The Sliding Regret

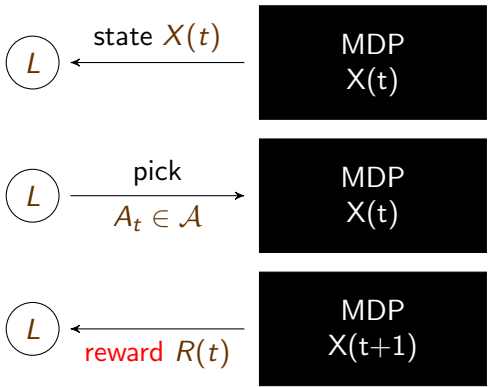
Victor Boone, Bruno Gaujal
Inria and Univ. Grenoble Alpes

Toulouse, June 2024

Reinforcement Learning in MDP: framework

MDP $M = (\mathcal{S}, \mathcal{A}, r, P)$. Let L be a learning algorithm.

1. Observe
2. Pick action
3. Get feedback



Regret and pseudo-regret

Measure of the efficiency of a learner L is the expected *regret* on MDP M after T steps:

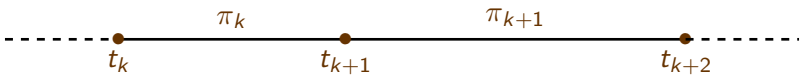
$$Reg(T) = g^* T - \mathbf{E}[\sum_{t=1}^{T} R_t].$$

The pseudo-regret measures the performance of the policy chosen by L:

$$Reg(T) = \mathbf{E} \sum_{t=1}^{T-1} \Delta_{X_t, A_t}, \quad \text{with } \Delta_{x,a} := h^*_x + g^*_x - r_{x,a} - p_{x,a} \cdot h^*.$$

The Bellman gap Δ_z is similar to a relative Q-value.
Differs from usual regret by an additive term $\leq sp(h^*)$.
$Reg(T) = \sum_{z \in \mathcal{Z}} \mathbf{E}[N_z(T)]\Delta_z$,
where $N_z(T)$ is the number of visits of the pair $z = (x, a)$ up to time T (excluded).

Model-based algorithm

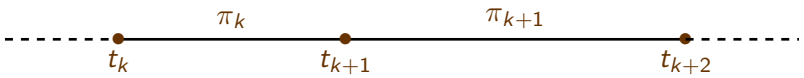


1. **How to compute π_k ?** *Optimism in Face of Uncertainty*: pick a policy that solves the optimist problem:

$$\tilde{g}_t^* = \max_{\pi} \max_{\widetilde{M} \in \widetilde{\mathcal{M}}_{t_k}} g(\pi; \widetilde{M}), \qquad (EVI)$$

$\widetilde{\mathcal{M}}_{t_k}$ is the confidence region at time t_k.

Model-based algorithm



1. **How to compute π_k ?** *Optimism in Face of Uncertainty*: pick a policy that solves the optimist problem:

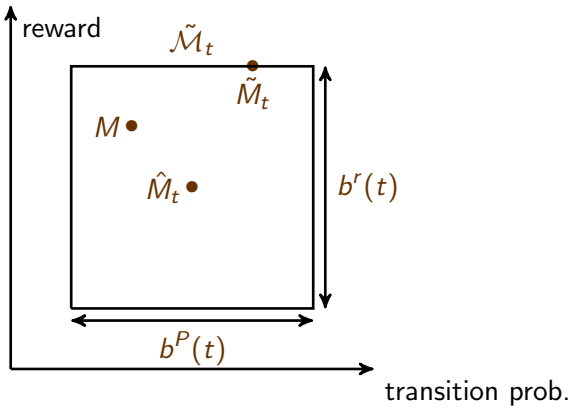
$$\tilde{g}_t^* = \max_{\pi} \max_{\widetilde{M} \in \widetilde{\mathcal{M}}_{t_k}} g(\pi; \widetilde{M}), \qquad (EVI)$$

$\widetilde{\mathcal{M}}_{t_k}$ is the confidence region at time t_k.

2. **How to choose episode ends t_{k+1} ?** In current algorithms, only based on the number of visits. The *doubling trick (DT)* rule: episode ends when a state-action pair is visited twice as often as at the start of the current episode: $N_{(x,a)}(t) = 2N_{(x,a)}(t_k)$.

OFU: Optimism in the Face of Uncertainty

Principle: the learner maintains a confidence set $\tilde{\mathcal{M}}_t$ for the unknown MDP M and uses an optimal policy of the best MDP in the confidence set.



Generic learning algorithm

In the following we consider EVI- based algorithms using confidence regions based on inequalities of the form:

$$N_z(t) d_p(\hat{p}_z(t), \tilde{p}_z) \leq \log(C_p t), \tag{1}$$
$$N_z(t) d_q(\hat{q}_z(t), \tilde{q}_z) \leq \log(C_q t) \tag{2}$$

Where $d_p(-,-)$ and $d_q(-,-)$ are "divergence" operators, for e.g., L_2-norm, squared L_1 norm or Kullback-Leibler divergence.
Remark: UCRL2, KL-UCRL, UCRL-2B and many others fit in this framework.
A notable exemption is EBF where the confidence region is not exactly of this form.

State of the Art: Near Optimal Regret

Quest for near-optimal minimax regret (matching the lower bound $\Omega(\sqrt{DSAT})$).

State of the Art: Near Optimal Regret

Quest for near-optimal minimax regret (matching the lower bound $\Omega(\sqrt{DSAT})$).

UCRL2 (Auer, Jaksch, Ortner, 2009)	$Reg(T) = O(SD\sqrt{AT\log(T)})$
KL-UCRL (Filippi, Cappé, Garivier, 2010) (Talebi, Maillard, 2018)	$Reg(T) = O(S\sqrt{DAT\log(T)})$
UCRL2-B (Fruit, Pirotta, Lazaric, 2018)	$Reg(T) = O(S\sqrt{DAT\log^2(T)})$
EBF (Zhang, Ji, 2019)	$Reg(T) = O(\sqrt{DSAT\log(T)})$
PM-EVI (Boone, Zhang, 2024)	$Reg(T) = O(\sqrt{DSAT\log(T)})$

State of the Art: Near Optimal Regret

Quest for near-optimal minimax regret (matching the lower bound $\Omega(\sqrt{DSAT})$).

UCRL2 (Auer, Jaksch, Ortner, 2009)	$Reg(T) = O(SD\sqrt{AT\log(T)})$	Hoeffding ineq.
KL-UCRL (Filippi, Cappé, Garivier, 2010) (Talebi, Maillard, 2018)	$Reg(T) = O(S\sqrt{DAT\log(T)})$	KL bounds
UCRL2-B (Fruit, Pirotta, Lazaric, 2018)	$Reg(T) = O(S\sqrt{DAT\log^2(T)})$	Bernstein ineq.
EBF (Zhang, Ji, 2019)	$Reg(T) = O(\sqrt{DSAT\log(T)})$	bounds on bias
PM-EVI (Boone, Zhang, 2024)	$Reg(T) = O(\sqrt{DSAT\log(T)})$	bounds on bias

Beyond Regret Bounds

All these algorithms are based on the episodic template.
The difference is on the choice of the confidence region.
But in all cases, the episode length is given by the doubling trick rule (DT).

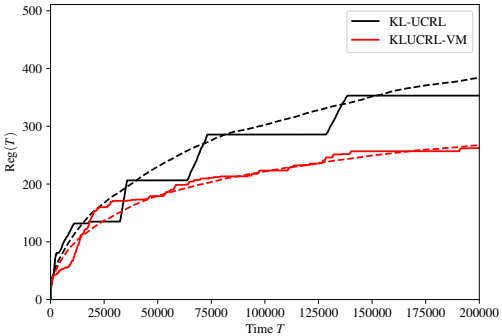
Beyond Regret Bounds

All these algorithms are based on the episodic template.
The difference is on the choice of the confidence region.
But in all cases, the episode length is given by the doubling trick rule (DT).

Is there any interest to replace DT by a refined rule for episode lengths?

No, if the only concern is the regret...
Could be, if you consider one run of your favorite algoithm.

Numerical Example



The regret is meant to measure the quality of a learning algorithm over a single run.
The variability under (DT) questions the pertinence of the expected regret in that case.
The behavior under the new rule (VM) is better in that respect.

Sliding Regret and Regret of Exploration

sliding (or local) regret:

$$Reg(t, t+T) := \mathbf{E} \sum_{u=t}^{t+T-1} \Delta_{Z_u}$$

Sliding Regret and Regret of Exploration

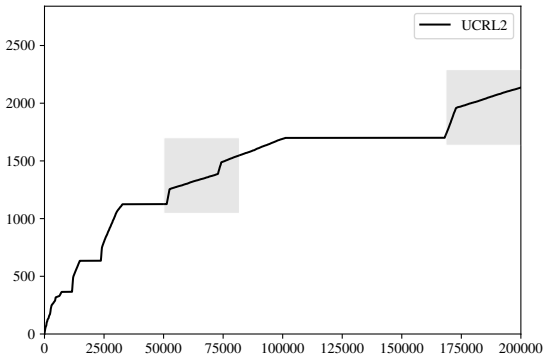
sliding (or local) regret:

$$Reg(t, t+T) := \mathbf{E} \sum_{u=t}^{t+T-1} \Delta_{Z_u}$$

If t is arbitrary, the sliding regret can be arbitrarily bad (linear in T)
Specific version of the sliding regret: measuring the sliding regret at exploration times.
Episode k is an *exploration episode* (denoted $\mathcal{E}^-$) if $\pi_{k-1} \in \Pi^*$ and $\pi_k \notin \Pi^*$.
This is the regret of exploration:

$$RegExp(T) := \limsup_{k\to\infty} \mathbf{E}\left[\mathrm{Reg}(t_k; t_k + T) \mid k \in \mathcal{E}^-\right].$$

Regret of Exploration: visual representation

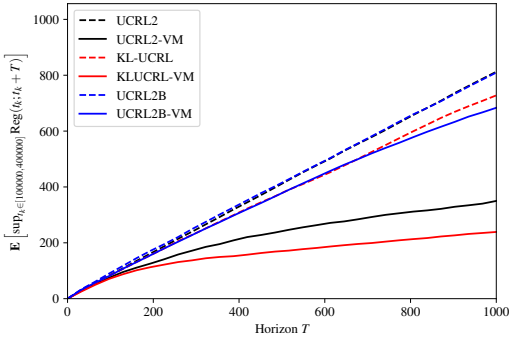


Sliding regret over $T = 25000$ steps at two successive explorations times.

Linear RegExp under DT

Theorem

Let M be a recurrent MDP that is not trivial to learn, then, any consistent algo using DT has a linear regret of exploration over M: $RegExp(T) = \Omega(T)$.



First Alternative to the Doubling Trick: Performance Test

Idea:

Dynamically check, over time, that the current policy is still good:

In the learning algorithm, (DT) is replaced by: (DT) or (PT): $g(\pi_k, \widetilde{\mathcal{M}}_t) + \sqrt{\frac{\theta \log(t)}{t}} < \tilde{g}_t^*$.

Theorem (Regret guarantees of PT)

Using the same confidence region $\widetilde{\mathcal{M}}_t$ as UCRL2, UCRL-PT has expected regret:
$Reg(T) = O(\left(DS\sqrt{A} + \theta^{-1/2} D^2 S^{3/2} A\right)\sqrt{T})$

First Alternative to the Doubling Trick: Performance Test

Idea:

Dynamically check, over time, that the current policy is still good:

In the learning algorithm, (DT) is replaced by: (DT) or (PT): $g(\pi_k, \widetilde{\mathcal{M}}_t) + \sqrt{\frac{\theta \log(t)}{t}} < \tilde{g}_t^*$.

Theorem (Regret guarantees of PT)

Using the same confidence region $\widetilde{\mathcal{M}}_t$ as UCRL2, UCRL-PT has expected regret:
$Reg(T) = O(\left(DS\sqrt{A} + \theta^{-1/2} D^2 S^{3/2} A\right)\sqrt{T})$

Theorem

UCRL-PT has **sub-logarithmic** *regret of exploration for any M in DMDPs:*

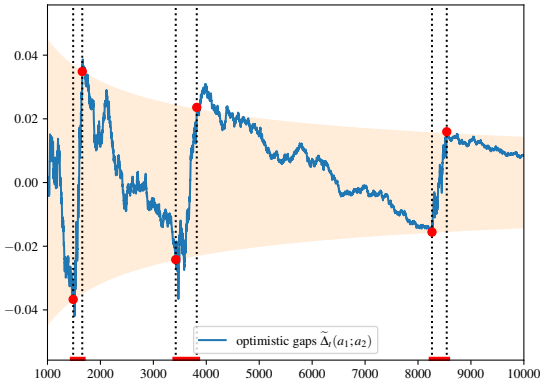
$$RegExp(T) = O(\log T)$$

Improvement: Lazy testing

Main drawback of (PT): EVI has to be solved at each time step. This can be alleviated:

- EVI converges faster when the optimal value at the previous time step is known (only P_{X_1,A_t} and r_{X_t,A_t} are modified).
- Lazy test: Performance test is not performed at each time step but according to a testing scheme that doubles the intervals between two tests.

Illustration of the behavior of UCRL-PT



Different behavior in sub-optimal / optimal episodes. This explains the good performance of UCRL-PT but is also related to the shrinking/shaking effect used in our 2nd solution.

Second Alternative to the doubling rule: Vanishing Multiplicative

Recall the Doubling Trick (DT) stopping rule:

$$N_{X_t,\pi^k(X_t)}(t) > (2)\, N_{X_t,\pi^k(X_t)}(t_k) \tag{EVI}$$

the Vanishing Multiplicative (VM) stopping rule:

$$N_{X_t,\pi^k(X_t)}(t) > (1 + f(t_k))\, N_{X_t,\pi^k(X_t)}(t_k) \tag{VM}$$

where $f(t)$ is a vanishing function of t. More precisely, we assume $1/\log(t) \gtrsim f(t) \gtrsim 1/\sqrt{t}$.

No degradation of the regret

Theorem (minimax regret)

Assume that the running algorithm is EVI-based with confidence region constructed as in (1). *Under (VM), the number of episodes in* $K(T) = O(SA\log(T)/f(T))$ *and , for all MDP* $M \in \mathcal{M}$ *such that* $D(M) < D$,

$$Reg(T) = O(DS\sqrt{AT\log(T/\delta)})$$

with probability $1-\delta$, *and in expectation*

$$\mathbf{E}Reg(T) = O(DS\sqrt{AT})$$

Theorem (Model dependent regret)

For any $M \in \mathcal{M}$, $\mathbf{E}^M[Reg(T)] = O(\log(T)/f(T))$, *with model dependent constants.*

Small increase in numerical complexity

The computational cost of any optimist algorithm comes almost exclusively from EVI. In UCRL2, the total cost of EVI is $O(D\sqrt{SAT})$ (Boone & Zhang, 2024).
Under (VM) the number of episodes can be as low as $O(\log^2(T)$ by choosing $f(t) = 1/\log(t)$. This implies that the number of calls to EVI is $O(\log^2(T))$, compared to $O(\log(T))$ under (DT) and $O(T)$ for (PT).

Small increase in numerical complexity

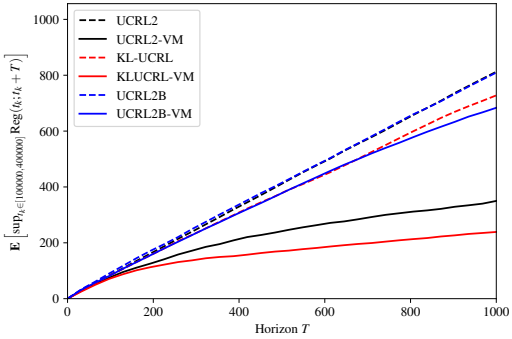
The computational cost of any optimist algorithm comes almost exclusively from EVI. In UCRL2, the total cost of EVI is $O(D\sqrt{S}AT)$ (Boone & Zhang, 2024).
Under (VM) the number of episodes can be as low as $O(\log^2(T)$ by choosing $f(t) = 1/\log(t)$. This implies that the number of calls to EVI is $O(\log^2(T))$, compared to $O(\log(T))$ under (DT) and $O(T)$ for (PT).

- The time complexity of UCRL2 is $O(D\sqrt{S}AT)$;
- The time complexity of UCRL2-VM is $O(D\sqrt{S}AT\log(T))$;
- The time complexity of UCRL2-PT is $O(T^2)$.

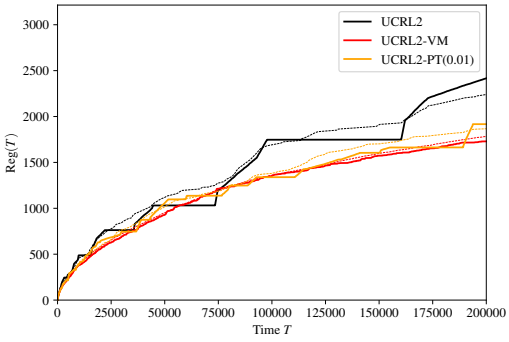
Big gain for the regret of exploration

Theorem

For any EVI-based algorithm with confidence region satisfying 1 and episodes following the (VM) rule, for all non-degenerate MDP in $\mathcal{M}$, $RegExp(T) = O(log(T))$.



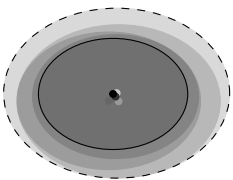
Comparison with (PT)



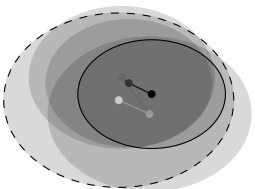
VM and PT have the same type of behavior over one run / expected regret. However (VM) runs 10 times faster here (riverswim 10 states).

Ideas of the proof (I): shrinking/shaking

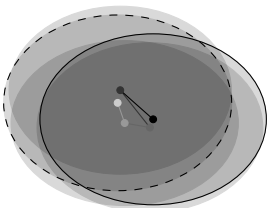
The behavior of the confidence region is very different when the current policy is sub-optimal and when the curremt policy is optimal.



$N_z(t) = o(\log(t))$



$N_z(t) = \Theta(\log(t))$



$N_z(t) = \Omega(\log(t))$

Ideas of the proof (II): Coherence

The shrinking/shaking effect implies that the algorithm has a coherent behavior:
An algorithm is coherent if under a good event and a stopping time τ,

$$\forall t \in [\tau, \tau + T], \left\{ g(\pi_t, S_t) < g^* \Rightarrow \exists z, S_t \xrightarrow{\pi_t} x : \; N_z(t) - N_z(\tau) \leq C \log(T) \right\}.$$

Coherence roughly says that if the current policy is sub-optimal, this implies that some reachable state (under the current policy) have been sub-sampled (in $O(\log(T))$).

Ideas of the proof (II): Coherence

The shrinking/shaking effect implies that the algorithm has a coherent behavior:
An algorithm is coherent if under a good event and a stopping time τ,

$$\forall t \in [\tau, \tau + T], \left\{ g(\pi_t, S_t) < g^* \Rightarrow \exists z, S_t \xrightarrow{\pi_t} x : \; N_z(t) - N_z(\tau) \leq C \log(T) \right\}.$$

Coherence roughly says that if the current policy is sub-optimal, this implies that some reachable state (under the current policy) have been sub-sampled (in $O(\log(T))$).

Finally, coherence implies that sub-optimal episodes are short and in turn this yields a logarithmic regret of exploration.

That's all folks!