Lecture II : Markov Decision Process

26/08/19

Markov Process (MP)

- formal description of a RL environment
- environment is fully observable
- current state completely characterizes the problem
- partially observable problems can be converted to MDP

eg: bandit problems ← MP with one state

The essential idea of a MP is :

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

↳ then S_t is a Markov state

ie. the state is a sufficient statistic of the future

State transition probability :

$$P_{ss'} \doteq P[S_{t+1} = s' \mid S_t = s]$$

State transition matrix :

$$P = \begin{matrix} \\ \text{FROM} \end{matrix} \overset{\text{TO}}{\begin{bmatrix} P_{11} & \cdots & P_{1n} \\ \vdots & & \\ P_{n1} & \cdots & P_{nn} \end{bmatrix}}$$

; each row sums to unity

Thus, formally, a Markov Process is a tuple $\langle S, P \rangle$

- S : finite set of states
- P : state transition probability matrix

Markov Reward Process:

- a tuple $\langle S, P, R, \gamma \rangle$
- R : immediate reward, $R_s = \mathbb{E}[R_{t+1} \mid S_t = s]$
- γ : discount factor

$$\text{Return}, G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$; no expectation here as it is just for one random sample from the MRP

$\gamma = 0$: maximally short sighted

$\gamma = 1$: maximally far sighted

Why discount?

- mathematically convenient
- avoid infinite return in cyclic Markov processes
- uncertainty about future may not be fully represented
- may earn more from immediate rewards (say in finance)
- undiscounted MRP ($\gamma = 1$)

Value function:

- long term value of a state s

$$v(s) = \mathbb{E}[G_t \mid S_t = s]$$

↑ as it is a stochastic process

Bellman Equation for MRP:

$$
\begin{aligned}
v(s) &= \mathbb{E}[G_t \mid S_t = s] \\
&= \mathbb{E}[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\
&= \mathbb{E}[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s]
\end{aligned}
$$

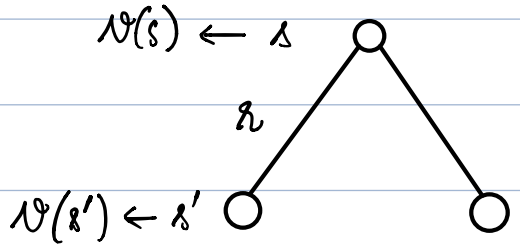
$\Updownarrow$: law of iterated expectations

R_{t+1} : immediate reward

$\gamma v(S_{t+1})$: discounted value of successor state

Reward for state S_t is obtained at time step $t+1$

↳ convention to separate agent environment interface



backup diagrams

single step lookahead search

$$
\therefore \quad v(s) = R_s + \gamma \sum_{s' \in S} P_{ss'} v(s')
$$

Bellman Equation in Matrix form:

$$
V = R + \gamma P V
$$

column vectors

$$
\text{or,} \quad (I - \gamma P) V = R
$$

$$
\text{or,} \quad V = (1 - \gamma P)^{-1} R
$$

- $O(n^3)$ for n states
- iterative methods : DP, Monte Carlo evaluation, TD learning

↖ more efficient

Markov Decision Process:

- tuple $\langle S, A, P, R, \gamma \rangle$
- A: finite set of actions
- P: $P^a_{ss'} = Pr[S_{t+1} = s' \mid S_t = s, A_t = a]$
- R: $R^a_s = E[R_{t+1} \mid S_t = s, A_t = a]$

Policy: $\pi(a|s)$: distribution of actions over states

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

- it defines the behaviors of an agent
- policies depend on current state only
- policies are stationary

State Value function: $(v_\pi(s))$

$$v_\pi(s) = E_\pi[G_t \mid S_t = s]$$

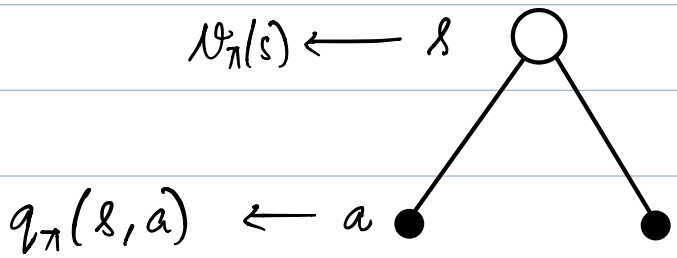
Action Value function: $(q_\pi(s,a))$

$$q_\pi(s,a) = E_\pi[G_t \mid S_t = s, A_t = a]$$

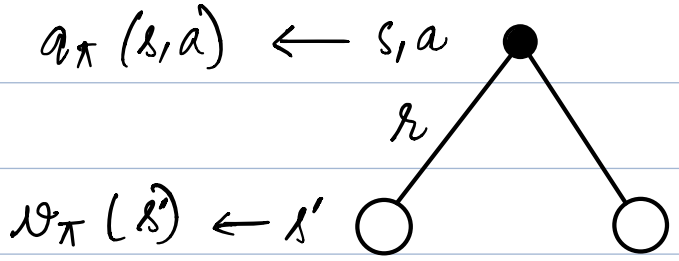
Bellman Expectation Equation:

$$v_\pi(s) = E_\pi[R_{t+1} + \gamma v_\pi(S_{t+1}) \mid S_t = s]$$

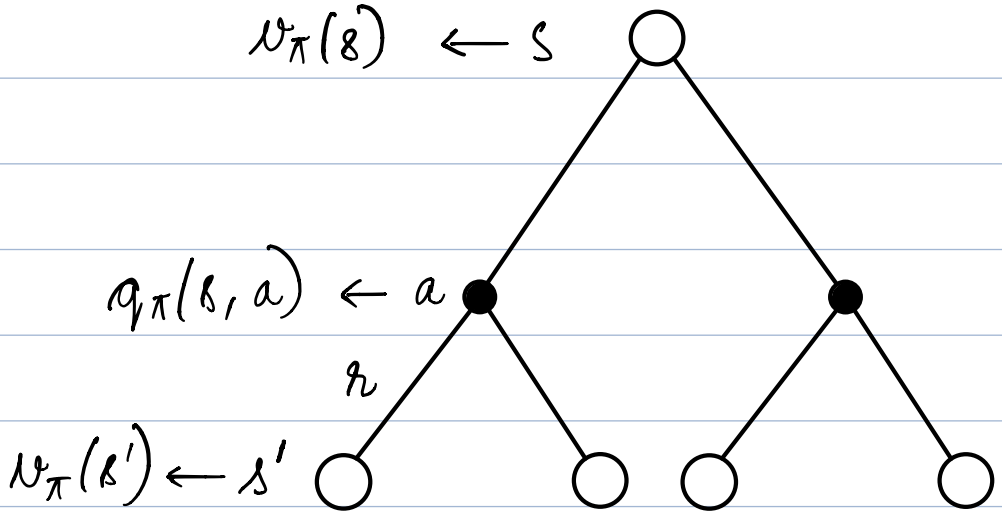
$$q_\pi(s,a) = E_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) \mid S_t = s, A_t = a]$$



$$v_\pi(s) = \sum_{a \in A} \pi(a|s)\, q_\pi(s,a)$$

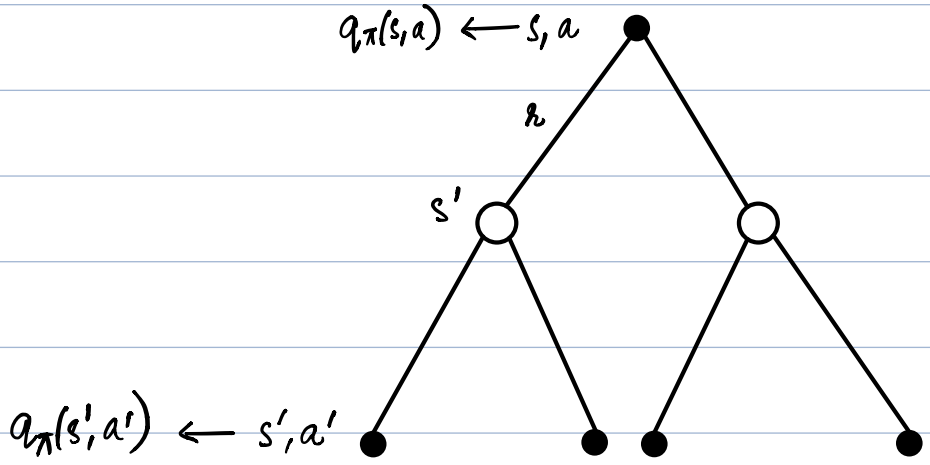


$$q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a\, v_\pi(s')$$



$$\therefore v_\pi(s) = \sum_{a \in A} \pi(a|s)\, q_\pi(s,a)$$

$$= \sum_{a \in A} \pi(a|s) \left(R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a\, v_\pi(s') \right)$$



$$\therefore \quad q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \, v_\pi(s')$$

$$= R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \sum_{a' \in A} \pi(a|s) \, q_\pi(s',a')$$

Optimal Value function:

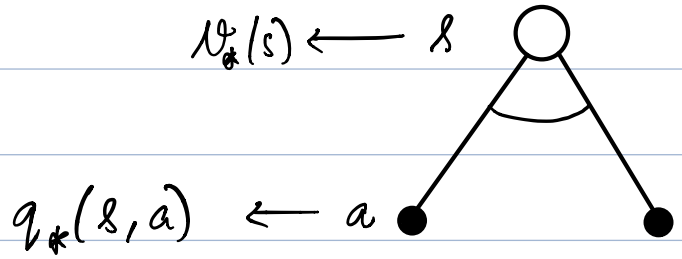
— $v_*(s) = \max_\pi v_\pi(s)$

— $q_*(s,a) = \max_\pi q_\pi(s,a)$

↑ if this is known, then the MDP is solved

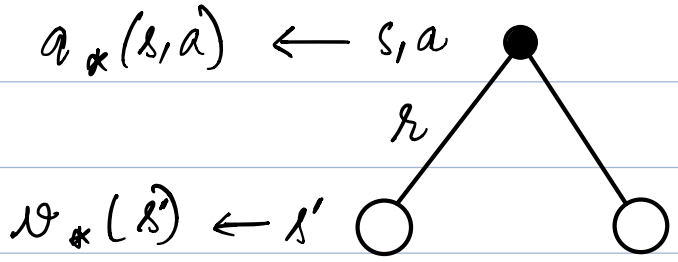
Bellman Optimality Equation:

V^* :

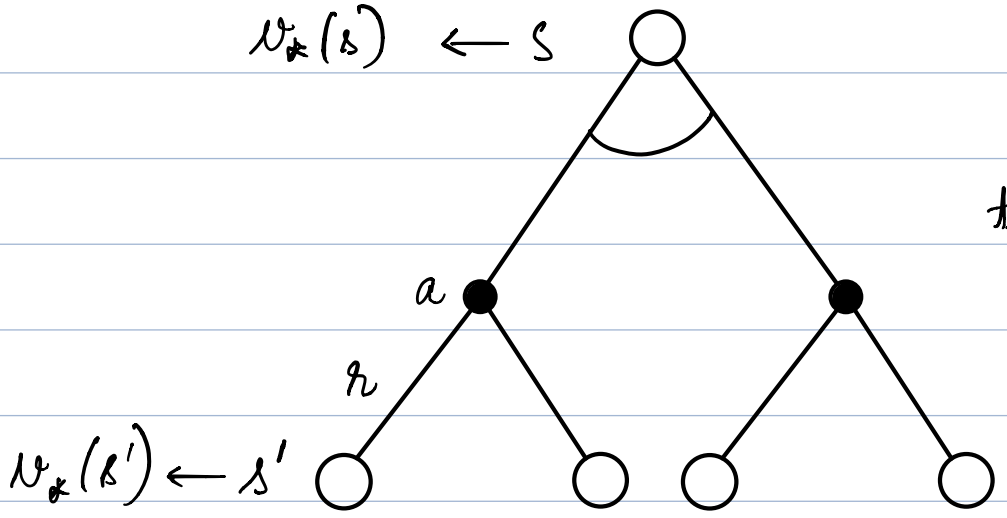


$$v_*(s) = \max_a q^*(s,a)$$

Q^* :

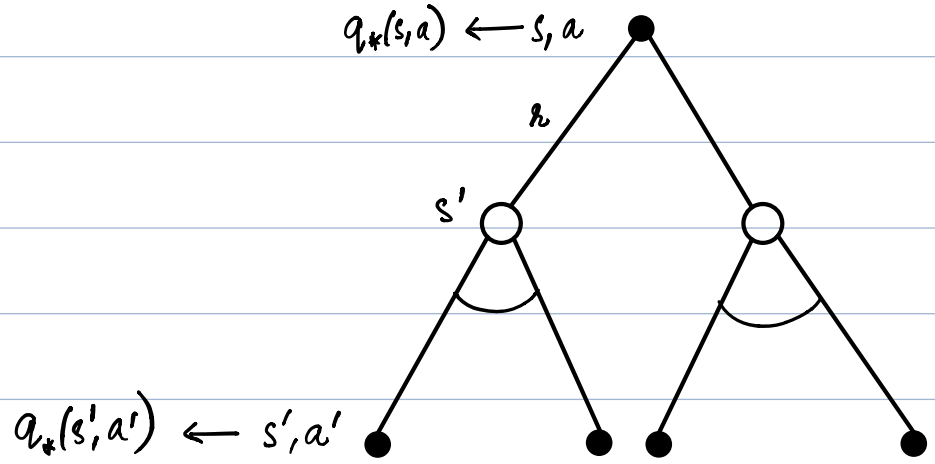


$$q_*(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \, v_*(s')$$



two step lookahead

$$\therefore v_*(s) = \max_a R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \, v_*(s')$$



$$q_{\pi}(s,a) = R_s^a + \gamma \sum P_{ss'}^a v_{\pi}(s')$$

$$= R_s^a + \gamma \sum P_{ss'}^a \max_{a'} q_{\pi}(s',a')$$

Solving the Bellman Optimality Equation:

- non linear equations
- no closed form solution in general
- requires iterative solutions
  - value iteration
  - policy iteration
  - } DP
  - Q-learning
  - Sarsa

Extensions to MDPs:

- infinite & continuous MDPs
- partially observable MDPs
- undiscounted, average reward MDPs