Dynamic programming and Markov decision process

Consider a stationary MDP given by $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$

* $\mathcal{S}$ is a finite set of states, $S_t \in \mathcal{S}$ denotes the state at time t.
* $\mathcal{A}$ is a finite set of actions, $A_t \in \mathcal{A}$ is the action taken at time t.
* P is a probability transition kernel: $P^a_{ss'} = \mathbb{P}[S_{t+1} = s' \mid S_t = s, A_t = a]$
* R is the reward function: $\mathcal{R}^a_s = \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a]$
* γ is the discount factor involved in the return: $G_t = \sum_{k \geq 0} \gamma^k R_{t+k+1}, \; 0 < \gamma < 1$

Optimal policies via Bellman optimality equations

value function → $v^*(s) = \max_a \left(\mathcal{R}^a_s + \gamma \sum_{s'} P^a_{ss'} \cdot v^*(s') \right)$

action value function → $q^*(s,a) = \mathcal{R}^a_s + \gamma \sum_{s'} P^a_{ss'} \max_{a'} q^*(s', a')$.

Solving a MDP consists in finding v^*: this problem may be decomposed in a collection of subproblems, whose solutions are used many times to build the full solution (optimal substructure + overlapping problems).

↳ This justifies applying dynamic programming (DP).

Concretely: value iteration algorithm = iterative application of the Bellman optimal operator:

$$v_{k+1}(s) = (T v_k)(s) = \max_a \left(\mathcal{R}^a_s + \gamma \sum_{s'} P^a_{ss'} \cdot v_k(s') \right)$$

Naive implementation involve synchronous back ups, where all states are updated during each sweep. Asynchronous back ups can substantially improve computational cost. Example: in-place value iteration with prioritized ordering according to Bellman error: $|Tv(s) - s| = \mathcal{E}(s)$.

Key limitation:

Dynamic programming methods assume knowledge of the environment dynamics. Monte Carlo methods offer to use sampling in order to improve policies without full knowledge of the environment. The central problem is then to balance exploitation and exploration.

Monte-Carlo method

improvement : greedy move (deterministic)

Policy iteration : $\pi_0 \xrightarrow{E} v_{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} v_{\pi_1} \xrightarrow{I} \pi_2 \dots$

evaluation : expectation back up (probabilistic)

Sampling idea : the value function is the expected return conditioned on a starting state. Monte Carlo methods replace expectation by sampling means during the evaluation stage, assumed to last for a full episode (many samples)

Monte-Carlo and policy iteration coincides under assumption 1) of exploring starts (each action-state pair has non zero probability to be chosen at the start of an episode) and under assumption 2) of an infinite number of samples per episode.

Assumption 2) can be relaxed naturally as in value iteration.

Assumption 1) is relaxed via two distinct approaches : "on-policy" and off-policy approach.

On-policy approach : the same policy π is used for evaluation and control, but the policy is soft : $\min_a \pi(a|s) \geq \varepsilon/|A|$.

Example : ε-greedy policy π' select a random action uniformly with probability ε and the greedy action with probability $1-\varepsilon$.

$$q_\pi(s,\pi'(s)) = \sum_a \pi'(a|s)\, q_\pi(s,a) = \frac{\varepsilon}{|A|} \sum_a q_\pi(s,a) + (1-\varepsilon) \max_a q_\pi(s,a)$$

$$\geq \frac{\varepsilon}{|A|} \sum_a q_\pi(s,a) + (1-\varepsilon) \sum_a \frac{\pi(s,a) - \varepsilon/|A|}{1-\varepsilon}\, q_\pi(s,a) = v_\pi(s)$$

Off-policy approach : distinct policies π and π' are used for evaluation and control. Typically π can be deterministic while π' ensures exploration.

$$v_\pi(s) \simeq \frac{1}{\sum_{i=1}^{n_s} \frac{p_i(s)}{p'_i(s)}} \sum_{i=1}^{n_s} \frac{p_i(s)}{p'_i(s)} R_i(s)$$

n_s : number of visit to s in an episode

$R_i(s)$: observed return from state s

$p_i(s)$: probability of a sequence path after the i-th visit to s

$$\frac{p_i(s)}{p'_i(s)} = \frac{\prod_{k=t_i}^{T-1} \pi(a_k|s_k)\, p^{a_k}_{s_k s_{k+1}}}{\prod_{k=t_i}^{T-1} \pi'(a_k|s_k)\, p^{a_k}_{s_k s_{k+1}}} = \prod_{k=t_i}^{T-1} \frac{\pi(a_k|s_k)}{\pi'(a_k|s_k)}$$

only depends on policy ratios

Temporal difference algorithm

Monte Carlo methods differ from DP methods because:

1) they operate on sample experience: no need for environment model
2) they do not update their value estimates on the basis of other value estimates, i.e. they do not bootstrap (as in DP).

Temporal difference learning, perhaps the central original idea of reinforcement learning (w.r.t. control theory) combines Monte Carlo and DP approaches.

Central idea: bootstrapping should allow for online policy updates, e.g., avoid waiting for the end of an episode to perform evaluation. A simple, every visit Monte Carlo scheme for evaluation is

instantaneous reward used as target = no bootstrapping

$$v_\pi(S_t) \leftarrow v_\pi(S_t) + \alpha \left(R_t^\pi - v_\pi(S_t) \right), \quad \alpha: \text{fixed learning rate}$$

Alternative:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi \left[G_t \mid S_t = s \right] \\ &= \mathbb{E}_\pi \left[\sum_{k \geq 0} \gamma^k R_{t+k+1} \mid S_t = s \right] \\ &= \mathbb{E}_\pi \left[R_{t+1} + \gamma G_{t+1} \mid S_t = s \right] \\ &= \mathbb{E}_\pi \left[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s \right] \end{aligned}$$

bootstrapping = using a target that depends on other estimates in the Monte-Carlo scheme

TD(0) rule:

$$\boxed{v_\pi(S_t) \leftarrow v_\pi(S_t) + \alpha \left[R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \right]}$$

Example of on-policy TD method: SARSA. For each episode

1) given current guess for $q(s,a)$, initialize s and choose a ε-greedily
2) observe r and s', choose a' ε-greedily given s'
3) update $q(s,a) \leftarrow q(s,a) + \alpha \left[r + \gamma q(s',a') - q(s,a) \right]$.
4) $s \leftarrow s'$, $a \leftarrow a'$ ...

Q-learning = off-policy TD algorithm (Watkins, 1989)

At the core of Q-learning is a simple value iteration update. That simplicity allows for simple convergence proof. SARSA is on policy because update can only be made on the basis of actions that are taken. Q-learning updates on the basis of the best available action possible.

For each episode:

1) initialize s

2) repeat for each step of the episode

   a) choose a given s ε-greedily

   b) take action a and observe s' and r

   c) $q(s,a) \leftarrow q(s,a) + \alpha\left[r + \gamma \max_{a'} q(s',a') - q(s,a)\right]$ ← best possible action

   d) $s \leftarrow s'$

3) terminate when halting condition is met.

Theorem: Given a finite MDP (S, A, P, R, γ), the Q-learning algorithm given by the update rule:

$$q_{t+1}(s_t,a_t) = q_t(s_t,a_t) + \alpha_t(s_t,a_t)\left[r_{t+1} + \gamma \max_a q_t(s_{t+1},a) - q_t(s_t,a_t)\right]$$

converges with probability 1 to the optimal action-value function as long as:

$$\left.\begin{aligned} &1) \sum_{t\geq 0} \alpha_t(s,a) = +\infty \\ &2) \sum_{t\geq 0} \alpha_t^2(s,a) < +\infty \end{aligned}\right\} \text{for all } (s,a)$$

Sketch of the proof: Rewrite the update rule as

$$q_{t+1}(s_t,a_t) = (1-\alpha_t)\, q_t(s_t,a_t) + \alpha_t\left[r_{t+1} + \gamma \max_a q_t(s_{t+1},a)\right]$$

* Define $\Delta_t(s,a) = q_t(s,a) - q^*(s,a)$ where q^* is the optimal value function. Then the update rule reads

$$\Delta_{t+1}(s_t,a_t) = (1-\alpha_t)\,\Delta_t(s_t,a_t) + \alpha_t\left[r_{t+1} + \gamma \max_a q(s_{t+1},a) - q^*(s_t,a_t)\right]$$

* The goal is to show that Δ_t converges to zero with probability one. The idea is the $F_t = r_{t+1} + \gamma \max_a q(s_{t+1},a) - q^*(s_t,a_t)$ represents a random error that can be controlled. If the amplitude of the error are small enough, the relaxation term can confine Δ_t to zero. However α_t must not decrease to zero too fast for the relaxation to be strong enough.

* These ideas can be made precise in the context of stochastic approximation theory. Namely we have (Dvoretzky 1956)
The random process Δ_t taking value in $\mathbb{R}^n$ and defined as

$$\Delta_{t+1}(s) = (1-\alpha_t(s))\,\Delta_t(s) + \alpha_t(s)\,F_t(s) \leftarrow \text{random noise}$$

converges to zero with probability one if:

1) $0 \le \alpha_t \le 1$ $\quad \sum_{t\ge 0} \alpha_t(s) = +\infty$ and $\sum_{t\ge 0} \alpha_t^2(s) < +\infty$

2) $\mathbb{E}\left[F_t(s) \mid \mathcal{F}_t\right] \le \gamma \|\Delta_t\|_\infty$, $\gamma < 1$

3) $\mathbb{V}\left[F_t(s) \mid \mathcal{F}_t\right] \le C\left(1 + \|\Delta_t\|_\omega^2\right)$, $C \ge 0$

* We only need to check 2) and 3). We will justify why 1) is a natural assumption heuristically.

First observe q^* is the fixed point of the contraction operator $(Tq)(s,a) = R_s^a + \gamma \sum_{s'} P_{ss'}^a \max_{a'} q(s',a')$

Indeed, we have:

$$|Tq(s,a) - Tq'(s,a)| = \gamma \left| \sum_{s'} P^a_{ss'} \left(\max_{a'} q(s',a') - \max_{a'} q'(s',a') \right) \right|$$

$$\leq \gamma \sum_{s'} P^a_{ss'} \left| \max_{a'} q(s',a') - \max_{a'} q'(s',a') \right|$$

triangular ineq. $\rightarrow$ $$\leq \gamma \sum_{s'} P^{a'}_{ss'} \max_{a'} |q(s',a') - q'(s',a')|$$

$$\leq \gamma \sum_{s'} P^a_{ss'} \|q - q'\|_\infty = \gamma \|q - q'\|_\infty$$

Thus, we can evaluate:

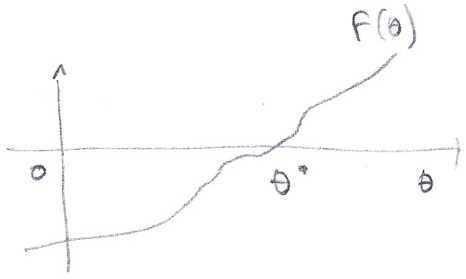
$$\left| \mathbb{E}\left[F_t(s,a) \mid \mathcal{F}_t \right] \right| = \left| R^a_s + \gamma \sum_{s'} P^a_{ss'} \max_{a'} q(s',a') - q^*(s,a) \right|$$

$$= \left| Tq(s,a) - Tq^*(s,a) \right| \leq \gamma \|q - q^*\|_\infty = \gamma \|\Delta_t\|_\infty$$

This establishes 2). 3) immediately follows from boundedness of the reward, which we assume.

Heuristic justification of 1)

Consider a non-decreasing function:



Goal: finding a sequence of estimate of the root θ^* when one has only access to noisy values.

Without noise: Newton-Raphson $\theta_{n+1} = \theta_n - \frac{f(\theta_n)}{f'(\theta_n)}$

or $\theta_{n+1} = \theta_n - \alpha f(\theta_n)$ for α small enough

Idea: Can one come up with a sequence $\theta_n \rightarrow \theta^*$ by a judicious choice of coefficient α_n such that

$$\theta_{n+1} = \theta_n - \alpha_n \left(f(\theta_n) - \gamma_n \right), \quad \gamma_n \text{ i.i.d noise.}$$

Intuition: If f is linear, averaging the noise should be beneficial. This corresponds to taking $\alpha_n = \frac{1}{n+1} \Rightarrow \theta_n = \frac{\sum_{i=1}^{n} \gamma_i}{n}$

If $\alpha_n = \frac{1}{n+1}$, one has $\sum_{n \geq 0} \alpha_n = +\infty$ and $\sum_{n \geq 0} \alpha_n^2 \leq +\infty$.

How general is this observation?

↳ consider $\theta_{n+1} = \theta_n - \alpha_n (\theta_n - \xi_n)$

$$= (1-\alpha_n)\theta_n + \alpha_n \xi_n$$

$$= (1-\alpha_n)(1-\alpha_{n-1})\theta_{n-1} + \alpha_n \xi_n + (1-\alpha_n)\alpha_{n-1}\xi_{n-1}$$

$$\vdots$$

$$= \underbrace{\prod_{i=1}^{n} (1-\alpha_i)\theta_0}_{\text{deterministic error}} + \underbrace{\sum_{i=1}^{n} \prod_{k=i+1}^{n} (1-\alpha_k)\alpha_i \xi_i}_{\text{noise error}}$$

for $\theta_n \xrightarrow{L^2} \theta^*$, we must have 1) $\lim_{n \to +\infty} \prod_{i=1}^{n} (1-\alpha_i)^2 = 0$

2) $\lim_{n \to +\infty} \sum_{i=1}^{n} \prod_{k=i+1}^{n} (1-\alpha_k)^2 \alpha_i^2 = 0$

1) is equivalent to $2 \sum_{i=1}^{n} \log(1-\alpha_i) \sim -2 \sum_{i=1}^{n} \alpha_i \xrightarrow[n \to +\infty]{} -\infty$

2) follows from: $\exists N_\varepsilon$, $n \geq N_\varepsilon \Rightarrow \sum_{i=n}^{+\infty} \alpha_i^2 < \varepsilon/2$

Thus taking α_i in $(0,1)$, we have, for $n \geq N_\varepsilon$

$$\left| \sum_{i=1}^{n} \prod_{k=i+1}^{n} (1-\alpha_k)^2 \alpha_i^2 \right| \leq \left| \sum_{i=1}^{N_\varepsilon} \prod_{k=i+1}^{n} (1-\alpha_k)^2 \alpha_i^2 \right| + \sum_{i=N_\varepsilon}^{+\infty} \alpha_i^2$$

$$\leq N_\varepsilon \underbrace{\prod_{i=N_\varepsilon}^{n} (1-\alpha_k)^2}_{\xrightarrow[n \to +\infty]{} 0} + \varepsilon/2 \leq \varepsilon/2$$