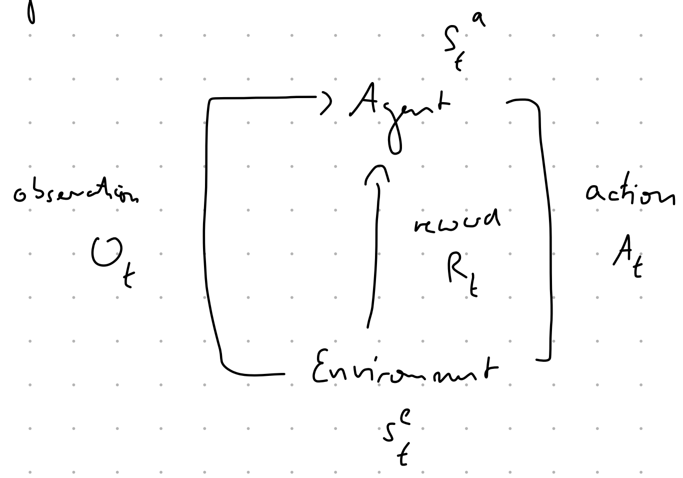
2019/11/20

What is RL?

- Agent & Environment



$\mathbb{P}[S_{t+1} \mid S_t]$
$= \mathbb{P}[S_{t+1} \mid S_1, \dots S_t]$

- RL agent components
  - ↳ policy
  - ↳ value function
  - ↳ model → model-based; model-free
  - ↳ prediction & control

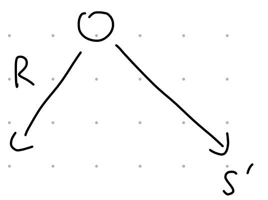
- Markov decision processes (MDP)
  - ↳ (S, P, R, A, γ) — S: states; P: state transition probability; R: reward $R_s = \mathbb{E}[R_{t+1} \mid S_t = s]$; A: actions; γ: discount factor $\in [0,1]$
  - ↳ return $G_t = R_{t+1} + \gamma R_{t+2} + \dots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$
  - ↳ state-value function

$$V(s) = \mathbb{E}[G_t \mid S_t = s] = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s\right]$$
$$= \mathbb{E}[R_{t+1} + G_{t+1}] = \mathbb{E}[R_{t+1} + \gamma V(S_{t+1}) \mid S_t = s]$$

Bellmann equation

Nice fact: Bellmann equation is recursive & linear

$V(S) \longleftarrow S$



$V(S') \longleftarrow S'$

$$V(S) = R_{t+1} + \gamma \sum_{s' \in S} P_{ss'} V(s')$$

↳ policy π:

$$\pi(a|s) = \mathbb{P}[A_t = a \mid S_t = s]$$

$$V_\pi(s) = \mathbb{E}_\pi [G_t \mid S_t = s]$$ value - function

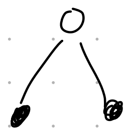
$$q_\pi(s,a) = \mathbb{E}_\pi [G_t \mid S_t = s, A_t = a]$$ action - value function

↳ return to Bellmann equation

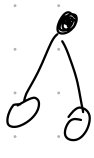
$$V_\pi(s) = \mathbb{E}_\pi [R_{t+1} + \gamma V_\pi(S_{t+1}) \mid S_t = s]$$

$$q_\pi(s,a) = \mathbb{E}_\pi [R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a]$$

$V_\pi(s) \longleftarrow s$



$q_\pi(s,a)$ a



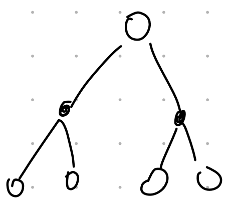
$s,a \rightarrow q_\pi(s,a)$

$s' \rightarrow V_\pi(s')$

$$V_\pi(s) = \sum_{a \in A} \pi(a|s)\, q_\pi(s,a) \qquad q_\pi(s,a) = R^a_{t+1} + \gamma \sum_{s'} P_{ss'} V_\pi(s')$$

$$= R_{t+1} + \gamma V_\pi(s')$$

⇓ combine



$s \rightarrow V_\pi(s)$

a

$s' \rightarrow V_\pi(s')$

$$V_\pi(s) = \sum_a \pi(a|s) \left(R^a_s + \gamma \sum_{s' \in S} P^a_{ss'} V_\pi(s') \right]$$

ex: $\left(R^a_s + \gamma V_\pi(s') \right]$

↳ optimality

$$V_*(s) = \max_\pi (V_\pi(s))$$

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

π_* for any MDP

1) there is always an optimal policy π^*

2) all optimal policies achieve the V_* and q_*

$$\pi_*(a|s) = \begin{cases} 1 & \text{if } a = \arg\max_a q_*(s,a) \\ 0 & \text{else} \end{cases}$$

Dynamic programming

1) optimal substructure
2) overlapping subproblems

} MDP ✓ because of Bellman equation

- model-based (update based on estimate)
- bootstraps

Monte Carlo

value = empirical mean return

- model-free (no knowledge of environment required)
- sampling method
- inherently slow (one episode ≙ one sample)

TD learning

- model free
- bootstraps
- online update

$$V(S_t) \leftarrow V(S_t) + \overset{\text{learning rate}}{\alpha} \underbrace{\Big(\underbrace{R_{t+1} + \gamma V(S_{t+1})}_{\text{target}} - V(S_t) \Big)}_{\delta_t}$$

$\Downarrow$

prediction problem

Control problem

on-policy (on the job) vs off-policy (from observing)

↳ learn about policy π from experience sampled from π

Ex: SARSA

- apply TD to $q(s, a)$
- every time step, evaluate & improve policy