MDP(Markov Decision Process)马尔科夫决策过程

Yanan LI(李亚男)

2017年3月7日

1 MDP空间结构

MDP是一个智能体 (Agent) 与环境 (Environment) 之间通过动作 (Action)、状态 (State) 和奖励 (Reward) 相互作用的循环过程。在 t 时刻，智能体根据从环境中得到的状态 S_t 和奖励 R_t，做出决策动作 (A_t)，在t+1时刻环境反馈给智能体新的状态 S_{t+1} 和奖励 R_{t+1}。

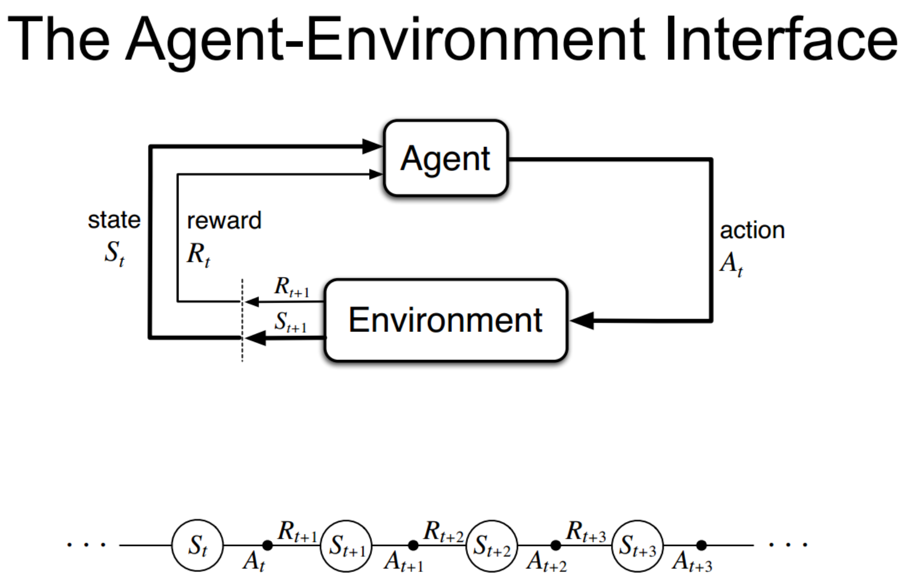

图 1: The Agent Environment Interface

- 奖励仅描述了智能体需要实现的目标，而不是如何实现。
- 强化学习可以视为开发利用已知策略(利用，exploitation)和探索新策略(探索，explortation)之间的权衡(利用多一点还是开发多一点)。

2 MDP

2.1 MDP definition

Definition

A *Markov Decision Process* is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$

- $\mathcal{S}$ is a finite set of states
- $\mathcal{A}$ is a finite set of actions
- $\mathcal{P}$ is a state transition probability matrix,
  $\mathcal{P}_{ss'}^{a} = \mathbb{P}\left[S_{t+1} = s' \mid S_t = s, A_t = a\right]$
- $\mathcal{R}$ is a reward function, $\mathcal{R}_s^a = \mathbb{E}\left[R_{t+1} \mid S_t = s, A_t = a\right]$
- γ is a discount factor $\gamma \in [0, 1]$.

图 2: Markov Decision Process

2.2 State

A state captures whatever information is available to the agent at step t about its environment. The state can include immediate "sensations," highly processed sensations, and structures built up over time from sequences of sensations, memories etc.

A state should summarize past sensations so as to retain all "essential" information, i.e., it should have the Markov Property:

$Pr\{R_{t+1} = r, S_{t+1} = s^{'}|S_0, A_0, R_1, ..., S_{t-1}, A_{t-1}, R_t, S_t, A_t\}$
$= Pr\{R_{t+1} = r, S_{t+1} = s^{'}|S_t, A_t\}$ for all $s^{'} \in S^{+}, r \in R$

We should be able to throw away the history once state is known.

2.3 Dynamics

- Model based: dynamics are known or are estimated.(知道并可以存储所有MDP信息，包括state,action,possibility and reward)
- Model free: we do not know the dynamics of the MDP.(只知道部分信息，包括state,action, 需要自己探索未知的MDP信息，包括possibility,reward)

2.4 Rewards

The definition of **rewards** as follows

Definition

The *return* G_t is the total discounted reward from time-step t.

$$G_t = R_{t+1} + \gamma R_{t+2} + ... = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

图 3: Rewards

- Reward 是每次采取action后获得的即时奖励
- the agent's goal is to maximize the total amount of reward it receives. This means maximizing not immediate reward, but cumulative reward in the long run.

2.5 Policy

策略的定义：智能体学到的策略π是指已知状态下可能产生的概率分布(即每个action应该分多少概率)。

At each time step, the agent implements a mapping from states to probabilities of selecting each possible action. This mapping is called the agent's *policy* and is denoted π_t, where $\pi_t(a|s)$ is the probability that $A_t = a$ if $S_t = s$.[1]

Definition

A *policy* π is a distribution over actions given states,

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

图 4: Policy

[1] Reinforcement Learning:An Introduction,Second edition

- A policy fully defines the behaviour of an agent(一旦policy确定后，每个action发生的概率都是确定的)
- MDP policies depend on the current state(not the history)
- i.e. Policies are stationary(time-independent),$A_t\ \pi(\cdot|S_t), \forall t > 0$

2.6 Probabilities and Rewards

$$p(s^{'}, r|s, a) = Pr\{S_{t+1} = s^{'}, R_{t+1} = r|S_t = s, A_t = a\}$$
$$r(s, a) = \mathbb{E}[R_{t+1}|S_t = s, A_t = a]$$
$$p(s^{'}|s, a) = Pr\{S_{t+1} = s^{'}|S_t = s, A_t = a\} = \sum_{r \in R} p(s^{'}, r|s, a)$$
$$r(s, a, s^{'}) = \mathbb{E}[R_{t+1}|S_t = s, A_t = a, S_{t+1} = s^{'}] = \frac{\sum_{r \in R} rp(s^{'}, r|s, a)}{p(s^{'}|s, a)}$$

2.7 Value functions

如图 5, Value functions are cumulative expected rewards.

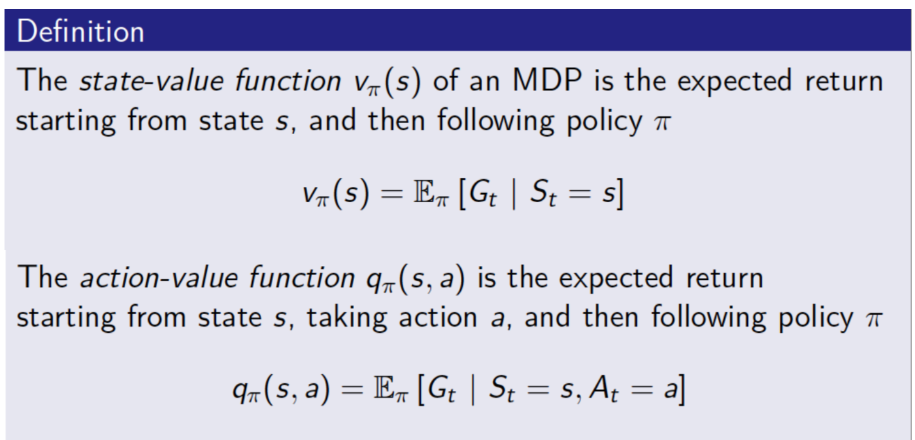

图 5: value functions

如图 6, Optimal value functions are best achievable cumulative expected rewards.

- 价值函数是用来衡量某一(s)或(s,a)的优劣.需要注意的是,这里价值函数是依赖于某一策略的。

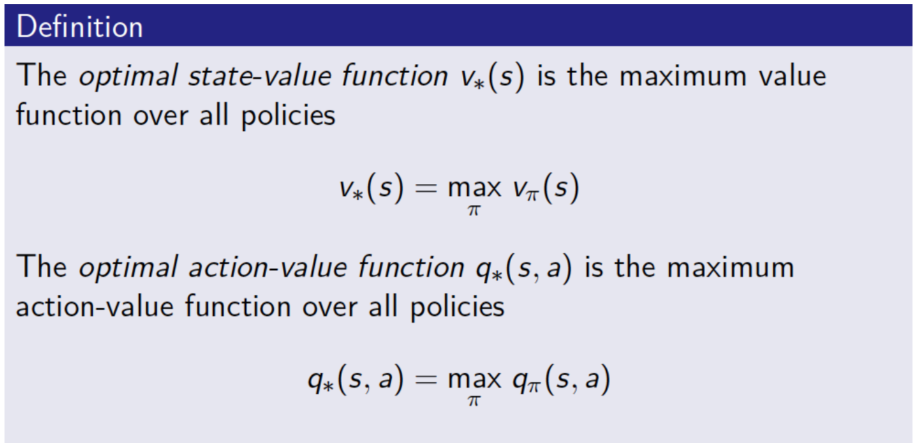

图 6: optimal value functions

- 最优价值函数是在所有策略下的某一(s)或(s,a)的最优值，它不依赖于策略。

3 Bellman Expectation Equation

3.1 Value function

$$
\begin{aligned}
v_\pi(s) &= \mathbb{E}_\pi[G_t|S_t = s] \\
&= \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1}|S_t = s\right] \\
&= \mathbb{E}_\pi\left[R_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+2}|S_t = s\right] \\
&= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s^{'}, r|s, a)\left[r + \gamma \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+2}|S_{t+1} = s^{'}\right]\right] \\
&= \sum_a \pi(a|s) \sum_{s',r} p(s^{'}, r|s, a)\left[r + \gamma v_\pi(s^{'})\right], \forall s \in S
\end{aligned}
$$

3.1.1 Looking Inside the Expectations

如图7所示，value function 的例外一种表达方式是：

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

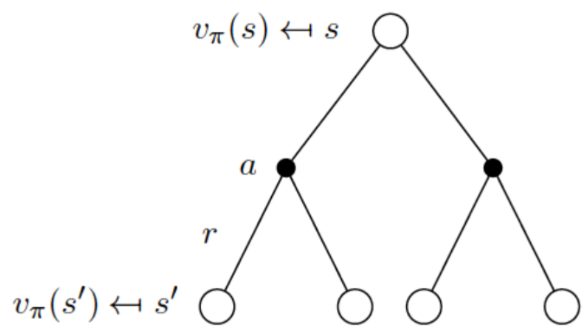

图 7: Value function

3.2 Action Value function

The action-value function can similarly be decomposed as follows,
$q_\pi(s,a) = \mathbb{E}_\pi[R_{t+1} + \gamma q_\pi(S_{t+1}, A_{t+1}) | S_t = s, A_t = a]$

3.2.1 Looking Inside the Expectations

如图8所示，action value function 的例外一种表达方式是:
$q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{ss'}^a \sum_{a' \in A} \pi(a'|s') q_\pi(s', a')$

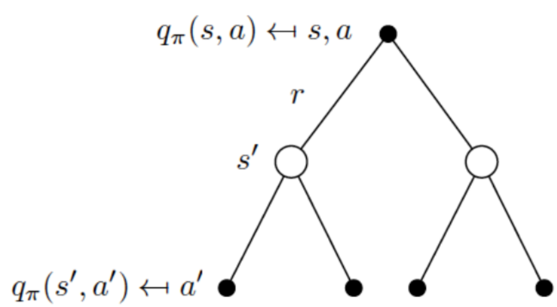

图 8: Action value function

3.3 Relating state and action value functions

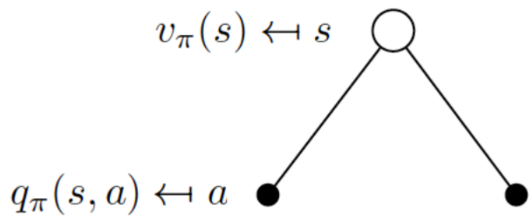

图 9: Relating state and action value functions

如图 9 所示，value function 和 action value function 的关系:

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

3.4 Optimal value function

如图 10 所示，optimal value function 可以表示如下：

$v_*(s) = \max\limits_a R_s^a + \gamma \sum\limits_{s' \in S} P_{ss'}^a v_*(s')$

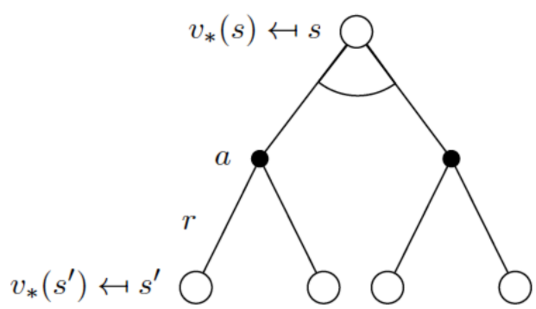

图 10: Optimal value function

3.5 Optimal action value function

如图 11 所示，optimal action value function 可以表示如下：

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

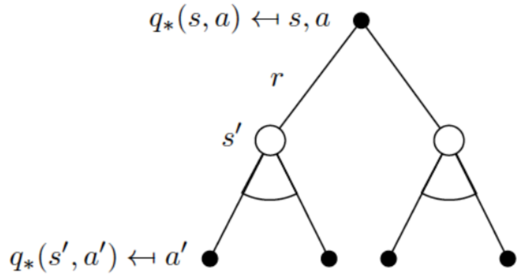

图 11: Optimal action value function

3.6 Relating optimal state and action value functions

如图 12 所示，optimal value function 和 optimal action value function 的关系:

$$\begin{aligned}
v_*(s) &= \max_{a \in A(s)} q_{\pi^*}(s,a) \\
&= \max_a \mathbb{E}_{\pi^*}\left[G_t | S_t = s, A_t = a\right] \\
&= \max_a \mathbb{E}_{\pi^*}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s, A_t = a\right] \\
&= \max_a \mathbb{E}_{\pi^*}\left[R_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+2} | S_t = s, A_t = a\right] \\
&= \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) | S_t = s, A_t = a] \\
&= \max_{a \in A(s)} \sum_{s^{'}, r} p(s^{'}, r | s, a)[r + \gamma v_*(s^{'})]
\end{aligned}$$

$$\begin{aligned}
q_*(s,a) &= \mathbb{E}\left[R_{t+1} + \gamma \max_{a^{'}} q_*(S_{t+1}, a^{'}) | S_t = s, A_t = a\right] \\
&= \sum_{s^{'}, r} p(s^{'}, r | s, a)\left[r + \gamma \max_{a^{'}} q_*(s^{'}, a^{'})\right]
\end{aligned}$$

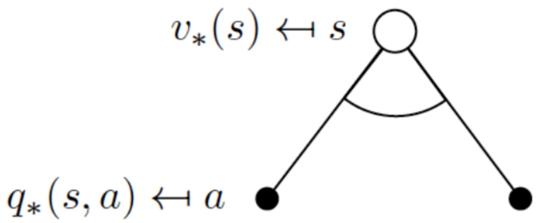

图 12: Optimal action value function

4 Optimal policy

4.1 Definition

Define a partial ordering over policies:
$\pi \geq \pi^{'}$ if $v_\pi(s) \geq v_{\pi^{'}}(s), \forall s$

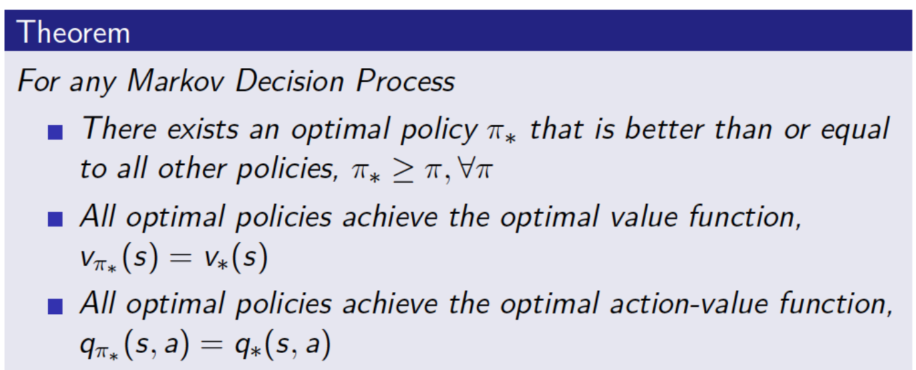

图 13: optimal policy

4.2 Find an optimal policy

A optimal policy can be found by maximising over $q_*(s,a)$,

$$\pi_*(a|s) = \begin{cases} 1 & \text{if } a = \arg\max\limits_{a \in A} q_*(s,a) \\ 0 & otherwise \end{cases}$$

- There is always a deterministic optimal policy for any MDP

- If we know $q_*(s, a), weimmediatelyhavetheoptimalpolicy$