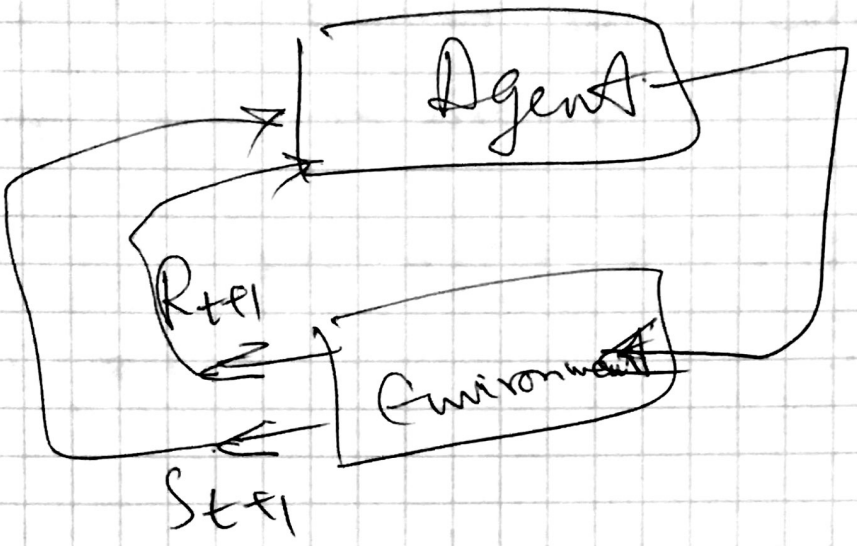
Ch. 3 Notes



$S_0, A_0, R_1, S_1, A_1, S_2, A_2, R_3 \cdots$

Finite MDP: sets of states, actions and rewards all have a finite number of elements

$$p(s', r \mid s, a) \doteq \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_{t-1} = a\}$$

↳ dynamics, model, plant (control)

people.

$$p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$$

We use R_{t+1} instead of R_t to denote reward due to A_t because it emphasizes that the

next reward and next state (R_{t+1}, S_{t+1}) are jointly determined. Both conventions are widely used in literature.

$$\sum_{s' \in S} \sum_{r \in R} p(s', r | s, a) = 1, \text{ for all } s \in S, a \in A(s)$$

* In an MDP once you know p, you can completely characterize the whole MDP. Where you go next is completely determined by where you are and actions you take. (Markov property)

* From the 4-argument dynamics function p, one can compute state-transition probabilities

$$p : S \times S \times A \to [0, 1]$$

$$p(s' | s, a) \doteq Pr\{S_t = s' | S_{t-1} = s, A_{t-1} = a\}$$
$$= \sum_{r \in R} p(s', r | s, a)$$

* Expected rewards for s-a pairs.

$$r : S \times A \to R.$$

$$r(s,a) \doteq E\left[R_t \mid S_{t-1} = s, A_{t-1} = a\right]$$

* Want to maximize return. (expected)

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} \cdots R_T.$$

T = final time step.

Episodic: When agent-environment interaction naturally breaks down into subsequences.

Usually we have some terminal state.

T can vary from episode to episode.

Continuing tasks:

No natural breakdown.

Lifelong learning.

This is usually problematic mathematically as $T = \infty$, and return could itself easily be ∞.

So we introduce discounting:-

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$

$$= \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

$0 \leq \gamma \leq 1$

γ discount rate.

$\{R_k\}$ should be bounded as well.

★ Value functions

Value of a state $V(s)$

$\doteq$ how good is it for the agent to be in that state.

"how good" = future rewards that can be expected.

But future rewards are consequences of actions taken in the future as well. → Policy → way of acting.

→ mapping from s to a.

$\pi(s) \rightarrow a$. (Deterministic)

$\pi(a|s)$ = prob. of taking a when in s. (stochastic)

$$v_\pi(s) \doteq \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right]$$

for all $s \in \mathcal{S}$.

$$q_\pi(s,a) \doteq \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t=s, A_t=a\right]$$

↑ action-value function for π.

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t=s]$$

$$= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t=s]$$

$$= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a) \cdot \left[r + \gamma \mathbb{E}_\pi[G_{t+1} \mid S_{t+1}=s']\right]$$

$$= \sum_a \pi(a|s) \sum_{s', r} p(s', r|s, a)\left[r + \gamma v_\pi(s')\right]$$

for all $s \in S$.

* Solving RL means finding a good policy.

~~best~~

Good policy ⇒ good cumulative returns.

$\pi \geq \pi'$ iff $v_\pi(s) \geq v_{\pi'}(s)$, $\forall s \in S$.

π_* = set of all optimal policies

(there may be more than 1)

$$v_*(s) \doteq \max_\pi v_\pi(s), \quad \forall s \in S.$$

$$q_*(s,a) \doteq \max_\pi q_\pi(s,a)$$

$$q_*(s,a) = \mathbb{E}_{\substack{a \sim \pi^*(a|s) \\ s' \sim p(s'|s,a)}} \left[r(s,a) + \gamma v_*(s') \right]$$

$$v_*(s) = \max_{a \in A(s)} q_{\pi_*}(s,a)$$

$$= \max_a \mathbb{E}_{\pi_*}\left[G_t \mid S_t = s,\ A_t = a\right]$$

$$= \max_a \mathbb{E}_{\pi_*}\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a\right]$$

$$= \max_a \mathbb{E}\left[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s,\ A_t = a\right]$$

Bellman Optimality Condition for Value Functions.

$$= \max_a \sum_{s', r} p(s', r \mid s, a)\left[r + \gamma v_*(s')\right]$$

$$q_*(s,a) = \mathbb{E}\left[R_{t+1} + \gamma \max_{a'} q_*(S_{t+1}, a') \mid S_t = s,\ A_t = a\right]$$

$$= \sum_{s', r} p(s', r \mid s, a)\left[r + \gamma \max_{a'} q_*(s', a')\right]$$

* For finite MDPs, Bellman optimality equation has a unique solution.
  System of 'n' equations in 'n' unknowns.
  (If dynamics p of the environment are known)

* If you have v_* just take 1-step greedy wrt it. You get optimal policy!

* If you have q_* life is very easy. don't even have to do 1-step search
  Just pick $\arg\max_a q_*(s,a)$.
  The q-function caches the results of all 1-step queries. More memory but allows optimal actions to be selected without having to know dynamics. (possible next states and their values.)

* other ways of writing V, Q (more common)

* $V_\pi(s) = r(s, \pi(s)) + \gamma E_{P(s'|s,\pi(s))}[V(s')]$

* $Q_\pi(s,a) = r(s,a) + \gamma E_{P(s'|s,a)}[\max_{a'} Q_\pi(s,a')]$

$$= r(s,a) + \gamma E_{P(s'|s,a)}[\max_{a'} Q_{\pi_1}(s',a')]$$

* $Q(s,a) \leq V(s)$

* $V(s) = \max_a Q(s,a)$

* $\pi^* = \arg\max_a Q^*(s,a)$

$\pi = \arg\max_a Q(s,a)$