Master research Internship

Internship report

Improved Strategies for Undiscounted Reinforcement Learning

Domain: Machine Learning - Artificial Intelligence

Author:
Hippolyte Bourel

Supervisor:
Odalric-Ambrym Maillard
Sadegh Talebi
Sequel team

Abstract: The upper confidence reinforcement learning (UCRL) strategy is a popular method to perform regret minimization in unknown discrete Markov Decision Processes (MDP) under the average-reward criterion when the learner interacts with the system in a single, infinite stream of observations. Its second version, called UCRL2 was introduced in the seminal paper from Jacksch et al. [1] and lead to several variants, motivated by the strong theoretical regret guarantees of this strategy. We present an exhaustive practical comparison of the State-of-the-Art's improvements of UCRL2, and introduce UCRL3 a refined version of the UCRL2 strategy having similar theoretical guarantees, but reducing the empirical regret by several orders of magnitudes on standard environments. Then we present C-UCRL and its variants that are improvements of UCRL2 exploiting an equivalence structure of the MDP. Finally, we present a short study of fundamental properties of the discrete MDP structure that may lead later to an alternative strategy to UCRL2.

Contents

1 Introduction

Reinforcement Learning (RL) is a specific area of Machine Learning (ML) based on the trial-and-error framework. The main differences between RL and others fields of ML such as Supervised or Unsupervised learning are the idea of reward signal replacing the supervisor, and the fact that the learning agent's actions have an effect on the received data.

The RL theory uses the Bellman operator contraction as a fundamental block. Markov Decision Processes (MDP), which are often used in fundamental RL works, used a discount factor $\gamma < 1$ to artificially reinforce the contraction but the contraction may also happen while $\gamma = 1$ which is the context of undiscounted MDPs where our work takes place.

1.1 The Reinforcement Learning problem

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 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. Finally, ν is a reward distribution function on $[0,1]$ with mean function denoted by μ.

The game proceeds as follows. The learner starts in some state $s_1 \in \mathcal{S}$ at time $t = 1$. At each time step $t \in \mathbb{N}$, the learner chooses one action $a \in \mathcal{A}$ in it's current state s_t based on it's 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. We refer to standard textbooks [2, 3] for background material on RL and MDPs.

The goal of the learner is to maximize the *cumulative reward* gathered in the course of interaction with the environment. As transition kernel p and reward function ν are unknown, 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*[1], which compares the cumulative reward gathered by an oracle, being aware of p and ν, to that gathered by the learner. More formally, under a learning algorithm $\mathbb{A}$, the regret is defined as

$$\mathfrak{R}(\mathbb{A}, T) := Tg^\star - \sum_{t=1}^{T} r_t(s_t, a_t)\,,$$

where $g^\star$ denotes *the optimal gain* in M, and where $(s_t, a_t)_t$ denotes a trajectory followed by the algorithm.

Add the definition of gain..

1.1.1 Additional definitions

We define a policy π as the function $\pi : \mathcal{A} \times \mathcal{S} \to [0,1]$ such as, having at time step t the state s_t and the action a_t:

$$\pi(a|s) = \mathbb{P}(a_t = a | s_t = s).$$

[1] We note that in the discounted setting, the quality of the learning algorithm is usually assessed through the notion of *sample complexity* as defined in [4].

It is the probability to chose the action a being in the state s. In most of the algorithm presented in this paper we only consider discrete policy where for a given $s \in \mathcal{S}$ there exist $a \in \mathcal{A}$ such that $\pi(a|s) = 1$.

Then we have the following definitions of MDP families on which we work.

Definition 1 (Communicating MDP) *An MDP $\mathcal{M}$ is communicating if for all pair of states (s_0, s_1) there exists a policy π which allows going from s_0 to s_1 with a non zero probability.*

Definition 2 (Ergodic MDP) *An MDP $\mathcal{M}$ is ergodic if for all pair of states (s_0, s_1) and for all policies there is non zero probability to go from s_0 to s_1. It implies that an ergodic MDP is communicating.*

We may remark that in practice the restriction to ergodic MDP is a serious limitation and cannot be applied in most real-life problems. About the communicating MDP, the restriction is much softer, we just want to be able to go in any state (even if it is difficult).

An other important notion in the context of RL is the value function. The value function, for a given policy π, denoted $V^\pi : \mathcal{S} \to \mathbb{R}$ is an estimation of the future reward for a given state s_t, such as:

$$V^\pi(s_t) = \mathbb{E}_\pi \left[\sum_{k=1}^{\infty} r(s_{t+k}, a_{t+k}) \right].$$

Finally, we introduce the notion of *optimal policy*, denoted π^*, which is one of the policies allowing to maximize the chosen notion of cumulative reward over the time, in our case the average reward criterion.

1.2 Contribution

To start this internship report we present an exhaustive State of the Art around the aforementioned RL problem. Then we make the following contributions: We present UCRL3, a modification of the UCRL2 algorithm from [1] using tighter confidence bounds on components of the transition probability similarly to [5]. These confidence bounds are uniform in time and are of independent interest. We present in Section B.1 such refined concentration inequalities. We further integrate a refinement of the inner maximization of the Extended Value Iteration, by considering a near-optimistic, as opposed to a fully optimistic optimization step. Then we propose a section discussing improvement of UCRL2 using an equivalence structure, this work has been done in collaboration with Mahsa Asadi and my supervisors and has recently been submitted and accepted to the 11th Asian Conference on Machine Learning (ACML 2019), the two first subsection of section 4 and the subsection 5.4 come from this submission. Finally, we propose a short study of specific properties of MDPs that could lead in further work to alternatives learning strategy to the one proposed by UCRL.

2 State of the Art

2.1 UCRL2 algorithm

In this subsection we will present the UCRL2 algorithm introduced by [1] in 2010, this algorithm adopts the paradigm of "optimism in face of uncertainty" proposed by its predecessors. It is based on the previous work on the Upper-Confidence Reinforcement Learning (UCRL) algorithm [6] itself based on the Upper-Confidence Bounds (UCB) algorithm [7] from the bandit's literature.

A specificity of the algorithm and its successors compared to other work in the RL field such as [2], is that instead of being focused on the regret of the learned policy obtained at the end of the learning, here the focus is on the regret during the learning, which means the performance of the algorithm during the learning, justifying the previous definition of the regret.

2.1.1 Algorithm

The UCRL2 algorithm uses a notion of diameter of the MDP in order to describe the transition structure of the MDP. Briefly an MDP M has a diameter D if for every pairs of states (s, s') there is a policy going from s to s' in at most D steps, which gives the formal definition:

Definition 3 *Consider the stochastic process defined by a stationary policy $\pi : \mathcal{S} \to \mathcal{A}$ operating on an MDP M with initial state s. Let $T(s'|M, \pi, s)$ be the random variable for the first time step in which state s' is reached in this process. Then the diameter of M is defined as:*

$$D(M) = \max_{s \neq s' \in \mathcal{S}} \min_{\pi:\mathcal{S}\to\mathcal{A}} \mathbb{E}[T(s'|M, \pi, s)]$$

We may observe that a communicating MDP necessarily has a finite diameter.

The UCRL2 algorithm, as UCRL and UCB, implements the paradigm of "optimism in face of uncertainty", that is, using the observations made so far, to determine at each step k a set of statistically possible MDPs and among them to select an optimistic MDP $\tilde{M}_k$ about the achievable average reward. Then the algorithm executes some nearly optimal (for $\tilde{M}_k$) policy $\tilde{\pi}_k$, where k is the current episode, because, in order to reduce the computational cost, the algorithm is decomposed in episodes of arbitrary length (depending on a criterion given in algorithm 1) and computes a new optimistic MDP $\tilde{M}_k$ and a new policy $\tilde{\pi}_k$ at the beginning of each episode k.

From a high-level perspective, we can describe an episode of the algorithm by decomposing it on 3 parts. The goal of the first part is to approximate the reward function and the transition probability using the known observations. Then in the second part, using the previous estimations, a set of plausible MDPs is selected, then the Extended Value Iteration presented in the next sub-section 2.1.2 is used in order to compute a near-optimal policy for an optimistic MDP while selecting this optimistic MDP. Finally, in the third part, the previous policy is executed in the optimistic MDP until a state s is visited and that the action induced by the current policy has been chosen in the current episode k equally often as before this episode. Because of this stopping criterion, the amount of observation is at most doubled at each episode, we will call this stopping criterion the doubling criterion in the following paper. The complete algorithm is presented in algorithm 1.

2.1.2 Extended Value Iteration

The Extended Value Iteration is introduced in order to solve the following problem: in the algorithm, we want to compute a near-optimal policy for an optimistic MDP having a set of plausible MDPs. The Extended Value Iteration allows finding both while a simple value iteration only allows computing an optimal policy for a given MDP.

We give us a set of MDP $\mathcal{M}$ with same set of states $\mathcal{S}$ and same set of actions $\mathcal{A}$, and containing at least one MDP with finite diameter. Another property of these MDPs is that they all have transition probability $\tilde{p}(.|s, a)$ and mean rewards $\tilde{\mu}(s, a)$ for all $(s, a) \in \mathcal{S} \times \mathcal{A}$ such that:

$$||\tilde{p}(.|s, a) - \hat{p}(.|s, a)||_1 \leq d(s, a), \tag{1}$$

Algorithm 1 UCRL2 [1], with input parameter $\delta \in (0, 1]$

Initialize: For all (s, a), set $N_0(s, a) = 0$ and $v_0(s, a) = 0$. Set $t = 1$, $k = 1$, and observe initial state s_1
for episodes $k \geq 1$ **do**
  Set $t_k = t$
  Set $N_k(s, a) = N_{k-1}(s, a) + v_{k-1}(s, a)$ for all (s, a)
  Find a $\frac{1}{\sqrt{t_k}}$-optimal policy $\tilde{\pi}_k$ and an optimistic MDP $\tilde{M}_k \in \mathcal{M}_k$ using EXTENDED VALUE ITERATION
  while $v_k(s_t, a_t) < N_k(s_t, a_t)^+$ **do**
    Play action $a_t = \tilde{\pi}_k(s_t)$, and observe the next state s_{t+1} and reward $r(s_t, a_t)$
    Update $N_k(s, a, x)$ and $v_k(s, a)$ for all actions a and states s, x
  end while
end for

$$|\tilde{\mu}(s, a) - \hat{\mu}(s, a)| \leq d'(s, a) \tag{2}$$

with $\hat{p}(.|s, a)$ given transition probability and $\hat{\mu}$ given mean rewards, d (positive) and d' (positive or null). d and d' are the confidence intervals used by the algorithm while estimating the transition probability and the reward function in the step 3 of figure 1.

Now we introduce the notation $\tilde{M}^+$ designing an MDP which is the combination of all the MDPs in $\mathcal{M}$ with an extended and continuous actions space $\mathcal{A}'$. This action space $\mathcal{A}'$ is build such that for all action $a \in \mathcal{A}$, for all admissible transition probability $\tilde{p}(.|s, a)$ according to equation (1) and for all admissible mean reward $\tilde{\mu}(s, a)$ there is a corresponding action in $\mathcal{A}'$. Having this definition, we have for each policy $\tilde{\pi}^+$ on $\tilde{M}^+$ the existence of policy $\tilde{\pi}$ and an MDP $\tilde{M} \in \mathcal{M}$ such that these policies induce same transition probability and mean reward on there respective MDPs. We also observe the reciprocal of this previous property. Finally having these properties, finding an MDP $\tilde{M} \in \mathcal{M}$, and an optimal policy $\tilde{\pi}$ on it, is equivalent to find one on $\tilde{M}^+$, the EVI is based on this property.

We denote the state value for the i-th iteration by $u_i(s)$. Then we got for undiscounted value iteration (see [3]) on $\tilde{M}^+$:

$$u_0 = 0,$$

$$u_{i+1}(s) = \max_{a \in \mathcal{A}} \left\{ \tilde{\mu}(s, a) + \max_{p(.) \in \mathcal{P}(s,a)} \left\{ \sum_{s' \in \mathcal{S}} p(s') \cdot u_i(s') \right\} \right\} \tag{3}$$

with $\tilde{\mu}(s, a)$ maximum mean reward accepted by 2 and $\mathcal{P}(s, a)$ set of transition probabilities accepted by equation (1). Detail convergence of EVI

We will call inner maximization of the EXTENDED VALUE ITERATION the maximization of the product between p and u_i, this can be computed in $O(S)$ computation step using the algorithm presented in algorithm 2 for a given pair (s, a), then to achieve a step of the EXTENDED VALUE ITERATION we choose the maximum plausible mean reward for (s, a) and perform an exhaustive comparison among the action a. Briefly the idea of the algorithm 2is, at each step i, to reinforce the transition probability for the state s having the best $u_i(s)$ and lowering the transition probability for states with lowest values while staying admissible by equation (1).

Algorithm 2 Algorithm to compute the inner maximization of the EXTENDED VALUE ITERATION (source [1]).

Input: $u(0) \geq u(1) \geq \ldots \geq u(n)$ and d the confidence bound on $\|\tilde{p}(\cdot|s,a) - \hat{p}(\cdot|s,a)\|_1$
output p_{max}
  $p_{max}(0) \leftarrow \min(1, \hat{p}(0|s,a) + d/2)$
  $\forall i \in [1, \ldots, n], p_{max}(i) \leftarrow \hat{p}(i|s,a)$
  $j \leftarrow n$
  while $\sum_{s \in \mathcal{S}} p_{max}(s) > 1$ **do**
    $p_{max}(j) \leftarrow \max(0, p_{max} - \sum_{k \neq j} p_{max}(k))$
    $j \leftarrow j - 1$
  end while

2.1.3 Regret of the UCRL2 algorithm

Bounds on the regret of the UCRL2 algorithm are proposed in its initial paper [1]. We consider that the algorithm is learning on an unknown MDP M, M having a finite diameter D, a number of states $S = |\mathcal{S}|$ and a number of actions $A = |\mathcal{A}|$. Only $\mathcal{S}$ and $\mathcal{A}$ are known by the algorithm, and a confidence parameter δ is given.

Theorem 1 *With probability of at least $1 - \delta$ it holds that for all initial state $s \in \mathcal{S}$ and any $T > 1$, the regret of UCRL2 is bounded by:*

$$\mathfrak{R}(\textit{UCRL2}, T) \leq 34DS\sqrt{AT\log\left(\frac{T}{\delta}\right)}$$

This result is the one we are the most interested in, and it is the result that the following works aim to improve most of the time, with theoretical bound or by experimental results. The theorem implies that the algorithm has a total regret $\tilde{O}(DS\sqrt{AT})$ after T steps. It was a good improvement compared to the state of the art in 2010 and having this control over the total regret of the algorithm motivate our interest in it.

2.2 Improvements and modifications on UCRL2

In this section, we introduce several improvements, of the UCRL2 algorithm, proposed by the literature during the last decade.

2.2.1 KL-UCRL

The first improvement of the UCRL2 algorithm we present is the one introduced in 2010 by [8]. The idea of this improvement is to replace the usage of the L_1 norm while estimating the transition probability in the UCRL2 algorithm by the Kullback-Leibler (KL) divergence. The resulting modified algorithm, called KL-UCRL is proved to verify the same upper bound on the regret that the initial UCRL2 algorithm. Another result of this paper is that in the presented experiments KL-UCRL outperform UCRL2 in terms of regret, particularly when the MDP has reduced connectivity.

The most important modification of KL-UCRL compared to UCRL2 is the modification of the confidence bounds while generating $\mathcal{M}_k$. The new bounds used are the following, for all a and s:

$$\mathcal{K}(\tilde{p}_k(.|s,a), \hat{p}_k(.|s,a)) \leq \frac{C_p}{N_k(s,a)}$$

$$|\tilde{\mu}(s,a) - \hat{\mu}(s,a)| \leq \frac{C_r}{N_k(s,a)}$$

where C_p and C_r are given constants and the other notations are the same as figure 1. And $\mathcal{K}(p,q)$ denote the KL-divergence between p and q, and is such that:

$$\mathcal{K}(p,q) = -\sum_x p(x) \log\left(\frac{q(x)}{p(x)}\right)$$

This usage of the KL-divergence is motivated by multiple observations on UCRL2 essentially based on the fact that the usage of the L_1 norm may compromise the EXTENDED VALUE ITERATION. The first problem is that small changes in the transition probability may bring very different optimistic models. Then, in the worse case, the optimistic model can become incompatible with the observations by assigning, for example, a probability of zero to a transition that has been observed. And the last observed problem is that in MDP with reduced connectivity while using L_1 norm, may result in giving a persistent bonus for some transition to some states with high values even when significant evidence has been accumulated that these transitions are impossible. Using the KL-divergence allows solving these problems, for example, the first one because of the smoothness of the KL-divergence. And the last problem is shown to be solved in the paper, the paper shows that the KL-optimistic model results from a trade-off between the value of a state and its statistical reachability.

To conclude on this paper, just by replacing the L_1 norm in the UCRL2 algorithm by the KL-divergence, and modifying the confidence bounds, allows to obtains better experimental results, the major weakness of this paper is the lack of improvement on the theoretical bounds of the regret of KL-UCRL.

2.2.2 Improvement on regret bound of KL-UCRL

We define an *ergodic MDP* as an MDP in which every policy induces a single recurrent class, which means that it is possible to reach any state from any other state.

The second improvement about UCRL2 we present is immediately based on the previous KL-UCRL algorithm and is presented in [9]. The main result of this paper is an improvement on the KL-UCRL regret bound when applied to ergodic MDPs. The new proposed bound is $\mathcal{O}(\sqrt{S\sum_{s,a}\mathbf{V}^*_{s,a}T} + S\sqrt{T}$. Here, $\mathbf{V}^*_{s,a} = \mathbb{V}_{p(.|s,a)}(b*)$ denotes the variance of the optimal bias function $b^\star$ of the true, and unknown, MDP with respect to next state distribution under the couple of state and action (s, a). Where the bias function, for a given policy π, a given MDP M and a given state s_0 is such that:

$$b^\pi_M(s_0) = C - \lim_{T\to\infty} \mathbb{E}\left[\sum_{t=1}^{T}(r(s_t,a_t) - g^\pi_M(s_t))\right]$$

and the optimal bias is the bias of the optimal policy.

This aforementioned bound on the regret improves the previous bound $\mathcal{O}(DS\sqrt{AT})$ for KL-UCRL as $\sqrt{\mathbf{V}^*_{s,a}} \leq D$. In order to prove this result, a concentration inequality based on the KL-divergence is used. This paper, from 2018, presented in this subsection and its result allows to extend the results from the previous subsection around the KL-UCRL algorithm, but no other improvement, like the improvements presented in next subsections, is used. The main weakness of this result is the restriction to ergodic MDPs.

2.2.3 Improvement using posterior sampling

Now we focus on the improvement proposed by [10] in 2017. In this paper a new version of the UCRL2 algorithm using posterior sampling to compute the optimal policy in each episode is introduced and the main result of the paper is that a regret bound of $\tilde{O}(D\sqrt{SAT})$ for $T \geq S^5A$ is proved but it is important to be aware of the fact that a mistake has been identified in the proof next to the publication of the paper and remains unsolved so far.

Even if the proof does not hold, it is still interesting to consider the approach proposed by the article. Staying at a high level, the idea is to use posterior sampling, a sampling strategy based on Dirichlet distribution properties, to generate samples from the observations made, these samples are used as the set of plausible MDPs, to make it short posterior sampling is used to generate $\mathcal{M}_k$ instead of using the estimates and confidence intervals. Then the algorithm goes back to UCRL2 steps. Because the proof doesn't currently hold results from this paper are quite weak but this approach should be considered in future works.

2.2.4 Modification of the stopping criterion

The improvement presented in this subsection is introduced by the paper [11], this improvement is a modification of the stopping criterion of the algorithm, precisely the stopping criterion for an episode. The original stopping criterion is almost purely quantitative, when we are running a "bad" MDP we continue for a long time because in expectation the length of the episodes grows exponentially. The idea of this new stopping criterion is to check the consistency of the current optimistic MDP $\tilde{M}_k$ according to the observations (see [11]) by hypothesis testing. Intuitively this stopping criterion could easily prevent the algorithm from wasting time on "mistakes" (bad choices of policy and optimistic MDP). Some experimental results are provided in the paper, applying this stopping criterion to UCRL2 and the 3 variations of C-UCRL(see subsection 2.4.2 and section 4), globally it seems to be an improvement in these experiments.

Additional results with this modified stopping criterion are provided in section 5. Formally this stopping criterion is the following test, at time t in episode k that starts at time t_k the episode ends if the classical doubling criterion is reached of if:

$$|\hat{p}_{t_k}(s_t|s_{t-1}, a_{t-1}) - \hat{p}_t(s_t|s_{t-1}, a_{t-1})| > (1+\varepsilon)\beta(\hat{p}_t(s_t|s_{t-1}, a_{t-1}))$$

where $\beta(\hat{p}_t(s_t|s_{t-1}, a_{t-1}))$ is a confidence bound on $\hat{p}_t(s_t|s_{t-1}, a_{t-1})$. and ε a small arbitrary parameter (we use $\varepsilon = 0$ in this paper).

2.2.5 UCRL-Bernstein: Local bounds and Bernstein's bounds

A natural and intuitive way to improve the performance of UCRL2 is to use tighter confidence bounds: if the confidence bounds are tighter the optimistic MDP should be "less" optimistic and closer from the true MDP. Initially UCRL2 use non-optimal union-bounds, but in the literature for example in [12] other confidence bounds are used. In this paper, Bernstein's bounds are used. These bounds are the following (here the bounds are slightly refined using a recent work from my supervisors), for given states s' and s, and given action a:

$$|\widehat{p}_t(s'|s,a) - p(s'|s,a)| \leq \sqrt{\frac{p(s'|s,a)}{N_t(s,a)}\ell_{N_t(s,a)}\left(\tfrac{\delta}{2SA}\right)} + \frac{\ell_{N_t(s,a)}\left(\frac{\delta}{2SA}\right)}{3N_t(s,a)}, \tag{4}$$

with $\ell_n(\delta) := 2\max\log(\log(\max\{e,n\})) + \log(\frac{3}{\delta})$.

We denote by UCRL-Bernstein the variant of UCRL2 using these bounds and the necessary additional modification explained in the following paragraph.

These confidence bounds provide confidence bounds on $|p(s'|s,a) - \hat{p}_t(s'|s,a)|$ instead of bounds on the L_1 norm like in UCRL2 where bounds on $\|p(\cdot|s,a) - \hat{p}_t(\cdot|s,a)\|_1$ are used. In order to deal with this difference it is necessary to modify the inner maximization of the EXTENDED VALUE ITERATION, the maximization used has been introduced in [13] and is also used by [14] where UCRL-Bernstein is implicitly used as the baseline. The pseudo-code of the modification of the inner maximization is provided in algorithm 3. We name this kind of bounds on the component as element-wise bounds in this document, and we denote algorithms using these bounds as local algorithms (UCRL-Bernstein being the first example of such an algorithm).

Algorithm 3 Compute the inner maximum in the EVI with element-wise bounds (for given (s, a)).

Input: $u(0) \geq u(1) \geq \ldots \geq u(n)$ and for all s' $d(s')$ is the confidence bound on $|p(s'|s,a) - \hat{p}(s'|s,a)|$
output p_{max}
  $\delta \leftarrow 1$.
  for $j \in [1, \ldots, n]$ **do**
    $p_{max}(j) \leftarrow \max(0, \hat{p}(j) - d(j))$
    $\delta \leftarrow \delta - p_{max}(j)$
  end for
  $l \leftarrow 0$
  while $\delta > 0$ & $l \leq n-1$ **do**
    $\delta_{new} \leftarrow \min(\delta, \hat{p}(j) + d(j) - p_{max}(j))$
    $p_{max}(j) \leftarrow p_{max} + \delta_{new}$
    $\delta \leftarrow \delta - \delta_{new}$
    $l \leftarrow l + 1$
  end while

2.3 Similar algorithms

Other algorithms, such as REGAL (see section 2.3.1) or PSRL (see section 2.3.2), were introduced at the same period as UCRL2. REGAL allows to take off the dependency on the Diameter D on the regret asymptotic bound, but currently, no implementation of this algorithm exists. PSRL is not based on the optimistic strategy, it outperforms UCRL2 in experimental results presented in [15] but its theoretical guarantees are proved in a restricted problem when the time horizon is bounded by a natural τ which means that after τ time-steps the learner is sent back to the initial state.

2.3.1 REGAL algorithm

The REGAL algorithm introduced by [16] in 2009 is based on the same structure as the UCRL2 algorithm, with an episodic structure that determines $\mathbb{M}_k$ and the optimal policy of the optimistic algorithm. But the big difference between the two algorithms is that REGAL optimizes another problem instead of running the EVI. We will not detail entirely the REGAL algorithm. It is important to notice that there is currently no implementation known for this algorithm (difficulty to implement a solver for the inner optimization problem), explaining the lower impact of this algorithm compared to UCRL2. The most interesting result about REGAL is the bound on the regret depending on a notion

of span instead of the notion of Diameter previously seen. By abuse of language, we name in this paper bias span the span of the optimal bias (defined in sub-subsection 2.2.2).

Definition 4 *We denote b the bias function, so we have $sp\{b\}$ the bias span which is the range of the bias function:*

$$sp\{b\} = \max_s(b(s)) - \min_s(b(s))$$

We admit in this paper that the bias span of an MDP is in general much smaller than its diameter (which could easily be infinite, instead of the span). Then considering the bias span bounded by B the article proves that the regret of REGAL is bounded by $\tilde{O}(BS\sqrt{AT})$ which is really interesting, but the article does not give an implementation of the algorithm, and so far no implementation of it exists.

2.3.2 PSRL

The article [15] introduce the Posterior Sampling Reinforcement Learning (PSRL) algorithm. This algorithm uses the posterior sampling strategy, without the supplementary optimism proposed in [10], this choice is motivated by the good results of this method in the context of multi-armed bandits see for example [17] or [18]. In this article, a bound on the regret of PSRL is proved but in the context of finite time MDPs, this means that there exist some τ such that at time τ an episode end and the learner is sent back to the initial state. This learning context is simpler, so the theoretical guarantees of PSRL are weaker than the one on optimistic strategies. Additionally, the article proposes a few experimental results in a finite and infinite time horizon MDP, in both case PSRL outperform UCRL2.

Because of its weaker theoretical guarantees, our interest was not on PSRL during the internship, no experiments had been run with this algorithm but it will be necessary to compare, in further work, the performances of PSRL and optimistic strategies (such as UCRL3 that we introduce in subsection 3.2) on standard environments. This is not done in the following report but would probably be done by the end of the internship.

2.4 Improvements of UCRL2 exploiting additional knowledge

In this section, we present three algorithms that improve UCRL2 exploiting additional information gave as input. In most of the practical problems, it is possible to give additional knowledge to the learner, for example, the equivalence classes used by C-UCRL will allow the learner to exploit the fact that, for example, walking when there is no obstacle give the same result wherever it is. In other algorithms presented previously we consider that the learner has absolutely no knowledge, the result is that every step is an entire discovery for it, in practice for most of the real problems it would be possible to give to the learner additional information.

2.4.1 UCWM

The first improvement of UCRL2 presented in this section is the Upper Confidence with Model (UCWM) algorithm introduced in [19]. This modification is really simple and is just briefly introduced in the result section of the paper. The idea is to give a set of plausible MDP, the true MDP being inside, as an input of the algorithm. Then while computing $\mathcal{M}_k$ only MDPs in the input set are selected, if only one remains we have the true MDP. Even if this modification is simple it allows good

improvements of the results on the experiments shown in the paper, and intuitively this modification is a good idea when the context allows it.

This improvement of UCRL2 could easily be used in practical problem, it could be used for example to give topological information to the learner: if we consider an environment in which the learner can only perform moving action we know in practice that the learner cannot reach most of the position of the environment (the learner cannot go through walls, or use teleportation to go in the opposite side of a map for example) these are the kind of information that can be used by UCWM, or a very similar algorithm, just be always restricting the set of plausible MDP to the subset of MDP that does not consider impossible movement. This improvement of the algorithm is problem dependent, and the performances depend a lot on the information given as input, in general, it is impossible to prove an improvement on the regret bound of such an algorithm but it should obviously be used in any practical use of any algorithm presented in this paper.

2.4.2 C-UCRL

The paper [11] from 2018 introduces 2 improvements of the UCRL2 algorithm. The first one is the C-UCRL algorithm which is based on a notion of equivalence classes of action and state pairs, the other improvement introduced by this paper is presented in subsection 2.2.4. In this paper, many points about the three variants of C-UCRL were not clearly presented, and the version using clustering to estimate the equivalence classes, while it is unknown, was not working in practice. More work has been done on these algorithms during the internship by Mahsa Asadi, my supervisors and myself, a big part of this work has been published in the 11th Asian Conference on Machine Learning (ACML19) [20]. The section 4 (precisely the two first subsections) and the subsection 5.4 repeat the work presented in this article, and we present some additional results in the third subsection of section 4.

2.4.3 SCAL

Motivated by the results presented on REGAL, having an implementable algorithm with a regret bound based on the bias span instead of the diameter would be very interesting, it is where the article [14] takes place in 2018. The main contribution of this article is the SCAL algorithm inspired both from UCRL2 and REGAL, with implementation and with a regret bound depending on the bias span. The regret bound proved for SCAL is $\mathcal{O}(c\sqrt{\Gamma SAT})$ with c a bound on the span of the optimal bias function and $\Gamma \leq S$ number of possible next states, or support of the transition probability function. An important point about the SCAL algorithm is that the bound on the span c is an input parameter of the algorithm if a false bound is given (smaller than the real span) the algorithm may achieve a linear regret. Additionally, it is not clear that we can have such a tight bound on the span in practice, or without this bound SCAL performs similarly as UCRL2 has shown in subsection 5.5.

The SCAL algorithm is based on the REGAL.c variant of REGAL, see [16]. We briefly present the differences between this algorithm and UCRL2: the first point is that SCAL is based on UCRL-Bernstein instead of UCRL2. Then the optimization problem solved is the following:

$$(\tilde{M}_k, \tilde{\pi}_k) \in \arg\max_{M \in \mathbb{M}_k, \pi \in \Pi_c(M)} \rho(M, \pi, s)$$

where $\Pi_c(M)$ is a policy space defined with a span constraint over the policies (estimated span cannot be bigger than the input parameter c). This problem is highly inspired by the one of REGAL.c. Finally, the last point that needs to be highlighted in SCAL is the use of ScOpt algorithm in order to solve the previous optimization problem. ScOpt can be seen as a modified value iteration

maintaining span constraint (an upper bound) while running. As ScOpt does not obviously similar to its implementation provided in the article we present our implementation based on the one of the paper in appendix A.2.

There are many differences between SCAL and UCRL2, but the global idea remains the same. SCAL is interesting because it manages to bring regret bounds depending on the span as REGAL and SCAL has an implementation which is obviously a good improvement. Experimental results are presented in the paper [14], they show that SCAL outperforms UCRL2 when a "good" input parameter c is given (which is problem dependent) and when the Diameter of the MDP is infinite.

3 Refinements of the UCRL2 algorithm

In this section we introduce two algorithms the first one UCRL2-L is a simple refinement of UCRL2 using the Laplace bounds already used in [11], then we introduced the algorithm UCRL3. UCRL3 is a modification of the UCRL2 algorithm from [1] using tighter element-wise confidence bounds on components of transition probability similarly to UCRL-Bernstein. These confidence bounds are uniform in time and are of independent interest, they come from a work of my supervisor Odalric-Ambrym Maillard. We present in appendix B.1 such refined concentration inequalities. We further integrate a refinement of the inner maximization of the EXTENDED VALUE ITERATION, by considering a near-optimistic, as opposed to a fully optimistic optimization step.

3.1 The UCRL2-L Algorithm

Our first algorithm has the same design as UCRL2: similarly to UCRL2, it uses L_1 confidence bounds for the transition probabilities and computes an optimistic policy using the same EXTENDED VALUE ITERATION algorithm used in UCRL2. However, our algorithm relies on the following confidence bounds:

$$\begin{aligned} &\left\{\tilde{p} : \|\widehat{p}_t(\cdot|s,a) - \tilde{p}(\cdot|s,a)\|_1 \leq \beta_{N_t(s,a)}\left(\tfrac{\delta}{SA}\right),\, \forall s,a\right\}, \\ &\left\{\mu' : |\widehat{\mu}_t(s,a) - \mu'(\cdot|s,a)| \leq \beta'_{N_t(s,a)}\left(\tfrac{\delta}{SA}\right),\, \forall s,a\right\}, \end{aligned} \tag{5}$$

where

$$\beta_n(\delta) = \sqrt{\frac{2(1+\frac{1}{n})\log\left(\sqrt{n+1}\frac{2^S-2}{\delta}\right)}{n}} \quad \text{and} \quad \beta'_n(\delta) = \sqrt{\frac{(1+\frac{1}{n})\log(2\sqrt{n+1}/\delta)}{2n}}\,. \tag{6}$$

These confidence bounds were derived by combining [21] and [22] inequalities with the Laplace method [23, 24], which enables to handle the random stopping times $N_t(s,a)$ in a sharp way; we refer to [25] for further discussion. In particular, this ensures that the true transition function p and mean reward function μ are contained in the confidence bounds with probability at least $1-2\delta$, uniformly over all time t. We refer to this algorithm as UCRL2-L. It has globally been used as baseline algorithm in this work because the bound on the regret of UCRL2 holds immediately for it and it outperforms UCRL2 by several orders of magnitudes on standard environments, but in practice it is almost not used in the literature of the domain where UCRL2 is unfortunately still the standard baseline (or UCRL-Bernstein in [14]).

3.2 The UCRL3 Algorithm

Our second algorithm, which we call the UCRL3 algorithm, is another variant of UCRL2 with two main differences: (i) Unlike UCRL2 (and UCRL2-L), it does not use L_1 norm to define the confidence bound of transition probabilities $p(\cdot|s,a)$. Rather it defines confidence bounds for each transition probability $p(s'|s,a)$. To this end, it relies on tight confidence intervals with the time-uniform Bernoulli concentration bounds (presented in appendix B.1). (ii) As the second difference, UCRL3 relies on a modification of Extended Value Iterationto compute the optimistic policy at each round.

Confidence bounds. UCRL3 uses the following confidence bounds for transition probabilities: For all (s,a,s'), we require that $\tilde{p}(s'|s,a)$ satisfies:

$$|\widehat{p}_t(s'|s,a) - \tilde{p}(s'|s,a)| \leq \sqrt{\frac{\tilde{p}(s'|s,a)}{N_t(s,a)}\ell_{N_t(s,a)}\left(\tfrac{\delta}{2SA}\right)} + \frac{\ell_{N_t(s,a)}\left(\frac{\delta}{2SA}\right)}{3N_t(s,a)}\,, \tag{7}$$

with $\ell_n(\delta) := 2\log(\log(\max\{e,n\})) + \log(\frac{3}{\delta})$, and

$$-\sqrt{\underline{g}(\tilde{p})}\beta'_{N_t(s,a)}\left(\tfrac{\delta}{2SA}\right) \leq \widehat{p}_t(s'|s,a) - \tilde{p}(s'|s,a) \leq \sqrt{g(\tilde{p})}\beta'_{N_t(s,a)}\left(\tfrac{\delta}{2SA}\right), \tag{8}$$

where $\underline{g}(p) = \begin{cases} g(p) & \text{if } p < 1/2 \\ p(1-p) & \text{else} \end{cases}$, and $g(p) = \frac{1/2-p}{\log(1/p-1)}$.

Now, UCRL3 uses the following confidence bounds:

$$\left\{\tilde{p} : (8) \text{ and } (4) \text{ hold} \,\forall s,a,s'\right\}. \tag{9}$$

The element-wise confidence bound in (equation 7) comes from time-uniform concentration bounds of Bernoulli random variables as studied in, e.g., [13], whereas the one in (8) is inspired by the concentration inequalities for Bernoulli random variables in [26, 27]. We refer to Section B.1 for a thorough discussion on this matter. Using both bounds allows to have the best performance in experiments thanks to equation 8 which is much tighter for a reasonable number of samples, and the asymptotic bounds on the regret proved using equation 7 which is most of the time asymptotically better.

Near-optimistic optimization. Now we study how UCRL3 implements the inner maximization in the Extended Value Iteration. The point is that the proof on the regret of the algorithm relies on the optimism performed in this inner maximization, but being less optimistic in a good way could allow having an optimistic MDP closer to the true MDP, which means that the algorithm will perform better. In order to do so, it is interesting to deal with the support of the transition probabilities. In general in all the algorithm presented before being optimistic means to put as much probability mass (in the limit of the concentration bounds) to the states with the highest value, it often means to maximize a transition probability $\tilde{p}(s'|s,a)$ which is supposed to be equal to zero because in most of the problems the support of the transition probability is sparse, in general from a given state s while performing an action a it is truly possible to reach a subset of states $S' \subset \mathcal{S}$ where $|S'| \ll |\mathcal{S}|$, gridworlds used in section 5 are a good example of this assumption.

In order to take into account this property, we introduce the following modification of the inner maximization of the Extended Value Iteration: only transitions that are parts of the empirical

support of the most optimistic transition (which is the transition to the state with the highest value) can be non zero. Doing so we keep enough optimism to prove a bound on the regret (see appendix B.2 for the (partial presented) proof proposed by Odalric-Ambrym Maillard). The resulting pseudo-code for the inner-maximization is the algorithm 4.

Algorithm 4 Compute the inner maximum in the EVI with local bounds (for given (s, a)) and only by modifying the experimental support plus the argmax(V).

Input: $u(0) \geq u(1) \geq \ldots \geq u(n)$ and for all s' $\beta(s')$ is the bound on $|\tilde{p}(s'|s,a) - \hat{p}(s'|s,a)|$
output p_{max}
  $\delta \leftarrow 1$.
  for $j \in [1, \ldots, n]$ **do**
    $p_{\max}(j) \leftarrow \max(0, \hat{p}(j) - \beta(j))$
    $\delta \leftarrow \delta - p_{\max}(j)$
  end for
  $l \leftarrow 1$
  while $\delta > 0$ **and** $l \leq n$ **do**
    if $l = 1$ **or** $\hat{p}(l|s,a) > 0$ **then**
      $\delta_{new} \leftarrow \min(\delta, \hat{p}(l) + \beta(l) - p_{\max}(l))$
      $p_{\max}(l) \leftarrow p_{\max} + \delta_{new}$
      $\delta \leftarrow \delta - \delta_{new}$
    end if
    $l \leftarrow l + 1$
  end while

3.3 Regret Bounds

In this subsection we present regret bounds for UCRL2-L and UCRL3. A modification of the analysis from [1] yields:

Theorem 2 (Regret of UCRL2-L) *With probability higher than $1 - 2\delta$, uniformly over all time horizon T,*

$$\mathfrak{R}(\textit{UCRL2-L}, T) \leq 17DS\sqrt{AT\log(SA\sqrt{T}/\delta)}\,.$$

Wrong (or at least incomplete), update it when the proof is done

Theorem 3 (Regret of UCRL3) *With probability higher than $1 - 2\delta$, uniformly over all time horizon T,*

$$\mathfrak{R}(\textit{UCRL3}, T) \leq 24D\sqrt{KSAT\log(\log(T)/\delta)} + \mathcal{O}(SA\log(T))\,.$$

(The proof of this result is not yet properly wrote, we put in appendix C a partially adapted proof based on the proof of regret provided for C-UCRL in our ACML submission, the complete will be written by the end of the internship).

For UCRL3 the bound is asymptotically worse than usually (we lose the square root on the $\log(T)$), in order to have the best of both world we can perform the inner maximization of the EXTENDED VALUE ITERATION from UCRL2-L in UCRL3 when the following assumption on p_{max} returned by the inner maximization from UCRL3 does not hold:

$$\|\widehat{p}_t(\cdot|s,a) - p_{max}\|_1 \leq \beta_{N_t(s,a)}\left(\tfrac{\delta}{SA}\right) \tag{10}$$

This case correspond to the case where the algorithm with L_1 is less optimistic. In practice, it would almost never append, but it allows to ensure that the regret bound is the minimum of both (and the additional computational cost would be reasonable if applied in practice).

Because of the bounds, motivate the following UCRL3(K) algorithm

3.4 UCRL3(K): bound on the support of transition known

Stronger guarantees on the regret, replace 1 optimism by $\hat{K} - K$ optimism + pseudo-code + regret guarantee

4 C-UCRL: Exploiting equivalence structure of the MDP

In most practical situations, the state space of the underlying MDP is too large but often endowed with some *structure*. Directly applying state-of-the-art RL algorithms, for instance from the above works, and ignoring structure would lead to a prohibitive regret. In this section we are in particular interested in structure in the form of some *equivalence structure* in the state-action space. This is quite typical in many MDPs in various application domains. For instance, in a grid-world MDP when taking action 'up' from state s or 'right' from state s' when both are far from any wall results in similar transitions (typically, move to the target state with probability p and stay still or transit to other neighbors with the remaining probability).

In this section, we first present a notion of similarity between state-action pairs, which naturally yields a *partition* of the state-action space $\mathcal{S} \times \mathcal{A}$, and induces an equivalence structure in the MDP (see Definition 5–6). Then we present confidence bounds that incorporate equivalence structure of transition probabilities into their definition when the learner has access to such information. These confidence bounds are tighter than those obtained by ignoring equivalence structures. For the case of unknown equivalence structure, we present ApproxEquivalence, which uses confidence bounds of various state-action pairs as a proxy to estimate an empirical equivalence structure of the MDP. Finally, to demonstrate the application of the above equivalence-aware confidence bounds, we present C-UCRL, which is a natural modification of the UCRL2 algorithm [1] employing the presented confidence bounds. As shown in Theorem 4, when the learner knows the equivalence structure, C-UCRL achieves a regret which is smaller than that of UCRL2 by a factor $\sqrt{\frac{SA}{C}}$, where C is the number of classes, which can be a massive improvement when $C \ll SA$. We also verify the superiority of C-UCRL over UCRL2 in the case of unknown equivalence structure, through numerical experiments.

This section is almost entirely based on a recent submission to ACML [20]. The subsection 4.4 is the result of collaborative work with Sadegh Talebi, which pursue the aforementioned article presented in ACML, a part of this work could be used in the C-UCRL version with unknown equivalence structure in order to improve it. The work presented in subsection 4.3 is based on UCRL2-L (see subsection 3.1), because it has been done before the work on UCRL3(see subsection 3.2) but all this work could be applied to UCRL3.

4.1 Similarity and Equivalence Classes

We now present a precise definition of the equivalence structure considered in this paper. We first introduce a notion of similarity between state-action pairs of the MDP:

Definition 5 (Similar state-action pairs) *The pair (s', a') is said to be ε-similar to the pair (s, a), for $\varepsilon = (\varepsilon_p, \varepsilon_\mu) \in \mathbb{R}^2_+$, if*

$$\begin{aligned} \|p(\sigma_{s,a}(\cdot)|s,a) - p(\sigma_{s',a'}(\cdot)|s',a')\|_1 &\leq \varepsilon_p\,, \quad \textit{(similar profile)} \\ \textit{and} \qquad |\mu(s,a) - \mu(s',a')| &\leq \varepsilon_\mu\,, \quad \textit{(similar rewards)} \end{aligned}$$

where $\sigma_{s,a} : \{1, \ldots, S\} \to \mathcal{S}$ indexes a permutation of states such that $p(\sigma_{s,a}(1)|s,a) \geq p(\sigma_{s,a}(2)|s,a) \geq \cdots \geq p(\sigma_{s,a}(S)|s,a)$. We refer to $\sigma_{s,a}$ as a **profile mapping** *(or for short, profile) for (s,a), and denote by $\boldsymbol{\sigma} = (\sigma_{s,a})_{s,a}$ the set of profile mappings of all pairs.*

The notion of similarity introduced above naturally yields a *partition* of the state-action space $\mathcal{S} \times \mathcal{A}$, as detailed in the following definition:

Definition 6 (Equivalence classes) *$(0,0)$-similarity is an equivalence relation and induces a canonical partition of $\mathcal{S} \times \mathcal{A}$. We refer to such a canonical partition as* **equivalence classes** *or* **equivalence structure**, *denote it by $\mathcal{C}$, and let $C := |\mathcal{C}|$.*

To help understand Definitions 5 and 6, in Figure 1 we provide an example. Here, a MDP with 13 states is shown where the action-state pairs (6,Up) and (8,Right) are equivalent up to a permutation. Consider a permutation σ such that $\sigma(2) = 9$, $\sigma(6) = 8$, and $\sigma(i) = i$ for all $i \neq 2, 6$. Now $p(\sigma(x)|6, \mathsf{Up}) = p(x|8, \mathsf{Right})$ for all $x \in \mathcal{S}$, and thus, (8,Right) and (6,Up) belong to the same class.

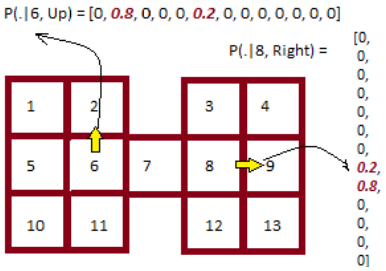


Figure 1: A grid-world MDP showing similar transitions from state-action pairs (6,Up) and (8,Right).

Remark 1 *Note, crucially, that the equivalence relation is not only stated about states but about* state-action pairs. *For instance, pairs (6,Up) and (8,Right) in this example are in the same class although corresponding to playing different actions in different states.*

Remark 2 *The profile mapping $\sigma_{s,a}$ in Definition 5 may not be unique in general, especially if distributions are sparse. For ease of presentation, in the sequel we assume that the restriction of $\sigma_{s,a}$ to the support of $p(\cdot|s,a)$ is uniquely defined. We also remark that Definition 5 easily extends to replacing the $\|\cdot\|_1$ norm with other contrasts, such as KL divergence, squared distance, etc.*

In many environments considered in RL with large state and action spaces, the number C of equivalent classes of state-action pairs using Definitions 5–6 stays small even for large SA, thanks to the profile mappings. This is the case in typical grid-world MDPs as well as in *RiverSwim* shown in Figure 2. For example in *Ergodic RiverSwim* with L states, we have $C = 6$ classes. This remarkable feature suggests that leveraging this structure may yield significant speed-up in terms of learning guarantees if well-exploited.

4.2 Equivalence-Aware Confidence Bounds

We are now ready to present an approach to defining confidence bounds for p and μ taking into account the equivalence structure in the MDP. The use of confidence bounds in a model-based approach is related to strategies implementing the *optimism in the face of uncertainty*, as in stochastic bandit problems [28, 29]. Such an approach relies on maintaining a set of plausible MDPs (models) that are consistent with the observations gathered, and contain the true MDP (model) with high probability. Exploiting equivalence structure of the MDP, one could obtain a more precise estimation of mean reward μ and transition kernel p of the MDP by *aggregating* observations from various state-action pairs in the same class. This, in turn, leads to *smaller* (hence, better) sets of models.

Notations. We introduce some necessary notations. Given an 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 $\widehat{\mu}_t(s,a)$ as the empirical mean reward built using $N_t(s,a)$ i.i.d. samples from $\nu(s,a)$, and $\widehat{p}_t(\cdot|s,a)$ as the empirical distribution built using $N_t(s,a)$ i.i.d. observations from $p(\cdot|s,a)$. For a set $c \subseteq \mathcal{S}\times\mathcal{A}$, we denote by $n_t(c)$ the total number of observations of pairs in c up to time t, that is $n_t(c) := \sum_{(s,a)\in c} N_t(s,a)$. For $c \subseteq \mathcal{S}\times\mathcal{A}$, we further denote $\widehat{\mu}_t(c)$ and $\widehat{p}_t(\cdot|c)$ as the empirical mean reward and transition probability built using $n_t(c)$ samples (we provide precise definitions of $\widehat{\mu}_t(c)$ and $\widehat{p}_t(\cdot|c)$ later on in this section).

For a given confidence parameter δ and time t, let $\mathcal{M}_{t,\delta}$ denote the set of plausible MDPs at time t, which may be expressed as

$$\mathcal{M}_{t,\delta} = \{(\mathcal{S},\mathcal{A},p',\nu') : p' \in \mathsf{CB}_{t,\delta} \text{ and } \mu' \in \mathsf{CB}'_{t,\delta}\},$$

where $\mathsf{CB}_{t,\delta}$ (resp. $\mathsf{CB}'_{t,\delta}$) denotes the confidence bound for p (resp. μ) centered at $\widehat{p}$ (resp. $\widehat{\mu}$). Note that both $\mathsf{CB}_{t,\delta}$ and $\mathsf{CB}'_{t,\delta}$ depend on $N_t(s,a), (s,a)\in\mathcal{S}\times\mathcal{A}$. For ease of presentation, in the sequel we consider confidence bounds defined using Hoeffding and Weismann concentration inequalities[2], as in several model-based RL algorithms, e.g., [1, 13, 16]:

$$\begin{aligned}
\mathsf{CB}_{t,\delta} &:= \left\{p' : \|\widehat{p}_t(\cdot|s,a) - p'(\cdot|s,a)\|_1 \le \mathbb{W}_{N_t(s,a)}\left(\tfrac{\delta}{SA}\right),\ \forall s,a\right\}, \qquad (11)\\
\mathsf{CB}'_{t,\delta} &:= \left\{\mu' : |\widehat{\mu}_t(s,a) - \mu'(\cdot|s,a)| \le \mathbb{H}_{N_t(s,a)}\left(\tfrac{\delta}{SA}\right),\ \forall s,a\right\}, \qquad \text{where}
\end{aligned}$$

$$\mathbb{W}_n(\delta) = \sqrt{\frac{2(1+\frac{1}{n})\log\left(\sqrt{n+1}\frac{2^S-2}{\delta}\right)}{n}} \quad \text{and} \quad \mathbb{H}_n(\delta) = \sqrt{\frac{(1+\frac{1}{n})\log(2\sqrt{n+1}/\delta)}{2n}}. \qquad (12)$$

These confidence bounds were derived by combining [21] and [22] inequalities with the Laplace method [23, 24], which enables to handle the random stopping times $N_t(s,a)$ in a sharp way; we refer to [25] for further discussion. In particular, this ensures that the true transition function p and mean reward function μ are contained in the confidence bounds with probability at least $1-2\delta$, uniformly over all time t.

Remark 3 *As the bounds for rewards and transitions are similar, from now on to simplify the presentation, we assume the mean reward function μ is known*[3]*.*

We provide modifications to the L_1 confidence bounds in (11) to exploit equivalence structure $\mathcal{C}$ when the learner knows $\mathcal{C}$ in advance. The case of unknown $\mathcal{C}$ is addressed in Section 4.2.3.

[2]The approach presented in this section can be extended to other concentration inequalities, as well.

[3]This is a common assumption in the RL literature; see, e.g., [16, 30].

4.2.1 Case 1: Known Classes and Profiles

Assume that an oracle provides the learner with a perfect knowledge of the equivalence classes $\mathcal{C}$ as well as profiles $\boldsymbol{\sigma} = (\sigma_{s,a})_{s,a}$. In this ideal situation, the knowledge of $\mathcal{C}$ and $\boldsymbol{\sigma}$ allows to straightforwardly aggregate observations from all state-action pairs in the same class to build more accurate estimates. Formally, for a class $c \subset \mathcal{C}$, we define

$$\widehat{p}_t^{\boldsymbol{\sigma}}(i|c) = \frac{1}{n_t(c)} \sum_{(s,a)\in c} N_t(s,a)\widehat{p}_t(\sigma_{s,a}(i)|s,a)\,, \qquad \forall i \in \mathcal{S}, \tag{13}$$

where we recall $n_t(c) = \sum_{(s,a)\in c} N_t(s,a)$. Then, we modify the confidence bound (11) by replacing the L_1 bound there as follows:

$$\|\widehat{p}_t^{\boldsymbol{\sigma}}(\boldsymbol{\sigma}^{-1}(\cdot)|c) - p'(\cdot|c)\|_1 \le \beta_{n_t(c)}\big(\tfrac{\delta}{C}\big)\,, \quad \forall c \in \mathcal{C}, \tag{14}$$

and further define: $\mathsf{CB}_{t,\delta}(\mathcal{C}, \boldsymbol{\sigma}) := \{p' : (14) \text{ holds}\}$, where $\mathsf{CB}_{t,\delta}(\mathcal{C}, \boldsymbol{\sigma})$ signifies that $\mathcal{C}$ and $\boldsymbol{\sigma}$ are provided as input. Then, for all time t and class $c \in \mathcal{C}$, by construction, the true transition p belongs to the set defined by (14), with probability greater than $1-\delta$.

Remark 4 *It is crucial to remark that the above confidence bound does not use elements of $\mathcal{C}$ as "meta states" (i.e., replacing the states with classes), as considered for instance in the literature on state-aggregation. Rather, the classes are only used to* group *observations from different sources and build more refined estimates for each pair: The plausible MDPs are built using the same underlying state $\mathcal{S}$ and action space $\mathcal{A}$, unlike, e.g., in [31].*

4.2.2 Case 2: Known Classes, Unknown Profiles

Now we consider a more realistic setting when the oracle provides $\mathcal{C}$ to the learner, but $\boldsymbol{\sigma}$ is not available. In this more challenging situation, we need to *estimate* mapping profiles as well. Given time t, we find an *empirical profile mapping* (or for short, empirical profile) $\sigma_t^{s,a}$ satisfying

$$\widehat{p}_t(\sigma_t^{s,a}(1)|s,a) \ge \widehat{p}_t(\sigma_t^{s,a}(2)|s,a) \ge \cdots \ge \widehat{p}_t(\sigma_t^{s,a}(S)|s,a)\,,$$

and let $\boldsymbol{\sigma}_t = (\sigma_t^{s,a})_{s,a}$. We then build the modified empirical estimate in a similar fashion to (13): For any $c \in \mathcal{C}$,

$$\widehat{p}_t^{\boldsymbol{\sigma}_t}(i|c) = \frac{1}{n_t(c)} \sum_{(s,a)\in c} N_t(s,a)\widehat{p}_t(\sigma_t^{s,a}(i)|s,a)\,, \quad \forall i \in \mathcal{S}.$$

Now, we may modify the L_1 inequality in (11) as follows:

$$\|\widehat{p}_t^{\boldsymbol{\sigma}_t}(\boldsymbol{\sigma}_t^{-1}(\cdot)|c) - p'(\cdot|c)\|_1 \le \beta_{n_t(c)}\big(\tfrac{\delta}{C}\big)\,, \quad \forall c \in \mathcal{C}, \tag{15}$$

which further leads to the following modified confidence bound that uses only $\mathcal{C}$ as input: $\mathsf{CB}_{t,\delta}(\mathcal{C}) := \{p' : (15) \text{ holds}\}$. The above construction is justified by the following non-expansive property of the ordering operator, as it ensures the Weissman concentration inequality also applies to the ordered empirical distribution:

Lemma 1 (Non-expansive ordering) *Let p, q be two discrete distributions, defined on the same alphabet $\mathcal{S}$, with respective profile maps σ_p and σ_q. Then,*

$$\|p(\sigma_p(\cdot)) - q(\sigma_q(\cdot))\|_1 \le \|p - q\|_1\,.$$

The proof of Lemma 1 is provided in the supplementary. The following corollary is an immediate consequence of Lemma 1:

Corollary 1 *The confidence set $\mathsf{CB}_{t,\delta}(\mathcal{C})$ contains the true transition function p with probability at least $1-\delta$, uniformly over all time steps.*

4.2.3 Unknown Classes: The ApproxEquivalence Algorithm

In this section, we turn to the most challenging situation when both $\mathcal{C}$ and $\boldsymbol{\sigma}$ are unknown to the learner. To this end, we first introduce an algorithm, which we call ApproxEquivalence, that finds an approximate equivalence structure in the MDP through grouping transition probabilities based on statistical tests. ApproxEquivalence is inspired from [32] that provides a method for clustering time series. Interestingly enough, ApproxEquivalence does not require the knowledge of number of clusters in advance.

Before presenting ApproxEquivalence, we introduce some definitions. ApproxEquivalence relies on finding subsets of $\mathcal{S}\times\mathcal{A}$ that are *statistically close* in terms of some distance function between $p(\cdot|s,a)$ for various (s,a), defined below. Given $x,y\subseteq\mathcal{S}\times\mathcal{A}$, we define *penalized distance function* between x,y as

$$d(x,y) := d_{t,\delta}(x,y) := \|\widehat{p}_t(\cdot|x)-\widehat{p}_t(\cdot|y)\|_1 - \varepsilon(n_t(x)) - \varepsilon(n_t(y))\,,$$

where for $n\in\mathbb{N}$, we define $\varepsilon(n) := \mathbb{W}_n(\frac{\delta}{3SA})$, where $\mathbb{W}_n$ is defined in (12). Here we omitted the dependence of empirical transition probabilities on $\boldsymbol{\sigma}_t$ to simplify the notation.

Definition 7 (Statistically Plausible Neighbor) *For a given equivalence structure $\mathcal{C}$, and given $c\in\mathcal{C}$, we say that $c'\in\mathcal{C}$ is a* statistically plausible neighbor *of c if it satisfies: (i) $d(c,c')\le 0$; (ii) $d(j,j')\le 0$, for all $j\in c$ and $j'\in c'$; and (iii) $d(j,c\cup c')\le 0$, for all $j\in c\cup c'$. We further define $\mathcal{N}(c) := \{c'\in\mathcal{C}\setminus\{c\} : \text{(i)–(iii) hold}\}$ as the set of all statistically plausible neighbors of c.*

Definition 8 (Statistically Closest Neighbor) *For a given partition $\mathcal{C}$ and $c\in\mathcal{C}$, we define the* statistically closest neighbor *of c (when it exists) as:*

$$\mathsf{Near}(c,\mathcal{C}) \in \arg\min_{x\in\mathcal{N}(c)} d(c,x)\,.$$

ApproxEquivalence proceeds as follows. At time t, it receives as input a parameter $\alpha>1$ that controls the level of aggregation, as well as $N_t(s,a)$ for all pairs (s,a). Starting from the trivial partition of $\{1,\dots,SA\}$ into $\mathcal{C}^0 := \{\{1\},\dots,\{SA\}\}$, the algorithm builds a coarser partition by iteratively merging elements of $\mathcal{C}^0$ that are *statistically close*. More precisely, the algorithm sorts elements of $\mathcal{C}^0$ in a decreasing order of $n_t(c)$ so as to promote pairs with the tightest confidence intervals. Then, starting from c with the largest $n_t(c)$, it finds the statistically closet neighbor c' of c, that is $c'=\mathsf{Near}(c,\mathcal{C}^0)$. If $\frac{1}{\alpha}\le\frac{n_t(c)}{n_t(c')}\le\alpha$, the algorithm merges c and c', thus leading to a novel partition $\mathcal{C}^1$, which contains the new cluster $c\cup c'$, and removes c and c'. The algorithm continues this procedure with the next set in $\mathcal{C}^0$, until exhaustion, thus finishing the creation of the novel partition $\mathcal{C}^1$ of $\{1,\dots,SA\}$. The algorithm continues this refinement process, by ordering the elements of $\mathcal{C}^1$ in a decreasing order, and carrying out similar steps as before, which leads to the new partition $\mathcal{C}^2$. The algorithm continues the same procedure until iteration k when $\mathcal{C}^{k+1}=\mathcal{C}^k$ (convergence). The pseudo-code of ApproxEquivalence is shown in Algorithm 5.

The purpose of condition $\frac{1}{\alpha} \leq \frac{n_t(c)}{n_t(c')} \leq \alpha$ is to ensure the stability of the algorithm. It prevents merging pairs whose number of samples differ a lot. We note that a very similar condition is considered in [31] for state-aggregation (with $\alpha = 2$). Nonetheless, we believe such a condition could be removed.

Algorithm 5 `ApproxEquivalence`

Input: N, α

Initialization: $\mathcal{C}^0 \leftarrow \{\{1\}, \{2\}, \ldots, \{SA\}\}$; $n \leftarrow N$; $L \leftarrow \mathbf{1}_{SA}$

`changed` $\leftarrow$ True; $k \leftarrow 1$;

while `changed` **do**

$\mathcal{C}^{k+1} \leftarrow \mathcal{C}^k$;

`changed` $\leftarrow$ False;

`Index` $\leftarrow$ argsort(n);

for all $i \in$ `Index` **do**

if $n(i) = 0$ **then**

Break;

end if

if Near$(i, \mathcal{C}^{k-1}) \neq \emptyset$ **then**

$j \leftarrow$ Near$(i, \mathcal{C}^{k-1})$;

if $\frac{1}{\alpha} \leq \frac{n(i)/L(i)}{n(j)/L(j)} \leq \alpha$ **then**

$\widehat{p}(j) \leftarrow \dfrac{n(j)\widehat{p}(j) + n(i)\widehat{p}(i)}{n(j) + n(i)}$

$L(i) \leftarrow L(j) + L(i)$; $n(i) \leftarrow n(j) + n(i)$;

$n(j) \leftarrow 0$, $L(j) \leftarrow 0$;

$\mathcal{C}^{k+1} \leftarrow \mathcal{C}^{k+1} \setminus (\{i\}, \{j\}) \cup \{i, j\}$;

`changed` $\leftarrow$ True;

end if

end if

end for

$k \leftarrow k + 1$;

end while

output $\mathcal{C}^k$

Remark 5 *Since at each iteration, either two or more clusters are merged, Algorithm 5 converges after, at most,* $SA - 1$ *steps.*

We provide a theoretical guarantee for the correctness of `ApproxEquivalence` for the case when α tends to infinity. The result relies under the following separability assumption:

Assumption 1 (Separability) *There exists some* $\Delta > 0$ *such that*

$$\forall c \neq c' \in \mathcal{C}, \forall (s,a) \in c, (s',a') \in c', \quad \|p(\sigma_{s,a}(\cdot|s,a)) - p(\sigma_{s',a'}(\cdot|s',a'))\|_1 \geq \Delta\,.$$

Proposition 1 *Under Assumption 1, provided that* $\min_{s,a} N_t(s,a) > f^{-1}(\Delta)$, *where* $f : n \mapsto 2\mathbb{W}_n(\frac{\delta}{3SA})$, *`ApproxEquivalence` with the choice* $\alpha \to \infty$ *outputs the correct equivalence class* $\mathcal{C}$ *of state-action pairs with probability at least* $1 - \delta$.

The proof of Proposition 1 is provided in [20]. We note that Assumption 1 bears some similarity to the separability assumption used in [33]. Note further, although the proposition relies on

Assumption 1, we believe that one may be able to derive a similar result under a weaker assumption as well. We leave this as a future work.

Now we turn to define the confidence intervals. Given time t, let $\mathcal{C}_t$ denote the equivalence class output by the algorithm. We may consider the following L_1 inequality

$$\|\widehat{p}_t^{\boldsymbol{\sigma}_t}(\sigma_t^{-1}(\cdot)|c) - p'(\cdot|c)\|_1 \leq \mathbb{W}_{n_t(c)}\left(\tfrac{\delta}{3SA}\right), \quad \forall c \in \mathcal{C}_t\,, \tag{16}$$

and define $\mathsf{CB}_{t,\delta}(\mathcal{C}_t) := \{p' : \text{ (16) holds}\}$.

4.3 Application: The C-UCRL Algorithm

This subsection is devoted to presenting some possible applications of our equivalence-aware confidence bounds introduced in Section 4.2. We present a natural modification of UCRL2 [1], which we call the C-UCRL algorithm. C-UCRL is capable of exploiting the equivalence structure of the MDP. We consider variants of C-UCRL depending on which information is available to the learner in advance.

4.3.1 C-UCRL: Known Equivalence Structure

Here we assume that the learner knows equivalence structure $\mathcal{C}$ and $\boldsymbol{\sigma}$ in advance, and provide a variant of UCRL2, referred to as C-UCRL$(\mathcal{C}, \boldsymbol{\sigma})$ capable of exploiting the knowledge on $\mathcal{C}$ and $\boldsymbol{\sigma}$. Given δ, at time t, C-UCRL$(\mathcal{C}, \boldsymbol{\sigma})$ uses the following set of models

$$\mathcal{M}_{t,\delta}(\mathcal{C}, \boldsymbol{\sigma}) = \left\{(\mathcal{S}, \mathcal{A}, p', \nu) : p' \in \mathsf{Pw}(\mathcal{C}) \text{ and } \forall c \in \mathcal{C}, \forall (s,a) \in c, \text{ (14) holds}\right\},$$

where $\mathsf{Pw}(\mathcal{C})$ denotes the state-transition functions that are piece-wise constant on $\mathcal{C}$ (that is $p'(\cdot|s,a)$ has same value for all $(s,a) \in c$). Moreover, C-UCRL$(\mathcal{C}, \boldsymbol{\sigma})$ uses the following stopping criterion as

$$t_{k+1} = \min\left\{t > t_k : \exists c \in \mathcal{C} : \nu_{t_k:t}(c) \geq \max\{n_{t_k}(c), 1\}\right\}.$$

The precise modified steps of the algorithm are presented in the supplementary material. A modification of the analysis from [1] yields:

Theorem 4 (Regret of C-UCRL$(\mathcal{C}, \sigma)$) *With probability higher than $1 - 2\delta$, uniformly over all time horizon T,*

$$\mathfrak{R}(\textit{C-UCRL}(\mathcal{C}, \sigma), T) \leq 17D\sqrt{CTK \ln(C\sqrt{T}/\delta)}\,,$$

where K bounds the support of the transition maps.

The proof of Theorem 4 is provided in [20]. This theorem shows that the regret of this algorithm that knows the equivalence structure (but not the distributions) thus scales with the number of classes C, hence achieving a massive reduction when $C \ll SA$. This is the case in many grid-worlds thanks to our definition using orderings; see [20]. We remark that if only $\mathcal{C}$ is provided, we can use the confidence bounds defined using (15) in lieu of (14), and the corresponding set is referred to as $\mathcal{M}_{t,\delta}(\mathcal{C})$.

4.3.2 C-UCRL: Unknown Equivalence Structure

Now we consider the case where $\mathcal{C}$ is unknown to the learner. To accommodate this situation, we use ApproxEquivalence in order to estimate the equivalence structure.

We introduce C-UCRL, which proceeds similarly to C-UCRL$(\mathcal{C}, \boldsymbol{\sigma})$. At each time t, ApproxEquivalence outputs $\mathcal{C}_t$ as an estimate of equivalence structure. Then, C-UCRL uses the following set of models taking $\mathcal{C}_t$ as input:

$$\mathcal{M}_{t,\delta}(\mathcal{C}_t) = \left\{(\mathcal{S}, \mathcal{A}, p', \nu) : p' \in \mathsf{Pw}(\mathcal{C}_t) \text{ and } \forall c \in \mathcal{C}_t, \forall (s,a) \in c, \ (16) \text{ holds}\right\}$$

Further, it uses the following stopping criterion:

$$t_{k+1} = \min\left\{t > t_k : \exists c \in \mathcal{C}_{t_k} : \nu_{t_k:t}(c) \geq \max\{n_{t_k}(c), 1\} \text{ or } \ \exists s, a : \nu_{t_k:t}(s,a) \geq \max\{N_{t_k}(s,a), 1\}\right\}.$$

Remark 6 *Note that $\mathcal{M}_{t,\delta}(\mathcal{C}) \neq \mathcal{M}_{t,\delta}(\mathcal{C}_t)$ as we may have $\mathcal{C}_t \neq \mathcal{C}$. Nonetheless, the design of ApproxEquivalence, which relies on confidence bounds, ensures that $\mathcal{C}_t$ is informative enough, in the sense that $\mathcal{M}_{t,\delta}(\mathcal{C}_t)$ could be much smaller (hence, better) than a set of models that one would obtain by ignoring equivalence structure; this is also validated by the numerical experiments in ergodic environments in subsection 5.4.*

4.4 C-UCRL: Known classes and unknown profile mapping

This sub-section is too dense (reviewers complained), cut it two parts: dealing with bias & avoiding the bias, with clear transitions and motivations of these sub-sections

A case has been excluded from the previous work of this section, it is the case where the classes are known but profile mapping are unknown, we denote this algorithm C-UCRL$(\mathcal{C})$. In this subsection, we justify why this algorithm is much more difficult than the case with complete knowledge C-UCRL$(\mathcal{C}, \boldsymbol{\sigma})$, and we propose some first solutions to this problem.

In [34] this algorithm was wrongly proved to verify the same regret bound as C-UCRL$(\mathcal{C}, \boldsymbol{\sigma})$ using the estimated profile mapping which is just an argsort of the transition probabilities at time t, $p_t(.|s,a)$ for given (s,a). Our experiments quickly showed that it is not true: this algorithm achieves a linear regret in all our non-ergodic environment with more than 20 states. This is due to the fact that the bias brought by the estimated profile mapping (instead of the true one) has not been considered. Basically, it means that using the same confidence bounds as C-UCRL$(\mathcal{C}, \boldsymbol{\sigma})$ on $p(\hat{\sigma}^t_{s,a}(.)|c_{s,a})$ is wrong. In practice it could lead to a situation where a pair (s,a) that has never been visited is considered to be well known because another pair of the same class has been visited a lot, but because the profile mapping of this pair is unknown we, in fact, have no real valuable information on the transition probabilities of this pair.

We have to keep in mind in this section that when the profile mapping is unknown it does not change anything about the reward, and so we can use aggregation on the reward as in C-UCRL$(\mathcal{C}, \boldsymbol{\sigma})$, this raise a question about the interest of using equivalence classes based on both transition probability and reward: for simplicity we continue to consider this type of equivalence class in this document but in practice it will more interesting to have separated equivalence classes for the reward and for the transition probability (for example in case of sparse reward it allows to have a big class of 0 reward which would probably be cut when classes also depends on the transition probability) it would also be interesting when classes are only known for transition probability (conversely the reward) and not for reward (conversely the transition probability). An additional remark is that

this separation of the equivalence class in two equivalence classes would be very relevant for the C-UCRL in the clustering algorithm (and it would probably be a good improvement of it).

The first solution to this problem that we tried is to take into account the bias brought by the estimated profile mapping. It leads to the following lemma proved by Sadegh Talebi (it can, in fact, be easily proved by decomposing over pairs in the class and by using the Jensen inequality):

Lemma 2 *For a given class $c \in \mathcal{C}$ we have the following confidence bound on the sorted transition probabilities:*

$$\|p(.|c) - \hat{p}_t(.|c)\|_1 \leq \frac{1}{n_t(c)} \sum_{(s,a)\in c} N_t(s,a)\beta_t(s,a) \tag{17}$$

with $\beta_t(s,a)$ a confidence bound on $\|p(.|s,a) - \hat{p}_t(.|s,a)\|_1$.

This lead to the two following observations, first using the aggregated estimate $\hat{p}_t(.|c)$ for all $(s,a) \in c$ decrease the quality of the estimate for most of the pairs $(s,a) \in c$, and the second observation is that having a bound on $\|p(.|c) - \hat{p}_t(.|c)\|_1$ is not enough we need confidence bounds on: $\|p(\sigma_{s,a}.|c) - \hat{p}_t(\hat{\sigma}^t_{s,a}(.)|c)\|_1$ in order to perform the Extended Value Iteration properly and to keep theoretical bounds on the cumulative regret. We can remark that this previous lemma raises the question of the validity of the C-UCRL algorithm (with equivalence classes unknown) as it is currently proposed, instead of using the aggregated confidences bounds used in C-UCRL$(\mathcal{C}, \boldsymbol{\sigma})$ we should at least use the bound of the previous lemma, but as explained we will still ignore a part of the biased brought by the estimated profile mapping. We no longer insist on it in this section but all the proposition of this subsection could and should be applied to the C-UCRL algorithm with unknown classes (and the current clustering algorithm should be seriously modified) in further work.

The previous first observation lead us to the following strategy: for all (s,a) in a given class c we use the estimated ordered transition probability $\hat{p}_t(.|{c^t_{s,a}}')$ build on the subclass ${c^t_{s,a}}' \subset c$ where ${c^t_{s,a}}' = \{(s',a') \in c | N_t(s,a) \leq N_t(s',a')\}$, doing that and because of lemma 2 for all $(s,a) \in c$ the confidence bounds on $\|p(.|c) - \hat{p}_t(.|{c^t_{s,a}}')\|_1$ can only be tighter than $\|p(.|c) - \hat{p}_t(\hat{\sigma}^t_{s,a}(.)|s,a)\|_1$. With this strategy we know that we improve the precision of the estimate of the ordered transition probability. But, following the previous second observation it does not imply that we have an improvement when coming back to $\|p(\sigma_{s,a}.|c) - \hat{p}_t(\hat{\sigma}^t_{s,a}(.)|{c^t_{s,a}}')\|_1$ because the confidence bound on it is biased by $\hat{\sigma}$, in order to solve this problem we tried to add optimism over the $\hat{\sigma}^t_{s,a}$ in the Extended Value Iteration (we precisely used optimism on all transition to s' from (s,a) when $N_t(s',s,a) = 0$) this strategy brought us good results in numerical experiments, but because we are currently unable to control the bias brought by $\hat{\sigma}$ (even if this bias is probably considered by the optimism on $\hat{\sigma}$) we cannot control the regret of this algorithm (the proof of upper bounds on the regret does not old because of this uncontrolled bias) results of this algorithm are not presented in this paper.

As the previous proposition cannot currently be controlled in general, even if it performs well in practice, it is not interesting in itself. This desire to have a control in general on the regret leads us to the following strategy: we decide to reduce the use of the knowledge of the classes on the transition probability, now aggregation is only performed when the profile mapping is known with high probability. Doing so it is almost trivial to adapt the proof of the bound on the regret of C-UCRL$(\mathcal{C}, \boldsymbol{\sigma})$ but instead of $\mathcal{C}$ we will have a $\hat{\mathcal{C}}_t$ such as $\mathcal{C} \leq \hat{\mathcal{C}}_t \leq \mathcal{SA}$.

To ensure that the profile mapping is known with high probability we ask that for a given pair (s,a) and for pairs of states $s_1, s_2 \in \mathcal{S} \times \mathcal{S}$ such as $s1 \neq s_2$ the element-wise confidence intervals around $\hat{p}_t(s_1|s,a)$ and $\hat{p}_t(s_2|s,a)$ have empty intersection. This criterion is really hard to reach and

cannot be reached when there exist $s_1, s_2 \in \mathcal{S} \times \mathcal{S}$ such as $s1 \neq s_2$ and $p(s_1|s,a) = p(s_2|s,a)$, which in practice almost always append because the support is sparse. To avoid this issue we propose the following strategy: instead of only performing aggregation for all transition starting from a pair (s,a) we use element-wise aggregation when profile mapping is known with high probability for this element.

Formally, C-UCRL($\mathcal{C}$) become a local algorithm where for given class $c \in \mathcal{C}$ and a given state $s' \in \mathcal{S}$, for all $(s,a) \in \hat{c}_t^{s'}$ we use the aggregated estimate $\hat{p}(\hat{\sigma}_{s,a}^t(s')|\hat{c}_t^{s'})$ instead of $\hat{p}(\hat{\sigma}_{s,a}^t(s')|s,a)$ with:

$$\hat{c}_t^{s'} = \left\{(s,a) \in c | \forall s'' \in \mathcal{S}/\{s'\}, \delta(\hat{p}(\hat{\sigma}_{s,a}^t(s')|s,a)) \cap \delta(\hat{p}(\hat{\sigma}_{s,a}^t(s'')|s,a)) = \emptyset\right\}$$

where $\delta(\hat{p})$ is the element-wise confidence interval of our choice around $\hat{p}$.

Finally we use this strategy as an improvement of UCRL3 which suit well to tit because all element outside from the support (which are in practice the elements that cannot be aggregated using this strategy) have a reduced effect on the regret due to the modification of the inner maximization used in the algorithm 4. We denote the resulting algorithm C-UCRL($\mathcal{C}$).

Experimental results obtained with this strategy are not shown in this report, because of a lack of time to run enough samples. But globally results are similar to the one obtained by C-UCRL($\mathcal{C}, \boldsymbol{\sigma}$) when applied to UCRL3 (which is also a result that, unfortunately, we cannot present in this document, but globally we have a gain of an order of magnitude in our testing environments with around 20 states).

5 Numerical experiments

We present in this section some of the experimental results obtained during the internship. As most of the literature has been implemented we do not propose an exhaustive presentation of our results but only the most relevant ones. The implementation is done in python, and our testing base is built using the gym package [35] [4].

Our only remark about the computational time is the following: there is no big difference of execution time between our algorithm (we remain in the same order) except for the optimistic Thompson sampling proposed by [10] which is very slow probably due to the really bad constant proposed in the paper. The variant of C-UCRL using clustering is also slower than other algorithms even if it stays in the same order of time (a factor 3 approximately in practice compared to closer UCRLalgorithms). Finally an easy way to improve the performances of the algorithms in practice is to increase the number of episodes (a trivial modification of the doubling stopping criterion by adding a square root, for example, allow to do this) but the improvement does not seem to be really important in general (it seems to be problem dependent) and the increase of computational cost due to such a modification is huge (with this square root, for example, an experiment of 1 minute can quickly go to almost 1 hour depending on the implementation). Globally to give an order of the execution time of our implementations (which are absolutely not optimized) the execution time with 6 states of one run for one algorithm is in the order of the second, with 20 state it is in the order of the minute and with 50 states it is in the order of the hour (on a personal computer). Further experiments in a more challenging environment (around 150 states) should be performed in grid5000 by the end of the internship. The quick increase in computation time comes both from the

[4] The full code will be available on https://github.com/HippolyteBourel/UCRL_implementation by the end of the internship (as well as a version of this report with completed Appendix).

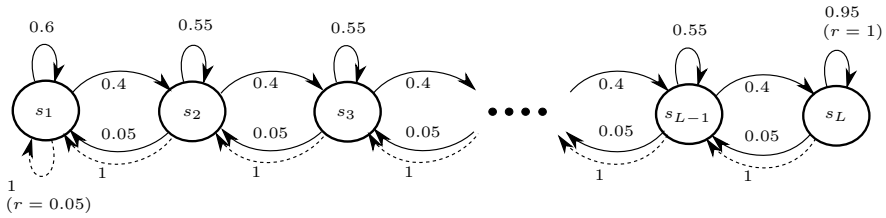


Figure 2: The L-state *RiverSwim* MDP

increase in computational cost with more states but also from the longer time horizon that should be considered due to the increase of difficulty for learners.

An additional remark about our experiments is that all our algorithms know that the maximum reward in our environments is 1 and all exploit it by doing truncation on the maximum plausible reward in the EXTENDED VALUE ITERATION. This does not seem to have any effect in practice on all algorithms excepted for SCAL where it affects the experimental span and so the results (it decreases slightly the improvement compared to UCRL2, or even brought linear regret for small but not too small input parameter c).

5.1 Testing environments

Our experiments are done on standard environments. Among these environments, we have the riverSwim and gridworld which are usual environments in the literature of the domain. In the following subsection, we precisely describe the set of environments used in experiments.

5.1.1 RiverSwims environments

The first type of environment that we use in our experiments are riverSwim environments. The riverSwim is a classical benchmark of the domain, usually used with 6-states in the literature considered in our State-of-the-Art. The principle of these environments is simple: the environment is a chain of states (representing a river) the learner can perform two actions going left or right, while going left the learner swims with the flow of the river so the action is easy, but, while going write the learner swim against the flow so it has a high probability to fail. In these environments, the learner starts at the bottom (or left) of the river, in this state while going left the learner have a small reward but its objective is to reach the top (or right) of the river by swimming against the flow to have a big reward. The interest of such an environment is the exploration challenge, the best policy is to exploit the big reward in the hardest-to-reach state.

In our experiments we use two riverSwim the classical communicating one shown in figure 2 and a less usual riverSwim Ergodic shown in figure 3. These environment being used with a various number of states.

5.1.2 Gridworlds

Gridworlds are usual benchmark environment in the domain, these environments are grid in which the learner can perform moving actions (choosing the direction) and sometimes additional actions are added to increase the complexity of the task. For implementation's simplicity walls are unreachable states, but our algorithm cannot be performed in such an MDP because it breaks the communicating property of our gridworlds. It is why, in the point of view of the learner, walls are not in its set of

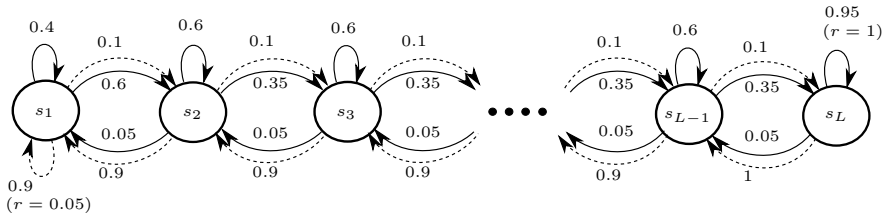
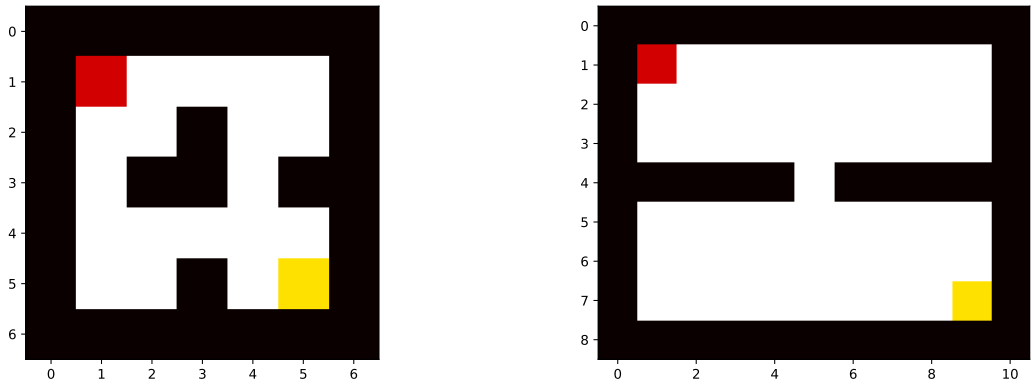


Figure 3: The L-state *RiverSwim Ergodic* MDP

states, and its why we are interested in the number of states while taking off walls. We perform experiments in two gridworlds a 7×7 4-room (20 states while taking off walls) that is shown in figure 4(a), and a 9×11 2-room (55 states while taking off walls) that is shown in figure 4(b). In these two gridworlds the learner start in the upper-left corner, a reward of 1 is placed in the lower-right corner when this reward is reached the agent is sent back to the initial state. The learner can perform 4 actions, going up, left, down or right for all of these actions there are probabilities of 0.1 to stay in the same state and to go in one of the perpendicular direction if there is no wall, giving a 0.7 probability to go in the good direction when there is no wall around the learner.



(a) 20-states 4-room

(b) 55-states 2-room

Figure 4: Gridworlds used in numerical experiments, the starting state is in red, the reward in the yellow state. From the yellow state all actions bring to the red state.

5.1.3 Others environments

In order to test SCAL another environment is introduced, this environment called three-states MDP is introduced by [14] in order to test SCAL. The three-state MDP is a toy example which has the good property to have an almost infinite diameter but a really small bias span, a good property to test SCAL. This environment depends on a parameter δ and we have, for small δ a bias span $sp(b^\star) \approx \frac{1}{1-\delta}$ and a diameter $D \approx \frac{1}{\delta}$. The gap between Diameter and bias span allows putting in evidence the difference between UCRL strategies and SCAL as shown in subsection 5.5.

In further experiments we plan to compare algorithms in more challenging environments, to do

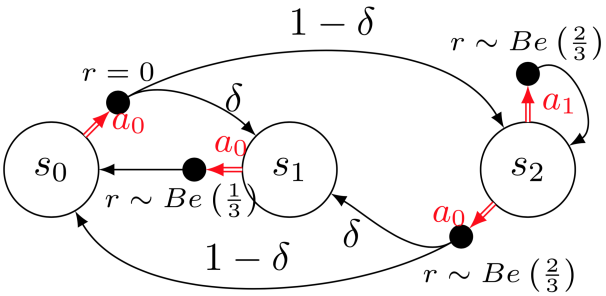


Figure 5: The three-states MDP from [14].

Learner	avg Regret ± 95% confidence
UCRL3	1367 ± 61
UCRL2-L	2287 ± 104
UCRL2	37556 ± 1184
UCRL-Bernstein	5114 ± 139
KL-UCRL	5229 ± 210

Figure 6: Table of results in 6-state riverSwim

so, we may use bigger gridworlds and probably a discretized *mountain car* problem similar to the one presented in [2], using the one available in gym [35], this problem is a classical benchmark of the domain.

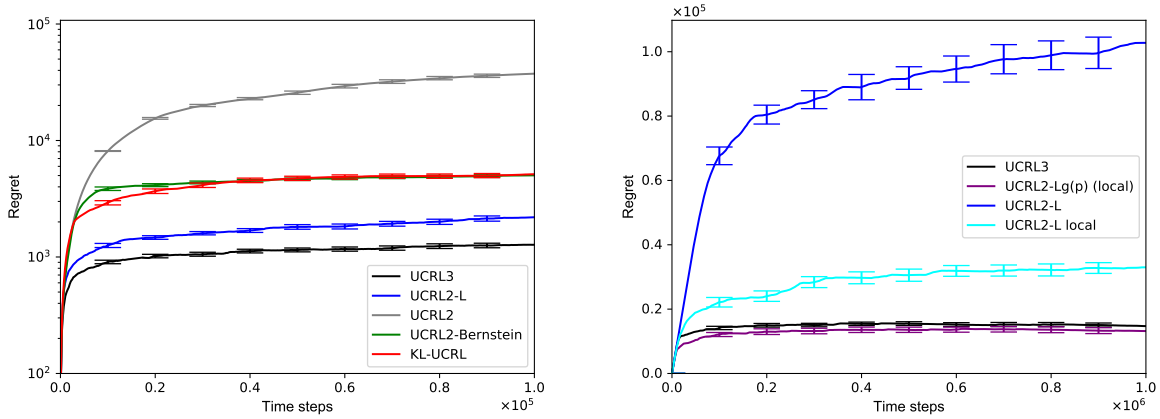
5.2 Primary results on the State of the Art

We conduct numerical experiments to examine the performance of the proposed variants of UCRL2. First, we compare the performance of State of the Art algorithms such as UCRL2, KL-UCRL and UCRL-Bernstein to that of UCRL2 with L_1 Laplace bounds and UCRL3.

In the first experiment, we study the regret of the State-of-Art Algorithms compared to UCRL2-L and UCRL3 in a 6-states riverSwim environment built as shown in Figure 2. In Figure 7(a), we plot the regret against time steps under UCRL2, KL-UCRL, UCRL-Bernstein, UCRL2-L and UCRL3examined in the aforementioned environment, results are also in the figure 5.2. The results are averaged over 60 independent runs, and the 95% confidence intervals are shown and all algorithms use $\delta = 0.05$. As the curves show, UCRL2-L and UCRL3 algorithms significantly outperform UCRL2 and achieve a much better regret than KL-UCRL and UCRL-Bernstein.

About KL-UCRL we put in the appendix A.1 the pseudo-code of our implementation, the fact is that many points are not well defined in this algorithm and it is necessary to filter a lot of definition error (most of the time error of the type divide by zero).

Modifying the stopping criterion could be a way to improve all UCRL algorithms, but, as explained earlier just increasing the number of episodes may lead to a serious increase in computational time for a negligible gain on the regret. As presented in the State-of-the-Art the paper [34] introduce a modified stopping criterion based on hypothesis testing. We present in figure 5.2 a comparison between UCRL2-L and UCRL2-L with this modified stopping criterion conjugate with the classical doubling criterion (the algorithm is denoted UCRL2-L MSC in the legend). These results are averaged over 60 independent runs, and the 95% confidence intervals are shown and all algorithms use $\delta = 0.05$.



(a) Baselines in 6-state riverSwim

(b) Refinements in 25-state riverSwim

Figure 7: Regret of various algorithms for riverSwim environments.

This modified stopping criterion has been tested on several algorithm and environments, similar results are observed in general but sometimes there is no gain. In practice we do the following observation: most of the time episodes end because of the classical doubling criterion (in average 2 or 3 times in the presented experiment) the hypothesis testing stop early a bad episode. In practice, conjugate the classic doubling criterion and the hypothesis testing one can only improve the regret of the algorithm (or at least it does not change anything), the bad point about it is that we cannot improve the theoretical bounds on the regret using it because bounds are asymptotic and proved by exploiting the fact that bad episodes are negligible or the effect the hypothesis testing criterion is to early stop these rare bad episodes.

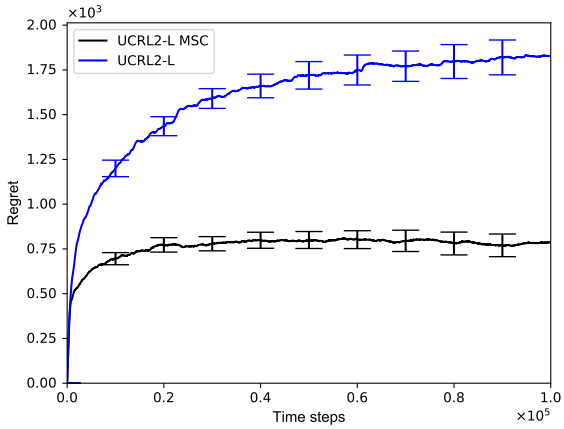


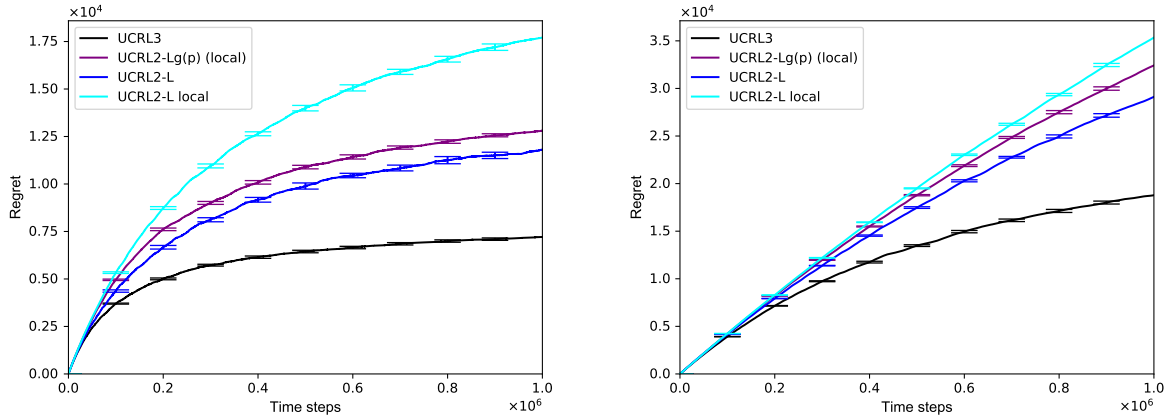
Figure 8: Comparison of regret of UCRL2-L and its variant with Modified Stopping Criterion (MSC), which is a conjugate of hypothesis testing and classical doubling criterion, on a 6-states riverSwim environment.

No result on the optimistic posterior sampling algorithm introduced by [10] and described in

subsection 2.2.3 are shown in this paper. This algorithm has been implemented and tested but additionally to its terribly bad computational time this algorithm achieves a linear regret in all experiments performed with it. This could come from an error in the code, an error in the algorithm (the proof is known to be wrong) or could come from the very non-optimal constant proposed by the paper (we suspect the last option). In any case, we decide to exclude this algorithm from our benchmarks.

5.3 Experimental validation of UCRL3

The second set of experiments focused on the intermediate improvement of UCRL2-L using element-wise confidence intervals (denoted UCRL2-L local), refined time-uniform Bernoulli concentration bounds (denoted UCRL2-Lg(p)) and finally UCRL3. UCRL2-L local and UCRL2-Lg(p) are intermediate steps between UCRL2-L and UCRL3, the comparison between these two intermediate algorithms allow to experimentally show the improvement brought by the time-uniform Bernoulli concentration bounds. These experiments are done in a 25-states riverSwim environment built as shown in Figure 2 and in our two gridworlds built as explain in sub-subsection 5.1.2. The results are averaged over 30 independent runs, the 95% confidence intervals are shown and all algorithms use $\delta = 0.05$. As the curves show in Figure 7(b), 9(a) and 9(b) using element-wise bounds in UCRL2-L is not necessarily an improvement of the algorithm, and relative result between UCRL2-L L_1 or local are problem dependent. But the tighter Laplace bounds using $g(p)$ brought significant improvement of the local UCRL2-L as do UCRL3. As the Time-uniform Bernoulli concentration bounds introduced in appendix B.1 are depending on the true p which is unknown in practice, having $\hat{p}$ we perform a dichotomic search over 16 steps to compute an accurate approximation of these confidence bounds.



(a) 20-states 4-room

(b) 55-states 2-room

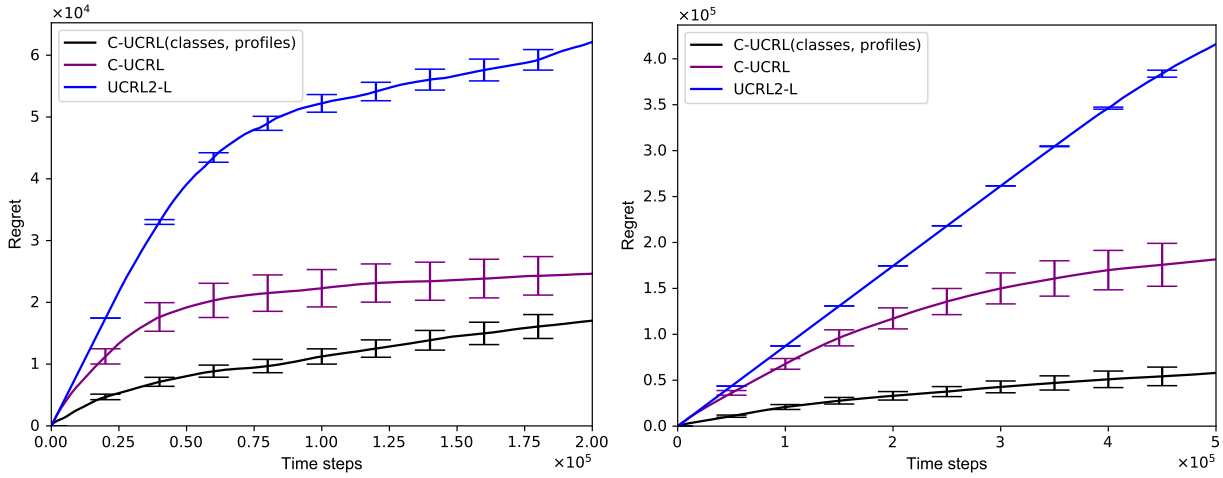
Figure 9: Regret of refined algorithms for gridworlds environments.

5.4 Preliminary results around C-UCRL

This subsection is the Numerical Experiments section of the aforementioned publication to ACML [20]. We conduct numerical experiments to examine the performance of the proposed variants of

C-UCRL. We compare the performance of C-UCRL and C-UCRL($\mathcal{C}, \boldsymbol{\sigma}$) to that of UCRL2-L[5].

In the first set of experiments, we examine the regret of various algorithms in ergodic environments. Specifically, we consider the ergodic *riverSwim* MDP, shown in Figure 2, with 25 and 50 states. In both environments, we have $C = 6$ classes. In Figure 5.4, we plot the regret against time steps under C-UCRL($\mathcal{C}, \boldsymbol{\sigma}$), C-UCRL, and UCRL2-L examined in the aforementioned environments. The results are averaged over 100 independent runs, and the 95% confidence intervals are shown. All algorithms use $\delta = 0.05$, and for C-UCRL, we use $\alpha = 4$. As the curves show, the proposed C-UCRL algorithms significantly outperform UCRL2-L, and C-UCRL($\mathcal{C}, \boldsymbol{\sigma}$) attains the smallest regret. In particular, in the 25-state environment and at the final time step, C-UCRL($\mathcal{C}, \boldsymbol{\sigma}$) attains a regret smaller than that of UCRL2-L by a factor of approximately $\sqrt{\frac{SA}{C}} = \sqrt{\frac{50}{6}} \approx 2.9$, thus verifying Theorem 4. Similarly, we may expect an improvement in regret by a factor around $\sqrt{\frac{SA}{C}} = \sqrt{\frac{100}{6}} \approx 4.1$ in the other environment. We, however, get a better improvement (by a factor around 8), which can be attributed to the increase in the regret of UCRL2-L due to a long linearly increasing phase, before the sub-linear phase kicks in.



(a) 25-state Ergodic RiverSwim

(b) 50-state Ergodic RiverSwim

Figure 10: Regret of various algorithms for Ergodic RiverSwim environments

We now turn our attention to examine the quality of approximate equivalence structure produced by ApproxEquivalence (Algorithm 5), which is run as a sub-routine of C-UCRL. To this aim, we introduce two performance measures to assess the quality of clustering: The first one is defined as the total number of pairs that are *mis-clustered*, normalized by the total number SA of pairs. We refer to this measure as the *mis-clustering ratio*. More precisely, consider $\mathcal{C}_t$ the empirical equivalence structure output by ApproxEquivalence at time t. For a given $c \in \mathcal{C}_t$, we consider the restriction of $\mathcal{C}$ to c, denoted by $\mathcal{C}|c$. We find $\ell(c) \in \mathcal{C}|c$ that has the largest cardinality: $\ell(c) \in \operatorname{argmax}_{x \in \mathcal{C}|c} |x|$. Now, we define

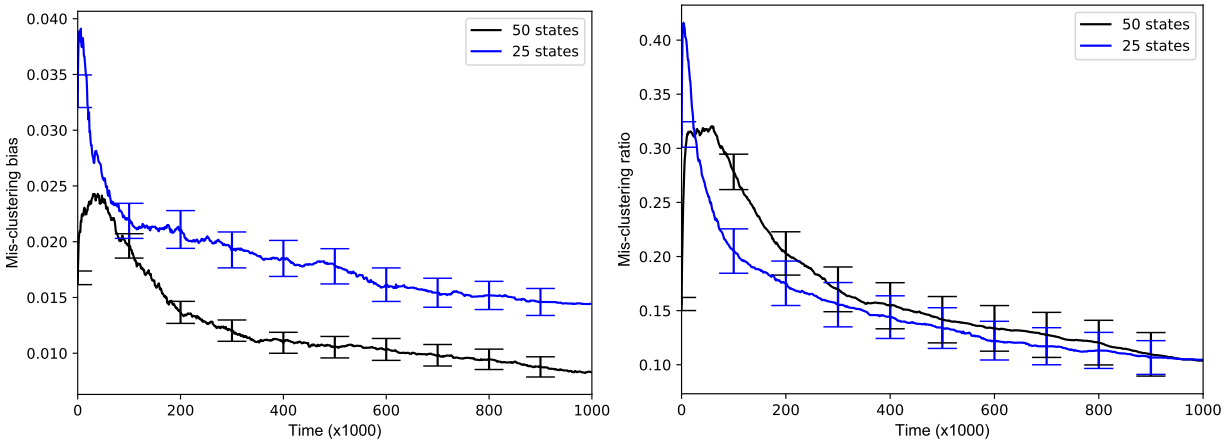
$$\text{mis-clustering ratio at time } t := \frac{1}{SA} \sum_{c \in \mathcal{C}_t} |c \setminus \ell(c)| \,.$$

[5]Use of 'L' signifies the use of Laplace method in the confidence bounds. Recall that UCRL2-L attains a strictly smaller regret than the original UCRL2 in [1].

Note that the mis-clustering ratio falls in $[0, 1]$ as $\sum_{c \in \mathcal{C}_t} |c| = SA$ for all t. The second performance measure accounts for the error in the aggregated empirical transition probability due to mis-clustered pairs. We refer to this measure as *mis-clustering bias*. Precisely speaking, for a given pair $e \in \mathcal{S} \times \mathcal{A}$, we denote by $c_e \in \mathcal{C}_t$ the set containing e in the output equivalence structure. Then, we define the mis-clustering bias at time t as

$$\text{mis-clustering bias at time } t := \sum_{c \in \mathcal{C}_t} \sum_{e \notin \ell(c)} \|\hat{p}_t(\cdot|c_e) - \hat{p}_t(\cdot|c_e \setminus \{e\})\|_1 \,.$$

In Figure 5.4, we plot the "mis-clustering ratio" and "mis-clustering bias" for the empirical equivalence structures produced in the previous experiments. As the figures show, when the time grows, the error in terms of performance measures considered become smaller. These errors do now vanish quickly, thus indicating that the empirical equivalence structures generated do not agree with the true one. Yet, they help reduce uncertainty in the transition probabilities, and in turn, reduce the regret; we refer to Remark 6 for a related discussion.



(a) Mis-clustering bias

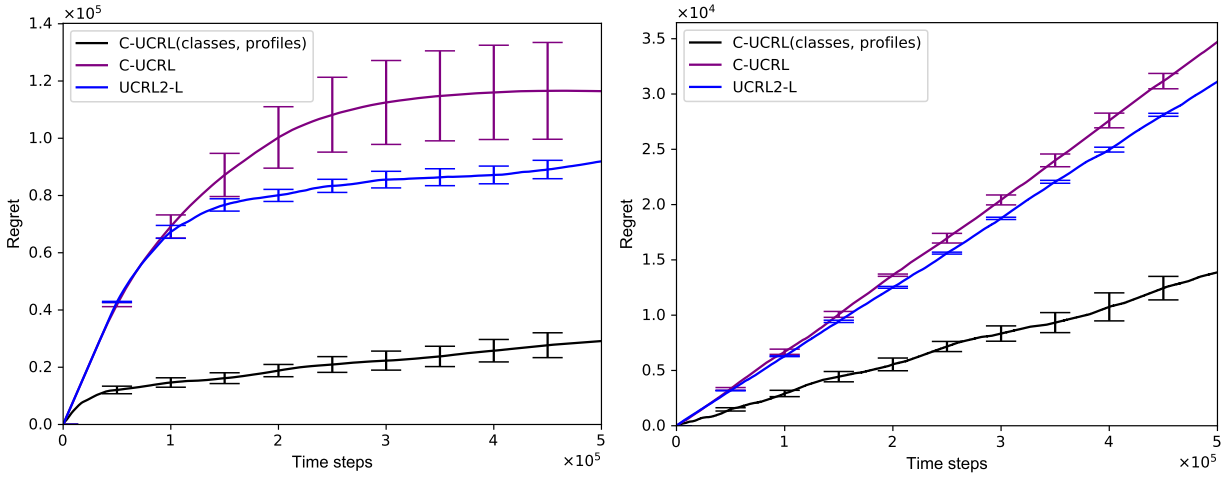
(b) Mis-clustering ratio

Figure 11: Error in clustering for Ergodic RiverSwim with 25 and 50 states

In the second set of experiments we consider two communicating environments: (non-ergodic) *riverSwim* (with 25 states) and *4-room gridworld* (with 49 states). In Figure 5.4, we plot the regret against time steps under C-UCRL($\mathcal{C}, \boldsymbol{\sigma}$), C-UCRL, and UCRL2-L, and similarly to the previous case, we set $\delta = 0.05$ and $\alpha = 4$. The results are averaged over 100 independent runs, and the 95% confidence intervals are shown. In both environments, C-UCRL($\mathcal{C}, \boldsymbol{\sigma}$) significantly outperforms UCRL2-L. However, C-UCRL attains a regret, which is slightly worse than that of UCRL2-L. The reason is that ApproxEquivalence is unable to find an accurate enough equivalence structure in non-ergodic environments.

5.5 Experimental results on SCAL

Our implementation of SCAL is based on the code used for the experiments presented in the paper [14], not on the article itself, our code for ScOpt is presented in appendix A.2. It is not entirely clear for us that this implementation is formally entirely equivalent to the one presented in the paper. Globally the paper introduces the ScOpt algorithm where the implementation is an EXTENDED VALUE



(a) 25-state RiverSwim (b) Grid-World

Figure 12: Regret in communicating environments.

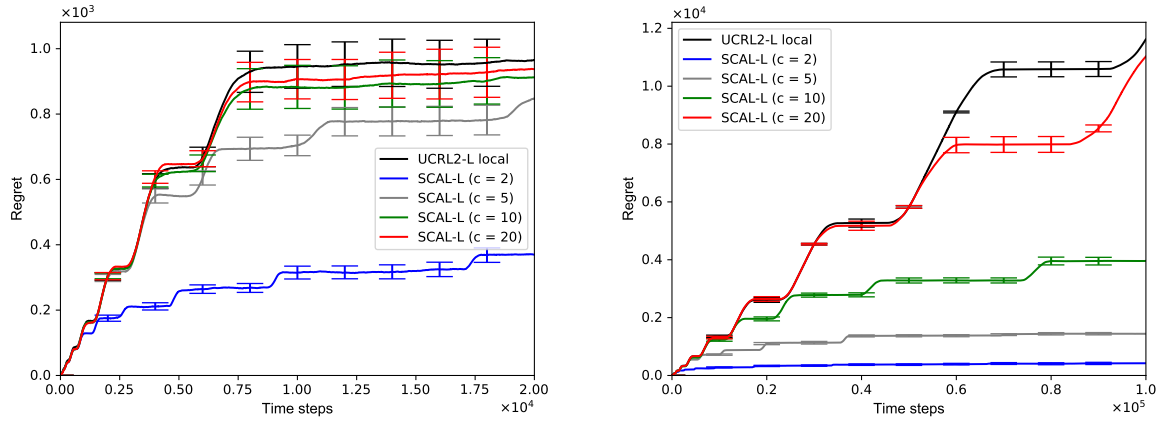
ITERATION with additionally span truncation (it is the idea behind ScOpt, but the equivalence is not entirely clear). In any case, thanks to a discussion with the authors of the paper that clarify some details about their implementation, our implementation works and provide similar results.

We conduct experiments with the SCAL algorithm applied as an improvement of UCRL2-Llocal instead of UCRL-Bernstein in the paper [14]. Globally the improvement brought by the change of bound is similar to the results observed on UCRL2 so this improvement is not shown again in this subsection. The results are averaged over 60 independent runs, and the 95% confidence intervals are shown. Provided results are on two three-states environment built as shown in figure 5.1.3 with $\delta = 0.005$ (see figure 13(a)) and $\delta = 0$ (see figure 13(b)). The three-states environment with $\delta = 0$ is non-communicating and has an infinite diameter, as expected UCRL2-L local achieve a linear regret in such a problem, one the best improvement of SCAL is its ability to learn is such an environment as curves show in figure 13(b). Otherwise, we observe as expected that the improvement brought by SCAL depends a lot on the input parameter, which would probably be a weakness in practical application (because span would probably be unknown).

Experiments with SCAL on our other environments have been done. Globally the conclusion is the following: our other environments have bias span almost equal to the Diameter, so globally results of SCAL are the same as UCRL2. Another remark about this algorithm is that giving it an input parameter c smaller than the true bound on the span breaks the proof of the regret bound and as expected it quickly decreases the performance of SCAL (worse regret than UCRL2) and could lead quickly to a linear regret.

6 New strategies based on MDP properties

In this section, we study two fundamental problem based on the structure of our MDPs, the first problem the research of most confusing MDP. Giving an MDP and a set of plausible MDP around it, the most confusing MDP is the plausible MDP which confuse as much as possible a learner. Being able to find such an MDP could be a way to efficiently explore an environment and may lead to an alternative strategy to UCRL2's. With the same objective in mind, we study the notion of self-hitting



(a) $\delta = 0.005$ (b) $\delta = 0$

Figure 13: Comparison of regret of UCRL2-L local and its variant with with SCAL (for several bound on the span c as input) on the three-states environment shown in figure 5.1.3 for $\delta = 0.005$ and $\delta = 0$.

times in an MDP, this self-hitting time is the number of time-steps necessary to go back to an initial state with high probability. We present some arguments showing that using this self-hitting time we may be able to build a new learning strategy and then we study some properties around this notion that may help to build later an algorithm based on such a strategy.

It is important to have in mind that the work presented in this section is much more "unfinished" than any other section in this document.

6.1 Confusing: 2-state MDPs

In this subsection, we present the notion of confusing MDP, similar to the notion of confusing environment defined in multi-armed bandits domain (see [36]). Then we propose a short study on the problem of the research of confusing MDP restricted to the case where the MDP has 2 states.

6.1.1 Notations and Definitions

Giving us a MDP $M_0 = (S^0, A^0, p^0, r^0)$ with $|S^0| = 2$, we denote, for a given action a: $p_a = p^0(1|1, a)$ and for a given action b: $q_b = p^0(2|2, b)$. We give us a set of MDP $\mathcal{M}$, built similarly as the set of plausible MDP in UCRL2, with for each $M \in \mathcal{M}$, $M = (S^0, A^0, p, r)$ with for all $(s, a) \in S^0 \times A^0$,$\|p(.|s, a) - p^0(.|s, a)\|_1 < \beta_p(s, a)$ for a given β_p and $|r(s, a) - r^0(s, a)| < \beta_r(s, a)$ for a given β_r.

Definition 9 (Optimal state) *We call an optimal state s for given MDP $\mathcal{M}$ a state which is visited by all optimal policies an asymptotically not negligible number of times in average: an optimal state is such that for all optimal policies $\pi^\star$:*

$$\lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[N_T^{\pi^\star}(s)\right] > 0$$

where $N_T^{\pi^\star}(s)$ is the number of visit of the optimal policy to the state s at time T.

In following paragraphs we denote:

$$\rho(s) = \min_{\pi^\star} \left(\lim_{T\to\infty} \frac{1}{T} \mathbb{E}\left[N_T^{\pi^\star}(s) \right] \right)$$

An optimal action for a given state s is an action a such that there exist an optimal policy $\pi^\star$ such that $\pi^\star(s) = a$. A pair (s, a) is optimal when the state and the action are optimal.

Our objective is to confuse the learner, to do so the objective in following sub-subsection is to make a non-optimal pair optimal bu modifying reward and transition probability of the MDP.

6.1.2 Optimize the (1, a) pair

To make the $(1, a)$ pair optimal we have to take care of 2 points: first the optimal policy $\pi^\star$ of the modified MDP have to verify that $\pi^*(1) = a$, that makes the action a optimal, then we want the state to be optimal, we want that there exist $\varepsilon > 0$ such that $\rho(1) > \varepsilon$.

To complete the first objective we have to choose a MDP $M \in \mathcal{M}$ such that $\arg\max_\pi(g(M, \pi)) \in \Pi_a$ with $\Pi_a = \{\pi | \pi(1) = a\}$, in the general case the only to do so is to run a policy-wise optimization: we have to chose a MDP M in the following set:

$$\mathcal{M}^{(1,a^*)} = \left\{ M \in \mathcal{M} | \arg\max_{\pi\in\Pi_a}(g(M, \pi) > \arg\max_{\pi\notin\Pi_a}(g(M, \pi)) \right\}$$

In this set of MDP, the action a is the optimal action, but 1 is not necessarily an optimal state (in practice that means that in our restricted case where $|S| = 2$ only the state 2 is visited, so if we accept to have an $\varepsilon > 0$ which could be arbitrarily small then we only have to add the following condition $q_b < 1$ with $b = \pi^*(2)$ in order to define the set of MDP were $(1, a)$ is optimal. But in practice we would probably favorise the possibility to have an arbitrary $\varepsilon > 0$, then the necessary condition will become (still having $b = \pi^\star(2)$) (it is important here to have in mind that p_a and q_b are components of M):

$$\mathcal{M}^{(1^*,a)} = \left\{ M \in \mathcal{M} | \frac{q_b}{1 - p_a + q_b} > \varepsilon \right\}$$

this result come from the stationary distribution over the Markov Chain defined by an MDP M when executing a policy $\pi^\star$.

Finally, it allows us to define the set of MDP in which the pair (1, a) is optimal (the set as the two previous ones could be empty):

$$\mathcal{M}^{(1^*,a^*)} = \mathcal{M}^{(1,a^*)} \bigcap \mathcal{M}^{(1*,a)}$$

We could also write the following optimization problem which will a return an element of the previous (if such an element exists), but in general case, this problem is computationally hard to solve.

$$M = \arg \max_{M\in\mathcal{M}^{(1^*,a)}} \left(\arg\max_{\pi\in\Pi_a}(g(M, \pi) - \arg\max_{\pi\notin\Pi_a}(g(M, \pi)) \right)$$

Globally, having this result, it seems that the only way to ensure that a pair is optimal in a modified MDP is to perform an exhaustive research over the policies, is due to the fact that most of the necessary modifications of the MDP to make a pair optimal could, in fact, make another pair (for example same state with another action) optimal instead, to control this an almost exhaustive search would be necessary. Having this for 2-states is not really a problem, but in general, it is computationally impossible to perform such exhaustive research, it is why we think that a strategy based on this idea of confusing MDP cannot be computationally efficient.

6.2 Hitting Times

The objective here is to use the properties of hitting times on finite MDPs in order to derive a new learning strategy. In a finite MDP, the optimal policy (here we will consider discrete policy) defines some returning times on a subset of states, globally the optimal behavior on the MDP will be a cycle over a set of optimal states. The objective here is to estimate this optimal cycle in order to have a good estimate of the asymptotic gain of the MDP and so defines a learning algorithm maximizing this estimate of the asymptotic gain.

6.2.1 Self-hitting time and approximation of the average gain

Having a state s_0, and a policy π, we denote $\tau^{\circlearrowleft}(s_0, \alpha, \pi)$ the (α-)self-hitting time of the state s_0 for policy π, such that:

$$\tau^{\circlearrowleft}(s_0, \alpha, \pi) = \min\left\{T | \sum_{t=0}^{T} \mathbb{P}_{\pi}(s_t = s_0) > 1 - \alpha\right\}$$

We know that an optimal state exist because we consider finite state-space MDP (then by definition it exists), then, we have, for s_0 an optimal state (by definition):

$$\lim_{T\to\infty} \sum_{t=0}^{T} \mathbb{P}_{\pi^\star}(s_t = s_0) = 1$$

justifying the existence of the α-self-hitting time when s_0 is an optimal state and when the policy is an optimal one.

Having an optimal state s_0, an optimal policy $\pi^\star$ and $\tau^{\circlearrowleft}$ its α-self-hitting time, for a given (small) α, we can compute the following gain:

$$g^{\circlearrowleft}(s_0, \alpha, \pi^\star) = \sum_{k=1}^{\tau^{\circlearrowleft}} \sum_{s_1,\dots,s_{k-1}\in\{\mathcal{S}/\{s_0\}\}^{k-1}} \left(\frac{1}{k}\mathbb{P}_{\pi^\star}(s_0, s_1, \dots, s_{k-1}, s_0) \sum_{t=0}^{k-1} \mu(s_t, \pi(s_t))\right)$$

Due to the previous definitions this gain is a tight estimation of the average gain $g^\star$ of the optimal policy on the MDP. We have the following result, for given optimal state s_0:

$$g^\star = \sum_{k=1}^{\infty} \sum_{s_1,\dots,s_{k-1}\in\{\mathcal{S}/\{s_0\}\}^{k-1}} \left(\frac{1}{k}\mathbb{P}_{\pi^\star}(s_0, s_1, \dots, s_{k-1}, s_0) \sum_{t=0}^{k-1} \mu(s_t, \pi(s_t))\right)$$

Using the fact that even if s_0 is not the true initial state, what's come before is negligible. Then using this definition we have:

$$\begin{aligned}
|g^{\circlearrowleft}(s_0, \alpha, \pi^\star) - g^\star| &= \sum_{k=\tau^{\circlearrowleft}+1}^{\infty} \sum_{s_1,\dots,s_{k-1}\in\{\mathcal{S}/\{s_0\}\}^{k-1}} \left(\frac{1}{k}\mathbb{P}_{\pi^\star}(s_0, s_1, \dots, s_{k-1}, s_0) \sum_{t=0}^{k-1} \mu(s_t, \pi(s_t))\right) \\
&\leq \sum_{k=\tau^{\circlearrowleft}+1}^{\infty} \sum_{s_1,\dots,s_{k-1}\in\{\mathcal{S}/\{s_0\}\}^{k-1}} (\mathbb{P}_{\pi^\star}(s_0, s_1, \dots, s_{k-1}, s_0)\mu_{max}) \\
&\leq \mu_{max} \sum_{k=\tau^{\circlearrowleft}+1}^{\infty} \sum_{s_1,\dots,s_{k-1}\in\{\mathcal{S}/\{s_0\}\}^{k-1}} (\mathbb{P}_{\pi^\star}(s_0, s_1, \dots, s_{k-1}, s_0)\mu_{max}) \\
&\leq \mu_{max}\alpha
\end{aligned}$$

then with the assumption $\mu_{max} = 1$, the gap between our estimate of the average gain and the average gain is bounded by α which is arbitrarily small.

These previous results motivate the following work, if we are able to compute an optimistic policy over a set of plausible MDPs that maximize an optimistic version of such a gain $g^{\circlearrowleft}$, we should have an efficient learning strategy. This is not a trivial problem and an there is an additional key point which is to identify an optimal state.

6.2.2 Identifying optimal states candidates

In this sub-subsection we consider that giving a state we know how to compute its self-hitting for a given policy and how to compute the MDP and policy that allows maximizing the gain estimated from this given state.

While running our hypothetical algorithm we want to almost surely identify states that are optimal ones without doing an exhaustive research. To do so an idea could be to look at the rewards that we have (here we consider reward known for simplicity of redaction, but it could be done with optimistic rewards (empirical estimate + confidence bound)): the idea is the following: we denote $S^{r_{max}} = \{s \in S | \exists a \in A, (s,a) \in \text{argmax}_{s,a}\, r(s,a)\}$ then having $\tau_{min}^{r_{max}} = \min_{s \in S^{r_{max}}} \{\tau^{\circlearrowleft}(s, \alpha, \tilde{\pi})\}$ with $\tilde{\pi} = \text{argmax}_{pi}\, g^{\circlearrowleft}(s, \alpha, \pi)$, we know that almost surely there exists a state s such that $\exists a \in A, r(s,a) \geq \frac{r_{max}}{\tau_{min}^{r_{max}}}$ which is optimal. So while running the algorithm looking at such states is enough to guarantee that an optimal state is considered while looking to the gain $g^{\circlearrowleft}(s, \alpha, \tilde{\pi})$ and so this gain should be a good estimate of the optimal one.

So if we can compute the self-hitting times and this new definition of optimistic policy, we only have to consider the previous subset of states in order to have the best optimistic estimated gain (and associated policy).

6.2.3 Estimate $\tau^{\circlearrowleft}$

Estimating the α-self-hitting time for small α may be arbitrarily long in general, even while knowing an optimal state, for example we give the following MDP M: in M only one state denoted s_r can give a positive reward, every other state give zero reward, so we know that s_r is an optimal state, and it seems to be the only state that can easily be identified as an optimal state (having no more knowledge than the reward). Now let's suppose that only transitions with probability ε allow reaching s_r then we know that the self-hitting time with a high probability of the state is at most $1/\varepsilon$ which can be arbitrarily big.

So considering this result an algorithm looking for the α-self-hitting time with small α cannot be performed in practice in general. It may be relevant to stop looking for this self-hitting time when it appears that it is bigger than $2|\mathcal{S}|$ but doing so our estimate of the gain can become arbitrarily bad. This let think that identifying an optimal state is not enough to build a viable strategy, but it will be necessary to identify a state which non-negligibly optimal (visited enough to have a reasonable α-self-hitting time).

More work is necessary for these ideas in order to build a real strategy. The following open questions remain: How to compute in practice optimistic $g^{\circlearrowleft}$ and associated policy and self-hitting time? And is it possible to easily identify a non-negligibly optimal state? For this last question, we think that the strategy proposed in last subsection is enough in practice (not in general) as we hardly ever consider problems where states with highest rewards are not non-negligibly optimal states (at least for one of these states with high reward). In further work, we think that it should

be possible to build an optimistic strategy based on self-hitting times which could be an alternative to UCRL's strategy.

7 Conclusion

In this report, many things were presented around our RL problem. First, we focused on the UCRL2 algorithm, this algorithm is poorly performing in practice but it has strong theoretical guarantees that are interesting. Many improvements and modifications of this algorithm exist, we can separate these into two big families: we have improvements that are technical improvements of the algorithm (such as modification of routines, or concentrations bounds), and the second family contains improvements based on additional information gave as an input of the algorithm.

In this first family of technical improvements of UCRL2 there are many algorithms, but a few of them should retain our attention. The first one is KL-UCRL, even if this algorithm poorly performs in practice, better confidence bounds on the KL-divergence could be derived in order to improve this algorithm, and we should do some well-justified modifications of the algorithm in order to avoid all the errors catching modifications that are currently necessary to run it. Then we have our contribution, the UCRL3 algorithm, even if this algorithm globally does not improve the asymptotic bounds on the regret, results on standards environment of this algorithm are promising. There is still a point that should be considered in further work around UCRL3 it is the stopping criterion, globally a modification of the stopping criterion could be a good way to improve the algorithm, the modification based on hypothesis testing performs well in experiments, in practice it seems interesting to add it to UCRL3 even if it seems impossible to improve the theoretical guarantees thanks to such a modification (or a new sketch of proof is necessary). Additional work on the C-UCRL algorithm using ApproxEquivalence could also lead to an improvement of the algorithm, currently this improvement is only observed in ergodic environment which is really restrictive.

The second family of improvements is the one based on additional information as input. Globally we have three modifications here SCAL C-UCRL and UCWM. About SCAL interesting points are the theoretical bounds relying on the bias span and the possibility to learn in some non-communicating MDPs. But, in practice, we doubt about the possibility to actually have this bound on the bias span which is necessary as input. Using known equivalence structure with some variant of C-UCRL seems to be easier to use in practice, for example knowing partial class on transition probabilities is a reasonable assumption, it could be equivalent in a labyrinth problem to give the following trivial information to a robot learner: "wherever you are, walking always bring the same result if there is no obstacle". Additionally, these improvements come with a significant improvement on the theoretical guarantees which is in itself interesting. Finally, we have the UCWM strategy which is globally the idea to restrict the set of plausible MDPs using input information. In practice, this is really interesting and even probably necessary. It is the easiest way to give topological, or dangers information to the learner and the gain in practice is obvious in such cases, but it is impossible to improve general theoretical guarantees with this.

At the end of the report a short study around alternatives strategies to UCRL strategy has been presented, we doubt that the confusing MDP idea could lead to something in itself, but in further work, it may be interesting to build an alternative strategy based on self-hitting times and optimism.

As explained in this conclusion there is still a lot of work that could be done to improve the existing strategies or to build a new one. Additionally, other problems exist around these algorithms two of them are quite relevant to, one day, being able to apply these algorithms to real-life problems: the first one is to deal with continuous state space (as studied in [37] or [38]) the second is to deal

with non-stationary environments.

References

[1] Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *The Journal of Machine Learning Research*, 11:1563–1600, 2010.

[2] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT Press Cambridge, 1998.

[3] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[4] Sham Kakade. *On the sample complexity of reinforcement learning*. PhD thesis, University of London London, England, 2003.

[5] Tor Lattimore and Marcus Hutter. Near-optimal PAC bounds for discounted MDPs. *Theoretical Computer Science*, 558:125–143, 2014.

[6] Peter Auer and Ronald Ortner. Logarithmic online regret bounds for undiscounted reinforcement learning. *Advances in Neural Information Processing Systems 19 (NIPS)*, 19:49, 2007.

[7] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.*, 3:397–422, March 2003.

[8] Sarah Filippi, Olivier Cappé, and Aurélien Garivier. Optimism in reinforcement learning and Kullback-Leibler divergence. In *Proceedings of the 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 115–122, 2010.

[9] Mohammad Sadegh Talebi and Odalric-Ambrym Maillard. Variance-aware regret bounds for undiscounted reinforcement learning in MDPs. In *International Conference on Algorithmic Learning Theory (ALT)*, pages 770–805, 2018.

[10] Shipra Agrawal and Randy Jia. Optimistic posterior sampling for reinforcement learning: Worst-case regret bounds. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 1184–1194, 2017.

[11] Odalric-Ambrym Maillard and Mahsa Asadi. Upper Confidence Reinforcement Learning exploiting state-action equivalence. working paper or preprint, December 2018.

[12] Ronan Fruit, Matteo Pirotta, and Alessandro Lazaric. Near optimal exploration-exploitation in non-communicating markov decision processes. *arXiv preprint arXiv:1807.02373*, 2018.

[13] Christoph Dann, Tor Lattimore, and Emma Brunskill. Unifying PAC and regret: Uniform PAC bounds for episodic reinforcement learning. In *Advances in Neural Information Processing Systems 30 (NIPS)*, pages 5711–5721, 2017.

[14] Ronan Fruit, Matteo Pirotta, Alessandro Lazaric, and Ronald Ortner. Efficient bias-span-constrained exploration-exploitation in reinforcement learning. *arXiv preprint arXiv:1802.04020*, 2018.

[15] Ian Osband, Dan Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. In *Advances in Neural Information Processing Systems 26 (NIPS)*, pages 3003–3011, 2013.

[16] Peter L Bartlett and Ambuj Tewari. REGAL: A regularization based algorithm for reinforcement learning in weakly communicating MDPs. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 35–42, 2009.

[17] Shipra Agrawal and Navin Goyal. Further optimal regret bounds for Thompson sampling. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 99–107, 2012.

[18] Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *Algorithmic Learning Theory*, pages 199–213. Springer, 2012.

[19] Mohammad Gheshlaghi Azar, Alessandro Lazaric, and Emma Brunskill. Regret Bounds for Reinforcement Learning with Policy Advice. In *ECML/PKDD - European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Prague, Czech Republic, September 2013.

[20] Mahsa Asadi, Sadegh Talebi, Hippolyte Bourel, and Odalric-Ambrym Maillard. Model-based reinforcement learning exploiting state-action equivalence. In *Asian Conference on Machine Learning, ACML 2019*, 2019.

[21] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[22] Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J Weinberger. Inequalities for the L1 deviation of the empirical distribution. *Hewlett-Packard Labs, Technical Report*, 2003.

[23] Victor H Peña, Tze Leung Lai, and Qi-Man Shao. *Self-normalized processes: Limit theory and Statistical Applications*. Springer Science & Business Media, 2008.

[24] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, pages 2312–2320, 2011.

[25] Odalric-Ambrym Maillard. *Mathematics of Statistiscal Sequential Decision Making*. PhD thesis, Université de Lille Nord de France, 2019.

[26] Daniel Berend and Aryeh Kontorovich. On the concentration of the missing mass. *Electronic Communications in Probability*, 18(3):1–7, 2013.

[27] Maxim Raginsky, Igal Sason, et al. Concentration of measure inequalities in information theory, communications, and coding. *Foundations and Trends® in Communications and Information Theory*, 10(1-2):1–246, 2013.

[28] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

[29] Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

[30] Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 263–272, 2017.

[31] Ronald Ortner. Adaptive aggregation for reinforcement learning in average reward Markov decision processes. *Annals of Operations Research*, 208(1):321–336, 2013.

[32] A. Khaleghi, D. Ryabko, J. Mary, and P. Preux. Consistent algorithms for clustering time series. *JMLR*, 17(1):94–125, 2016.

[33] Emma Brunskill and Lihong Li. Sample complexity of multi-task reinforcement learning. In *Proc. of UAI*, page 122, 2013.

[34] Odalric-Ambrym Maillard. Basic Concentration Properties of Real-Valued Distributions. Lecture, September 2017.

[35] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016.

[36] Odalric-Ambrym Maillard. *Mathematics of Statistiscal Sequential Decision Making*. Habilitation à diriger des recherches, Université de Lille Nord de France, February 2019.

[37] Ronald Ortner and Daniil Ryabko. Online regret bounds for undiscounted continuous reinforcement learning. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 1763–1771, 2012.

[38] Jian Qian, Ronan Fruit, Matteo Pirotta, and Alessandro Lazaric. Exploration bonus for regret minimization in undiscounted discrete and continuous markov decision processes. *CoRR*, abs/1812.04363, 2018.

[39] Michael Kearns and Lawrence Saul. Large deviation methods for approximate probabilistic inference. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 311–319. Morgan Kaufmann Publishers Inc., 1998.

A Pseudo-code of implemented algorithms

A.1 KL-UCRL modifications to prevent errors

This implemented algorithm is as similar as the algorithm introduced by [**?**] that we manage to do. The algorithm of the paper cannot be implemented as presented in the paper (precisely the big problem is the pseudo-code of the MaxKL function of this paper), at high level this algorithm cannot be implemented because it leads to many evaluation of undefined functions (precisely the "f" function introduced by the paper is often undefined in the algorithm context, mostly at the initialization, so we have to catch some errors here, then a Newton optimization is done on this algorithm and again it leads to many errors because the function can be constant or the optimization can go out from the interval of definition of the function again: error catching) see the pseudo-code 7 for the detail of the modifications.

Algorithm 6 Original MaxKL as introduced in [8]

Input: A value function V, a probability vector p, a constant ε

$Z = \{i | p_i = 0\}$ and $\bar{Z} = \{i | p_i > 0\}$

$I^* = Z \cap \text{argmax}_i \, V_i$

if $I^* \neq \emptyset$ and $\exists i \in I^*, f(V_i) < \varepsilon$ **then**

$\nu = V_i$ and $r = 1 - \exp(f(\nu) - \varepsilon)$

$\forall i \in I^*$, define q_i such that:

$$\sum_{i \in I^*} q_i = r$$

$\forall i \in Z / I^*, q_i = 0$

else

$\forall i \in Z, q_i = 0$ and $r = 0$

Find ν such that $f(\nu) = \varepsilon$ using Newton's method.

end if

$\forall i \in \bar{Z}, q_i = \frac{(1-r)\tilde{q}_i}{\sum_{i \in \bar{Z}} \tilde{q}_i}$ with $\tilde{q}_i = \frac{p_i}{\nu - V_i}$

The additional errors filtering in KL-UCRL are the following: the initialization of the Newton's method is arbitrarily fixed to $1.1 \max(V[p > 0])$, the one proposed in the paper is not necessarily in the interval of definition of the function f (only asymptotically). The Newton's method is modified to stays in the interval of definition (when going out the value is replaced by the min of the definition interval + some small τ margin. And finally all the possible cases of divide by zeros are filtered (for example in f or in the definition of q) and depending on the situation we arbitrarily return a big value (10^{10} when the result should be $+\infty$ or -10^{10} when the result should be $-\infty$), these are observed only at the initialization of the algorithm, and so it does affect the execution in practice.

In practice, when we define q_i such that: $\sum_{i \in I^*} q_i = r$ we do: $\forall i \in I^*, q_i = \frac{r}{|I^*|}$.

About the confidence bounds: we use in this implementation the one used in the regret proof of the paper. And about the initialization of the Newton optimization: we don't use the initialization proposed in the appendix of the paper because it is not well defined for small T, we experimentally chose the bound: $1.1 \times max(V)$.

Algorithm 7 MaxKL with error catching

Input: A value function V, a probability vector p, a constant ε

$Degenerate = \mathsf{False}$
$Z = \{i | p_i = 0\}$ and $\bar{Z} = \{i | p_i > 0\}$
$I^* = Z \cap \operatorname{argmax}_i V_i$
if $I^* \neq \emptyset$ and $\exists i \in I^*, f(V_i) < \varepsilon$ **then**
  $\nu = V_i$ and $r = 1 - \exp(f(\nu) - \varepsilon)$
  $\forall i \in I^*$, define q_i such that:

$$\sum_{i \in I^*} q_i = r$$

  $\forall i \in Z / I^*, q_i = 0$
else
  $\forall i \in Z, q_i = 0$ and $r = 0$
  if $\exists \alpha, i_0$ such that $\forall i \neq i_0, p_i = 0$ and $p_{i_0} = \alpha$ **then**
    $q = p$ and $degenerate = \mathsf{True}$
  else
    Find ν such that $f(\nu) = \varepsilon$ using Modified Newton's method.
  end if
end if
$\forall i \in \bar{Z}, q_i = \frac{(1-r)\tilde{q}_i}{\sum_{i \in \bar{Z}} \tilde{q}_i}$ with $\tilde{q}_i = \frac{p_i}{\nu - V_i}$

A.2 SCAL

Because including the code here is not readable, we encourage the learner to have a look on our code on github (`https://github.com/HippolyteBourel/UCRL_implementation`).

But globally ScOpt's implementation is an exact copy of the EXTENDED VALUE ITERATION where, when the span of the estimated value (not the bias which is a difference with article) is bigger than the constant gave as input the span is constrained by choosing an almost optimal stochastic policy (2 action possible).

B Proofs

B.1 Time-uniform Bernoulli concentration bounds

In this section, we discuss the concentration of Bernoulli random variables, and make use of the special structure of Bernoulli variables to derive refined time-uniform concentration inequalities. Let us first recall that if $(X_i)_{i \leq n}$ are i.i.d. according to a Bernoulli $\mathcal{B}(p)$ with parameter $p \in [0, 1]$, then it holds by the Chernoff-method that

$$\forall \varepsilon \geq 0, \quad \mathbb{P}\left(\frac{1}{n} \sum_{i=1}^{n} (X_i - p) \geq \varepsilon \right) \leq \exp\left(- n \mathtt{kl}(p + \varepsilon, p) \right),$$

where $\mathtt{kl}(p, q) = p \log(p/q) + (1 - p) \log((1 - p)/(1 - q))$ denotes the Bernoulli Kullback-Leibler divergence. The reverse map of the Cramer transform $\varepsilon \mapsto \mathtt{kl}(p + \varepsilon, p)$ is unfortunately not explicit, and one may consider Taylor approximation of it to derive approximate but explicit high-probability

confidence bounds, see [39, 22, 26, 27]. On the other hand, since Bernoulli random variables are bounded in $[0,1]$ with variance $p(1-p)$, a Bernstein bound leads

$$\forall \delta \in [0,1], \quad \mathbb{P}\left(\frac{1}{n} \sum_{i=1}^{n} (X_i - p) \geq \sqrt{\frac{2p(1-p)\log(1/\delta)}{n}} + \frac{\log(1/\delta)}{3n} \right) \leq \delta \,.$$

In order to build time-uniform concentration inequalities, one may resort to two main tools. Either a time-peeling proof technique, that is generic and yields asymptotically optimal rate, or a Laplace mixture method for martingales, specific to sub-Gaussian variables but tighter for small n. It is natural to ask whether the Laplace method applies to Bernoulli variables. It turns out this is indeed possible, thanks to the following known control:

Lemma 3 (Sub-Gaussianity of Bernoulli random variables, see e.g. [26]) *For all $\forall p \in [0,1]$, the left and right tails of the Bernoulli distribution are controlled in the following way*

$$\forall \lambda \in \mathbb{R}, \qquad \log \mathbb{E}_{X \sim \mathcal{B}(p)} \exp(\lambda(X-p)) \leq \frac{\lambda^2}{2} g(p), \text{ where } g(p) = \frac{1/2 - p}{\log(1/p - 1)} \,.$$

The control on right-tail can be further refined when $p \in [\frac{1}{2}, 1]$, as follows:

$$\forall \lambda \in \mathbb{R}^+, \qquad \log \mathbb{E}_{X \sim \mathcal{B}(p)} \exp(\lambda(X-p)) \leq \frac{\lambda^2}{2} p(1-p) \,.$$

As an immediate corollary, introducing the function $\underline{g}(p) = \begin{cases} g(p) & \text{if } p < 1/2 \\ p(1-p) & \text{else} \end{cases}$, we deduce that

Corollary 2 (Time-uniform Bernoulli concentration) *Let $(X_i)_{i \leq n} \overset{i.i.d.}{\sim} \mathcal{B}(p)$. Then $\forall \delta \in [0,1]$,*

$$\mathbb{P}\left(\forall n \in \mathbb{N}, \; -\sqrt{\frac{2g(p)(1+\frac{1}{n})\log(\sqrt{n+1}/\delta)}{n}} \leq \frac{1}{n}\sum_{i=1}^{n} X_i - p \leq \sqrt{\frac{2\underline{g}(p)(1+\frac{1}{n})\log(\sqrt{n+1}/\delta)}{n}} \right) \geq 1 - 2\delta \,.$$

B.2 Near-optimistic optimization

B.2.1 First proposition

We study how UCRL3 implements the inner maximization in the EVI. To this end, we study the solution of the following problem, for a given function f:

$$\max \left\{ \sum_{x \in \mathcal{X}} f(x)\tilde{p}(x) : \tilde{p} \text{ s.t. } \forall x \in \hat{\mathcal{X}}_n \cup \star_f, \; |\tilde{p}(x) - \hat{p}_n(x)| \leq \beta_n(\tilde{p}(x), \delta) \text{ and } \mathrm{supp}(\tilde{p}) \subset \hat{\mathcal{X}}_n \cup \star_f \right\},$$

where $\star_f = \mathrm{Argmax}_{x \in \mathcal{V}} f(x)$ and $\hat{\mathcal{X}}_n = \mathrm{supp}(\hat{p}_n)$ denotes the support of the empirical distribution.

The following proof, proposed by my supervisor Odalric-Ambrym Maillard should allow to adapt the first step of the usual proof on regret for UCRL2 provided by [1] when applying the near-optimism of UCRL3. Globally the idea is to ensure that the result of the inner-maximization performed in UCRL3 is optimistic enough to verify that: $g_k \overset{\text{def}}{=} g_{\pi_{t_k}^+}^{\widetilde{M}_{t_k}} \geq g_\star - \frac{1}{\sqrt{t_k}} - E$ Instead of: $g_k \overset{\text{def}}{=} g_{\pi_{t_k}^+}^{\widetilde{M}_{t_k}} \geq g_\star - \frac{1}{\sqrt{t_k}}$ for UCRL2, with E the penalty brought by the near-optimism and that have to be determined.

Let us denote by F^+ the value of the optimization problem under constraints. On the one hand, by a simple Bernstein inequality, it holds that uniformly over all x, then with probability higher than $1-\delta$,

$$p(x) - \hat{p}_n(x) \leq \sqrt{\frac{2p(x)(1-p(x))\log(|\mathcal{X}|/\delta)}{n}} + \frac{\log(|\mathcal{X}|/\delta)}{3n}\,.$$

In particular, for any $x \notin \hat{\mathcal{X}}_n$, it comes

$$p(x) \leq \sqrt{p(x)}a + b\,, \text{ where } \begin{cases} a = \sqrt{\frac{2\log(|\mathcal{X}|/\delta)}{n}} \\ b = \frac{\log(|\mathcal{X}|/\delta)}{3n} \end{cases}$$

Solving this in $p(x)$ yields $p(x) \leq (\sqrt{a/4}+\sqrt{a/4+b})^2 = \frac{\log(|\mathcal{X}|/\delta)}{n}(\frac{1}{2}+\sqrt{\frac{5}{6}})^2 \leq \frac{2\log(|\mathcal{X}|/\delta)}{n}$. Introducing $\mathcal{X}_p = \mathrm{supp}(p)$, we thus deduce that

$$\sum_{x\in\mathcal{X}} f(x)p(x) \leq \sum_{x\in\mathcal{X}_p\cap(\hat{\mathcal{X}}_n\cup\star_f)} f(x)p(x) + \frac{2\|f\|_\infty|\mathcal{X}_p\setminus(\hat{\mathcal{X}}_n\cup\star_f)|\log(|\mathcal{X}|/\delta)}{n}$$

On the other hand, if we denote by $u_\delta(x) = \max\{q : |q-\hat{p}_n(x)| \leq \beta_n(q,\delta)\}$ the entry-wise upper-confidence bound on $p(x)$, and $\ell_\delta(x)$ the corresponding lower-confidence bound, then it comes for each $\tilde{p}$ considered in the optimization set,

$$\sum_{x\in\mathcal{X}} f(x)\tilde{p}(x) \geq \sum_{x\in\hat{\mathcal{X}}_n\cup\star_f} f(x)p(x) - \sum_{x\in\hat{\mathcal{X}}_n\cup\star_f} f(x)(u(x)-\ell(x))\,.$$

Using the fact that $\hat{\mathcal{X}}_n \subset \mathcal{X}_p$, one can also derive the slightly more precise bound:

$$\sum_{x\in\mathcal{X}} f(x)\tilde{p}(x) \geq \sum_{x\in\hat{\mathcal{X}}_p\cap(\mathcal{X}_n\cup\star_f)} f(x)p(x) - \sum_{x\in\mathcal{X}_p\cap(\hat{\mathcal{X}}_n\cup\star_f)} f(x)(u_\delta(x)-\ell_\delta(x)) - \frac{2\|f\|_\infty|\star_f\setminus\mathcal{X}_p|\log(|\mathcal{X}|/\delta)}{n}\,.$$

Hence, combining these two simple bounds, we deduce that the value F^+ computed by the optimization problem satisfies with probability higher than $1-2\delta$,

$$F^+ \geq \sum_{x\in\mathcal{X}} f(x)p(x) - \|f\|_\infty\Big[\sum_{x\in\mathcal{X}_p\cup\star_f}(u_\delta(x)-\ell_\delta(x)) + \frac{2|\mathcal{X}_p|\log(|\mathcal{X}|/\delta)}{n}\Big]$$

B.2.2 Second proposition

The previous result may not really be used in the regret analysis of UCRL3, the reason is the dependency on $\|f\|_\infty$ in the final bound. f being the value function when going back to the regret analysis and so it is not bounded. We can propose the following bound which may be easier to use in the regret analysis, for given pair (s,a):

$$\sum_{s'} u(s')\tilde{p}(s'|s,a) - \sum_{s'} u(s')\tilde{p}_{NOO}(s'|s,a) \leq sp(u)(\hat{K}-1)\beta(s,a) \leq sp(u)K\beta(s,a)$$

with β confidence bounds (considering upper and lower equals, in practice we may use the Laplace bounds without the $g(p)$ refinement) and with $\tilde{p}$ the optimistic transition and $\tilde{p}_{NOO}$ the near-optimistic one. We denote by $sp(u)$ the span of the value function, this is bounded by the diameter of the MDP D (see [1]) and so it seems more interesting than the previous infinite norm. K is the size of the true support, $\hat{K}$ is the observed size of the support. Here K and $\hat{K}$ are for the given pair (s, a), but by abuse of notation we will consider the maximum for all (s, a) in the regret analysis.

To prove this result we just follow the following reasoning: while executing the optimistic or near-optimistic sub-routine to maximize the inner maximum in the `Extended Value Iteration`, globally we just distribute a mass of probability bounded by $\hat{K}\beta(s, a)$ to most optimistic transitions for optimistic case and in near-optimistic case to the most optimistic one plus the one that are on the experimental support. Then considering this problem of mass distribution, we immediately have a bound considering the worst case: the optimistic strategy put the all mass on transition with max value u_{max} while all transitions on the experimental support have the lowest value u_{min}, then it immediately give that an amount of probability mass equal to $(\hat{K} - 1)\beta(s, a)$ can be in worst case put to the minimum value instead of the maximum one, giving immediately the aforementioned bound.

We may notice that following the same reasoning a similar bound can be proved when considering p the true transition instead of $\tilde{p}$ (probably possible to introduce a factor $1/2$, using the fact that all observed transition are actually in the true support).

Including this result in the `Extended Value Iteration` context we keep the similar following result:

$$\sum_{s'} u(s')\tilde{p}(s'|s, \tilde{\pi}(s)) - \sum_{s'} u(s')\tilde{p}_{NOO}(s'|s, \tilde{\pi}_{NOO}) \leq sp(u)(\hat{K} - 1)\beta(s, \tilde{\pi}) \leq sp(u)K\beta(s, \tilde{\pi})$$

as the policy is chosen while maximizing through actions, this bound due to worse case is still true.

B.3 Near-optimism in the `Extended Value Iteration`

B.3.1 (Temporary) Remarks and To Do

When introducing NOO in the `Extended Value Iteration` we get a penalty in the order of $KDSAT \log(\sqrt{T})$ in the regret bound (coming from the fact that we will have a term $\sum_{s,a} \frac{\nu(s,a)}{\min_{s,a} N_k(s,a)}$). This penalty disappearing when the true support is (almost) known (all transition observed at least once, or all except one for all (s, a) pair), due to the fact that the true MDP M is then included in the NO plausible set of MDP $\mathcal{M}_k^{NOO}$.

It lead us to two regime: when support unknown: regret bounded by linear, and when support almost known:"good" bound on the regret.

In order to propose an algorithm with stronger regret-guarantees: we propose the following modification of `UCRL3`: as input the algorithm take K maximum size of support as input (and we will denote this algorithm `UCRL3`(K), then we use this knowledge by having this new NOO: We modify only experimental support **plus** $K - \hat{K}$ optimistic transitions. Having this the true MDP is in $\mathcal{M}_k^{NOO}$ so we have only one regime in the regret guarantees (but I expect slightly weaker results in practice).

B.3.2 Temporary (the true content of this subsection)

This section is based on the beginning of the chapter 6 of [36]. Let introduce the span of a function f: $\mathbb{S}(f) = \max_x f(x) - \min_x f(x)$. Then we denote $\mathbf{P}_\pi f$ the function such that for all $s \in \mathcal{S}$,

$(\mathbf{P}_\pi f)(s) = \sum_{s' \in \mathcal{S}} p(s'|s, \pi(s)) f(s')$. We also introduce the average transition operator of policy π denoted $\bar{\mathbf{P}}_\pi$ (when it exists) and such that: $\bar{\mathbf{P}}_\pi = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \mathbf{P}_\pi^{t-1}$. Various properties of these operator and the gain are presented in beginning of chapter 6 from [36].

When the true MDP M belongs to the set of plausible MDPs, $M \in \mathcal{M}_k$, we have two cases, denoting $\mathcal{M}_k^{NOO}$ the set of MDP that are considered by the NOO:

First case: $M \in \mathcal{M}_k^{NOO}$ If the true MDP belong to the set of MDP considered by the NOO that implies that the experimental support is the true support or the true support minus one element (for all pairs (s, a)). Then usual results around the EXTENDED VALUE ITERATION holds and we verify the following properties.

To complete: introduce EXTENDED VALUE ITERATION notations

Proposition 2 (Value and gain in favorable sub-case (Corollary 6.2 of [36]))

$$\forall n \in \mathbb{N}, \quad \bar{\mathbf{P}}_{\pi_{n+1}}[u_{n+1} - u_n] \leq g_{\pi_{n+1}} \leq g_\star \leq \bar{\mathbf{P}}_\star[u_{n+1} - u_n]$$

For any any n such that $\mathbb{S}(u_{n+1} - u_n) \leq \varepsilon$, then

$$g_\star - g_{\pi_{n+1}} \leq \varepsilon, \quad |u_{n+1} - u_n - g_\star| \leq \varepsilon, \quad |u_{n+1} - u_n - g_{\pi_{n+1}}| \leq \varepsilon$$

Second case: $M \notin \mathcal{M}_k^{NOO}$ If the true does not belong to the set of MDP $\mathcal{M}_k^{NOO}$ that is considered by the NOO, then we have an additional penalty term in the EXTENDED VALUE ITERATION. This penalty come from results of appendix B.2, and lead to the following results.

Proposition 3 (Value and gain in penalized case) *Add the penalty*

$$\forall n \in \mathbb{N}, \quad \bar{\mathbf{P}}_{\pi_{n+1}}[u_{n+1} - u_n] \leq g_{\pi_{n+1}} \leq g_\star \leq \bar{\mathbf{P}}_\star[u_{n+1} - u_n]$$

For any any n such that $\mathbb{S}(u_{n+1} - u_n) \leq \varepsilon$, then

$$g_\star - g_{\pi_{n+1}} \leq \varepsilon, \quad |u_{n+1} - u_n - g_\star| \leq \varepsilon, \quad |u_{n+1} - u_n - g_{\pi_{n+1}}| \leq \varepsilon$$

Proof: The following proof closely follow the one presented in chapter 6 of [36] for the Corollary 6.2. The global idea is to use results presented in appendix B.2 to modify the first steps of of the proof and introduce our penalty term.

We have... To finish □

C Regret analysis of UCRL3

This section is an unfinished draft of the proof of the regret bound on UCRL3. An updated version of this report will be put in our github code repository as soon as the proof is wrote.

The analysis is based on the analysis of UCRL2 given by [1]. We are going to investigate UCRL3 by analyzing the regret of the algorithm. Let $m(T)$ denote the number of episodes initiated by the

algorithm up to time T. The effective regret naturally decomposes episode-wise

$$\begin{aligned}\mathfrak{R}(\mathbb{A}, T) &= \sum_{t=1}^{T} g_\star(s_1) - \sum_{t=1}^{T} \mu(s_t, a_t) \\ &= \sum_{k=1}^{m(T)} \sum_{s,a \in \mathcal{S}\times\mathcal{A}} \underbrace{\sum_{t=t_k+1}^{t_{k+1}} \mathbb{I}s_t = s, a_t = a}_{\nu_k(s,a)} \left(g_\star - \mu(s,a)\right) \\ &= \sum_{k=1}^{m(T)} \Delta_k \,.\end{aligned}$$

where we introduced the effective regret in episode k

$$\Delta_k = \sum_{s,a \in \mathcal{S}\times\mathcal{A}} \nu_k(s,a)(g_\star - \mu(s,a)) \,.$$

We say an episode is *good* if $M \in \mathcal{M}_{t_k}$ (that is, the set of plausible MDPs contains the true model), and bad otherwise.

Control of the regret due to bad episodes: $M \notin \mathcal{M}_{t_k}$. Due to using time-uniform instead of time-instantaneous confidence bounds, we can show that with high probability, all episodes are good for all horizons. More precisely, with probability higher than $1-\delta$, for all T, bad episodes do not contribute to the regret:

$$\sum_{k=1}^{m(T)} \Delta_k \mathbb{I}M \notin \mathcal{M}_{t_k} = 0 \,.$$

Control of the regret due to good episodes: $M \in \mathcal{M}_{t_k}$. We closely follow [1] and decompose the regret to control the transition and reward functions. But we also have to distinguish two sub-case due to our Near Optimistic Optimization:

- First sub-case: $M \in \mathcal{M}_k^{NOO}$; in this case the proof is a simple adaptation of the usual one using the new confidence bounds and dealing with the element-wise bounds.
- Second sub-case: $M \notin \mathcal{M}_k^{NOO}$; in this case a penalty term brought by the Near Optimistic Optimization in the Extended Value Iteration appears (see section B.3).

The key point around these two sub-case is that it is impossible in general to prove that our algorithm will reach with high probability the first and favorable sub-case, and in practice it is wrong: the algorithm may never observe enough some state-actions pairs in order to reach this first sub-case (because we have to observe true support minus one element to reach it). This lead to the following observation: UCRL3 will have two asymptotical regime, depending the sub-case. This lead us to the introduction of UCRL3(K) which is always in the favorable sub-case.

In the following proof we consider the second sub-case which brought a penalty to the regret bound of UCRL3, for the favorable sub-case (which will be quickly reached in practice when $K \ll S$) we refer to the regret analysis of UCRL3(K) (see appendix D).

At a high level, the first modification of the usual proof that we do is to use a time-uniform bound to control the martingale difference sequence that appears in the proof, using the following result:

Lemma 4 (Time-uniform Azuma-Hoeffding) *For any martingale difference sequence $(X_t)_t$ bounded by D (that is, $|X_t| \leq D$ for all t) it comes, by application of time-uniform Laplace concentration inequality for bounded variables,*

$$\mathbb{P}\Big(\exists T \in \mathbb{N} : \sum_{t=1}^{T} X_t \geq D\sqrt{2(T+1)\log(\sqrt{T+1}/\delta)}\Big) \leq \delta\,.$$

Details Since $M \in \mathcal{M}_{t_k}$ by assumption and by choosing $\pi^+_{t_k}$ and $\widetilde{M}_{t_k}$ by following the algorithm, we get that $g_k \overset{\text{def}}{=} g^{\widetilde{M}_{t_k}}_{\pi^+_{t_k}} \geq g_\star - \frac{1}{\sqrt{t_k}}$. It then follows that To modify, need to consider the Near Optimistic Strategy

$$\Delta_k \leq \sum_{s,a\in\mathcal{S}\times\mathcal{A}} \nu_k(s,a)(g_\star - \mu(s,a)) \leq \sum_{s,a\in\mathcal{S}\times\mathcal{A}} \nu_k(s,a)(g_k - \mu(s,a)) + \sum_{s,a\in\mathcal{S}\times\mathcal{A}} \frac{\nu_k(s,a)}{\sqrt{t_k}} \tag{18}$$

To modify, need proof of EVI for UCRL3 Besides, according to the proof of EVI in `UCRL2`, we also have, for the output value at iteration i,

$$\max_s u_i(s) - \min_s u_i(s) \leq D \tag{19}$$

where D is the diameter of the MDP. Moreover, when the convergence criterion of EVI holds, we have:

$$|u_{i+1}(s) - u_i(s) - g_k| \leq \frac{1}{\sqrt{t_k}}\,, \qquad \forall s \in S \tag{20}$$

where g_k is the average reward of the policy $\pi^+_{t_k}$ chosen on the optimistic MDP $\widetilde{M}_{t_k}$. Using Bellman operator on the optimistic MDP:

$$u_{i+1}(s) = \mu^+_{t_k}(s,\pi^+_{t_k}(s)) + \sum_{s'} p^+_{t_k}(s'|s,\pi^+_{t_k}(s))u_i(s')\,,$$

Employing the above equation alongside with (20) leads to:

$$|\Big(g_k - \mu^+_{t_k}(s,\pi^+_{t_k}(s))\Big) - \Big(\sum_{s'} p^+_{t_k}(s'|s,\pi^+_{t_k}(s))u_i(s') - u_i(s)\Big)| \leq \frac{1}{\sqrt{t_k}}$$

By defining the column vector of $\mathbf{g}_k = g_k\mathbf{1}$, $\tilde{\mathbf{r}}_k := \big(\mu^+_{t_k}(s,\pi^+_{t_k}(s))\big)_s$ for $\pi^+_{t_k}$ policy, $\widetilde{\mathbf{P}}_k := \big(p^+_{t_k}(s'|s,\pi^+_{t_k}(s))\big)_{s,s'}$ and $\nu_k := \big(\nu_k(s,\pi^+_{t_k}(s))\big)$, we can rewrite the above equation as:

$$|\Big(\mathbf{g}_k - \tilde{\mathbf{r}}_k\Big)_s - \Big((\widetilde{\mathbf{P}}_k - \mathbf{I})u_i\Big)_s| \leq \frac{1}{\sqrt{t_k}}$$

Therefore, (18) can be rewritten as:

$$\begin{aligned}
\Delta_k &\leq \sum_{s,a\in\mathcal{S}\times\mathcal{A}} \nu_k(s,a)(g_k - \mu(s,a)) + \sum_{s,a\in\mathcal{S}\times\mathcal{A}} \frac{\nu_k(s,a)}{\sqrt{t_k}} \\
&= \sum_{s,a\in\mathcal{S}\times\mathcal{A}} \nu_k(s,a)(g_k - \mu^+_{t_k}(s,a)) + \sum_{s,a\in\mathcal{S}\times\mathcal{A}} \nu_k(s,a)(\mu^+_{t_k}(s,a) - \mu(s,a)) + \sum_{s,a\in\mathcal{S}\times\mathcal{A}} \frac{\nu_k(s,a)}{\sqrt{t_k}} \\
&\leq \nu_k(\widetilde{\mathbf{P}}_k - \mathbf{I})u_i + \sum_{s,a\in\mathcal{S}\times\mathcal{A}} \nu_k(s,a)(\mu^+_{t_k}(s,a) - \mu(s,a)) + 2\sum_{s,a\in\mathcal{S}\times\mathcal{A}} \frac{\nu_k(s,a)}{\sqrt{t_k}}\,.
\end{aligned} \tag{21}$$

Since each row of $\widetilde{\mathbf{P}}_k$ sums up to one, we can add a constant to u_i and still have the same equation. So, we are going to replace it with w_k:

$$w_k(s) := u_i(s) - \frac{\min_s u_i(s) + \max_s u_i(s)}{2}\,.$$

At this point we are going to bound the reward part using reward confidence bound. Knowing that $M \in \mathcal{M}_{t_k}$, then $\mu^+_{t_k}(s,a) - \mu(s,a) \leq |\mu^+_{t_k}(s,a) - \widehat{\mu}_{t_k}(s,a)| + |\widehat{\mu}_{t_k}(s,a) - \mu(s,a)|$ can be bounded by $2\beta'_{N_{t_k}}(s,a)$. Therefore,

$$\begin{aligned}
\Delta_k &\leq \nu_k(\widetilde{\mathbf{P}}_k - \mathbf{I})w_k + 2\sum_{s,a\in\mathcal{S}\times\mathcal{A}} \nu_k(s,a)\mathbb{H}_{N_{t_k}(s,a)}(\frac{\delta}{2SA}) + 2\sum_{s,a\in\mathcal{S}\times\mathcal{A}} \frac{\nu_k(s,a)}{\sqrt{t_k}} \\
&\leq \nu_k(\widetilde{\mathbf{P}}_k - \mathbf{I})w_k + 2\sum_{s,a\in\mathcal{S}\times\mathcal{A}} \nu_k(s,a)\sqrt{\frac{(1+\frac{1}{N_{t_k}(s,a)})\log(4SA\sqrt{N_{t_k}(s,a)+1}/\delta)}{2N_{t_k}(s,a)}} + 2\sum_{s,a\in\mathcal{S}\times\mathcal{A}} \frac{\nu_k(s,a)}{\sqrt{t_k}}
\end{aligned}$$

Since $\max\{1, N_{t_k}(s,a)\} \leq t_k \leq T$,

$$\Delta_k \leq \nu_k(\widetilde{\mathbf{P}}_k - \mathbf{I})w_k + \Big(\sqrt{4\ln(\frac{2SA\sqrt{T}}{\delta})} + 2\Big)\sum_{s,a\in\mathcal{S}\times\mathcal{A}} \frac{\nu_k(s,a)}{\sqrt{\max\{1, N_{t_k}(s,a)\}}} \tag{22}$$

Afterwards we have to bound the first part of the above equation, $\nu_k(\widetilde{\mathbf{P}}_k - \mathbf{I})w_k$, which is going to be done by using the confidence bound on the transition probability distribution. Having defined $\mathbf{P}_k := \big(p(s'|c_{s,\pi^+_{t_k}(s)})\big)_{s,s'}$:

$$\nu_k(\widetilde{\mathbf{P}}_k - \mathbf{I})w_k = \nu_k(\widetilde{\mathbf{P}}_k - \widehat{\mathbf{P}}_k + \widehat{\mathbf{P}}_k - \mathbf{P}_k + \mathbf{P}_k - \mathbf{I})w_k = \nu_k(\widetilde{\mathbf{P}}_k - \widehat{\mathbf{P}}_k)w_k + \nu_k(\widehat{\mathbf{P}}_k - \mathbf{P}_k)w_k + \nu_k(\mathbf{P}_k - \mathbf{I})w_k \tag{23}$$

Assuming that $M \in \mathcal{M}_{t_k}$ while knowing that $\|w_k\| \leq \frac{D}{2}$ and taking advantage from the Jensen inquality:

$$\begin{aligned}
\nu_k(\widetilde{\mathbf{P}}_k - \widehat{\mathbf{P}}_k)w_k &= \sum_{s,s'} \nu_k(s,\pi^+_{t_k}(s))\Big(\tilde{p}^+_{t_k}(s'|s,\pi^+_{t_k}(s)) - \hat{p}(s'|s,\pi^+_{t_k}(s))\Big)w_k(s') \\
&= \sum_{s} \sum_{s'\in\mathrm{supp}(\hat{p}(s'|s,pi^+_{t_k}(s)))} \nu_k(s,\pi^+_{t_k}(s))\Big(p^+_{t_k}(s'|s,\pi^+_{t_k}(s)) - \hat{p}(s'|s,\pi^+_{t_k}(s))\Big)w_k(s')
\end{aligned}$$

Using Jensen's inequality and Lemma 6 we have: Unfinished modification below

$$
\begin{aligned}
&\sum_{s'} \tilde{p}^+_{t_k}(s'|s,\pi^+_{t_k}(s)) - \hat{p}(s'|s,\pi^+_{t_k}(s) \\
\leq\ & \sum_{s'} \min\Bigg(\sqrt{\frac{g(p)\log(\sqrt{N_{t_k}(s,\pi^+_{t_k}(s))+1})\left(1+\frac{1}{N_{t_k}(s,\pi^+_{t_k}(s))}\right)}{N_{t_k}(s,\pi^+_{t_k}(s))}}, \\
& \sqrt{\frac{\hat{p}(s'|s,\pi^+_{t_k}(s))}{N_{t_k}(s,\pi^+_{t_k}(s))}\ell_{N_{t_k}(s,\pi^+_{t_k}(s))}(\tfrac{\delta}{2SA})} + \frac{\ell_{N_{t_k}(s,\pi^+_{t_k}(s))}(\frac{\delta}{2SA})}{3N_{t_k}(s,\pi^+_{t_k}(s))}\Bigg) \\
\leq\ & \sum_{s'}\left(\sqrt{\frac{\hat{p}(s'|s,\pi^+_{t_k}(s))}{N_{t_k}(s,\pi^+_{t_k}(s))}\ell_{t_k}(\tfrac{\delta}{2SA})} + \frac{\ell_{t_k}(\frac{\delta}{2SA})}{3t_k}\right) \\
\leq\ & \sqrt{\sum_{s'}\frac{\hat{p}(s'|s,\pi^+_{t_k}(s))}{N_{t_k}(s,\pi^+_{t_k}(s))}\ell_{N_{t_k}(s,\pi^+_{t_k}(s))}(\tfrac{\delta}{2SA})} + \frac{\ell_{N_{t_k}(s,\pi^+_{t_k}(s))}(\frac{\delta}{2SA})}{3N_{t_k}(s,\pi^+_{t_k}(s))} \\
\leq\ & \sqrt{\frac{\ell_{N_{t_k}(s,\pi^+_{t_k}(s))}(\frac{\delta}{2SA})}{N_{t_k}(s,\pi^+_{t_k}(s))}\sum_{s'\in\mathrm{supp}(\hat{p}(s'|s,pi^+_{t_k}(s)))}\hat{p}(s'|s,\pi^+_{t_k}(s)) + \frac{\ell_{N_{t_k}(s,\pi^+_{t_k}(s))}(\frac{\delta}{2SA})}{3N_{t_k}(s,\pi^+_{t_k}(s))}} \\
\leq\ & \sqrt{\frac{\ell_{N_{t_k}(s,\pi^+_{t_k}(s))}(\frac{\delta}{2SA})}{N_{t_k}(s,\pi^+_{t_k}(s))}K + \frac{\ell_{N_{t_k}(s,\pi^+_{t_k}(s))}(\frac{\delta}{2SA})}{3N_{t_k}(s,\pi^+_{t_k}(s))}} \\
\leq\ & \sqrt{\frac{\ell_{N_{t_k}(s,\pi^+_{t_k}(s))}(\frac{\delta}{2SA})}{N_{t_k}(s,\pi^+_{t_k}(s))}K + 2\log(N_{t_k}(s,\pi^+_{t_k}(s)))+1}
\end{aligned}
$$

Similarly, we also have:

$$
\sum_{s'}\hat{p}^+_{t_k}(s'|s,\pi^+_{t_k}(s)) - p(s'|s,\pi^+_{t_k}(s) \leq \sqrt{\frac{\ell_{N_{t_k}(s,\pi^+_{t_k}(s))}(\frac{\delta}{2SA})}{N_{t_k}(s,\pi^+_{t_k}(s))}K + 2\log(N_{t_k}(s,\pi^+_{t_k}(s)))+1}
$$

Then we have:

$$
\begin{aligned}
\nu_k(\widetilde{\mathbf{P}}_k - \widehat{\mathbf{P}}_k)w_k \ &\leq\ \sum_s \nu_k(s,\pi^+_{t_k}(s))\left(\sqrt{\frac{\ell_{N_{t_k}(s,\pi^+_{t_k}(s))}(\frac{\delta}{2SA})}{N_{t_k}(s,\pi^+_{t_k}(s))}K + 2\log(N_{t_k}(s,\pi^+_{t_k}(s)))+1}\right)\frac{D}{2} \\
&\leq
\end{aligned}
$$

To pursue

To bound the third term in (23), we are going to use Lemma 4 and Proposition 18 in [1]. Let us define $X_t := (p(\cdot|s_t,a_t) - \mathbf{e}_{s_{t+1}})w_k(t)\mathbf{I}_{M\in\mathcal{M}_{t_k}}$, $\forall t = 1,\ldots,T$. For any k with $M\in\mathcal{M}_{t_k}$, we have

that:

$$\nu_k(\mathbf{P}_k - \mathbf{I})w_k = \sum_{t=t_k}^{t_{k+1}-1} (p(\cdot|s_t,a_t) - \mathbf{e}_{s_t})w_k = \sum_{t=t_k}^{t_{k+1}-1} \Big(p(\cdot|s_t,a_t) - \mathbf{e}_{s_{t+1}} + \mathbf{e}_{s_{t+1}} - \mathbf{e}_{s_t}\Big)w_k$$
$$= \sum_{t=t_k}^{t_{k+1}-1} X_t + w_k(s_{t+1}) - w_k(s_t) \leq \sum_{t=t_k}^{t_{k+1}-1} X_t + D\,.$$

Knowing that $\|w_k\|_\infty = \frac{D}{2}$ and using the Hölder inequality,

$$|X_t| \leq \|p(\cdot|s_t,a_t) - \mathbf{e}_{s_{t+1}}\|_1 \frac{D}{2} \leq \Big(\|p(\cdot|s_t,a_t)\|_1 + \|\mathbf{e}_{s_{t+1}}\|_1\Big)\frac{D}{2} = D\,.$$

So, X_t is bounded by D, and also $\mathbb{E}[X_t|s_1,a_1,\ldots,s_t,a_t] = 0$, so that $(X_t)_t$ is a sequence of martingale differences. Therefore, by using Lemma 4, we get:

$$\mathbb{P}(\exists T : \sum_{t=1}^{T} X_t \geq \underbrace{D\sqrt{2(T+1)\log(\sqrt{T+1}/\delta)}}_{\varepsilon_{p_2}}\Big) \leq \delta\,.$$

Using Proposition 18 in [1]., we know that $m \leq SA\log_2\left(\frac{8T}{SA}\right)$. By summing over all episodes we get:

$$\sum_{k=1}^{m(T)} \nu_k(\mathbf{P}_k - \mathbf{I})w_k\mathbf{I}\{M \in \mathcal{M}_{t_k}\} \leq \sum_{t=1}^{T} X_t + m(T) \leq \varepsilon_{p_2} + m(T)D \leq \varepsilon_{p_2} + DSA\log_2(\tfrac{8T}{SA}) \qquad (24)$$

with probability $1-2\delta$.

Final control. Using a union bound over the event that $\mathcal{M}$ is plausible at any time, and over the control of the martingale difference sequence $(X_t)_t$ appearing in the control of good episodes, we deduce that the regret is controlled with probability higher than $1-2\delta$, uniformly over all $T \in \mathbb{N}$. More precisely, using (22)–(24) and summing over all episodes: To modify

$$\begin{aligned}
\sum_{k=1}^{m} \Delta_k \mathbf{I}_{M\in\mathcal{M}_{t_k}} &\leq \sum_{k=1}^{m} \nu_k(\widetilde{\mathbf{P}}_k - \mathbf{P}_k)w_k\mathbf{I}_{M\in\mathcal{M}_{t_k}} + \sum_{k=1}^{m} \nu_k(\mathbf{P}_k - \mathbf{I})w_k\mathbf{I}_{M\in\mathcal{M}_{t_k}} \\
&\quad + \sum_{k=1}^{m} \Big(\sqrt{4\ln(\frac{2SA\sqrt{T+1}}{\delta})} + 2\Big) \sum_{s,a\in\mathcal{S}\times\mathcal{A}} \frac{\nu_k(s,a)}{\max\{1, N_{t_k}(s,a)\}} \\
&\leq \Big(D\sqrt{4\ln\big(\frac{2SA\sqrt{T+1}(2^S-2)}{\delta}\big)} + \sqrt{4\ln(\frac{2SA\sqrt{T+1}}{\delta})} + 2\Big) \sum_{k=1}^{m}\sum_{s,a} \frac{\nu_k(s,a)}{\max\{1, N_{t_k}(s,a)\}} \\
&\quad + D\sqrt{2(T+1)\log(\sqrt{T+1}/\delta)} + DSA\log_2\left(\frac{8T}{SA}\right).
\end{aligned}$$

To bound the above equation, we are going to introduce the below lemma:

Lemma 5 ([1, Lemma 19]) *For any sequence of numbers $z_1, z_2, \ldots, z_n$ with $0 \leq z_k \leq Z_{k-1} := \max\{1, \sum_{i=1}^{k-1} z_i\}$,*

$$\sum_{k=1}^{n} \frac{z_k}{\sqrt{Z_{k-1}}} \leq (\sqrt{2}+1)\sqrt{Z_n}\,.$$

Knowing that $N_T(s,a) := \sum_k \nu_k(s,a)$, $\sum_{s,a} N_T(s,a) = T$ and $N_{t_k}(s,a) = \sum_{i<k} \nu_i(s,a)$, and using the above lemma we get:

$$\sum_{s,a\in\mathcal{S}\times\mathcal{A}} \sum_{k=1}^{m} \frac{\nu_k(s,a)}{\max\{1, N_{t_k}(s,a)\}} \leq \sum_{s,a\in\mathcal{S}\times\mathcal{A}} \Big(\sqrt{2}+1\Big)\sqrt{N_T(s,a)}$$

On the other hand, using the Jensen's inequality we get:

$$(\sqrt{2}+1) \sum_{s,a\in\mathcal{S}\times\mathcal{A}} \sqrt{N_T(s,a)} \leq (\sqrt{2}+1) SA \frac{\sum_{s,a\in\mathcal{S}\times\mathcal{A}} \sqrt{N_T(s,a)}}{SA} \leq (\sqrt{2}+1)\sqrt{SAT}$$

Therefore,

$$\sum_{s,a\in\mathcal{S}\times\mathcal{A}} \sum_{k=1}^{m} \frac{\nu_k(s,a)}{\max\{1, N_{t_k}(s,a)\}} \leq (\sqrt{2}+1)\sqrt{SAT}$$

And by using the above result in (25), we get:

$$\begin{aligned}\sum_{k=1}^{m(T)} \Delta_k \mathbf{I}_{M\in\mathcal{M}_{t_k}} &\leq D\sqrt{2(T+1)\log(\sqrt{T+1}/\delta)} + DSA\log_2(\frac{8T}{SA}) \\ &+ \Big(D\sqrt{4\ln\big(\frac{2SA\sqrt{T+1}(2^S-2)}{\delta}\big)} + \sqrt{4\ln(\frac{2SA\sqrt{T+1}}{\delta})} + 2\Big)(\sqrt{2}+1)\sqrt{SAT}\,,\end{aligned} \tag{25}$$

with probability of at least $1-\delta-\delta$. Finally, the regret of UCRL3 is controlled on an event of probability higher than $1-\delta$, uniformly over all time T, by

$$\begin{aligned}\mathfrak{R}(T) \;\leq\; &\Big(D\sqrt{4\ln\big(\frac{2SA\sqrt{T+1}(2^S-2)}{\delta}\big)} + \sqrt{4\ln(\frac{2SA\sqrt{T+1}}{\delta})} + 2\Big)\Big(\sqrt{2}+1\Big)\sqrt{SAT} \\ &+D\sqrt{2(T+1)\log(\sqrt{T+1}/\delta)} + DSA\log_2(\tfrac{8T}{SA})\,.\end{aligned}$$

We conclude by polishing the bound to highlight the main terms of the regret:

$$\begin{aligned}\mathfrak{R}(T) \;&\leq\; [6(\sqrt{2}+1)+\sqrt{2}+1]D\sqrt{S^2AT\ln(SA\sqrt{T}/\delta)} \\ &\leq\; 17D\sqrt{S^2AT\ln(SA\sqrt{T}/\delta)}\,.\end{aligned}$$

The bound in the main body of the paper removes the $\sqrt{4\ln(\frac{2SA\sqrt{T+1}}{\delta})}$ when assuming the mean rewards are known. □

Lemma 6 *Consider a sequence $(z_k)_{1\leq k\leq n}$ with $0 \leq z_k \leq Z_{k-1} := \max\{1, \sum_{i=1}^{k-1} z_i\}$ for $k \geq 1$ and $Z_0 = z_1$. Then,*

$$\sum_{k=1}^{n} \frac{z_k}{Z_{k-1}} \leq 2\log(Z_n) + 1\,.$$

Theorem 5 (Regret of UCRL3) *With probability higher than $1-2\delta$, uniformly over all time horizon T,*

$$\mathfrak{R}(\textit{UCRL3}, T) \leq 24D\sqrt{KSAT\log(\log(T)/\delta)} + \mathcal{O}(SA\log(T))\,.$$

D Regret analysis of UCRL3