Online Reinforcement Learning in Average-Reward MDPs

Mohammad Sadegh Talebi
m.shahi@di.ku.dk
Department of Computer Science

Online Average-Reward RL: Setting and Performance Metrics

Setting

Online Average-Reward RL. An **agent** interacts with an average-reward MDP $M = (\mathcal{S}, \mathcal{A}, P, R)$ for T rounds (potentially unbounded) without any reset.

At each time step $t = 1, 2, \ldots$:

- The agent observes the current state s_t and takes an action $a_t \in \mathcal{A}$
- M decides a reward $r_t \sim R(s_t, a_t)$ and a next state $s_{t+1} \sim P(\cdot|s_t, a_t)$
- The agent receives r_t (any time in step t before start of $t+1$)

M is unknown (beyond $\mathcal{S}$ and $\mathcal{A}$), and the goal is to maximize $\sum_{t=1}^{T} r_t$ (in expectation) using collected experience (history):

$$h_t = (s_1, a_1, r_1, \ldots, s_{t-1}, a_{t-1}, r_{t-1}, s_t)$$

Need for balancing exploration and exploitation.

Setup

The goal is to maximize T-step total reward

$$\sum_{t=1}^{T} r_t$$

Recall that under $\pi^\star$,

$$\sum_{t=1}^{T} r_t^\star = Tg^\star + \mathcal{O}\Big(\mathrm{sp}(b^\star)\sqrt{T\log(T/\delta)}\Big), \quad \text{w.p. } \geq 1-\delta$$

Hence, the agent can resort to learning a gain-optimal policy $\pi^\star \in \Pi^{\mathsf{SD}}$ in M:

$$\pi^\star(s) \in \arg\max_{\pi^\star \in \Pi^{\mathsf{SD}}} g^\pi(s)$$

Hence, in online average-reward RL, the goal can be set to learn $\pi^\star$ from collected experience.

Online RL: Performance Metrics

- Many offline algorithms can be made online with some tricks.
- But will they explore well?

For online RL, we need performance metrics to measure the quality of exploration-exploitation tradeoff.

Online RL: Performance Metrics

The performance of a learning algorithm $\mathbb{A}$ can be measured through:

- **Convergence:** Whether $\mathbb{A}$ converges to an optimal (or near-optimal) policy.
- **PAC Sample Complexity:** The number of steps where the value of the current policy output by $\mathbb{A}$ is not near-optimal with high-probability.
- **Regret:** The amount of reward lost due to choosing sub-optimal actions by $\mathbb{A}$.

In fact these metrics measure how exploration-exploitation tradeoff is implemented.

Regret

In online average-reward RL, the **Regret** of an algorithm $\mathbb{A}$ is the difference between cumulative reward of the optimal policy $\pi^\star$ (oracle) and that gathered by $\mathbb{A}$:

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

- Agent $\mathbb{A}$'s trajectory:

$$\forall t : \qquad a_t = \mathbb{A}(h_t), \quad r_t \sim R(\boldsymbol{s_t}, a_t), \quad \boldsymbol{s_{t+1}} \sim P(\cdot|\boldsymbol{s_t}, a_t)$$

- Oracle's trajectory:

$$\forall t : \qquad r_t^\star \sim R\big(s_t^\star, \pi^\star(s_t^\star)\big), \quad \boldsymbol{s_{t+1}^\star} \sim P\big(\cdot\,|\boldsymbol{s_t^\star}, \pi^\star(\boldsymbol{s_t^\star})\big)$$

Regret is directly connected to the agent's goal (maximizing $\sum_{t=1}^{T} r_t$).
Alternatively, the objective of the agent is to minimize the regret.

No-Regret Algorithm

- $\mathfrak{R}(\mathbb{A}, T)$ is a r.v., and we wish to control it in expectation or with high probability.
- An algorithm $\mathbb{A}$ is a no-regret learning algorithm if either

$$\mathbb{E}[\mathfrak{R}(\mathbb{A}, T)] = o(T) \quad \text{or} \quad \mathfrak{R}(\mathbb{A}, T) = o(T) \quad \text{w.h.p.}$$

No-Regret Algorithm

An algorithm $\mathbb{A}$ is called no-regret if there exists a deterministic function f with $\frac{f(T)}{T} \to_{T\to\infty} 0$, such that one of the following holds:

$$\mathbb{E}[\mathfrak{R}(\mathbb{A}, T)] \le f(T)$$
$$\mathfrak{R}(\mathbb{A}, T) \le f(T) \quad \text{with high probability.}$$

- f could be MDP-dependent but must be deterministic.
- Note that a high-probability bound on $\mathfrak{R}(\mathbb{A}, T)$ implies a bound on $\mathbb{E}[\mathfrak{R}(\mathbb{A}, T)]$, but not the other way around.

Exploration vs. Exploitation

The key difficulty to do so is to balance *exploration and exploitation*:

- Play the best action so far, ...
- ... or rather explore a different action?

Warm-up: A Simple Algorithm

Empirical MDP

For any $t \geq 1$, define

- $N_t(s,a,s')$: number of visits, up to t, to (s,a) followed by a visit to s'

$$N_t(s,a,s') = \sum_{i=1}^{t-1} \mathbb{I}\{s_i = s, a_i = a, s_{i+1} = s'\}$$

- $N_t(s,a) = \sum_{s' \in \mathcal{S}} N_t(s,a,s')$

Empirical Estimator for P:

$$\forall s' \in \mathcal{S}: \quad \widehat{P}_t(s'|s,a) = \begin{cases} \frac{N_t(s,a,s')}{N_t(s,a)} & \text{if } N_t(s,a) > 0 \\ \frac{1}{S} & \text{otherwise} \end{cases}$$

Empirical Estimator for R:

$$\widehat{R}_t(s,a) = \frac{1}{N_t(s,a)} \sum_{i=1}^{t-1} r_i \mathbb{I}\{s_i = s, a_i = a\}$$

Empirical MDP

The empirical MDP:

$$\widehat{M}_t = (\mathcal{S}, \mathcal{A}, \widehat{P}_t, \widehat{R}_t)$$

Why not only using $\widehat{M}_t$. I.e., finding the optimal policy in $\widehat{\pi}_t^\star$ and taking $a_t = \widehat{\pi}_t^\star(s_t)$ each step.

$\implies$ No exploration-exploitation tradeoff. Will not lead to a no-regret algorithm.

A better proposal. At each time t,

$$a_t = \begin{cases} \widehat{\pi}_t^\star(s_t) & \text{w.p. } 1-\varepsilon_t \\ \text{chosen uniformly at random over } \mathcal{A} & \text{w.p. } \varepsilon_t \end{cases}$$

Despite its simplicity, it will enjoy a sublinear regret for suitably chosen ε_t —E.g., $\varepsilon = \frac{1}{\sqrt{t}}$.

A Simple Algorithm

- **input:** $(\varepsilon_t)_{t\geq 1}$
- **initialization:** For all (s,a,s'), $N(s,a,s')=0$
- **for** $t=1,2,\dots,T$
  - Compute estimates $\widehat{P}_t$ and $\widehat{R}_t$
  - Find $\widehat{\pi}_t^\star$ using `VI` with accuracy $\frac{1}{\sqrt{t}}$
  - Take action

$$a_t = \begin{cases} \widehat{\pi}_t^\star(s_t) & \text{w.p. } 1-\varepsilon_t \\ \text{chosen uniformly at random over } \mathcal{A} & \text{w.p. } \varepsilon_t \end{cases}$$

  - Receive reward $r_t \sim R(s_t,a_t)$ and next-state $s_{t+1} \sim P(\cdot|s_t,a_t)$
  - Update $N(s_t,a_t,s_{t+1})$

UCRL2: Upper Confidence Reinforcement Learning

The simple algorithm implements

the certainty equivalence principle + exploration.

- Intuitive design (+)
- For suitable ε_t, it becomes no-regret (+)
- Tuning ε_t is not easy, and may require prior knowledge to obtain sublinear regret (−)
- Weak empirical performance (−).

We need more powerful principle.

OFU Principle

Optimism in the Face of Uncertainty (OFU)

- A well-known principle in balancing exploration-exploitation in bandits and online RL dating back to (Lai & Robbins, 1985).
- Also known as the Optimism principle

The OFU Principle: In an uncertain world, suppose that the environment is the best possible (in terms of rewards)!

- If the chosen action is optimal $\Longrightarrow$ no penalty
- If sub-optimal $\Longrightarrow$ reducing uncertainty

Optimism in the Face of Uncertainty (OFU)

In bandits, OFU prescribes replacing unknown mean rewards by their corresponding high-probability UCBs. the most prominent example is the `UCB` algorithm.

In MDPs, different implementations exist depending on the approach

- Model-based: Select the best candidate environment (among all plausible models/MDPs), i.e. the one leading to the highest possible value function.
- Model-free: When updating the Q-function, be optimistic. Initialize all Q-values to their highest possible value and use "reward + exploration bonus" instead of "reward" alone.

This lecture: A no-regret algorithm (`UCRL2`) based on OFU.

OFU: Model-Based

`UCRL2` (Jaksch et al., 2010):

- Stands for Upper Confidence Reinforcement Learning
- A model-based algorithm for average-reward designed based on OFU.

Model-based recipe for the optimism principle (OFU):

- **Step 1:** Maintains a set of plausible MDPs (models) (i.e., consistent with history h_t). This can be done by defining high-probability confidence sets for R and P, and forming a corresponding set of MDPs.
- **Step 2:** Choose an optimistic model (among all models) and an optimistic policy leading to the highest gain.

Step 1: Confidence Sets

$\delta \in (0,1)$ is given.

Confidence Set for R:

- Define a confidence set for $R(s,a)$ as

$$C_{s,a} = \left\{ \lambda \in [0,1] : |\widehat{R}_t(s,a) - \lambda| \le \beta_{N_t(s,a)} \right\}$$

  for some suitable function $\beta_{N_t(s,a)}$.

- For example, using Hoeffding's inequality (combined with Laplace's methods):

$$\beta_n = \sqrt{\tfrac{1}{2n}\left(1+\tfrac{1}{n}\right)\log \tfrac{SA\sqrt{n+1}}{\delta}}, \quad n \in \mathbb{N}.$$

$$\mathbb{P}\Big(\forall t,\, \forall (s,a):\, R(s,a) \in C_{s,a}\Big) \ge 1-\delta$$

Step 1: Confidence Sets

$\delta \in (0,1)$ is given.

Confidence Set for P:

- Define a confidence set for $P(\cdot|s,a)$ as

$$C'_{s,a} = \left\{ q \in \Delta(\mathcal{S}) : \left\| \widehat{P}_t(\cdot|s,a) - q \right\|_1 \le \beta'_{N_t(s,a)} \right\}$$

  for some suitable function $\beta'_{N_t(s,a)}$.

- For example, using Weissman'ss inequality (combined with Laplace's methods):

$$\beta'_n = \sqrt{\tfrac{2}{n}\left(1 + \tfrac{1}{n}\right) \log \tfrac{SA(2^S - 2)\sqrt{n+1}}{\delta}}$$

$$\mathbb{P}\Big(\forall t,\, \forall (s,a) :\ P(\cdot|s,a) \in C'_{s,a} \Big) \ge 1 - \delta$$

Step 1: Set of Models

Confidence sets $\{C_{s,a}, C'_{s,a}\}_{s\in\mathcal{S},a\in\mathcal{A}}$ yield a set of models consistent with h_t:

$$\mathcal{M}_t = \Big\{ M' = (\mathcal{S}, \mathcal{A}, P', R') : P'(\cdot|s,a) \in C'_{s,a} \text{ and } R'(s,a) \in C_{s,a},\ \forall (s,a) \Big\}$$

- $\mathcal{M}_t$ collects *all MDPs* that could be a candidate for the true Model M (in view of h_t).
- M is trapped in $\mathcal{M}_t$ with high probability, simultaneously for all t:

$$\mathbb{P}(\forall t : M \in \mathcal{M}_t) \geq 1 - 2\delta$$

- $\mathcal{M}_t$ is called a bounded parameter MDP.

Step 2: Planning

Step 2: Planning. To implement OFU, we wish to find

$$\pi_t \in \arg \max_{M' \in \mathcal{M}_t} \max_{\pi \in \Pi^{\mathsf{SD}}} g^{\pi}_{M'}$$

and then we choose $a_t = \pi_t(s_t)$.

Alternatively, by Bellman's optimality equation, we wish to find $\widetilde{g}$ and $\widetilde{b}$ satisfying: For all s,

$$\widetilde{g} + \widetilde{b}(s) = \max_{a \in \mathcal{A}} \left(\max_{R'(s,a) \in C_{s,a}} R'(s,a) + \max_{P'(\cdot|s,a) \in C'_{s,a}} \sum_x P'(x|s,a)\widetilde{b}(s) \right)$$

Step 2: Planning

$$\widetilde{g} + \widetilde{b}(s) = \max_{a \in \mathcal{A}} \left(\max_{R'(s,a) \in C_{s,a}} R'(s,a) + \max_{P'(\cdot|s,a) \in C'_{s,a}} \sum_x P'(x|s,a) \widetilde{b}(s) \right)$$

Compared to optimality equations for MDPs, we have two extra maximizations.

- The one in blue admits a closed-form solution:

$$\max_{R'(s,a) \in C_{s,a}} R'(s,a) = \widehat{R}_t(s,a) + \beta_{N_t(s,a)}$$

- No closed-form solution to the second. However, for a fixed $u \in \mathbb{R}^S$, the problem

$$\max_{p \in C'(s,a)} \sum_x p(x) u(x)$$

  can be solve using a simple procedure thanks to the shape of $C'_{s,a}$.

The second optimization problem can be efficiently solved using `Extended Value Iteration (EVI)`

Step 2: Planning

In summary,

- To implement OFU, we wish to find

$$\pi_t \in \arg \max_{M' \in \mathcal{M}_t} \max_{\pi \in \Pi^{\mathsf{SD}}} g_{M'}^{\pi}$$

- It suffices to find a $\frac{1}{\sqrt{t}}$-optimal policy π_t:

$$g^{\pi_t} \geq \arg \max_{M' \in \mathcal{M}_t} \max_{\pi \in \Pi^{\mathsf{SD}}} g_{M'}^{\pi} - \frac{1}{\sqrt{t}}$$

- This can be done efficiently by `Extended Value Iteration (EVI)` —see next slides for the pseudo-code.

UCRL2-L: Planning

For technical issues with regret analysis, UCRL2-L does not update policy π_t at each step. Rather it proceeds in *internal* epochs:

- In each epoch, the policy will be kept unchanged.
- An epoch stops as soon as $N_t(s, a)$ for some (s, a) is doubled (compared to its number before the episode).

To implement this, UCRL2-L maintains two sets of counters:

- Global counters:
  $N(s, a, s')$ for each (s, a, s'), and $N(s, a) = \max 1, \sum_{s'} N(s, a, s')$
- Per-epoch counters: $\nu(s, a, s')$, which count the number of visits within an epoch. Further define:

$$\nu(s, a) = \sum_{s'} \nu(s, a, s')$$

UCRL2-L

- **input:** δ
- **initialization:** For all (s, a),
  - $N(s, a) = 0$, $v(s, a) = 0$
- **for** epochs $k = 1, 2, \ldots$
  - $N(s, a, s') \leftarrow N(s, a, s') + \nu(s, a, s')$ for all (s, a)
  - Compute estimates $\widehat{P}_t$ and $\widehat{R}_t$
  - Find π_k using `EVI` with accuracy $\frac{1}{\sqrt{t}}$
  - $\nu(s, a, s') = 0$ for (s, a, s')
  - **repeat**
    - Choose $a_t = \pi_k(s_t)$
    - Receive reward $r_t \sim R(s_t, a_t)$ and next-state $s_{t+1} \sim P(\cdot|s_t, a_t)$
    - Update $\nu(s_t, a_t, s_{t+1}) \leftarrow \nu(s_t, a_t, s_{t+1}) + 1$
    - Increment t
  - **until** $\nu(s, a) = N(s, a)$ for some (s, a)

UCRL2-L: EVI

- **input:** ε
- **initialization:** Select $u_0 \in \mathbb{R}^S$ arbitrarily. Set $n = -1$.
- **repeat:**
  - Increment n
  - Compute, for each (s, a),

$$R'(s,a) = \widehat{R}_t(s,a) + \beta_{N(s,a)}$$

$$P'(\cdot|s,a) \in \operatorname{argmax}\left\{ \sum_{x\in\mathcal{S}} q(x)u_n(x) : q \in C'_{s,a} \right\}$$

  - Update, for each (s, a),

$$u_{n+1}(s) = \max_a \left(R'(s,a) + \sum_{x\in\mathcal{S}} P'(x|s,a)u_n(x) \right)$$

  until $\max_s\left(u_{n+1}(s) - u_n(s)\right) - \min_s\left(u_{n+1}(s) - u_n(s)\right) < \varepsilon$
- **output:** Policy π_k,

$$\pi_k(s) \in \operatorname*{argmax}_a \left(R'(s,a) + \sum_{x\in\mathcal{S}} P'(x|s,a)u_n(x) \right), \quad \forall s$$

UCRL2-L: Inner Maximization in EVI

Algorithm for solving

$$\max_{q \in C'_{s,a}} \sum_{x \in \mathcal{S}} q(x) u(x)$$

Index $\mathcal{S} = \{s_1, s_2, \ldots, s_S\}$, and assume w.l.o.g. that

$$u(s_1) \geq u(s_2) \geq \ldots \geq u(s_S)$$

- **initialization:** $q = \widehat{P}_t(\cdot|s,a)$
- Set $q(s_1) = \widehat{P}_t(s_1|s,a) + \frac{1}{2}\beta'_{N_t(s,a)}$
- $\ell = S$
- **while:** $\sum_{x \in \mathcal{S}} q(x) > 1$
  - Set $q(s_\ell) = \max\left\{0, 1 - \sum_{x \neq s_\ell} q(x)\right\}$
  - Decrement ℓ
- **output:** q

UCRL2-L: Regret Guarantee

UCRL2-L: Regret

Regret bound for UCRL2-L in any communicating MDP with S states, A actions, and diameter D:

Theorem (Regret of UCRL2-L)

Let $\delta \in (0,1)$. The regret under UCRL2-L satisfies

$$\mathfrak{R}(T) \leq 24DS\sqrt{AT\log(T/\delta)},$$

with probability at least $1-\delta$, and uniformly for all $T \geq 2$.

- Expected regret:

$$\mathbb{E}[\mathfrak{R}(T)] \leq (1-\delta) \times 24DS\sqrt{AT\log(T/\delta)} + \delta T$$

  Setting $\delta = 1/\sqrt{T}$ gives: $\mathbb{E}[\mathfrak{R}(T)] \leq 31DS\sqrt{AT\log(T)}$

- For rewards supported on $[a,b]$ (instead of $[0,1]$), scale $\mathfrak{R}(T)$ by $(b-a)$.

UCRL2-L: Regret

The theorem tells us that UCRL2-L is no-regret. More precisely:

$$\mathbb{P}\left\{ \sum_{t=1}^{T} r_t^\star > \sum_{t=1}^{T} r_t + 24DS\sqrt{AT\log(T/\delta)} \right\} < \delta,$$

Note this is a worst-case bound:

$$\sup_{M'} \mathfrak{R}_{M'}(T) \le 24DS\sqrt{AT\log(T/\delta)}, \quad \text{w.p.} \; \ge 1-\delta$$

where M' is any communicating MDP with S states, A actions, and diameter D. In particular, it holds for *hardest-to-learn* M'.

Regret: UCRL2-L vs. UCB

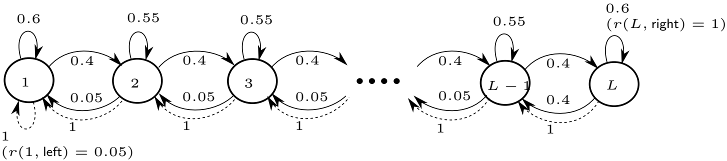
Regret bound in the theorem resembles that for UCB in K-armed bandits:

$$\underbrace{\mathcal{O}\big(DS\sqrt{AT\log(T)}\big)}_{\texttt{UCRL2-L}} \quad \text{vs.} \quad \underbrace{\mathcal{O}\big(\sqrt{KT\log(T)}\big)}_{\texttt{UCB}}$$

In K-armed bandits, we have K unknowns distributions, whereas in MDPs there are $2SA$ unknown distributions (one reward dist. and one transition dist. per state-action pair).

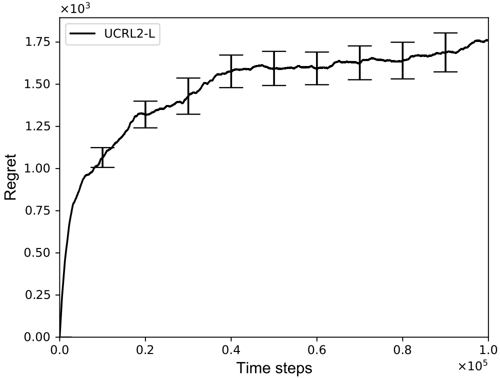
Why does regret must depend on diameter D?

Intuitively, D captures the price to navigate in the MDP due to learning.



UCRL2-L: Empirical Performance

Numerical Experiments

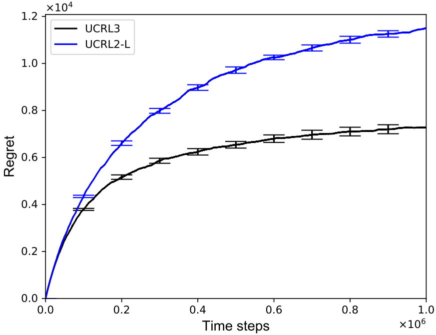


`UCRL2-L` in 6-state RiverSwim: Average regret shown with 95% CIs for $n = 30$ experiments, $\delta = 0.01$.

Numerical Experiments

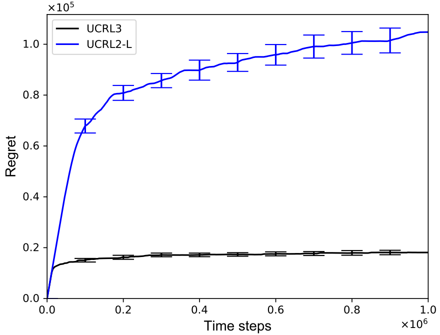
- `UCRL2-L` brings massive improvement over `UCRL2` in almost any MDP, yet achieving a smaller regret bound.
- This is due to using tighter confidence sets derived using a more advanced tool than vanilla union bounds.
- Can we do better empirically than `UCRL2-L`?

Numerical Experiments



- UCRL2-L in 4-room grid-world: Average regret shown with 95% CIs for $n = 30$ experiments).
- The black curve shows the result of UCRL3 (Bourel et al., 2020).

Numerical Experiments



- `UCRL2-L` in 25-state RiverSwim: Average regret shown with 95% CIs for $n = 30$ experiments)
- The black curve shows the result of `UCRL3` (Bourel et al., 2020).

Worst-Case Regret Lower Bound

Worst-Case Lower Bound

How good is the regret bound of UCRL2? Could it be improved?

To answer these, we need to derive lower bounds on regret.

- Problem-dependent lower bound
- Worst-case lower bound

Worst-Case Lower Bound

Theorem (Worst-Case Regret Lower Bound)

Let $S, A \geq 5$, $D \geq 20 \log_A S$, and $T \geq DSA$. For any learning algorithm $\mathbb{A}$, there exists an MDP M with S states, A actions, and diameter D such that for any initial state, the T-step expected regret under $\mathbb{A}$ satisfies

$$\mathbb{E}[\mathfrak{R}(\mathbb{A}, T)] \geq 0.015\sqrt{DSAT}$$

- A regret of $\Omega(\sqrt{DSAT})$ is a fundamental performance limit for communicating MDPs, which no algorithm can beat.
- Compare it to the minimax lower bound of $\Omega(\sqrt{KT})$ for stochastic K-armed bandits.

Worst-Case Lower Bound

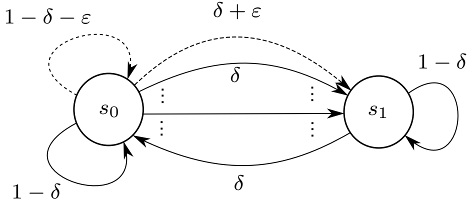
$$\underbrace{\Omega\left(\sqrt{DSAT}\right)}_{\text{worst-case LB}} \quad \text{vs.} \quad \underbrace{\widetilde{\mathcal{O}}\left(DS\sqrt{AT\log\frac{T}{\delta}}\right)}_{\text{UCRL2-L's regret}}$$

UCRL2 is rate-optimal (i.e., its regret has optimal dependence on T, up to logarithmic factors).

- There is a gap of $\sqrt{DS\log(T/\delta)}$ between the LB and the UB.
- The gap is reduced by improved variants of UCRL2 and UCRL2-L.

Worst-Case Lower Bound: Proof

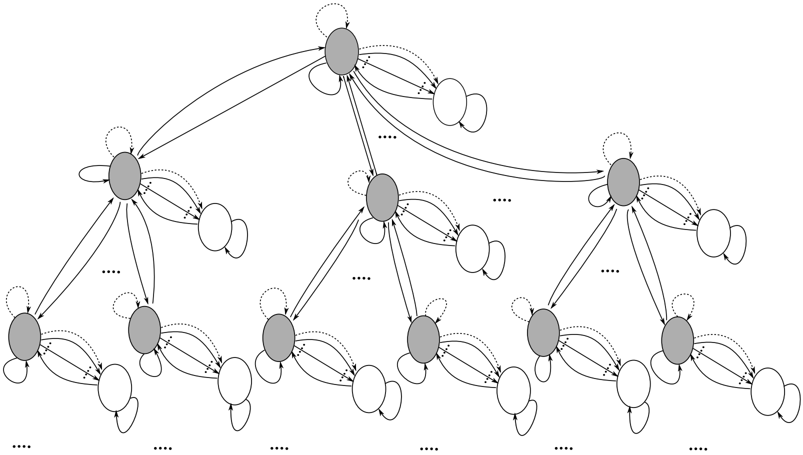
A family of worst-case 2-state MDPs, parameterized by $\delta \in (0, \frac{1}{3})$ and $\varepsilon \le \delta$:



- A actions per state, all with identical rewards and transitions. Only one action in s_0 has a slightly different transition.
- For all actions, in s_0 the reward is 0, but in s_1 the reward is 1.
- Choosing $\varepsilon \propto \sqrt{\frac{A}{TD}}$ leads to a worst-case MDP with 2 states, A actions, and diameter D for any algorithm.

Worst-Case Lower Bound: Proof

A worst-case instance for $S > 2$ —for details, see (Jaksch et al., 2010).



Remarks

UCRL2 is a model-based algorithm for regret minimization in average-reward MDPs.

- Several variants of UCRL2 exist that improve upon its theoretical and/or empirical regret.
- State-of-the-art UCRL2-style algorithms achieve a regret bound *almost* matching the LB.
- These algorithms outperform model-free algorithms empirically, often by a large margin.
- Logarithmic regret bounds are mostly open.

Remarks

UCRL2 is a model-based algorithm for regret minimization in average-reward MDPs.

Key questions:

- Does it find a near-optimal policy?
- Does it output an accurate estimation $\widehat{M}$ of the true MDP?
- Is it capable of doing generalization in MDPs (when possible)?

Remarks

UCRL2 is a model-based algorithm for regret minimization in average-reward MDPs.

Key questions:

- Does it find a near-optimal policy? It does not have a policy recommendation. In general, regret minimization is different than best policy identification.
- Does it output an accurate estimation $\widehat{M}$ of the true MDP? Not necessarily. Some (rewarding) part of state-space could be visited much more than other parts.
- Is it capable of doing generalization in MDPs (when possible)? It doesn't. It assumes that various state-action pairs are unrelated in terms of p and r.

Remarks

(Near)-optimal behavior is easier to learn than the truth.

Minds & Machines (2016) 26:243–252
DOI 10.1007/s11023-016-9389-y

CrossMark

Optimal Behavior is Easier to Learn than the Truth

Ronald Ortner[1]

Received: 26 February 2014 / Accepted: 9 January 2016 / Published online: 3 February 2016

Abstract We consider a reinforcement learning setting where the learner is given a set of possible models containing the true model. While there are algorithms that

References

- Tze Leung Lai and Herbert Robbins, "Asymptotically efficient adaptive allocation rules," *Advances in Applied Mathematics*, 1985.
- Thomas Jaksch, Ronald Ortner, and Peter Auer, "Near-optimal regret bounds for reinforcement learning," *Journal of Machine Learning Research*, 2010.
- Hippolyte Bourel, Odalric-Ambrym Maillard, and Mohammad Sadegh Talebi, "Tightening exploration for upper confidence reinforcement learning," in Proc. *International Conference on Machine Learning*, 2020.