RL for MDP
Chap. 3: Value-based algorithms

Idris Kharroubi

LPSM
Sorbonne Université

DUFE

Outline

- We recall that the **state value function** is given by

$$V_{n,\pi}(x) \quad = \quad \mathbb{E}^{\pi}_{n,x}\Big[\sum_{k=n}^{N-1} r_k(X_k,\alpha_k) + g_N(X_N)\Big]$$

for $x \in E$, $n \leqslant N$ and a strategy π with $\mathbb{E}^{\pi}_{n,x}$ the expectation operator under the conditions $X_n = x$ and $\alpha_k \sim f_k(.|X_k)$ for $k = n, \ldots, N-1$.

- We next define the Q-function by

$$Q_{n,\pi}(x,a) \quad = \quad \mathbb{E}^{\pi}_{n,x,a}\Big[\sum_{k=n}^{N-1} r_k(X_k,\alpha_k) + g_N(X_N)\Big]$$

for $x \in E$, $n \leqslant N$ and a strategy π. Here $\mathbb{E}^{\pi}_{n,x,a}$ is the expectation under conditions $(X_n, \alpha_n) = (x, a)$ and $\alpha_k \sim f_k(.|X_k)$ for $k = n+1, \ldots, N-1$.

- The link between these tow functions is the following

$$V_{n,\pi}(x) \quad = \quad \mathbb{E}_{a\sim f_n(x)}[Q_{n,\pi}(x,a)] \ .$$

- Our goal : to find an optimal policy π^* that maximizes V^π.

$\Rightarrow$ we define the optimal value and Q functions by

$$\begin{aligned} V_n &:= \sup_\pi V_{n,\pi} \ , \\ Q_n &:= \sup_\pi Q_{n,\pi} \ . \end{aligned}$$

- We notice that the link between these two functions is given by

$$V_n(x) \quad = \quad \sup_{a\in A} Q_n(x,a) \ .$$

From **RI** and **Structure** Theorems of Chap. 1, we know that V_π and V satisfy the following Bellman equations

$$\begin{aligned} V_{N,\pi} &= g_N \\ V_{n,\pi}(x) &= \mathbb{E}_{a\sim f_n(x)}\Big[r_n(x,a) + \mathbb{E}_{x'\sim P_n(\cdot|x,a)}\big[V_{n+1,\pi}(x')\big]\Big] \end{aligned}$$

for $n = 0, \dots, N-1$, and

$$\begin{aligned} V_N &= g \\ V_n &= \sup_{a\in A}\Big\{r_n(x,a) + \mathbb{E}_{x'\sim P_n(\cdot|x,a)}\Big[V_{n+1}(x')\Big]\Big\} \end{aligned}$$

for $n = 0, \dots, N-1$.

This leads to the following **Bellman equations** for the Q-functions:

$$\begin{aligned} Q_{N,\pi}(x,a) &= g(x) \\ Q_{n,\pi}(x,a) &= r_n(x,a) + \mathbb{E}_{X' \sim P_n(\cdot|x,a), a' \sim f_{n+1}(x')} \left[Q_{n+1,\pi}(x', a') \right] \end{aligned}$$

for $n = 0, \ldots, N-1$, and

$$\begin{aligned} Q_N(x,a) &= g_N(x) \\ Q_n(x,a) &= r_n(x,a) + \mathbb{E}_{X' \sim P_n(\cdot|x,a)} \left[\sup_{a' \in A} Q_{n+1}(x', a') \right] \end{aligned}$$

for $n = 0, \ldots, N-1$.

Outline

Temporal difference learning (TD for short) is an algorithm that estimates the function V_π by stochastic approximation as follows.

- Initialize at episode $k = 0$ an estimate $\hat{V}_\pi$ of V_π.
- Generate (from real system or simulator) episodic trajectories/reward samples (x, a, r_n, x') from policy π, $n = 0, \ldots, N-1$, and final reward g
- Update estimate $\hat{V}_\pi$ at the next episode by

$$\hat{V}_{n,\pi}(x) \quad \leftarrow \quad (1-\eta)\hat{V}_{n,\pi}(x) + \eta\big(r_n + \hat{V}_{n+1,\pi}(x')\big) \qquad (1)$$

with $\hat{V}_{n,\pi}(x') = g$ when $t = N-1$, and $\eta = \eta_n(x, a) \in (0, 1)$ is the learning rate.

The updating rule (1) can also be written

$$\hat{V}_{n,\pi}(x) \quad \leftarrow \quad \hat{V}_{n,\pi}(x) + \eta\delta_n \tag{2}$$

where δ_n is referred to as TD error and is given by

$$\delta_n \quad = \quad r_n + \hat{V}_{n+1,\pi}(x') - \hat{V}_{n,\pi}(x)$$

This algorithm is usually referred to as TD(0).

There are other variations of TD(0) given by

- TD(1)(Monte-Carlo method) : update value function estimate according to

$$\hat{V}_{n,\pi}(x) \quad \leftarrow \quad \hat{V}_{n,\pi}(x) + \eta \sum_{k=n}^{N-1} \delta_k \tag{3}$$

- More generally TD(λ) for $\lambda \in [0,1]$

$$\hat{V}_{n,\pi}(x) \quad \leftarrow \quad \hat{V}_{n,\pi}(x) + \eta \sum_{k=n}^{N-1} \lambda^{k-n} \delta_k \tag{4}$$

We notice that a choice of λ close to 1 permits a **more accurate** estimation of value function, but induces a **higher variance**.

Outline

Q-learning algorithm consists in a stochastic approximation of the Bellman equation for the Q-function :

$$Q_n(x,a) \quad = \quad r_n(x,a) + \mathbb{E}_{x' \sim P_n(\cdot|x,a)} \left[\sup_{a' \in A} Q_{n+1}(x', a') \right] .$$

Q-learning algorithm can be described by the following steps.

- Initialize at episode $k = 0$ an estimate $\hat{Q}$ of Q.
- Generate trajectories/reward samples (x, a, r_n, x') from a **suitable** policy π.
- Update estimate $\hat{Q}$ at the next episode by

$$\hat{Q}_n(x,a) \quad \leftarrow \quad \hat{Q}_n(x,a) + \eta \delta_n \tag{5}$$

with

$$\delta_n \quad = \quad r_n + \sup_{a' \in A} \hat{Q}_{n+1}(x', a') - \hat{Q}_n(x,a)$$

Under the Robbins-Monro conditions on the learning rate, and if all states are visited infinitely often, then

$$\hat{Q}_n(x,a) \quad \rightarrow \quad Q_n(x,a)$$

for all $(n,x,a) \in \{0,\ldots,N\} \times S \times A$ More precisely, we have the following result from [Watkins&Dayan92]

Theorem

Assume A and S finite. Let $k_i(s,a)$ be the index of the i-th time that the action a is used in state s. Suppose that the reward functions are bounded by some constant R. Given learning rate $\eta_k \in [0,1)$ such that

$$\sum_{i=0}^{+\infty} \eta_{k_i(s,a)} \;=\; +\infty \quad \textit{and} \quad \sum_{i=0}^{+\infty} |\eta_{k_i(s,a)}|^2 \;<\; +\infty$$

for all $(s,a) \in S \times A$ then

$$Q^k(s,a) \quad \xrightarrow[k \to +\infty]{} \quad Q(s,a) \text{ a.s.}$$

Algorithm 1: Q-learning algorithm.

Data: Initialize $\hat{Q}_0(x,a),\ldots,\hat{Q}_N(x,a)$ for $x \in S$ and $a \in A$

Result: Approximated Q-functions.

for $k = 0$ **to** $k = K$ *(number of episodes)* **do**

  Set **initial state** x

  for $n = 0$ **to** $N-1$ *(length of episode)* **do**

    Take action a according to a suitable exploration policy

    Observe reward r_n and next state x'

    Compute temporal difference :

$$\delta_n \quad = \quad r_n + \sup_{a' \in A} \hat{Q}_{n+1}(x',a') - \hat{Q}_n(x,a)$$

    Update Q-function

$$\hat{Q}_n(x,a) \quad \leftarrow \quad \hat{Q}_n(x,a) + \eta\delta_n$$

    $x \leftarrow x'$

Remark

The condition of visiting "Infinitely often" all the states requires a **suitable exploration policy**. *We shall give later an example of such a suitable policy.*

Once the approximation $\hat{Q}$ is computed, we get an approximated optimal policy $\hat{\pi}$ by an off-line computation:

$$\hat{f}_n(x) \quad \in \quad \arg\max_{a \in A} \hat{Q}_n(x, a)$$

for $n = 0, \ldots, N-1$.

Outline

In contrast to Q-learning, SARSA learns policy π^k on-line by iterations of the two following steps :

- **Policy evaluation (PE).** Current estimate by TD stochastic approximation of $Q\pi^k$ satisfying the Bellman equation :

$$Q_{n,\pi^k}(x,a) \quad = \quad r_n(x,a) + \mathbb{E}_{x' \sim P_n(\cdot|x,a), a' \sim f_{n+1}^k(x')} \left[Q_{n+1,\pi^k}(x', a') \right] .$$

- **Policy improvement/iteration (PI).** Compute the greedy policy

$$\hat{f}_n^{k+1}(x) \quad \in \quad \arg\max_{a \in A} \hat{Q}_{n,\pi^k}(x,a)$$

Algorithm 2: SARSA.

Data: Initialize $\hat{Q}_0(x,a),\ldots,\hat{Q}_N(x,a)$ for $x \in S$ and $a \in A$

Result: Approximated Q-functions.

for $k = 0$ **to** $k = K$ *(number of episodes)* **do**

  Set **initial state** x

  for $n = 0$ **to** $N-1$ *(length of episode)* **do**

    Observe reward r_n and next state x'

    Take action a' from PI based on $\hat{Q}$

    Compute temporal difference :

$$\delta_n \quad = \quad r_n + \hat{Q}_{n+1}(x',a') - \hat{Q}_n(x,a)$$

    Update Q-function

$$\hat{Q}_n(x,a) \quad \leftarrow \quad \hat{Q}_n(x,a) + \eta\delta_n$$

    $x \leftarrow x'$

Outline

For a policy, the adjective suitable refers to the tradeoff between exploitation and exploration. There two common randomized suitable policies in the case where A is finite.

ε-**greedy policy.** it consists in taking action according to the following formula

$$\alpha_n \sim \begin{cases} \arg\max_{a \in A} \hat{Q}_n(x, a) & \text{with probability} \quad 1 - \varepsilon \\ \mathcal{U}(A) & \text{with probability} \quad \varepsilon \end{cases}$$

$\varepsilon \in (0, 1)$ is a parameter of exploration in the action space A via the uniform distribution $\mathcal{U}(A)$.

- **Soft-max policy.** $\alpha_n \sim f_n(x)$ where

$$f_n(x,a) \quad = \quad \frac{\exp(Q_n(x,a)/\lambda)}{\sum_{a' \in A} \exp(Q_n(x,a')/\lambda)}$$

The higher $Q_n(x,a)$, the more probability to take action a in state x, and this is weighted by the temperature parameter $\lambda > 0$:

- $\lambda \to +\infty$: one chooses each action with same probability .
- $\lambda \to 0$: greedy policy.

This last policy is actually related to Shannon Entropy problem. Consider indeed the optimization problem

$$\max_{p \in \mathcal{P}(A)} \mathbb{E}_{\alpha \sim p}[Q(\alpha)] + \lambda \mathcal{E}(p) \tag{6}$$

where $\lambda > 0$, and the entropy $\mathcal{E}$ is the strictly concave function on the set $\mathcal{P}(A)$ of probability measures on A defined by

$$\mathcal{E}(\pi) \quad = \quad -\mathbb{E}_{\alpha \sim p}[\log(p(\alpha))] \, .$$

Then the solution to (6) is given by the Soft-max policy

$$p^*(a) \quad = \quad \frac{\exp(Q(a)/\lambda)}{\sum_{a' \in A} \exp(Q(a')/\lambda)} \, , \quad a \in A.$$