(18) RL framework: The solution

↓

once a problem is properly defined, the solution can be thought of

↓

solution to a RL problem can be considered as a series of actions that need to be learned (or taken) by the agent in order to maximize the reward signal.

↓

these actions depend on the state of the task and hence being able to come up with the best possible action for a given state, can be considered to be a solution to the RL problem.

↓

motivates the idea of policy i.e. mapping b/w the states and actions.

↓

obviously, there will be many possible, but not all will be optimal

→ finding this is the key requirement

↓

⇒ optimal policy for a RL problem is considered as its solution

policy

Deterministic policy

an action for state

$\pi : S \to A$

π (low) → recharge

π (high) → search

stochastic policy

probability based

probability of taking a certain action if a given state is encountered

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

$\pi(a/s) = p(A_t = a / S_t = s)$

π (recharge/low) = 0.5

π (wait/low) = 0.4

π (search/low) = 0.1

(20)

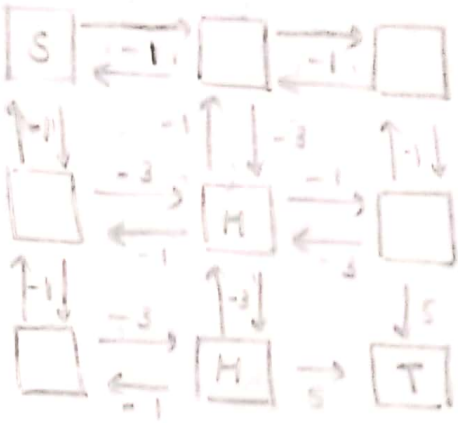
* Finding the optimal policy

↓

based on grid world example

↓

agent needs to reach the target location T, starting at some location in the grid



↓

episodic : ends with T is reached

↓

state space : grid cells

action space : movement to adjacent cells (state dependent of course)

↓

reward structure : −1 : non terminal state transition (except ones into holes/mountains)

−3 : for mountain state transition

+5 : terminal transition

$\gamma \rightarrow 1$ (no discounting)

⇓

we start with a bad policy (presumably)

↓

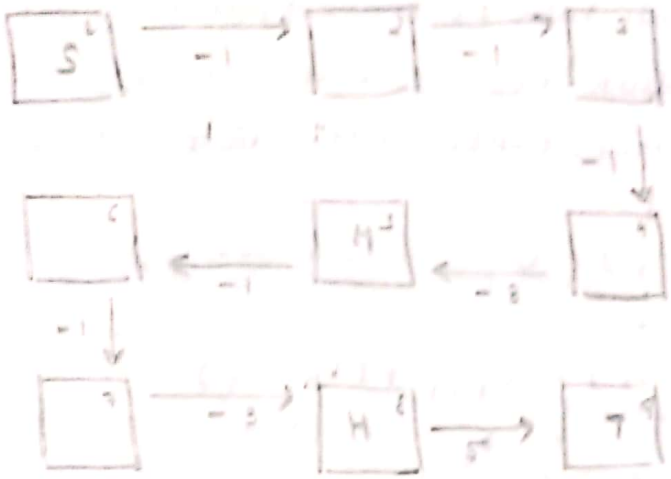
try to understand why it is bad

↓

then try to improve it.

⇓

to try to understand why a policy is bad, we start from the initial state and find the cumulative reward, if we move following the given policy



approximation only
how do we know if it's good?
CR = -6
better
policy → action
value (to improve policy)
state value function (to compare with other policies)

State value function:

The process is repeated for the cell right adjacent to S, and then get CR = -5, and all cells soon.

↓

for terminal state, R = 0, since if agent starts there, the episode ends straight away.

↓

Accordingly, we have a number associated to each state, this number is called the output of the state value function.

↓

state value function yields the reward expected if the agent were to start at that position & follow the given policy.

↓

$$v_\pi(s) = E_\pi \left[\underset{s_t = s}{G_t} \right]$$ → Mathematical notation for state value function

↓

** for each state s, it yields the expected discounted return if the agent starts in state (s) and then uses the policy to choose its actions for all time steps.

(21) State value function have a recursive property where the state value function output is equal to the immediate reward added to the discounted reward of the next state ($\gamma = 1$)

$\downarrow$

$v_\pi(s_9) = 0$ (since terminal)

$\downarrow$

$v_\pi(s_8) = r_{t+1} + \gamma v_\pi(s_9)$

$= 5$

$\downarrow$

$v_\pi(s_7) = r_{t+1} + \gamma v_\pi(s_8)$

$= -3 + 1(5) = 2$

$\downarrow$

$v_\pi(s_6) = r_{t+1} + \gamma v_\pi(s_7)$

$= 2 + 1(-1) = 1$

$\downarrow$

$v_\pi(s_1) = r_{t+1} + \gamma v_\pi(s_2)$

$= -1 + 1(-5) = -6$

$\downarrow$

$\therefore v_\pi(s) = E_\pi \left[\underline{R_{t+1} + \gamma v_\pi(s+1)} \right]$

$S_t = s$

$\downarrow$

Bellman equation

$\downarrow$

since the next state on taking a certain action depends on one step dynamics and our state value is dependent on the next state, then it should be accounted in the expected reward. and the corresponding rewards

* Also the action taken at a given state, which eventually decides the next possible state, may vary depending on the type of policy which can be either deterministic or stochastic.

↳ so that needs to be taken into account as well

i) deterministic → one action per state

↓

now the probability factor comes in for the resulting state, reward for that action, which is again given by the one step dynamics of the environment

future reward → depends on the state, action pair and that is also dependent on the type of policy.

$$v_\pi(s) = \sum_{\substack{s' \in S^+, \\ r \in R}} p\left(\underset{s, \pi(s)}{s', r}\right)\left(r + \gamma v_\pi(s')\right)$$

$v_\pi(s)$ → state value for current

values as weighted sum of possible next states given the current state

one step dynamics.

↓

probability of s' with corresponding reward when current state is s and action $a \to \pi(s)$, which is deterministic

ii) stochastic → probability based action, given a state, adds another probability factor for selection of action.

↓

once action is selected, the further process is same.

↓

(weighted sum)

reward for (s, a)

$$v_\pi(s) = \sum_{\substack{s' \in S^+, \\ r \in R, \\ a \in A(s)}} \pi(a/s)\, p\left(\underset{s, \pi(s)}{s', r}\right)\left(r + \gamma v_\pi(s')\right)$$

$\pi(a/s)$ → prob of action

$p\left(\underset{s, \pi(s)}{s', r}\right)$ → prob of new state for the action

(24) * Comparing policies

↓

given two policies π_1 and π_2, we can say that

π_1 is better than π_2, if $v_{\pi_1}(s) > v_{\pi_2}(s)$

for all $s \in S$.

↓

in some situations, it might be hard to compare policies.

↓

But an optimal policy will always be available for a problem, denoted by v_*

↓

next problem is, how do we determine this better policy from bad policy

↓

this is done through action value function

Action value function

↓

value of taking action a in state s under policy π

↓

If action a is taken at state s, then what is the reward, if the policy π is followed, for the next state.

↓

a value for each possible action in a state s.

↓

$$q_\pi(s,a) = E_\pi \left[\frac{G_t}{S_t, A_t} \right]$$

(S_t → s, A_t → a)

↓

By choosing the best action value per state, we move from q_* to π_*