Conservative Exploration in Reinforcement Learning

Evrard Garcelon **Mohammad Ghavamzadeh** **Alessandro Lazaric** **Matteo Pirotta**
Facebook AI Reasearch

Abstract

While learning in an unknown Markov Decision Process (MDP), an agent should trade off exploration to discover new information about the MDP, and exploitation of the current knowledge to maximize the reward. Although the agent will eventually learn a good or optimal policy, there is no guarantee on the quality of the intermediate policies. This lack of control is undesired in real-world applications where a minimum requirement is that the executed policies are guaranteed to perform at least as well as an existing baseline. In this paper, we introduce the notion of conservative exploration for average reward and finite horizon problems. We present two optimistic algorithms that guarantee (w.h.p.) that the conservative constraint is never violated during learning. We derive regret bounds showing that being conservative does not hinder the learning ability of these algorithms.

1 Introduction

While Reinforcement Learning (RL) has achieved tremendous successes in simulated domains, its use in real system is still rare. A major obstacle is the lack of guarantees on the learning process, that makes difficult its application in domains where hard constraints (e.g., on safety or performance) are present. Examples of such domains are digital marketing, healthcare, finance, and robotics. For a vast number of domains, it is common to have a known and reliable *baseline policy* that is potentially suboptimal but satisfactory. Therefore, for applications of RL algorithms, it is important that are guaranteed to perform at least as well as the existing baseline.

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

In the offline setting, this problem has been studied under the name of *safety w.r.t. a baseline* (Bottou et al., 2013; Thomas et al., 2015a,b; Swaminathan and Joachims, 2015; Petrik et al., 2016; Laroche et al., 2019; Simão and Spaan, 2019). Given a set of trajectories collected with the baseline policy, these approaches aim to learn a policy –without knowing or interacting with the MDP– that is guaranteed (e.g., w.h.p.) to perform at least as good as the baseline. This requires that the set of trajectories is sufficiently reach in order to allow to perform counterfactual reasoning with it. This often implies strong requirements on the ability of exploration of the baseline policy. These approaches can be extended to a semi-batch settings where phases of offline learning are alternated with the executing of the improved policy. This is the idea behind conservative policy iteration (e.g., Kakade and Langford, 2002; Pirotta et al., 2013b) where the goal is to guarantee a monotonic policy improvement in order to overcome the policy oscillation phenomena Bertsekas (2011). These approaches has been successively extended to function approximation preserving theoretical guarantees (e.g., Pirotta et al., 2013a; Achiam et al., 2017). A related problem studied in RL is the one of safety, where the algorithm is forced to satisfy a set of constraints, potentially not directly connected with the performance of a policy (e.g., Altman, 1999; Berkenkamp et al., 2017; Chow et al., 2018).

In the online setting, which is the focus of this paper, the learning agent needs to trade-off exploration and exploitation while interacting with the MDP. Opposite to offline learning, the agent has direct control over exploration. Exploration means that the agent is willing to give up rewards for policies improving his knowledge of the environment. Therefore, there is no guarantee on the performance of policies generated by the algorithm, especially in the initial phase where the uncertainty about the MDP is maximal and the algorithm has to explore multiple options (almost randomly). To increase the application of exploration algorithm, it is thus important that the policies selected by the algorithm are (cumulatively) guaranteed to perform as well as the baseline by making exploration more *conservative*. This setting has been studied in multiarmed bandits (Wu et al., 2016), contextual linear bandits (Kazerouni et al., 2017), and stochastic combinatorial semi-bandits (Katariya et al., 2019). These papers formulate the problem using a constraint defined based on the performance of the baseline policy (mean of the baseline arm in the multi-armed bandit case), and modify the corresponding UCB-type algorithm (Auer et al., 2002) to satisfy this constraint. Another algorithm in the online setting is by (Mansour et al., 2015) that balances exploration and exploitation such that the actions taken are compatible with the agent's (customer's) incentive formulated as a Bayesian prior.

While the conservative exploration problem is well-understood in bandits, little is known about this setting to RL, where the actions taken by the learning agent affect the system state. This dynamic component makes the definition of the conservative condition much less obvious in RL. While in the bandit case it is sufficient to look at (an estimate of) the immediate reward to perform a conservative decision, in MDPs acting greedily may not be sufficient since an action can be "safe" in a single step but lead to a potentially dangerous state space where it will not be possible to satisfy the conservative constraint. Moreover, after t steps, the action followed by the learning agent may lead to a state that is possibly different from the one observed by following the baseline. This dynamical aspect is not captured by the bandit problem and should be explicitly taken into account by the learning agent in order to perform a meaningful decision. This, together with the problem of counterfactual reasoning in an unknown MDP, make the conservative exploration problem is much more difficult (and interesting) in RL than in bandits.

This paper aims to provide the first analysis of conservative exploration in RL. In Sec. 3 we explain the design choices that lead to the definition of the conservative condition for RL (both in average reward and finite horizon settings), and discuss all the issues introduced by the dynamical nature of the problem. Then, we provide the first algorithm for efficient conservative exploration in average reward and analyze its regret guarantees. The variant for finite-horizon problems is postponed to the appendix. We conclude the paper with synthetic experiments.

2 Preliminaries

We consider a Markov Decision Process (Puterman, 1994, Sec. 8.3) $M = (\mathcal{S}, \mathcal{A}, p, r)$ with state space $\mathcal{S}$ and action space $\mathcal{A}$. Every state-action pair (s, a) is characterized by a reward distribution with mean $r(s, a)$ and support in $[0, r_{\max}]$, and a transition distribution $p(\cdot|s, a)$ over next states. We denote by $S = |\mathcal{S}|$ and $A = |\mathcal{A}|$ the number of states and action A stationary Markov randomized policy $\pi : \mathcal{S} \to P(\mathcal{A})$ maps states to distributions over actions. The set of stationary randomized (resp. deterministic) policies is denoted by Π^{SR} (resp. Π^{SD}). Any policy $\pi \in \Pi^{\mathrm{SR}}$ has an associated *long-term average reward* (or gain) and a *bias function* defined as

$$g^\pi(s) := \lim_{T\to+\infty} \mathbb{E}_s^\pi\left[\frac{1}{T}\sum_{t=1}^T r(s_t, a_t)\right] \quad \text{and}$$

$$h^\pi(s) := C\text{-}\lim_{T\to+\infty} \mathbb{E}_s^\pi\left[\sum_{t=1}^T \big(r(s_t, a_t) - g^\pi(s_t)\big)\right],$$

where $\mathbb{E}_s^\pi$ denotes the expectation over trajectories generated starting from $s_1 = s$ with $a_t \sim \pi(s_t)$. The bias $h^\pi(s)$ measures the expected total difference between the reward and the stationary reward in *Cesaro-limit* (denoted by C-lim). We denote by $sp(h^\pi) := \max_s h^\pi(s) - \min_s h^\pi(s)$ the *span* (or range) of the bias function.

Assumption 1. *The MDP M is* ergodic.x

In ergodic MDPs, any policy $\pi \in \Pi^{\mathrm{SR}}$ has *constant* gain, i.e., $g^\pi(s) = g^\pi$ for all $s \in \mathcal{S}$. There exists a policy $\pi^\star \in \arg\max_\pi g^\pi$ for which $(g^\star, h^\star) = (g^{\pi^\star}, h^{\pi^\star})$ satisfy the *optimality equations*,

$$h^\star(s) + g^\star = Lh^\star(s) := \max_{a\in\mathcal{A}}\{r(s, a) + p(\cdot|s, a)^\mathsf{T} h^\star\},$$

where L is the *optimal* Bellman operator. We use $D = \max_{s\neq s'} \min_{\pi\in\Pi^{\mathrm{SD}}} \mathbb{E}[\tau_\pi(s'|s)]$ to denote the diameter of M, where $\tau_\pi(s'|s)$ is the hitting time of s' starting from s. We introduce the "worst-case" diameter

$$\Upsilon = \max_{s\neq s'} \max_{\pi\in\Pi^{\mathrm{SD}}} \ \mathbb{E}\left[\tau_\pi(s'|s)\right], \tag{1}$$

which defines the worst-case time it takes for any policy π to move from any state s to s'. Asm. 1 guarantees that $D \leq \Upsilon < \infty$.

Exploration in RL. Let $M^\star$ be the true *unknown* MDP. We consider the learning problem where $\mathcal{S}$, $\mathcal{A}$ and $r_{\max}$ are *known*, while rewards r and transition probabilities p are *unknown* and need to be estimated online. We evaluate the performance of a learning algorithm $\mathfrak{A}$ after T time steps by its cumulative *regret*

$$R(\mathfrak{A}, T) = Tg^\star - \sum_{t=1}^T r_t(s_t, a_t). \tag{2}$$

The exploration-exploitation dilemma is a well-known problem in RL and (nearly optimal) solutions have been proposed in the literature both base on optimism-in-the-face-of-uncertainty (OFU, e.g., Jaksch et al., 2010; Bartlett and Tewari, 2009; Fruit et al., 2018a) and Thompson sampling (TS, e.g., Gopalan and Mannor, 2015; Osband and Roy, 2016). Refer to (Lazaric et al., 2019) for more details.

3 Conservative Exploration in RL

In conservative exploration, a learning agent is expected to perform as well as the optimal policy over time (i.e., regret minimization) under the *constraint* that at no point in time its performance is significantly worse than a known baseline policy $\pi_b \in \Pi^{\mathrm{SR}}$. This problem has been studied in the bandit literature (Wu et al., 2016; Kazerouni et al., 2017), where the conservative constraint compares the cumulative *expected* reward obtained by the actions $a_1, a_2, \ldots, a_t$ selected by the algorithm to the one of the baseline action a_b,

$$\forall t > 0, \quad \sum_{i=1}^{t} r(a_i) \geq (1-\alpha)\, t\, r(a_b), \tag{3}$$

where $r(a)$ is the expected reward of action a. At any time t, conservative exploration algorithms first query a standard regret minimization algorithm (e.g., UCB) and decide whether to play the proposed action $\widetilde{a}_t$ or the baseline a_b based on the accumulated budget (i.e., past rewards) and whether the estimated performance of $\widetilde{a}_t$ is sufficient to guarantee that the conservative constraint is satisfied at $t+1$ after $\widetilde{a}_t$ is executed. While (3) effectively formalizes the objective of constraining an algorithm to never perform much worse than the baseline, in RL it is less obvious how to define such constraint. In the following we review three possible directions, we point out their limitations, and we finally propose a conservative condition for RL for which we derive an algorithm in the next section.

Gain-based condition. Instead of actions, RL exploration algorithms (e.g., UCRL2), first select a policy and then execute the corresponding actions. As a result, a direct way to obtain a conservative condition is to translate the reward of each action in (3) to the *gain* associated to the policies selected over time, i.e.,

$$\forall t > 0, \quad \sum_{i=1}^{t} g^{\pi_i} \geq (1-\alpha)\, t\, g^{\pi_b}. \tag{4}$$

The main drawback of this formulation is that the gain g^{π_i} is the expected *asymptotic* average reward of a policy and it may be very far from the *actual* reward accumulated while executing π_i in the specific state s_i achieved at time i. The same reasoning applies to the baseline policy, whose cumulative reward up to time t may significantly different from t times its gain. As a result, an algorithm that is conservative in the sense of (4) may still perform quite poorly in practice depending on t, the initial state, and the actual trajectories observed over time.

Reward-based condition. In order to address the concerns about the gain-based condition, we could define the stronger condition

$$\forall t > 0, \quad \sum_{i=1}^{t} r_i \geq (1-\alpha) \sum_{i=1}^{t} r_i^b, \tag{5}$$

where r_i is the sequence of rewards obtained while executing the algorithm and r_i^b is the reward obtained by the baseline. While this condition may be desirable in principle (the learning algorithm never performs worse than baseline), it is impossible to achieve. In fact, even if the optimal policy $\pi^\star$ is executed for all t steps, the condition may still be violated because of an unlucky realization of transitions and rewards. If we wanted to accounting for the effect of randomness, we would need to introduce an additional slack of order $O(\sqrt{t})$ (i.e., the cumulative deviation due to the randomness in the environment), which would make the condition looser and looser over time.

Condition in expectation. The previous remarks could be solved by taking the expectation of both sides

$$\begin{aligned}\forall t > 0,\ \mathbb{E}_{\mathfrak{A}}&\left[\sum_{i=1}^{t} r_i(s_i,a_i)\middle| s_1 = s\right] \\ &\geq (1-\alpha)\mathbb{E}\left[\sum_{i=1}^{t} r_i(s_i,a_i)\middle| s_1 = s, \pi_b\right],\end{aligned} \tag{6}$$

where $\mathbb{E}_{\mathfrak{A}}$ denotes the expectation w.r.t. the trajectory of states and actions generated by the learning algorithm $\mathfrak{A}$, while the RHS is simply the expected reward obtain by running the baseline for t steps. Condition (6) effectively captures the nature of the RL problem w.r.t. the bandit case. In fact, after t steps, the actions followed by the learning algorithm may lead to a state that is possibly very different from the one we would have reached by playing only the baseline policy from the beginning. This deviation in the state dynamics needs to be taken into account when deciding if an exploratory policy is safe to play in the future. In the bandit case, selecting the baseline action contributes to *build a conservative budget* that can be *spent* to play explorative actions later on (i.e., by selecting a_b, the LHS of (3) in increased by $r(a_b)$, while only a fraction $1-\alpha$ is added to the RHS, thus increasing the margin that may allow playing alternative actions later). In the RL case, selecting policy π_b at time t may not immediately contribute to increasing the conservative budget. In fact, the state s_t where π_b is applied may significantly differ from the state that π_b *would have achieved* had we selected it from the beginning. As a result, a conservative RL algorithm should be extra-cautious when selecting policies different from π_b since their execution may lead to unfavorable states, where it is difficult to recover good performance, even when selecting the baseline policy.

While this may seem a reasonable requirement, unfortunately it is impossible to build an empirical estimate of (6) that a conservative exploration algorithm could use to guide the choice of policies to execute. In fact, the LHS *averages* the performance of the algorithm over multiple executions, while in practice we have only access to a single realization of the algorithm's process. This prevents from constructing accurate estimates of such expectation directly from the data observed up to time t. A possible approach would be to construct an estimate of the MDP and use it to *replay* the algorithm itself for t steps. Beside prohibitive computational complexity, the resulting estimate of the expected cumulative reward of $\mathfrak{A}$ would suffer from an error that increases with t, thus making it a poor proxy for (6).[1]

Condition with conditional expectation. Let t be a generic time and $\mu_t = (\pi_1, \pi_2, \dots, \pi_t)$, the non-stationary policy executed up to t. We require the algorithm to satisfy the following *conditional* conservative condition

$$\begin{aligned}\forall t > 0,\ \mathbb{E}\left[\sum_{i=1}^{t} r_i(s_i, a_i) | s_1 = s, \mu_t\right] \\ \geq (1-\alpha)\mathbb{E}\left[\sum_{i=1}^{t} r_i(s_i, a_i) | s_1 = s, \pi_b\right].\end{aligned} \tag{7}$$

where the expectations are taken w.r.t. the trajectories generated by a *fixed* non-stationary policy μ_t (i.e., we ignore how rewards affect μ_t). Notice that this condition is now stochastic, as μ_t itself is a random variable and thus we require to satisfy (7) with *high probability*. This formulation can be seen as relying on a *pseudo*-performance evaluation of the algorithm instead of the actual expectation as in (6)[2] and it is similar to (3), which takes the expected performance of each of the (random) actions, thus ignoring their correlation with the rewards. This formulation has several advantages w.r.t. the conditions proposed above: **1)** it considers the sum of rewards rather than the gain as (4), thus capturing the dynamical nature of RL, **2)** it contains expected values, so as to avoid penalizing the algorithm by unlucky noisy realizations as (5), **3)** as shown in the next section, it can be verified using the samples observed by the algorithm unlike (6).

[1]More precisely, let $\widehat{M}_t$ be an estimate of $M^\star$ and ϵ_t be the largest error in estimating its dynamics at time t. Estimating the expected cumulative reward by running (an infinite number of) simulations of $\mathfrak{A}$ in $\widehat{M}_t$ would suffer from an error scaling as $t\epsilon_t$. For any regret minimization algorithm, ϵ_t cannot decrease linearly with t and thus the estimation of $\mathbb{E}_{\mathfrak{A}}$ would have an error increasing with t.

[2]We use *pseudo*-performance to stress the link the *pseudo*-regret formulations used in bandit (e.g., Auer et al., 2002)

The finite-horizon case. We conclude the section, by reformulating (7) in the finite-horizon case. In this setting, the learning agent interacts with the environment in episodes of fixed length H. Let $\overline{s}$ be the initial state, π_j be the policy proposed at episode j and let $t = (k-1)H + 1$ be beginning of the k-th episode. Then μ_t is a sequence of policies π_j, each executed for H steps. In this case, condition (7) can be conveniently written as

$$\begin{aligned}(1-\alpha)kV_1^{\pi_b}(\overline{s}) \leq \mathbb{E}\left[\sum_{i=1}^{t} r_i(s_i, a_i) | s_1 = s, \mu_t\right] \\ = \sum_{j=1}^{k} \mathbb{E}\left[\sum_{i=1}^{H} r_i^j(s_i^j, a_i^j) | s_1 = s, \pi_j\right] = \sum_{j=1}^{k} V_1^{\pi_j}(\overline{s})\end{aligned} \tag{8}$$

where V_1^π is the H step value function of π at the first stage. In this formulation, the conservative condition has a direct interpretation, as it directly mimics the bandit case (3). In fact, the performance of the algorithm up to episode k is simply measured by the sum of the value functions of the policies executed over time (each for H steps) and it is compared to the value function of the baseline itself. Note that this definition is compatible with the regret: $R_{\mathrm{FH}}(\mathfrak{A}, K) = \sum_{k=1}^{K} V^\star(\overline{s}) - V^{\pi_k}(\overline{s})$. Indeed, the regret defined in expectation w.r.t. the stochasticity of the model but not w.r.t. the algorithm, there is no expectation w.r.t. the possible sequence of policies generated by $\mathfrak{A}$.

4 Conservative UCRL

In this section, we introduce *conservative upper-confidence bound for reinforcement learning* (CUCRL2), an efficient algorithm for exploration-exploitation in average reward that both minimize the regret (2) and satisfy condition (7).

CUCRL2 builds on UCRL2 in order to perform efficient *conservative* exploration. At each episode k, CUCRL2 builds a bounded parameter MDP $\mathcal{M}_k = \{M = (\mathcal{S}, \mathcal{A}, r, p), r(s,a) \in B_r^k(s,a), p(\cdot|s,a) \in B_p^k(s,a)$, where $B_r^k(s,a) \in [0, r_{\max}]$ and $B_p^k(s,a) \in \Delta_S$ are high-probability confidence intervals on the rewards and transition probabilities such that $M^\star \in \mathcal{M}_k$ w.h.p. and Δ_S is the S-dimensional simplex. This confidence intervals can be built using Hoeffding or empirical Bernstein inequalities by using the samples available at episode k (e.g., Jaksch et al., 2010; Fruit et al., 2018b). CUCRL2 computes an optimistic policy $\widetilde{\pi}_k$ in the same way as UCRL2: $(\widetilde{M}_k, \widetilde{\pi}_k) \in \arg\max_{M \in \mathcal{M}_k, \pi \in \Pi^{\mathrm{SD}}} \{g^\pi(M)\}$. This problem can be solved using EVI (see Fig. 3 in appendix) on the optimistic optimal Bellman operator $\mathcal{L}_k^+$ of $\mathcal{M}_k$ (Jaksch et al., 2010).[3] Then, it needs to decide whether policy

[3]$\mathcal{L}_k^+ v(s) = \max_a\{\max_{r \in B_k^r(s,a)}\{r\} + \max_{p \in B_k^p(s,a)} p^\mathsf{T} v\}$.

Input: $\pi_b \in \Pi^{\mathrm{SR}}$, $\delta \in (0,1)$, $r_{\max}$, $\mathcal{S}$, $\mathcal{A}$, $\alpha \in (0,1)$
For episodes $k = 1, 2, ...$ **do**

1. Set $t_k = t$ and episode counters $\nu_k(s,a) = 0$.
2. Compute estimates $\widehat{p}_k(s'|s,a)$, $\widehat{r}_k(s,a)$ and a confidence set $\mathcal{M}_k$.
3. Compute an $r_{\max}/\sqrt{t_k}$-approximation $\widetilde{\pi}_k$ of the optimistic planning problem $\max_{M\in\mathcal{M}_k,\pi\in\Pi^{\mathrm{SD}}}\{g^\pi(M)\}$.
4. Compute $(g_k^-, h_k^-) = \mathrm{EVI}(\mathcal{L}_k^{\widetilde{\pi}_k}, r_{\max}/\sqrt{t_k})$, see Eq. 11.
5. **if** Eq. 15 is true **then** $\pi_k = \widetilde{\pi}_k$ **else** $\pi_k = \pi_b$
6. Sample action $a_t \sim \pi_k(\cdot|s_t)$.
7. **While** $\nu_k(s_t,a_t) \le N_k^+(s_t,a_t) \wedge t \le t_k + T_{k-1}$ **do**
   (a) Execute a_t, obtain reward r_t, and observe s_{t+1}.
   (b) Set $\nu_k(s_t,a_t) = \nu_k(s_t,a_t) + 1$.
   (c) Sample action $a_{t+1} \sim \pi_k(\cdot|s_{t+1})$ and set $t = t+1$.
8. Set $N_{k+1}(s,a) = N_k(s,a) + \nu_k(s,a)$, $\Lambda_k = \Lambda_{k-1} \cup \{k\} \cdot \mathbb{1}_{(Eq.\ 15)}$ and $\Lambda_k^c = \Lambda_{k-1}^c \cup \{k\} \cdot \mathbb{1}_{(\neg Eq.\ 15)}$

Figure 1: CUCRL2 algorithm.

π_k is "safe" to play by checking a conservative condition $f_c(\mathcal{H}_k)$ (see Sec. 4.1) where $\mathcal{H}_k$ contains all the information (samples and chosen policies) available at the beginning of episode k, including the optimistic policy π_k. If $f_c(\mathcal{H}_k) \ge 0$, the UCRL2 policy $\widetilde{\pi}_k$ is "safe" to play and CUCRL2 plays $\pi_k = \widetilde{\pi}_k$ until the end of the episode. Otherwise, CUCRL2 executes the baseline π_b, i.e., $\pi_k = \pi_b$. We denote by Λ_k the set of episodes (k included) where UCRL2 executed an optimistic policy and by $\Lambda_k^c = \{1, \dots, k\} \setminus \Lambda_k$ its complement. Formally, if $f_c(\mathcal{H}_k) \ge 0$ we set $\Lambda_k = \Lambda_{k-1} \cup \{k\}$ else $\Lambda_k = \Lambda_{k-1}$. The pseudocode of CUCRL2 is reported in Fig. 1.

Note that, contrary to what happens in conservative (linear) bandits, the statistics of the algorithm are updated continuously, i.e., using also the samples collected by running the baseline policy. This is possible since UCRL2 is a model-based algorithm and any off-policy sample can be used to update the estimates of the model. To have a better estimate of the conservative condition, it is possible to use the model available at episode k to re-evaluate the policies $(\pi_l)_{l<k}$ at previuous episodes (change line 3 in Fig. 1). This will improve the empirical performance of CUCRL2 but breaks the regret analysis.

4.1 Algorithmic Conservative Condition

We now derive a *checkable* conservative condition that can be incorporated in the UCRL2 structure illustrated in the previous section. In the bandit setting, it is relatively straightforward to turn (3) into a condition that can be checked at any time t using estimates and confidence intervals build from the data collected so far. On the other hand, while condition (7) effectively formalizes the requirement that the learning algorithm should constantly perform almost as well as the baseline policy, we need to consider the specific RL structure to obtain a condition that can be verified *during the execution* of the algorithm itself. In order to simplify the derivation, we rely on the following assumption.

Assumption 2. *The gain and bias function (g^{π_b}, h^{π_b}) of the baseline policy are known.*

As explained in (Kazerouni et al., 2017), this is a reasonable assumption since the baseline policy is assumed to be the policy currently executed by the company and for which historical data are available. We will mention how to relax this assumption in Sec. 4.1.

We follow two main steps in deriving a checkable condition. **1)** We need to estimate the cumulative reward obtained by each of the policies played by the learning algorithm directly from the samples observed so far. We do this by relating the cumulative reward to the gain and bias of each policy and then building their estimates. **2)** It is necessary to evaluate whether the policy proposed by UCRL2 is safe to play w.r.t. the conservative condition, before actually executing it. While this is simple in bandit, as each action is executed for only one step. In RL, policies cannot be switched at each step and need to be played for a *whole* episode. Nonetheless, the length of a UCRL2 episode is not known in advance and this requires predicting for how long the explorative policy could be executed in order to check its performance.

Step 1: Estimating the conditional conservative condition from data. In order to evaluate (7) from data, one may be tempted to first replace the sum of rewards obtained by each policy π_j in μ_t on the lhs side by its gain g^{π_j}, similar to the gain-based condition in (4). Indeed, under Asm. 1 any stationary policy π receives *asymptotically* an expected reward g^π at each step. Unfortunately, in our case $\mathbb{E}\left[\sum_{i=1}^t r_i \middle| \mu_t\right] \neq \sum_{j=1}^k T_j g^{\pi_j}$. In fact, when evaluating a policy for a finite number of steps, we need to account for the time required to reach the steady regime (i.e., mixing time) and, as such, the influence of the state at which the policy is started. The notion of reward collected during the transient regime is captured by the bias function.[4] In particular, for any stationary (unichain) policy $\pi \in \Pi^{\mathrm{SR}}$ with gain g^π and gain function h^π executed for t steps, we have that:

$$\mathbb{E}\left[\sum_{i=1}^t r_i \middle| s_1 = s, \pi\right] = t\, g^\pi + h^\pi(s) - P_\pi^t(\cdot|s)^\mathsf{T} h^\pi. \quad (9)$$

[4] Puterman (1994, Sec. 8.2.1) refers to the gain as "stationary" reward while to the bias as "transient" reward.

As a result, we have the bounds

$$t\, g^\pi - sp(h^\pi) \leq \mathbb{E}\left[\sum_{i=1}^{t} r_i \big| s_1 = s, \pi\right] \leq t\, g^\pi + sp(h^\pi).$$

Leveraging prior knowledge of the gain and bias of the baseline, we can use the second inequality to directly upper bound the baseline performance as

$$\mathbb{E}\left[\sum_{i=1}^{t} r_i \big| s_1 = s, \pi_b,\right] \leq sp(h^{\pi_b}) + t\, g^{\pi_b}. \tag{10}$$

On the other hand, for a generic policy π, the gain and bias cannot be directly computed since $M^\star$ is unknown. To estimate the cumulative reward of the algorithm we resort to the estimate of the true MDP build by UCRL2 to construct a pessimistic estimate of the cumulative reward for any policy π_j (i.e., to perform counterfactual reasoning).

Given a policy π and the bounded-parameter MDP $\mathcal{M}_k$, we are intersted in finding $\underline{g}^\pi$ such that: $\underline{g}^\pi := \min_{M \in \mathcal{M}_k}\{g^\pi(M)\}$. Define the Bellman operator $\mathcal{L}_k^\pi$ associated to $\mathcal{M}_k$ as: $\forall v \in \mathbb{R}^S, \forall s \in \mathcal{S}$

$$\mathcal{L}_k^\pi v(s) := \min_{r \in B_r^k(s,a)} r + \min_{p \in B_p^k(s,a)} \{p^\mathsf{T} v\} \tag{11}$$

Then, there exists $(\underline{g}^\pi, \underline{h}^\pi)$ such that, $\forall s \in \mathcal{S}$, $\underline{g}^\pi e + \underline{h}^\pi = \mathcal{L}_k^\pi \underline{h}^\pi$ where $e = (1, \dots, 1)$ (see Lem. *5.1* in App. A). Similarly to what is done by UCRL2, we can use EVI with $\mathcal{L}_k^\pi$ to build an ϵ_k-approximate solution of the Bellman equations. Let $(g_n, v_n) = \mathrm{EVI}(\mathcal{L}_k^\pi, \varepsilon_k)$, then $g_n - \varepsilon_k \leq \underline{g}^\pi \leq g^\pi(M^\star)$. The values computed by the pessimistic policy evaluation can be then used to bound the cumulative reward of any stationary policy.

Lemma 1. *Consider a bounded parameter MDP $\mathcal{M}$ such that $M^\star \in \mathcal{M}$ w.h.p., a policy π and let $(g_n, v_n) = \mathrm{EVI}(\mathcal{L}^\pi, \varepsilon)$. Then, under Asm. 1 for any state $s \in \mathcal{S}$:*

$$\mathbb{E}\left[\sum_{i=1}^{t} r_i | s_1 = s, \pi\right] \geq t(g_n - \varepsilon) - sp(v_n).$$

Step 2: Test safety of optimistic policy. Let t_k be the time when episode k starts. Policies $\pi_1, \dots, \pi_{k-1}$ have been executed until $t_k - 1$ and UCRL2 computed an optimistic policy $\widetilde{\pi}_k$. In order to guarantee that (7) is verified the algorithm needs to anticipate how well $\widetilde{\pi}_k$ may perform if executed for the next episode. For any policy $(\pi_j)_{j<k} \cup \{\widetilde{\pi}_k\}$, we first compute $(g_j^-, h_j^-) = \mathrm{EVI}(\mathcal{L}_j^{\pi_j}, \varepsilon_j)$.[5] If $\pi_j = \pi^b$ (i.e., the baseline was executed at episode j), we let $(g_j^-, h_j^-) = (g^{\pi_b}, h^{\pi_b})$ and $\varepsilon_j = 0$. Then

$$\begin{aligned}
&\mathbb{E}\left[\sum_{i=1}^{t} r_i \big| s_1 = s, \mu_t\right] \\
&= \sum_{j=1}^{k} \sum_{y \in \mathcal{S}} \mathbb{P}\left(s_{t_j} = y \big| s, \mu_t\right) \cdot \mathbb{E}\left[\sum_{i=1}^{T_j} r_i \mid y, \pi_j\right] \quad (12) \\
&\geq \sum_{j=1}^{k} T_j (g_j^- - \varepsilon_j) - sp(h_j^-)
\end{aligned}$$

where t_j is time at which episode j started, $\mathbb{P}\left(s_{t_j} = y \big| s_1 = s, \mu_t\right)$ is the probability of reaching state y after t_j steps starting from state s following policy μ_t. The inequality follows from Lem. 1. By lower bounding the LHS of (7) by (12) and upper bounding the RHS by (10), the conservative condition becomes:

$$\begin{aligned}
\sum_{j=1}^{k-1} \Big(T_j (g_j^- - \varepsilon_j - g^{\pi_b}) - sp(h_j^-)\Big) - sp(h^{\pi_b}) \quad (13) \\
+ T_k (g_k^- - \varepsilon_k - (1-\alpha) g^{\pi_b}) \geq 0
\end{aligned}$$

Note that the algorithm should check this condition at the beginning of episode k in order to understand if the policy $\widetilde{\pi}_k$ is safe or if it should resort to playing policy π_b. In many OFU algorithms, including UCRL2, the length of episode k (i.e., T_k) is not known at the beginning of the episode. As a consequence, condition (13) is not directly computable. To overcome this limitation, we consider the dynamic episode condition introduced by (Ouyang et al., 2017). This stopping condition provides an upper-bound on the length of each episode as $T_k \leq T_{k-1} + 1$, without affecting the regret bound of UCRL2 (up to constants). This condition can be used to further lower-bound the last term in (13) by

$$\begin{aligned}
&T_k (g_k^- - \varepsilon_k - (1-\alpha) g^{\pi_b}) \quad (14) \\
&\geq (T_{k-1} + 1)(g_k^- - \varepsilon_k - (1-\alpha) g^{\pi_b}) \cdot \mathbb{1}_{\left((1-\alpha) g^{\pi_b} \geq g_k^- - \varepsilon_k\right)}.
\end{aligned}$$

Plugging this lower bound into (13) gives the final conservative condition

$$\begin{aligned}
&\sum_{j=1}^{k-1} \Big(T_j (g_j^- - \varepsilon_j - g^{\pi_b}) - sp(h_j^-)\Big) - sp(h^{\pi_b}) + \\
&(T_{k-1} + 1)(g_k^- - \varepsilon_k - (1-\alpha) g^{\pi_b}) \cdot \mathbb{1}_{\left((1-\alpha) g^{\pi_b} \geq g_k^- - \varepsilon_k\right)} \\
&\geq 0, \quad (15)
\end{aligned}$$

tested by CUCRL2 at the beginning of each episode. *Unknown* (g^{π_b}, h^{π_b}). If the gain and bias of the baseline are unknown, we can use EVI on $\mathcal{L}_l^{+,\pi_b}$ (Eq. 11 with max instead of min) to compute an optimistic

[5] The subscript j in the operator $\mathcal{L}_j^{\pi_j}$ denotes the fact that it is computed using the samples observed up to t_j. For each episode, we need to compute the estimate only for the new UCRL2 policy. In order to have a tighter estimate of the conservative condition it possible to recompute the gain and bias of the past policies at every episode (or periodically) by using all the available samples (i.e., using $\mathcal{L}_k^\pi$). However, this will break the current regret proof.

estimate of the cumulative reward of the baseline up to time $t_l + T_{l-1} + 1$. While this account for the RHS of Eq. 7, we simply define $(g_l^-, h_l^-) = \mathrm{EVI}(\mathcal{L}_l^{\pi_b}, \epsilon_l)$ for every episode $l \in \Lambda_{k-1}^c$ to compute a lower bound to the cumulative reward obtained by the algorithm by playing the baseline in episode before k. Clearly, this approach is very pessimistic and it may be possible to design better strategies for this case.

The finite-horizon case. We conclude this section with a remark on the finite horizon case. This case is much simpler and resemble the bandit setting. We can directly build a lower bound $\underline{v}_{l,1}$ to the value function $V_1^{\pi_l}$ by using the model estimate and its uncertainty at episode l. This estimate can be computed via extended backward induction –see (Azar et al., 2017, Alg. 2)– simply subtracting the exploration bonus, see Lem. 10 in App. C. The same approach can be used to construct and optimistic and pessimistic estimate of $V_1^{\pi_b}$ when it is unknown. This values can be directly plug in (8) to define a checkable condition for the algorithm.

4.2 Regret Guarantees

We start providing an upper-bound to the regret of CUCRL2 showing the dependence on UCRL2 and on the baseline π_b. Since the set Λ_k is updated at the end of the episode, we denote by $\Lambda_T = \Lambda_{k_T} \cup \{k_T\} \cdot \mathbb{1}_{(Eq.\ (15))}$ the set containing all the episodes where CUCRL2 played an optimistic policy. The set Λ_T^c is its complement.

Lemma 2. *Under Asm. 1 and 2, for any T and any conservative level α, there exists a numerical constant $\beta > 0$ such that the regret of* CUCRL2 *is upper-bounded as*

$$R(\mathrm{CUCRL2}, T) \leq \beta \cdot \Bigg(R_{\mathrm{UCRL2}}(T|\Lambda_T) + (g^\star - g^{\pi_b}) \sum_{l \in \Lambda_T^c} T_l + sp(h^{\pi_b}) \sqrt{SAT \ln(T/\delta)} \Bigg),$$

and the conservative condition (7) *is met at every step $t = 1, \ldots, T$ with probability at least $1 - \frac{2\delta}{5}$.*[6]

$R_{\mathrm{UCRL2}}(T|\Lambda_T)$ denotes the regret of UCRL2 over an horizon T conditioned on the fact that the UCRL2 policy is executed only at episodes $i \in \Lambda_T$. During the other episodes, the internal statistics of UCRL2 are updated using the samples collected by the baseline policy π_b. This does not pose any major technical challenge and, as shown in App. B, the UCRL2 regret can be bounded as follows.

[6]The probability refers to both events: the regret bound and the conservative condition.

Lemma 3 ((Jaksch et al., 2010))**.** *Let $L_T = \ln\left(\frac{5T}{\delta}\right)$, for any T, there exists a numerical constant $\beta > 0$ such that, with probability at least $1 - \frac{2\delta}{5}$,*

$$R_{\mathrm{UCRL2}}(T|\Lambda_T) \leq \beta DS\sqrt{ATL_T} + \beta DS^2 AL_T$$

The second term in Lem. 2 represents the regret incurred by the algorithm when playing the baseline policy π_b. The following lemma shows that the total time spent executing conservative actions is sublinear in time (see Lem. 8 in App. B for details).

Lemma 4. *For any $T > 0$ and any conservative level α, with probability at least $1 - \frac{2\delta}{5}$, the total number of play of conservative actions is bounded by:*

$$\sum_{l \in \Lambda_T^c} T_l \leq 2\sqrt{SAT\ln(T)} + \frac{112 SAL_T}{(\alpha g^{\pi_b})^2}(1 + S(D + \Upsilon)^2) + \frac{16\sqrt{TL_T}}{\alpha g^{\pi_b}} \Big[(D + \Upsilon)\sqrt{SA} + r_{\max} + \sqrt{SA} sp(h^{\pi_b}) \Big]$$

where $L_T = \ln\left(\frac{5SAT}{\delta}\right)$ and $\Upsilon < \infty$ as in Eq. 1

Proof. Let τ be the last episode played conservatively: $\tau = \sup\{k > 0 : k \in \Lambda_k^c\}$. This means that at the beginning of episode τ the conservative condition was not verified. By rearranging the terms in Eq. 15 and using simple bounds, we can write that:

$$\Delta_t + 4k_T\left(sp(g^{\pi_b}) + \Upsilon + (1-\alpha) r_{\max}\right) \geq \alpha \sum_{l=1}^{\tau-1} T_l g^{\pi_b}$$

where $\Delta_\tau := \sum_{l \in \Lambda_{t-1}} T_l(\widetilde{g}_l - \underline{g}_l)$ and $(\widetilde{g}_l, \underline{g}_l)$ are optimistic and pessimistic gain of policy $\pi_l = \widetilde{\pi}_l$. Note that both satisfies the Bellman equation: $\widetilde{g}_l + \widetilde{h}_l = \mathcal{L}_l^+ \widetilde{h}_l$ (see footnote 3) and $\underline{g}_l + \underline{h}_l = \mathcal{L}_l^{\pi_l} \underline{h}_l$ (see Eq. 11). At this point, the important terms in upper-bounding Δ_τ are similar to the one analysed in UCRL2. In particular, we have a term depending on the confidence intervals $\Delta_{ci}^{l,t} := 2\beta_r^l(s_t, a_t) + \beta_p^l(s_t, a_t)\big(sp(\widetilde{h}_l) + sp(\underline{h}^{\pi_l})\big)$ and one depending on the transitions $\Delta_p^{l,t} := p^\star(\cdot|s_t, a_t)^\mathsf{T}\left(\widetilde{h}_l + \underline{h}^{\pi_l}\right) - (\widetilde{h}_l(s_{t+1}) - \underline{h}^{\pi_l}(s_{t+1}))$. Let $X = \sum_{l=1}^{\tau-1} T_l$. By using the definition of the confidence intervals, it is easy to show that $\sum_{l \in \Lambda_{\tau-1}} \sum_{t=t_l}^{t_{l+1}-1} \Delta_{ci}^{l,t} \lesssim \sqrt{SATX} + (D + \Upsilon)\sqrt{S^2AX}$. Define the σ-algebra based on past history at t: $\mathcal{F}_t = \sigma(s_1, a_1, r_1, \ldots, s_t, a_t, r_t, s_{t+1})$. The sequence $(\Delta_p^{l,t}, \mathcal{F}_t)_{l,t}$ is an MDS. Thus, using Azuma inequality we have that $\sum_{l \in \Lambda_{\tau-1}} \sum_{t=t_l}^{t_{l+1}-1} \Delta_{ci}^{l,t} \lesssim (D + \Upsilon)\sqrt{T}$. Putting everything together we have a quadratic form in X and solving it we can write that $\alpha g^{\pi_b} X \lesssim b_T + S^2 AL_T^\delta (D + \Upsilon)^2/(\alpha g^{\pi_b})$ where $b_T = \widetilde{O}((D + \Upsilon)\sqrt{SAT})$ (see App. B). The result follows noticing that $\sum_{l \in \Lambda_\tau^c} T_l \leq \sum_{l=1}^{\tau-1} T_l + T_\tau$. $\square$

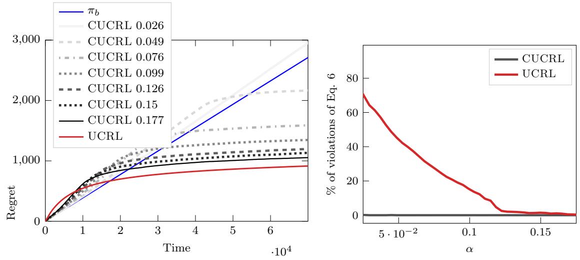

Figure 2: Inventory control problem.

Combining the results of Lem. 3 and Lem. 4 into Lem. 2 leads to an overall regret of order $\widetilde{O}(\sqrt{T})$, which matches the regret of UCRL2. This shows that CUCRL2 is able to satisfy the conservative condition without compromising the learning performance. Nonetheless, the bound in Lem. 4 shows how conservative exploration is more challenging in RL compared to the bandit setting. While the dependency on the conservative level α is the same, the number of steps the baseline policy is executed can be as large as $\widetilde{O}(\sqrt{T})$ instead of constant as in CUCB (Wu et al., 2016). Furthermore, Lem. 2 relies on an ergodicity assumption instead of the much milder communicating assumption needed by UCRL2 to satisfy Lem. 3. Asm. 1 translates into the bound through the "worst-case" diameter Υ, which in general is much larger than the diameter D. This dependency is due to the need of computing a lower bound to the reward accumulated by policies π_j in the past (see Lem. 1). In fact, UCRL2 only needs to compute *upper bounds* on the gain and the value function returned by EVI by applying the optimistic Bellman operator $\mathcal{L}_k^+$ has span bounded by the diameter D. This is no longer the case for computing pessimistic estimates of the value of a policy. Whether Asm. 1 and the worst-case diameter Υ are the unavoidable price to pay for conservative exploration in infinite horizon RL remains as an open question.

The finite-horizon case. App. C shows how to modify UCB-VI (Azar et al., 2017) to satisfy the conservative condition in Eq. 8. In this setting, it is possible to show (see Prop. 1 in appendix) that the number of conservative episodes is simply logarithmic in $T = KH$. Formally, $|\Lambda_T^c| = O(H^5 S^2 A \ln(T/\delta)/(\alpha r_b(\underline{\Delta}_b + \alpha r_b))$ where $0 < r_b \leq r(s,a)$, for all (s,a), and $\underline{\Delta}_b = \min_s\{V_1^\star(s) - V_1^{\pi_b}(s)\}$ is the optimality gap. This problem dependent terms resemble the one in the bandit analysis. The regret of conservative UCB-VI is thus dominated by the UCB-VI term $\widetilde{O}(H\sqrt{SAT})$.

5 Experiments

In this section, we report results in the inventory control problem to illustrate the performance of CUCRL2 compared to unconstrained UCRL2 and how it varies with the conservative level. See App. D for additional experiments for both average reward and finite horizon. In order to have a better estimate of the budget, we re-evaluate past policies at each episode. We start considering the *stochastic inventory control* problem (Puterman, 1994, Sec. 3.2.1) with capacity $M = 6$ and uniform demand. At the beginning of a month t, the manager has to decide the number of items to order in order to satisfy the random demand, taking into account the cost of ordering and maintainance of the inventory (see App. D). Since the optimal policy is a threshold policy, as baseline we consider a (σ, Σ) policy (Puterman, 1994, Sec. 3.2.1) with target stock $\Sigma = 4$ and capacity threshold $\sigma = 4$. Note that $g^\star = 0.603$ and $(g^{\pi_b}, sp(h^{\pi_b})) = (0.565, 0.651)$. We use this domain to perform an ablation study w.r.t. the conservative level α. We have taken $T = 70000$ and the results are averaged over 100 realizations.

Fig. 2(*left*) shows that the regret of CUCRL2 grows at the same speed as the one of the baseline policy π_b at the beginning (the conservative phase), because during this phase CUCRL2 is constrained to follow π_b to make sure that constraint (7) is satisfied. Clearly, the duration of this conservative phase is proportional to the conservative level α. As soon as CUCRL2 has built margin, it starts interleaving exploratory (optimistic) policies with the baseline. After this phase, CUCRL2 has learn enough about the system and has a sufficient margin to behave as UCRL2. As expected, Fig. 2(*left*) confirms that the convergence to the UCRL2 behavior happens more quickly for larger values of α, i.e., when the conservative condition is relaxed and CUCRL2 can explore more freely. On the other hand, UCRL2 converges faster since it is agnostic to the safety constraint and may explore very poor policies in the initial phase. To better understand this condition, Fig. 2(*right*) shows the percentage of time the constraint was violated in the first 15000 steps (about 20% of the overall time T). CUCRL2 always satisfies the constraint for all values of α while UCRL2 fails a significant number of times, especially when the conditions is tight (small values of α).

6 Conclusion

We presented algorithms for conservative exploration for both finite horizon and average reward problems with $O(\sqrt{T})$ regret. We have shown that the non-episodic nature of average reward problems makes the definition of the conservative condition much harder than in finite horizon problems. In both cases, we used a model-based approach to perform counterfactual reasoning required by the conservative condition. Recent papers have focused on model-free exploration in tabular settings or linear function approximation (Jin et al., 2018; Yang and Wang, 2019; Jin et al., 2019), thus a question is if it is possible for model-free algorithms to be conservative and still achieve $\widetilde{O}(\sqrt{T})$ regret.

References

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 22–31. PMLR, 2017.

Eitan Altman. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.

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

Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 263–272. PMLR, 2017.

Peter L. Bartlett and Ambuj Tewari. REGAL: A regularization based algorithm for reinforcement learning in weakly communicating MDPs. In *UAI*, pages 35–42. AUAI Press, 2009.

Felix Berkenkamp, Matteo Turchetta, Angela P. Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In *NIPS*, pages 908–918, 2017.

Dimitri P Bertsekas. *Dynamic programming and optimal control. Vol II.* Number 2. Athena scientific Belmont, MA, 1995.

Dimitri P Bertsekas. Approximate policy iteration: A survey and some new methods. *Journal of Control Theory and Applications*, 9(3):310–335, 2011.

L. Bottou, J. Peters, J. Quinonero-Candela, D. Charles, D. Chickering, E. Portugaly, D. Ray, P. Simard, and E. Snelson. Counterfactual reasoning and learning systems: The example of computational advertising. *Journal of Machine Learning Research*, 14:3207–3260, 2013.

Yinlam Chow, Ofir Nachum, Edgar A. Duéñez-Guzmán, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. In *NeurIPS*, pages 8103–8112, 2018.

Ronan Fruit, Matteo Pirotta, and Alessandro Lazaric. Near optimal exploration-exploitation in non-communicating markov decision processes. In *NIPS*, 2018a.

Ronan Fruit, Matteo Pirotta, Alessandro Lazaric, and Ronald Ortner. Efficient bias-span-constrained exploration-exploitation in reinforcement learning. In *ICML*, Proceedings of Machine Learning Research. PMLR, 2018b.

Aditya Gopalan and Shie Mannor. Thompson sampling for learning parameterized markov decision processes. In *COLT*, volume 40 of *JMLR Workshop and Conference Proceedings*, pages 861–898. JMLR.org, 2015.

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

Chi Jin, Zeyuan Allen-Zhu, Sébastien Bubeck, and Michael I. Jordan. Is q-learning provably efficient? *CoRR*, abs/1807.03765, 2018.

Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I. Jordan. Provably efficient reinforcement learning with linear function approximation. *CoRR*, abs/1907.05388, 2019.

Sham M. Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *ICML*, pages 267–274. Morgan Kaufmann, 2002.

Sumeet Katariya, Branislav Kveton, Zheng Wen, and Vamsi K. Potluru. Conservative exploration using interleaving. In *AISTATS*, volume 89 of *Proceedings of Machine Learning Research*, pages 954–963. PMLR, 2019.

Abbas Kazerouni, Mohammad Ghavamzadeh, Yasin Abbasi, and Benjamin Van Roy. Conservative contextual linear bandits. In *NIPS*, pages 3910–3919, 2017.

Romain Laroche, Paul Trichelair, and Remi Tachet des Combes. Safe policy improvement with baseline bootstrapping. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 3652–3661. PMLR, 2019.

Alessandro Lazaric, Matteo Pirotta, and Ronan Fruit. Regret minimization in infinite-horizon finite markov decision processes. Tutorial at ALT'19, 2019. URL `https://rlgammazero.github.io/`.

Yishay Mansour, Aleksandrs Slivkins, and Vasilis Syrgkanis. Bayesian incentive-compatible bandit exploration. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pages 565–582. ACM, 2015.

Minou Catharina Anselma Olde Keizer. *Condition-based maintenance for complex systems: Coordinating maintenance and logistics planning for the process industries.* PhD thesis, University of Groningen, 2016.

Ian Osband and Benjamin Van Roy. Posterior sampling for reinforcement learning without episodes. *CoRR*, abs/1608.02731, 2016.

Yi Ouyang, Mukul Gagrani, Ashutosh Nayyar, and Rahul Jain. Learning unknown markov decision processes: A thompson sampling approach. In *NIPS*, pages 1333–1342, 2017.

M. Petrik, M. Ghavamzadeh, and Y. Chow. Safe policy improvement by minimizing robust baseline regret. In *Advances in Neural Information Processing Systems*, pages 2298–2306, 2016.

Matteo Pirotta, Marcello Restelli, and Luca Bascetta. Adaptive step-size for policy gradient methods. In *NIPS*, pages 1394–1402, 2013a.

Matteo Pirotta, Marcello Restelli, Alessio Pecorino, and Daniele Calandriello. Safe policy iteration. In *ICML (3)*, volume 28 of *JMLR Workshop and Conference Proceedings*, pages 307–315. JMLR.org, 2013b.

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

Thiago D. Simão and Matthijs T. J. Spaan. Safe policy improvement with baseline bootstrapping in factored environments. In *AAAI*, pages 4967–4974. AAAI Press, 2019.

A. Swaminathan and T. Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *Proceedings of The 32nd International Conference on Machine Learning*, 2015.

P. Thomas, G. Theocharous, and M. Ghavamzadeh. High confidence off-policy evaluation. In *Proceedings of the Twenty-Ninth Conference on Artificial Intelligence*, 2015a.

P. Thomas, G. Theocharous, and M. Ghavamzadeh. High confidence policy improvement. In *Proceedings of the Thirty-Second International Conference on Machine Learning*, pages 2380–2388, 2015b.

Yifan Wu, Roshan Shariff, Tor Lattimore, and Csaba Szepesvári. Conservative bandits. In *ICML*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1254–1262. JMLR.org, 2016.

Lin F. Yang and Mengdi Wang. Reinforcement leaning in feature space: Matrix bandit, kernels, and regret bound. *CoRR*, abs/1905.10389, 2019.

Andrea Zanette and Emma Brunskill. Tighter problem-dependent regret bounds in reinforcement learning without domain knowledge using value function bounds. In *ICML*, volume 97 of *Proceedings of Machine Learning Research*, pages 7304–7312. PMLR, 2019.