ORF 505: Statistical Analysis of Financial Data

Tuesday Precept 9: Finite Markov Decision Processes (MDPs)

Nov 12, 2024

Lecturer: Qishuo Yin *Scribe: Qishuo Yin*

1 Agent-Environment Interface

The learner and decision-maker is called the agent. The thing it interacts with, comprising everything outside the agent, is called the environment. These interact continually, the agent selecting actions and the environment responding to those actions and presenting new situations to the agent. For details please refer to section 3.1 of [1]. And there is also a figure to demonstrate the relation in figure 1.

2 Basic Definition

A reinforcement learning task that satisfies the Markov property is called a **Markov Decision Process (MDP)**. If the state and action spaces are finite, it is referred to as a finite MDP. An MDP is defined by:

- A set of states: S
- A set of actions: $A(s)$, available from each state $s \in S$
- One-step dynamics of the environment: $p(s', r|s, a)$. This is the probability of transitioning to state s' and receiving reward r upon taking action a in state s.

Following the definition of the Markov Decision Process (MDP), we can define a few concepts about the reward:

- Expected rewards for state-action pairs:

$$r(s, a) = \mathbb{E}[R_{t+1}|S_t = s, A_t = a] = \sum_{r\in\mathcal{R}} r \sum_{s'\in\mathcal{S}} p(s', r|s, a)$$

- State-transition probabilities:

$$p(s'|s, a) = \mathbb{P}(S_{t+1} = s'|S_t = s, A_t = a) = \sum_{r\in\mathcal{R}} p(s', r|s, a)$$

- Expected rewards for state-action-next-state triples:

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

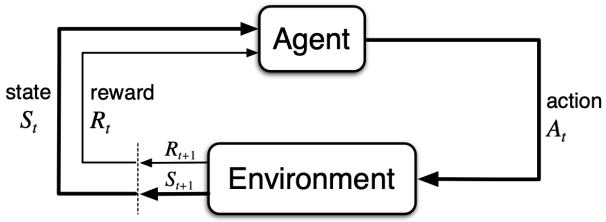

Figure 1: The agent-environment interaction in reinforcement learning

3 Value Functions

Policy π (sometimes called strict policy) is a (deterministic) function $\pi_t : S \ni s \mapsto \pi(s) \in A$. Value functions estimate the expected return from a state (or state-action pair), crucial in evaluating the quality of policies.

- Value of a state s under a policy π: This is the expected return of when starting in s and following π thereafter.
- State-value function for policy π:

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

  where $\mathbb{E}_\pi[\cdot]$ denotes the expected value of a random variable given that the agent follows policy π, and t is any time step.
- Action-value function for policy π:

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

- Bellman Equations for MDPs: it quantifies a relationship between the value of a state and the values of its successor states.

$$\begin{aligned}
v_\pi(s) &= \mathbb{E}_\pi \left[G_t \mid S_t = s \right] && (1) \\
&= \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s \right] && (2) \\
&= \mathbb{E}_\pi \left[R_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k R_{t+k+2} \mid S_t = s \right] && (3) \\
&= \sum_a \pi(a|s) \sum_{s',r} p(s', r \mid s, a) \left[r + \gamma \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+2} \mid S_{t+1} = s' \right] \right] && (4) \\
&= \sum_a \pi(a|s) \sum_{s',r} p(s', r \mid s, a) \left[r + \gamma v_\pi(s') \right] && (5)
\end{aligned}$$

4 Optimal Value Functions

- Optimal policy: π_*
  A policy π is defined to be better than or equal to a policy π' if its expected return is greater than or equal to that of π' for all states. In other words, $\pi \geq \pi'$ if and only if $v\pi(s) \geq v\boldsymbol{p}'(s)$ for all $s \in \mathcal{S}$. There is always at least one policy that is better than or equal to all other policies, which is the optimal policy.

- Optimal state-value function: v_*. They are optimal policies sharing the same state-value function.
$$v_*(s) = \max_\pi v_\pi(s)$$

- Optimal action-value function: q_*. They are optimal policies sharing the same action-value function.
$$q_*(s,a) = \max_\pi q_\pi(s,a)$$

- Bellman Optimality Equation for v_*:
$$v_*(s) = \max_{a \in A(s)} \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_*(s')\right] \tag{6}$$

- Bellman Optimality Equation for q_*:
$$q_*(s,a) = \sum_{s',r} p(s',r|s,a)\left[r + \gamma \max_{a'} q_*(s',a')\right] \tag{7}$$

References

[1] R. S. Sutton. Reinforcement learning: An introduction. *A Bradford Book*, 2018.