it takes to get to $s'$ starting from $s$ and following policy $\pi$ (Jaksch et al., 2010).

3Near-Optimistic Support Optimization
Figure 1. Regret bounds of state-of-the-art algorithms for average-reward reinforcement learning. Here, $x \lor y$ denotes the maximum between $x$ and $y$. For KL-UCRL, $V_{s,a}$ denotes the variance of optimal bias function of the true MDP, when the state is distributed according to $p(\cdot|s,a)$. For UCRL3, $L_{s,a} := (\sum_{a \in S} p(x|s,a)(1 - p(x|s,a))^2)$ denotes the local effective support of $p(\cdot|s,a)$.

Presented in (Auer & Ortner, 2007; Jaksch et al., 2010; Bartlett & Tewari, 2009; Filippi et al., 2010; Maillard et al., 2014; Talebi & Maillard, 2018; Fruit et al., 2018a;b; Zhang & Ji, 2019). Among these works, (Filippi et al., 2010) introduces KL-UCRL, which is a variant of UCRL2 that uses the KL divergence to define confidence bounds. Similarly to UCRL2, KL-UCRL achieves a regret of $O(D\sqrt{AT})$ for the class of communicating MDPs. A more refined regret bound for KL-UCRL for ergodic MDPs is presented in (Talebi & Maillard, 2018). (Bartlett & Tewari, 2009) presents REGAL and report a $O(D'\sqrt{AT})$ regret with high probability in the larger class of weakly communicating MDPs, provided that the learner knows an upper bound $D'$ on the span of the bias function of the true MDP. (Fruit et al., 2018b) presents SCAL, which similarly to REGAL works for weakly communicating MDPs, but admits an efficient implementation. A similar algorithm called SCAL$^+$ is presented in (QIAN et al., 2019). Both SCAL and SCAL$^+$ admit a regret scaling as $O(D\sqrt{\sum_{s,a} K_{s,a} T})$. Recently, (Zhang & Ji, 2019) presents EBF achieving a regret of $O(\sqrt{HSA})$ assuming that the learner knows an upper bound $H$ on the span of the optimal bias function of the true MDP. EBF does not admit a computationally efficient implementation. Another related line of works considers posterior sampling methods such as (Osband et al., 2013) inspired from Thompson sampling (Thompson, 1933). For average-reward RL, existing works on these methods report Bayesian regret bounds, with the exception of (Agrawal & Jia, 2017a), whose corrected regret, reported in (Agrawal & Jia, 2017b), scales as $O(D\sqrt{AT} \log^3(T))$ and is valid for $T \geq S^4A^3$.

We finally mention that some studies consider regret minimization in MDPs in the episodic setting, with a fixed and known horizon; see, e.g., (Osband et al., 2013; Gheshlaghi Azar et al., 2017; Dann et al., 2017; Efroni et al., 2019; Zanette & Brunskill, 2019). Despite some similar-

Notations. We introduce some notations that will be used throughout. For $x, y \in \mathbb{R}$, $x \lor y$ denotes the maximum between $x$ and $y$. $\Delta_S$ represents the set of all probability distributions defined on $S$. For a distribution $p \in \Delta_S$ and a vector-function $f = (f(s))_{s \in S}$, we denote by $P_f$ its application on $f$, defined by $P_f = \mathbb{E}_{X \sim p}(f(X))$. We introduce $\Delta_S^{S \times A} := \{q : q(\cdot|s,a) \in \Delta_S, (s,a) \in S \times A\}$, and for $p \in \Delta_S^{S \times A}$, we define the corresponding operator $P$ such that $P_f : s,a \mapsto \mathbb{E}_{X \sim p(\cdot|s,a)}(f(X))$. We also introduce $S(f) = \max_s f(s) - \min_s f(s)$.

Under a given algorithm and for a pair $(s,a)$, we denote by $N_t(s,a)$ the total number of observations of $(s,a)$ up to time $t$. Let us define $\tilde{\mu}_t(s,a)$ as the empirical mean reward built using $N_t(s,a)$ i.i.d. samples from $\nu(s,a)$, and $\hat{p}_t(\cdot|s,a)$ as the empirical distribution built using $N_t(s,a)$ i.i.d. observations from $p(\cdot|s,a)$. We further define $N_t(s,a)^+ := \max\{N_t(s,a), 1\}$.

2. Background: The UCRL2 Algorithm

Before presenting UCRL3 in Section 3, we briefly present UCRL2 (Jaksch et al., 2010). To this end, let us introduce
the following two sets: For each \((s, a) \in S \times A\),
\[
c^\text{UCRL2}_{t, \delta}(s, a) = \left\{ \mu' \in [0, 1] \mid |\tilde{\mu}_t(s, a) - \mu'| \leq \sqrt{\frac{3.5 \log \left( \frac{2SA}{\delta} \right)}{N_t(s, a)}} \right\},
\]
\[
c^\text{UCRL2}_{t, \delta}^{\text{opt}}(s', s, a) = \left\{ \mu' \in \Delta_S : \|\tilde{\mu}_t(s, a) - \mu'\|_1 \leq \sqrt{\frac{14SL \log \left( \frac{2A^2}{\delta} \right)}{N_t(s, a)}} \right\}.
\]

At a high level, UCRL2 maintains the set of MDPs \(M_{t, \delta} = \{\hat{M} = (S, A, \tilde{\pi}, \tilde{\nu})\}\), where for each \((s, a) \in S \times A\), \(\tilde{\mu}(s, a) \in c^\text{UCRL2}_{t, \delta}(s, a)\) (with \(\tilde{\mu}\) denoting the mean of \(\tilde{\nu}\)) and \(\tilde{\pi}(s, a) \in c^\text{UCRL2}_{t, \delta}^{\text{opt}}(s', s, a)\). It then implements the optimistic principle by trying to compute \(\tilde{\pi}^+_t\) approximately by \(\arg\max_{\pi \in \Delta_A} \max \{\tilde{\mu}^\pi_M : M \in M_{t, \delta}\}\), where \(\tilde{\mu}^\pi_M\) is the average-gain for policy \(\pi\) in MDP \(M\). This is carried out approximately by EVI that builds a near-optimal policy \(\pi^+_t\) and an MDP \(\hat{M}_t\) such that \(\tilde{\mu}^\pi_{\hat{M}_t} \geq \max_{\pi, M \in M_{t, \delta}} \tilde{\mu}^\pi_M - \frac{1}{\sqrt{t}}\).

Finally, UCRL2 does not recompute \(\pi^+_t\) at each time step. Instead, it proceeds in internal episodes, indexed by \(k \in \mathbb{N}\), where a near-optimistic policy \(\pi^+_k\) is computed only at the starting time of each episode. Letting \(t_k\) denote the starting time of episode \(k\), the algorithm computes \(\pi^+_k := \pi^+_{t_k}\) and applies it until \(t = t_k + 1\), where the sequence \((t_k)_{k \geq 1}\) is defined as follows: \(t_1 = 1\), and for all \(k > 1\),
\[
t_k = \min \left\{ t > t_{k-1} : \max_{s, a} \frac{\nu_{t_{k-1}:t}(s, a)}{N_{t_{k-1}}(s, a)} \geq 1 \right\},
\]
where \(\nu_{t_{k-1}:t}(s, a)\) denotes the number of observations of pair \((s, a)\) between time \(t_1\) and \(t_2\), and where we recall that for \(x \in \mathbb{N}\), \(x^+ := \max\{x, 1\}\). The EVI algorithm writes as presented in Algorithm 1.

**Algorithm 1** Extended Value Iteration (EVI)

**Input:** \(\epsilon_t\)

1. Let \(u_0 = 0, u_{-1} \equiv -\infty, n = 0\)
2. While \(E(u_n - u_{n-1}) > \epsilon_t\) do
3. Compute \(\left\{ \mu^+_t : s, a \mapsto \max_{\mu' \in \mu^+_t} \left| \tilde{\mu}(s, a) - \mu' \right| \leq \frac{1}{\sqrt{t}} \right\}\)
4. Compute \(\left\{ p^+_t : s, a \mapsto \arg\max_{P_n u_n \in \mu^+_t} \left| P_n u_n(s, a) - a \in A \right\}\)
5. Update \(\left\{ \pi^+_t(s, a) = \max\{\mu^+_t(s, a) + P_n u_n(s, a) : a \in A\} \right\}\)
6. \(n = n + 1\)
7. End while

### 3. The UCRL3 Algorithm

In this section, we introduce the UCRL3 algorithm, a variant of UCRL2 that relies on two main ideas motivated as follows:

(i) While being a theoretically appealing strategy, UCRL2 suffers from conservative confidence intervals, yielding bad empirical performances. Indeed the random stopping times \(N_t(s, a)\) are handled using simple union bounds, causing large confidence bounds. The first modification we introduce has thus the same design as UCRL2, but it replaces these confidence bounds with tighter time-uniform concentration inequalities. Further, unlike UCRL2, it does not use the \(L_1\) norm to define the confidence bound of transition probabilities \(p\). Rather it defines confidence bounds for each transition probability \(p(s' | s, a)\), for each pair \((s, a)\), similarly to SCAL or UCRL2B. Indeed one drawback of \(L_1\) confidence bounds is that they require an upper bound on the size of the support of the distribution. Without further knowledge, only \(S\) can be provided. In UCRL2, this causes the factor \(S\) to appear inside the square-root, due to a union bound over \(2^S\) terms. Deriving \(L_1\) confidence bounds adaptive to the support size seems challenging. In stark contrast, entry-wise confidence bounds can be used without knowing the support: when \(p(s' | s, a)\) has a support much smaller than \(S\), this may lead to a substantial improvement. Hence, the modified strategy UCRL3 relies on time-uniform Bernoulli concentration bounds (presented in Section 3.1 below).

(ii) In order to further tighten exploration, the second idea is to revisit EVI to compute a refined optimistic policy at each round. Indeed, the optimization procedure used in EVI considers all plausible transition probabilities without support restriction, causing unwanted exploration. We introduce a novel value iteration procedure, which we call EVI-NOSS, which uses a restricted support optimization, where the considered support is chosen adaptively in order to retain near-optimistic guarantees.

We discuss these two modifications below in greater detail.

#### 3.1. Confidence Bounds

We now introduce the following high probability confidence sets for the mean rewards: For each \((s, a) \in S \times A\),
\[
c_{t, \delta}(s, a) = \left\{ \mu' \in [0, 1] : |\tilde{\mu}_t(s, a) - \mu'| \leq \frac{b^r_{t, \delta}(S, A)}{N_t(s, a)} \right\},
\]
where we introduced the notation
\[
b_{t, \delta}(S, A) = \max \left\{ \frac{1}{2} \delta_{N_t(s, a)} \left( \frac{\delta_t}{2A^2} \right), \sqrt{\frac{2\tilde{\sigma}^2_t(s, a) \ell_{N_t(s, a)}(\delta_t)}{N_t(s, a)}}, \frac{\sqrt{7} \delta_{N_t(s, a)} \delta_t}{3N_t(s, a)} \right\},
\]
with \(\tilde{\sigma}^2_t(s, a)\) denoting the empirical variance of the reward function of \((s, a)\) built using the observations gathered up to time \(t\), and where \(\ell_{\alpha}(\delta) = \eta \log \left( \frac{\log(n) \log(m)}{\log^2(\eta \delta)} \right)\) with \(\eta = 1.12\).

Any \(\eta > 1\) is valid, and \(\eta = 1.12\) yields a small bound.
Gaussian distributions\(^6\), modified to hold for an arbitrary random stopping time, using the method of mixtures (a.k.a. the Laplace method) from (\textit{Peña et al.}, 2008). This satisfies by construction that
\[
\mathbb{P}\left( \exists t \in \mathbb{N}, (s, a) \in \mathcal{S} \times \mathcal{A}, \mu(s, a) \notin c_{t, \delta_0}(s, a) \right) \leq 3\delta_0.
\]

We recall the proof of this powerful result for completeness in Appendix A. Regarding the transition probabilities, we introduce the two following sets: For each \((s, a, s') \in \mathcal{S} \times \mathcal{A} \times \mathcal{S}\),
\[
C_{t, \delta_0}(s, a, s') = \left\{ q \in [0, 1] : \right. \\
\left. |\hat{p}_t(s'|s, a) - q| \leq \sqrt{\frac{2q(1-q)}{N_t(s,a)}} + \frac{\ell_{N_t(s,a)}(\frac{\delta_0}{\mathcal{S}\mathcal{A}})}{3N_t(s,a)}, \right.
\]
and
\[
\left. -\sqrt{g(q)} \leq \hat{p}_t(s'|s, a) - q \leq g(q) \right\},
\]
where \( g(p) = \begin{cases} g(p) & \text{if } p < 0.5, \\ p(1-p) & \text{else}, \end{cases} \) with \( g(p) = \frac{1 - 2p}{\log(1/1-p)}. \)
The first inequality comes from Bernstein concentration inequalities, modified using a peeling technique in order to handle arbitrary random stopping times. We refer the interested reader to (\textit{Maillard}, 2019) for the generic proof technique behind this result. In (\textit{Dann et al.}, 2017), the authors used similar proof techniques for Bernstein’s concentration, however the resulting bounds are looser; we discuss this more in Appendix A.3. The last two inequalities are obtained by applying again the method of mixture (a.k.a. the Laplace method) for sub-Gaussian random variables, with a modification: Indeed Bernoulli random variables are not only 1/2-sub-Gaussian, but satisfy a stronger sub-Gaussian tail property, already observed in (\textit{Berend & Kontorovich}, 2013; \textit{Raginsky & Sason}, 2013). We discuss this in great detail in Appendix A.2.

\textbf{UCRL3} finally considers the set of plausible MDPs \( \mathcal{M}_{t, \delta} = \{ \tilde{M} = (\mathcal{S}, \mathcal{A}, \tilde{p}, \tilde{v}) \} \), where for each \((s, a) \in \mathcal{S} \times \mathcal{A}\),
\[
\tilde{\mu}(s, a) \in c_{t, \delta_0}(s, a), \quad \text{if } \mu(s, a) \notin c_{t, \delta_0}(s, a),
\]
\[
\tilde{p}(s'|s, a) \in C_{t, \delta_0}(s, a) = \left\{ \tilde{p}' \in \Delta_{\mathcal{S}} : \forall s', \tilde{p}'(s') \in C_{t, \delta_0}(s, a, s') \right\}.
\]

Finally, the confidence level is chosen as\(^7\) \( \delta_0 = \delta/(3 + 3\mathcal{S}) \).

\textbf{Lemma 1 (Time-uniform confidence bounds)} For any MDP with rewards bounded in \([0, 1]\), mean function \( \mu \) and transition function \( p, \) for all \( \delta \in (0, 1), \) it holds
\[
\mathbb{P}\left( \exists t \in \mathbb{N}, (s, a) \in \mathcal{S} \times \mathcal{A}, \mu(s, a) \notin c_{t, \delta_0}(s, a) \text{ or } p(\cdot|s, a) \notin C_{t, \delta_0}(s, a) \right) \leq \delta.
\]

\textbf{3.2. Near-Optimistic Support-Adaptive Optimization}

Last, we revisit the EVI procedure of \textit{UCRL2}. When computing an optimistic MDP, EVI uses for each pair \((s, a)\) an optimization over the set of all plausible transition probabilities (that is, over all distributions \( q \in C_{t, \delta}(s, a) \)). This procedure comes with no restriction on the support of the considered distributions. In the case where \( p(\cdot|s, a) \) is supported on a sparse subset of \( \mathcal{S} \), this may however lead to computing an optimistic distribution with a large support, which in turn results in unnecessary exploration, and thereby degrades the performance. The motivation to revisit \textit{EVI} is to provide a more adaptive way of handling sparse supports.

Let \( \tilde{\mathcal{S}} \subset \mathcal{S} \) and \( f \) be a given function (intuitively, the value function \( u_i \) at the current iterate \( i \) of \textit{EVI}), and consider the following optimization problem for a specific state-action pair \((s, a)\):
\[
\mathcal{J}_{s,a}(\tilde{\mathcal{S}}) = \max_{\tilde{p} \in \mathcal{X}} \sum_{s' \in \tilde{\mathcal{S}}} f(s')\tilde{p}(s'), \quad \text{where} \quad (2)
\]
\[
\mathcal{X} = \left\{ \tilde{p} : \forall s' \in \tilde{\mathcal{S}}, \tilde{p}(s') \in C_{t, \delta}(s, a, s') \text{ and } \sum_{s' \in \tilde{\mathcal{S}}} \tilde{p}(s') \leq 1 \right\}.
\]

\textbf{Remark 1 (Optimistic value)} The quantity \( \mathcal{J}_{s,a}(\tilde{\mathcal{S}}) \) is conveniently defined by an optimization over positive measures whose mass may be less than 1. The reason is that \( p(\tilde{\mathcal{S}}|s, a) \leq 1 \) in general. This ensures that \( p(\cdot|s, a) \in \mathcal{X} \) indeed holds with high probability, and thus \( \mathcal{J}_{s,a}(\tilde{\mathcal{S}}) \geq \sum_{s' \in \tilde{\mathcal{S}}} f(s'|s, a) \) as well.

The original \textit{EVI} (Algorithm 1) computes \( \mathcal{J}_{s,a}(\mathcal{S}) \) for the function \( f = u_i \) at each iteration \( i \). When \( p(\cdot|s, a) \) has a sparse support included in \( \tilde{\mathcal{S}}, \) \( C_{t, \delta}(s, a, s') \) often does not reduce to \( \{ 0 \} \) for \( s' \notin \tilde{\mathcal{S}} \), while one may prefer to force a solution with a sparse support. A naïve way to proceed is to define \( \tilde{\mathcal{S}} \) as the empirical support (i.e., the support of \( \tilde{p}_t(\cdot|s, a) \)). Doing so, one however solves a different optimization problem than the one using the full set \( \mathcal{S} \), which means we may lose the optimistic property (i.e., \( \mathcal{J}_{s,a}(\tilde{\mathcal{S}}) \geq \mathbb{E}_{X \sim p(|s, a)}[f(X)] \) may not hold) and get an uncontrolled error. Indeed, the following decomposition
\[
\mathbb{E}_{X \sim p}[f(X)] = \sum_{s' \in \tilde{\mathcal{S}}} f(s')p(s') + \sum_{s' \notin \tilde{\mathcal{S}}} f(s')p(s'),
\]
shows that computing an optimistic value restricted on \( \tilde{S} \) only upper bounds the first term in the right-hand side. The second term (the error term) needs to be upper bounded as well. Consider a pair \((s, a), t \geq 1, \) and let \( n := N_t(s, a) \). Provided that \( \tilde{S} \) contains the support of \( \tilde{p}_n \), thanks to Bernstein’s confidence bounds, it is easy to see⁸ that the first term in the above decomposition contains terms scaling as \( \tilde{O}(n^{-1/2}) \), while the error term contains only terms scaling as \( \tilde{O}(n^{-1}) \). On the other hand, the error term sums \(|S|/|\tilde{S}| \) many elements, which can be large in case \( \rho \) is sparse, and thus may even exceed \( \tilde{T}_{s,a}(\tilde{S}) \) for small \( n \). To ensure the error term does not dominate the first term, we introduce the Near-Optimistic Support-adaptive Optimization (NOSS) procedure, whose generic pseudo-code is presented in Algorithm 2. For instance, for a given pair \((s, a)\) and time \( t \), NOSS takes as input a target function \( f = u_t \) (i.e., the value function at iterate \( t \)), the support \( \hat{S} \) of the empirical distribution \( \hat{p}_t(\cdot | s, a) \), high-probability confidence sets \( C := \{ C_{t,s,a}(s,a,s', s' \in \tilde{S}) \} \) and a parameter \( \kappa \in (0, 1) \). It then adaptively augments \( \tilde{S} \) in order to find a set \( \hat{S} \), whose corresponding value function \( \tilde{T}_{s,a}(\hat{S}) \) is near-optimistic, as formalized in the following lemma:

**Algorithm 2 NOSS**

Let \( \tilde{S} = \tilde{S} \cup \arg\max_{s \in \tilde{S}} f(s) \).

Define \( \tilde{T} \) using \( f \) and confidence sets \( C \) (see (2)).

while \( \tilde{T}(\tilde{S} \setminus \tilde{S}) \geq \min(\kappa, \tilde{T}(\tilde{S})) \) do

Let \( \tilde{s} \in \arg\max_{s \notin \tilde{T}} f(s) \)

\( \tilde{S} = \tilde{S} \cup \{ \tilde{s} \} \)

end while

return \( \tilde{S} \)

**Algorithm 3 EVI-NOSS**

Let \( \bar{u}_0 \equiv 0, u_{-1} \equiv -\infty, n = 0 \)

while \( \bar{S}(u_n - u_{n-1}) > \varepsilon \) do

Compute for all \((s, a), \)

\( \tilde{S}_{s,a} = \text{NOSS}((u_n - \min_{u_n} u_n, \sup(s'|s, a), \mathcal{C}, \kappa)), \) with

\( \kappa = 105(u_n)/\sup(s'|s, a))/\sqrt{n/2} \)

\( \bar{C}(s, a) = (p' \in \bar{C}(s, a) : p'(x) = 0, \forall x \in \tilde{S}_{s,a}) \)

Compute

\( \begin{cases} a^*: s, a \mapsto \max\{a': \bar{C}(s, a), P_{u_n}^* : s, a \} \\ P_{u_n}^* : s, a \mapsto \arg\min\{p' \in \bar{C}(s, a) \} \end{cases} \)

Update

\( u_{n+1}(s) = \max\{u^*(s, a) + P_{u_n}^* u_n(s, a), a \in A\} \)

\( \pi_{n+1}(s) \in \text{Argmax}\{u^*(s, a) + P_{u_n}^* u_n(s, a), a \in A\} \)

\( n = n + 1 \)

end while

**Lemma 2 (Near-optimistic support selection)** Let \( \tilde{S} \) be a set output by NOSS. Then, with probability higher than \( 1 - \delta \),

\[ \tilde{T}_{s,a}(\tilde{S}) \geq \mathbb{E}_{X \sim P(s|s, a)}[f(X)] - \min\{\kappa, \tilde{T}_{s,a}(\tilde{S}), \tilde{T}_{s,a}(\tilde{S} \setminus \tilde{S})\} \]

In other words, the value function \( \tilde{T}_{s,a}(\tilde{S}) \) is near-optimistic.

**Near-optimistic value iteration: The EVI-NOSS algorithm.** In UCRL3, we thus naturally revisit the EVI procedure and combine the following step in EVI:

\[ p_n^+ : s, a \mapsto \arg\max\{P^* u_n, p' \in \tilde{C}_{t,\delta}(s, a)\} \]

with NOSS: For a state-action pair \((s, a)\) and at each iterate \( n \) of EVI, UCRL3 applies NOSS (Algorithm 2) to the function \( u_n - \min_{u_n} u_n(s) \) (that is, the relative optimistic value function), and empirical distribution \( \tilde{p}_t(\cdot | s, a) \). We refer to the resulting algorithm as EVI-NOSS, as it combines EVI with NOSS, and present its pseudo-code in Algorithm 3. Finally, for iterate \( n \) in EVI-NOSS, we set the value of \( \kappa \) to

\[ \kappa = \kappa_{t,n}(s, a) = \frac{\gamma S(u_n)}{\max_{s,a} N_t(s, a)^{2/3}} \]

The scaling with the size of the support and span of the considered function is intuitive. The reason to further normalize by \( \max_{s', a} N_t(s', a)^{2/3} \) is to deal with the case when \( N_t(s, a) \) is small: First, in the case of Bernstein’s bounds, and since \( \tilde{S} \) contains at least the empirical support, \( \min \{\tilde{T}_{s,a}(\tilde{S}), \tilde{T}_{s,a}(\tilde{S} \setminus \tilde{S})\} \) should essentially scale as \( \tilde{O}(N_t(s, a)^{-1}) \). Hence for pairs such that \( N_t(s, a) \) is large, \( \kappa \) is redundant. Now for pairs that are not sampled a lot, \( N_t(s, a)^{-1} \) may still be large even for large \( t \), resulting in a possibly uncontrolled error. Forcing a \( \max_{s,a} N_t(s, a)^{2/3} \) scaling ensures the near-optimality of the solution is preserved with enough accuracy to keep the cumulative regret controlled. This is summarized in the following lemma, whose proof is deferred to Appendix B.

**Lemma 3 (Near-optimistic value iteration)** The EVI-NOSS algorithm satisfies that, using the stopping criterion \( \tilde{S}(u_{n+1} - u_n) \leq \varepsilon \), the average-reward gain \( g_{n+1} \) corresponding to the policy \( \pi_{n+1} \) and the MDP \( M = (\mathcal{S}, \mathcal{A}, \mu_{\pi_{n+1}}, P_{\pi_{n+1}}^+) \) computed at the last iteration \( n + 1 \), is near-optimistic, in the sense that with probability higher than \( 1 - \delta \), uniformly over all \( t \), \( g_{n+1} \geq g^* - \varepsilon - \bar{r} \), where \( \bar{r} = \frac{\gamma \sum_{u_n} K}{\max_{s,a} [N_t(s, a)^{2/3}]^{2/3}} \).

The pseudo-code of UCRL3 is provided in Algorithm 4.

### 3.3. Regret Bound of UCRL3

We are now ready to present a finite-time regret bound for UCRL3. Before presenting the regret bound in Theorem 1 below, we introduce the notion of local effective support. Given a pair \((s, a)\), we define the local effective support \( L_{s,a} \) of \((s, a)\) as:

\[ L_{s,a} := \left( \sum_{x \in \tilde{S}} \sqrt{p(x|s, a)(1 - p(x|s, a))} \right)^2 , \]
1, the latter algorithm attains a regret bound of

\[ \sum_{s,a} \min \{ D_s^{2/3}, T^{1/3} \} + O(D^2 A \log^2 (T \delta / T)). \]

We now discuss the regret bound of \textsc{UCRL3} with respect to that of \textsc{UCRL2B}. As shown in Table 1, the latter algorithm attains a regret bound of \( O(\sqrt{D \sum_{s,a} K_{s,a} T \log(T)} \log(T) / \delta) \). The two regret bounds are not directly comparable: The regret bound of \textsc{UCRL2B} depends on \( \sqrt{D} \) whereas that of \textsc{UCRL3} has a term scaling as \( D \). However, the regret bound of \textsc{UCRL2B} suffers from an additional \( \sqrt{\log(T)} \) term. Let us compare the two bounds for MDPs where quantities such as \( K_{s,a} \),

We recall that for a pair \( (s,a) \), we define \( K_{s,a} := \text{supp}(p(\cdot | s,a)) \), and denote its cardinality by \( K_{s,a} \).

<table>
<thead>
<tr>
<th>( S )</th>
<th>( D )</th>
<th>( \min { D_s^{2/3}, T^{1/3} } )</th>
<th>( \max { D_s^{2/3}, T^{1/3} } )</th>
<th>( \min { D_s^{2/3}, T^{1/3} } )</th>
<th>( \max { D_s^{2/3}, T^{1/3} } )</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>14.72</td>
<td>1.67</td>
<td>6.67</td>
<td>0</td>
<td>1.40</td>
</tr>
<tr>
<td>12</td>
<td>34.72</td>
<td>1.67</td>
<td>6.67</td>
<td>0</td>
<td>1.40</td>
</tr>
<tr>
<td>20</td>
<td>61.39</td>
<td>1.67</td>
<td>6.67</td>
<td>0</td>
<td>1.40</td>
</tr>
<tr>
<td>40</td>
<td>128.06</td>
<td>1.67</td>
<td>6.67</td>
<td>0</td>
<td>1.40</td>
</tr>
<tr>
<td>70</td>
<td>228.06</td>
<td>1.67</td>
<td>6.67</td>
<td>0</td>
<td>1.40</td>
</tr>
<tr>
<td>100</td>
<td>328.06</td>
<td>1.67</td>
<td>6.67</td>
<td>0</td>
<td>1.40</td>
</tr>
</tbody>
</table>

Table 1. Problem-dependent quantities for \( S \)-state RiverSwim.

\( L_{s,a} \) and \( D_s \) are local parameters in the sense that they do not scale with \( S \), but where \( D \) could grow with \( S \). For instance, in 100-state RiverSwim — in other words, \( K_{s,a}, L_{s,a}, \) and \( D_s \) scale as \( o(S) \).

In the first set of experiments, we consider a \( S \)-state RiverSwim environment (corresponding to the MDP shown in Figure 2). To better understand Theorem 1 for such environments, we report in Table 1 a computation of some of the key quantities appearing in the regret bounds, as well as the diameter \( D \), for several values of \( S \).

We further provide in Table 2 a computation of the leading terms of several regret analyses. More precisely, for a given algorithm \( \tilde{A} \), we introduce \( \mathfrak{R}(\tilde{A}) \) to denote the regret bound normalized by \( \sqrt{T \log(T) / \delta} \) ignoring universal constants. For instance, \( \mathfrak{R}(\textsc{UCRL2B}) = D \sqrt{SA} \). In Table 2, we compare \( \mathfrak{R} \) for various algorithms, for \( S \)-state RiverSwim for several values of \( S \). Note that \( \mathfrak{R}(\textsc{UCRL2B}) \) grows with \( T \) unlike \( \mathfrak{R} \) for \( \textsc{UCRL2, SCAL+} \), and \textsc{UCRL3}. Note that even

\( A \Gamma \) Ignoring universal constants here provides a more fair comparison; for example the final regret bound of \textsc{UCRL2} has no second-order term at the expense of a rather large universal constant. Another reason in doing so is that for \textsc{UCRL2B} and \( \textsc{SCAL+} \), universal constants in their corresponding papers are not reported.

\[ \mathfrak{R}(\textsc{UCRL2B}) \]
We further provide results in larger MDPs. We consider two
starts learning after close to 10
mic scale). Figure 4 shows similar results on larger
25
UCRL2
KL-UCRL
UCRL3
3
0.2
0.8
6
800000
200000

We provide below, an illustration of a randomly-generated
MDP, with 15 states and 3 actions (blue, red, green). Such
an MDP is a type of Garnet (Generalized Average Reward
Non-stationary Environment Test-bench) (Bhatnagar et al.,
2009), in which we can specify the number of states, actions,
average size of the support of transition distributions, spar-
sity of the reward function, as well as the minimal non-zero
probability mass and minimal non-zero mean-reward.

### Table 2. Comparison of the quantity \( \bar{\mathbb{R}} \) of various algorithms for 6-state RiverSwim:

<table>
<thead>
<tr>
<th>S</th>
<th>( \bar{\mathbb{R}}(\text{SCAL}^+) )</th>
<th>( \bar{\mathbb{R}}(\text{UCRL}2) )</th>
<th>( \bar{\mathbb{R}}(\text{UCRL}2B) )</th>
<th>( \bar{\mathbb{R}}(\text{UCRL}3) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>124.9</td>
<td>69.1</td>
<td>38.6</td>
<td>30.0</td>
</tr>
<tr>
<td>12</td>
<td>589.3</td>
<td>235.5</td>
<td>85.8</td>
<td>59.5</td>
</tr>
<tr>
<td>20</td>
<td>1736.3</td>
<td>542.2</td>
<td>148.5</td>
<td>94.9</td>
</tr>
<tr>
<td>40</td>
<td>7243.9</td>
<td>1699.6</td>
<td>305.3</td>
<td>176.9</td>
</tr>
<tr>
<td>70</td>
<td>22576.2</td>
<td>3802.4</td>
<td>540.0</td>
<td>293.6</td>
</tr>
<tr>
<td>100</td>
<td>46394.3</td>
<td>6544.7</td>
<td>775.3</td>
<td>407.6.2</td>
</tr>
</tbody>
</table>

choosing \( T \) as small \( T = 100 \), and ignoring universal con-
stants (which disadvantage \( \text{UCRL3} \)), we observe a benefit.

In Figure 3, we plot the regret under \( \text{UCRL2} \), \( \text{KL-UCRL} \),
\( \text{UCRL2B} \), and \( \text{UCRL3} \) examined in a RiverSwim environ-
ment with \( S = 6 \) states. The curves show the results aver-
aged over 50 independent runs along with the first and third
quartiles. We observe that \( \text{UCRL3} \) achieves the smallest
regret amongst these algorithms and significantly outperforms
\( \text{UCRL2} \), \( \text{KL-UCRL} \) and \( \text{UCRL2B} \) (note the logarith-
mic scale). Figure 4 shows similar results on larger 25-state
RiverSwim environment; On this environment, \( \text{UCRL2} \) only
starts learning after close to \( 10^7 \) time steps.

We further provide results in larger MDPs. We consider two
frozen lake environments of respective sizes \( 7 \times 7 \) and \( 9 \times 11 \)
as shown in Figure 5, yielding MDPs with \( A = 4 \) actions
and \( S = 20 \) states (respectively \( S = 55 \)) when taking off
walls. In such grid-worlds, the learner starts in the upper-left
corner. A reward of 1 is placed in the lower-right corner, when
this rewarding state is reached, the learner is sent back to the initial state. The learner can perform 4 actions:
Going up, left, down, or right, there are probabilities of
0.1 to stay in the same state, and 0.1 to go in each of the
two perpendicular directions (or stay if this leads to a wall),
giving a 0.7 probability to go in the chosen direction.

**Remark 2** Importantly, \( \text{UCRL2} \) and its variants are
generic purpose algorithms, and as such are not aware
of the specific structure of the MDP, such as being a grid-
world. In particular, no prior knowledge is assumed on the
support of the transition distributions by any of the algo-
rithms, which makes it a highly non-trivial learning task,
since the number of unknowns (i.e., problem dimension) is
then \( S^2 A (S A(S - 1)) \) for the transition function, and \( S A \)
for the rewards). For instance, a 4-room MDP is really seen
as a problem of dimension 1600 by these algorithms, and a
2-room MDP as a problem of dimension 12100.

Figures 6 (respectively 7) show the regret performance of
\( \text{UCRL2} \), \( \text{KL-UCRL} \), \( \text{UCRL2B} \), and \( \text{UCRL3} \) on these 2-room
(respectively 4-room) grid-worlds MDPs. The full code and
implementation details are made available to the community.
Finally, since all these algorithms are generic-purpose
MDP learners, we provide numerical experiments in a large
randomly-generated MDP consisting of 100 states and 3
actions, hence seen as being of dimension \( 3 \times 10^3 \). \( \text{UCRL3} \)
still outperforms other state-of-the-art strategies by a large
margin consistently in all these environments.

We provide below, an illustration of a randomly-generated
MDP, with 15 states and 3 actions (blue, red, green). Such
an MDP is a type of Garnet (Generalized Average Reward
Non-stationary Environment Test-bench) (Bhatnagar et al.,
2009), in which we can specify the number of states, actions,
average size of the support of transition distributions, spar-
sity of the reward function, as well as the minimal non-zero
probability mass and minimal non-zero mean-reward.
Comparing UCRL3 against UCRL2B in experiments reveal the gain achieved is not only due to Bernstein’s confidence intervals. Let us recall that on top of using Bernstein’s confidence intervals, UCRL3 also uses a refinement using sub-Gaussianity of Bernoulli distributions as well as the EVI-NOSS instead of EVI for planning. Experimental results verify that both tight confidence sets (see also Figure 11 in the appendix) and EVI-NOSS play an essential role in achieving small empirical regret.

5. Conclusion

We studied reinforcement learning in finite Markov decision processes (MDPs) under the average-reward criterion, and introduced UCRL3, a refined variant of UCRL2 (Jaksch et al., 2010), that efficiently balances exploration and exploitation in communicating MDPs. The design of UCRL3 combines two main ingredients: (i) Tight time-uniform confidence bounds on individual elements of transition and reward functions, and (ii) a refined Extended Value Iteration procedure being adaptive to the support of transition function. We provided a non-asymptotic and high-probability regret bound for UCRL3 scaling as \( \tilde{O}\left((D + \sqrt{\sum_{s,a}(D_s^2 L_{s,a} + 1)}) \sqrt{T}\right) \), where \( D \) denotes the (global) diameter of the MDP, \( D_s \) denotes the local diameter of state \( s \), and \( L_{s,a} \) represents the local effective support of transition distribution for state-action pair \((s, a)\). We further showed that \( D_s \leq D \) and that \( L_{s,a} \) is upper bounded by the number of successor states of \((s, a)\), and therefore, the above regret bound improves on that of UCRL2. Through numerical experiments we showed that UCRL3 significantly outperforms existing variants of UCRL2 in standard environments.

Acknowledgement

This work has been supported by CPER Nord-Pas-de-Calais/FEDER DATA Advanced data science and technologies 2015-2020, the French Ministry of Higher Educa-
tion and Research, Inria, and the French Agence Nationale de la Recherche (ANR), under grant ANR-16-CE40-0002 (project BADASS). This work was done when M. S. Talebi was a postdoctoral researcher in Inria Lille – Nord Europe.

References


Tightening Exploration in Upper Confidence RL


