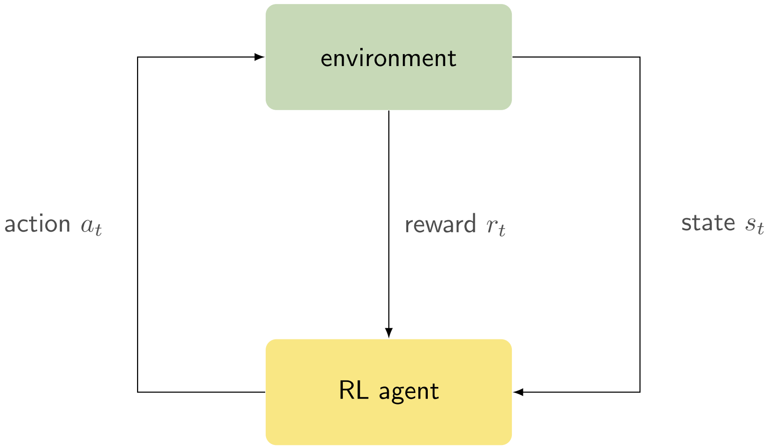
facebook
Artificial Intelligence Research

Exploration-Exploitation in Reinforcement Learning

Part 1 – Finite-Horizon MDPs

Mohammad Ghavamzadeh, Alessandro Lazaric and Matteo Pirotta
Facebook AI Research

RL Agent-Environment Interaction



Website
`https://rlgammazero.github.io`

Markov Decision Process

[Puterman, 1994]

A **finite-horizon** Markov decision process (MDP) is a tuple $M = \langle \mathcal{S}, \mathcal{A}, r_h, p_h, H \rangle$

- State space $\mathcal{S}$
- Action space $\mathcal{A}$
- Horizon H
- Transition distribution $p_h(\cdot|s,a) \in \Delta(\mathcal{S})$, $h = 1, \ldots, H$
- Reward distribution with expectation $r_h(s,a) \in [0,1]$, $h = 1, \ldots, H$

An agent acts according to a *time-variant policy*

$$\pi_h : \mathcal{S} \to \mathcal{A} \qquad h = 1, \ldots, H$$

Markov Decision Process

[Puterman, 1994]

A **finite-horizon** Markov decision process (MDP) is a tuple $M = \langle \mathcal{S}, \mathcal{A}, r_h, p_h, H \rangle$

- State space $\mathcal{S}$
- Action space $\mathcal{A}$
- Horizon H
- Transition distribution $p_h(\cdot|s,a) \in \Delta(\mathcal{S})$, $h = 1, \dots, H$
- Reward distribution with expectation $r_h(s,a) \in [0,1]$, $h = 1, \dots, H$

An agent acts according to a *time-variant policy*

$$\pi_h : \mathcal{S} \to \mathcal{A} \qquad h = 1, \dots, H$$

In (contextual) bandit, actions do not influence the evolution of states

Value Functions and Optimality

Value functions

$$Q_h^{\pi}(s,a) = r_h(s,a) + \mathbb{E}\Big[\sum_{l=h+1}^{H} r_l(s_l, \pi_l(s_l))\Big]$$

$$V_h^{\pi}(s) = Q_h^{\pi}(s, \pi_h(s))$$

Optimality

$$Q_h^{\star}(s,a) = \sup_{\pi} Q_h^{\pi}(s,a)$$

$$\pi_h^{\star}(s) = \arg\max_{a\in\mathcal{A}} Q_h^{\star}(s,a)$$

Value Functions and Optimality

Value functions

$$Q_h^{\pi}(s,a) = r_h(s,a) + \mathbb{E}\Big[\sum_{l=h+1}^{H} r_l(s_l, \pi_l(s_l))\Big]$$

$$V_h^{\pi}(s) = Q_h^{\pi}(s, \pi_h(s))$$

Optimality

$$Q_h^{\star}(s,a) = \sup_{\pi} Q_h^{\pi}(s,a)$$

$$\pi_h^{\star}(s) = \arg\max_{a\in\mathcal{A}} Q_h^{\star}(s,a)$$

Remark: given $r_h(s,a) \in [0,1]$, then $Q_h(s,a), V_h(s) \in [0, H-(h-1)]$

Bellman Equations

Policy Bellman equation

$$
\begin{aligned}
Q_h^\pi(s,a) &= r_h(s,a) + \mathbb{E}_{s'\sim p_h(\cdot|s,a)}\Big[Q_{h+1}^\pi(s', \pi_{h+1}(s'))\Big] \\
&= r_h(s,a) + \mathbb{E}_{s'\sim p_h(\cdot|s,a)}\bigg[V_{h+1}^\pi(s')\bigg]
\end{aligned}
$$

Optimal Bellman equation

$$
\begin{aligned}
Q_h^\star(s,a) &= r_h(s,a) + \mathbb{E}_{s'\sim p_h(\cdot|s,a)}\Big[\max_{a'\in\mathcal{A}} Q_{h+1}^\star(s', a')\Big] \\
&= r_h(s,a) + \mathbb{E}_{s'\sim p_h(\cdot|s,a)}\Big[V_{h+1}^\star(s')\Big]
\end{aligned}
$$

Value Iteration (aka Backward Induction)

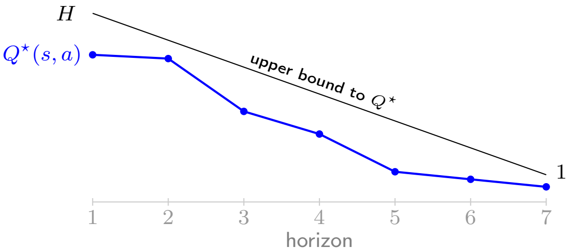
Input: $\mathcal{S}$, $\mathcal{A}$, r_h, p_h
Set $Q^\star_{H+1}(s,a) = 0$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$

for $h = H, \dots, 1$ **do**
  for $(s,a) \in \mathcal{S} \times \mathcal{A}$ **do**
    Compute

$$\begin{aligned} Q^\star_h(s,a) &= r_h(s,a) + \mathbb{E}_{s' \sim p_h(\cdot|s,a)}\Big[\max_{a' \in \mathcal{A}} Q^\star_{h+1}(s',a')\Big] \\ &= r_h(s,a) + \mathbb{E}_{s' \sim p_h(\cdot|s,a)}\Big[V^\star_{h+1}(s')\Big] \end{aligned}$$

  end
end
return $\pi^\star_h(s) = \arg\max_{a \in \mathcal{A}} Q^\star_h(s,a)$

Value Iteration (aka Backward Induction)



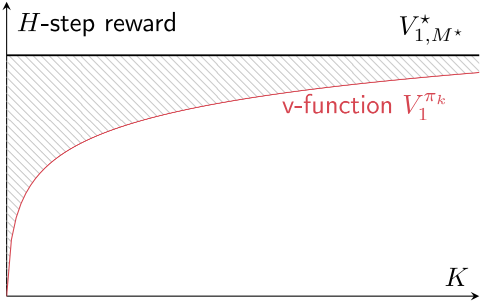
$$Q_h^\star(s,a) = \max_a \left\{ r_h(s,a) + \mathbb{E}_{s'|s,a}[V_{h+1}^\star(s')] \right\}$$

Online Learning Problem

Input: $\mathcal{S}$, $\mathcal{A}$ $\cancel{r_h, p_h}$
Initialize $Q_{h1}(s,a) = 0$ for all $(s,a) \in \mathcal{S} \times \mathcal{A}$ and $h = 1, \ldots, H$, $\mathcal{D}_1 = \emptyset$

for $k = 1, \ldots, K$ **do** `// episodes`
  Define π_k based on $(Q_{hk})_{h=1}^{H}$
  Observe initial state s_{1k} *(arbitrary)*
  for $h = 1, \ldots, H$ **do**
    Execute $a_{hk} = \pi_{hk}(s_{hk})$
    Observe r_{hk} and $s_{h+1,k}$
  end
  Add trajectory $(s_{hk}, a_{hk}, r_{hk})_{h=1}^{H}$ to $\mathcal{D}_{k+1}$
  Compute $(Q_{h,k+1})_{h=1}^{H}$ from $\mathcal{D}_{k+1}$
end

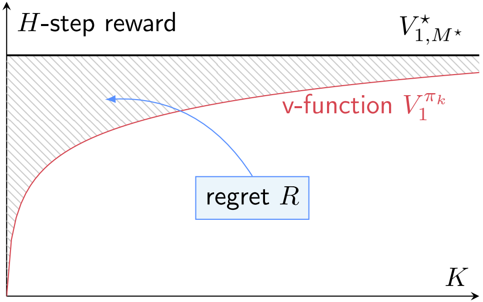
Frequentist Regret



$$R(K,\ M^{\star},\ \mathfrak{A}) = \sum_{k=1}^{K} \Big(V^{\star}(s_{1k}) - V^{\pi_k}(s_{1k}) \Big)$$

Let $T = HK$ total number of steps executed in the environment

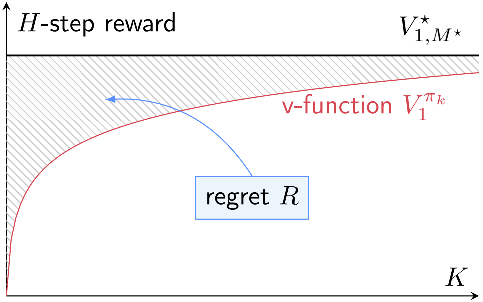
Frequentist Regret



$$R(K,\ M^\star,\ \mathfrak{A}) = \sum_{k=1}^{K} \Big(V^\star(s_{1k}) - V^{\pi_k}(s_{1k}) \Big)$$

Let $T = HK$ total number of steps executed in the environment

Frequentist Regret



unknown true MDP
$M^\star = \langle \mathcal{S}, \mathcal{A}, r, p, H \rangle$

algorithm $\mathfrak{A} = \{\pi_k\}_{k=1}^K$

$$R(K,\ M^\star,\ \mathfrak{A}) = \sum_{k=1}^{K} \Big(V^\star(s_{1k}) - V^{\pi_k}(s_{1k}) \Big)$$

policy selected by $\mathfrak{A}$

Let $T = HK$ total number of steps executed in the environment

Alternative Models

- Infinite-horizon undiscounted MDPs (average reward)
  $\Rightarrow$ regret minimization

- Infinite-horizon discounted MDPs
  $\Rightarrow$ PAC-MDPs

$$N(M^\star, \mathfrak{A}) = \sum_{t=0}^{\infty} \mathbb{I}\Big\{V^{\pi_t}(s_t) \leq V^\star(s_t) - \epsilon\Big\}$$

What is Wrong with Q-learning with ϵ-greedy?

- ϵ-greedy strategy

$$a_{hk} = \begin{cases} \arg\max_{a \in \mathcal{A}} Q_{hk}(s_{hk}, a) & \text{w.p. } 1 - \epsilon_{hk}, \\ \mathcal{U}(\mathcal{A}) & \text{otherwise.} \end{cases}$$

- Q-learning update

$$Q_{h,k+1}(s_{hk}, a_{hk}) = (1 - \alpha_t) Q_{hk}(s_{hk}, a_{hk}) + \alpha_t \big(r_{hk} + \max_{a' \in \mathcal{A}} Q_{h+1,k}(s_{h+1,k}, a') \big)$$

What is Wrong with Q-learning with ϵ-greedy?

- ϵ-greedy strategy

$$a_{hk} = \begin{cases} \arg\max_{a \in \mathcal{A}} Q_{hk}(s_{hk}, a) & \text{w.p. } 1 - \epsilon_{hk}, \\ \mathcal{U}(\mathcal{A}) & \text{otherwise.} \end{cases}$$

- Q-learning update

$$Q_{h,k+1}(s_{hk}, a_{hk}) = (1 - \alpha_t) Q_{hk}(s_{hk}, a_{hk}) + \alpha_t \Big(r_{hk} + \max_{a' \in \mathcal{A}} Q_{h+1,k}(s_{h+1,k}, a') \Big)$$

The exploration strategy relies on **biased** estimates Q_{hk}

What is Wrong with Q-learning with ϵ-greedy?

- ϵ-greedy strategy

$$a_{hk} = \begin{cases} \arg\max_{a \in \mathcal{A}} Q_{hk}(s_{hk}, a) & \text{w.p. } 1 - \epsilon_{hk}, \\ \mathcal{U}(\mathcal{A}) & \text{otherwise.} \end{cases}$$

- Q-learning update

$$Q_{h,k+1}(s_{hk}, a_{hk}) = (1 - \alpha_t) Q_{hk}(s_{hk}, a_{hk}) + \alpha_t \big(r_{hk} + \max_{a' \in \mathcal{A}} Q_{h+1,k}(s_{h+1,k}, a') \big)$$

- The exploration strategy relies on **biased** estimates Q_{hk}
- Samples are used **once**

What is Wrong with Q-learning with ϵ-greedy?

- ϵ-greedy strategy

$$a_{hk} = \begin{cases} \arg\max_{a \in \mathcal{A}} Q_{hk}(s_{hk}, a) & \text{w.p. } 1 - \epsilon_{hk}, \\ \mathcal{U}(\mathcal{A}) & \text{otherwise.} \end{cases}$$

- Q-learning update

$$Q_{h,k+1}(s_{hk}, a_{hk}) = (1 - \alpha_t) Q_{hk}(s_{hk}, a_{hk}) + \alpha_t \Big(r_{hk} + \max_{a' \in \mathcal{A}} Q_{h+1,k}(s_{h+1,k}, a') \Big)$$

- The exploration strategy relies on **biased** estimates Q_{hk}
- Samples are used **once**
- **Dithering effect:** exploration is not effective in covering the state space
- **Policy shift:** the policy changes at each step

What is Wrong with Q-learning with ϵ-greedy?

- ϵ-greedy strategy

$$a_{hk} = \begin{cases} \arg\max_{a \in \mathcal{A}} Q_{hk}(s_{hk}, a) & \text{w.p. } 1 - \epsilon_{hk}, \\ \mathcal{U}(\mathcal{A}) & \text{otherwise.} \end{cases}$$

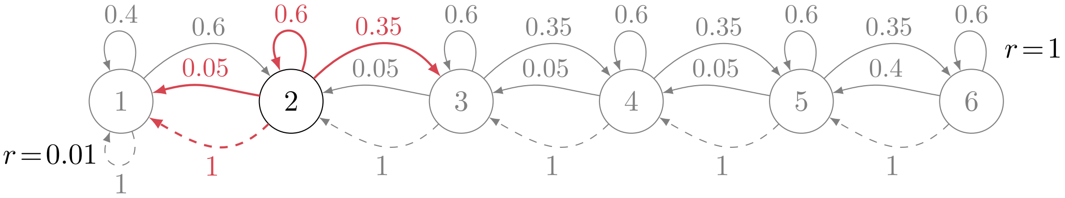
- Q-learning update

$$Q_{h,k+1}(s_{hk}, a_{hk}) = (1 - \alpha_t) Q_{hk}(s_{hk}, a_{hk}) + \alpha_t \Big(r_{hk} + \max_{a' \in \mathcal{A}} Q_{h+1,k}(s_{h+1,k}, a') \Big)$$

- The exploration strategy relies on **biased** estimates Q_{hk}
- Samples are used **once**
- **Dithering effect:** exploration is not effective in covering the state space
- **Policy shift:** the policy changes at each step
- **Regret:** $\Omega\Big(\min\{T, A^{H/2}\}\Big)$ [Jin et al., 2018]

River Swim: Markov Decision Processes

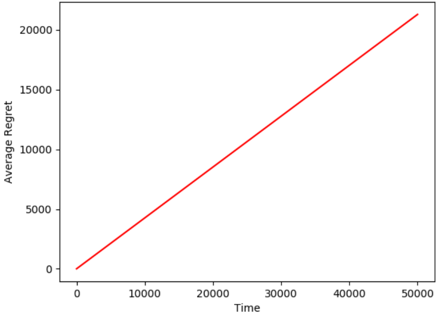
Strehl and Littman [2008]

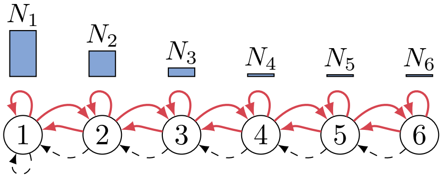


- $\mathcal{S} = \{1, 2, 3, 4, 5, 6\}$, $\mathcal{A} = \{L, R\}$
- $\pi_L(s) = L$, $\pi_R(s) = R$

River Swim: Q-learning w\ ϵ-greedy Exploration

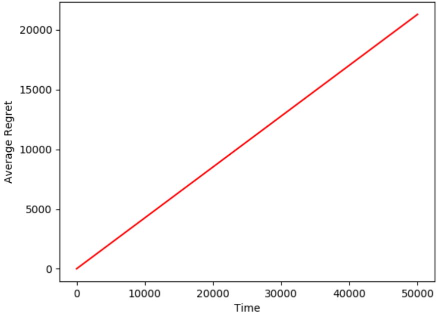
- $\epsilon_t = 1.0$

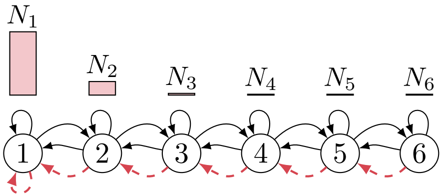




River Swim: Q-learning w\ ϵ-greedy Exploration

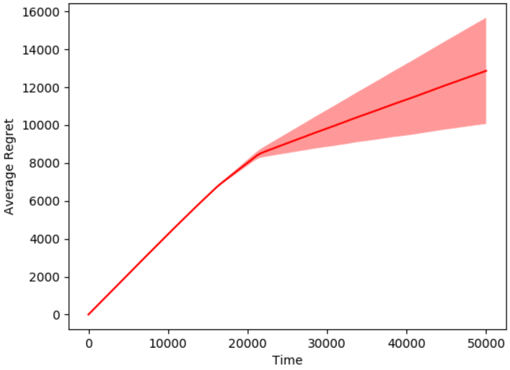
- $\epsilon_t = 1.0$
- $\epsilon_t = 0.5$





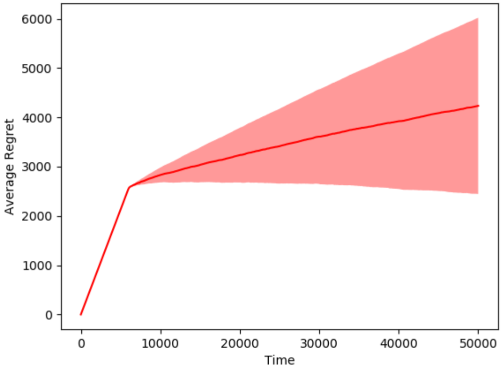
River Swim: Q-learning w\ ϵ-greedy Exploration

- $\epsilon_t = 1.0$
- $\epsilon_t = 0.5$
- $\epsilon_t = \dfrac{\epsilon_0}{(N(s_t) - 1000)^{2/3}}$



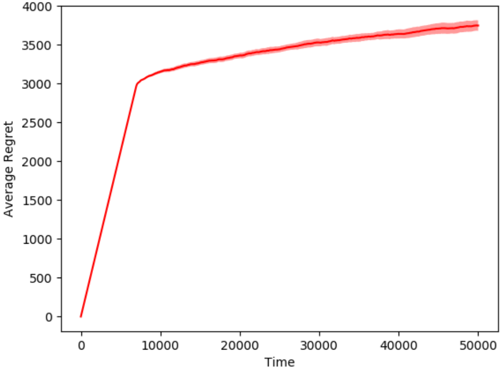
River Swim: Q-learning w\ ϵ-greedy Exploration

- $\epsilon_t = 1.0$
- $\epsilon_t = 0.5$
- $\epsilon_t = \dfrac{\epsilon_0}{(N(s_t) - 1000)^{2/3}}$
- $\epsilon_t = \begin{cases} 1.0 & t < 6000 \\ \dfrac{\epsilon_0}{N(s_t)^{1/2}} & \text{otherwise} \end{cases}$



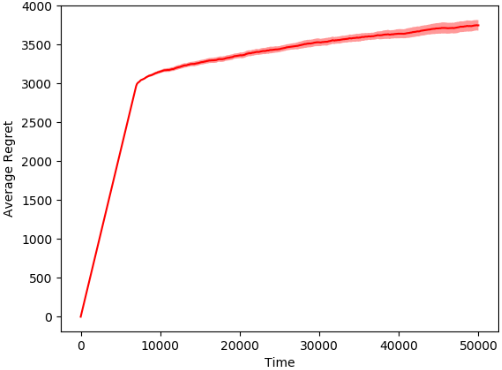
River Swim: Q-learning w\ ϵ-greedy Exploration

- $\epsilon_t = 1.0$
- $\epsilon_t = 0.5$
- $\epsilon_t = \dfrac{\epsilon_0}{(N(s_t) - 1000)^{2/3}}$
- $\epsilon_t = \begin{cases} 1.0 & t < 6000 \\ \dfrac{\epsilon_0}{N(s_t)^{1/2}} & \text{otherwise} \end{cases}$
- $\epsilon_t = \begin{cases} 1.0 & t < 7000 \\ \dfrac{\epsilon_0}{N(s_t)^{1/2}} & \text{otherwise} \end{cases}$



River Swim: Q-learning w\ ϵ-greedy Exploration

- $\epsilon_t = 1.0$
- $\epsilon_t = 0.5$
- $\epsilon_t = \dfrac{\epsilon_0}{(N(s_t) - 1000)^{2/3}}$
- $\epsilon_t = \begin{cases} 1.0 & t < 6000 \\ \dfrac{\epsilon_0}{N(s_t)^{1/2}} & \text{otherwise} \end{cases}$
- $\epsilon_t = \begin{cases} 1.0 & t < 7000 \\ \dfrac{\epsilon_0}{N(s_t)^{1/2}} & \text{otherwise} \end{cases}$



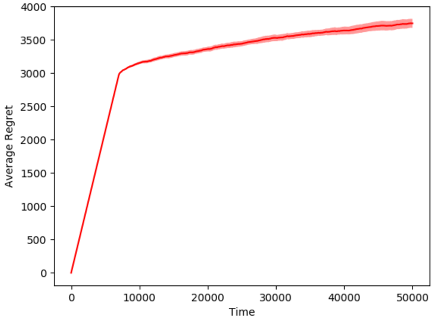
Tuning the ϵ schedule is **difficult and problem dependent**

River Swim: Q-learning w\ ϵ-greedy Exploration

Main drawbacks of Q-learning with ϵ-greedy

- ϵ-greedy performs *undirected* exploration
- *Inefficient use* of samples

Regret: $\Omega\Big(\min\{T, A^{H/2}\}\Big)$

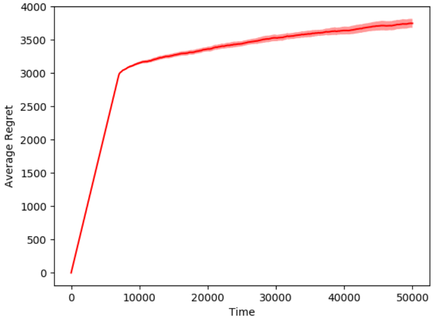


River Swim: Q-learning w\ ϵ-greedy Exploration

Main drawbacks of Q-learning with ϵ-greedy

- ϵ-greedy performs *undirected* exploration
- *Inefficient use* of samples

Regret: $\Omega\Big(\min\{T, A^{H/2}\}\Big)$



Uncertainty-driven exploration-exploitation

Chi Jin, Zeyuan Allen-Zhu, Sébastien Bubeck, and Michael I. Jordan. Is q-learning provably efficient? In *NeurIPS*, pages 4868–4878, 2018.

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

Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences*, 74(8):1309–1331, 2008.