Finite Markov Decision Processes

Some Important Notations in This Chapter

Four arguments dynamics of the MDP:

$$p(s', r \mid s, a) = \Pr\{S_t = s', R_t = r \mid S_{t-1} = s, A_t = a\} \tag{1.1}$$

This is the core equation of this chapter.

Some probability relationships between dynamics of MDP

$$p(s' \mid s, a) = \sum_{r \in \mathbb{R}} p(s', a \mid s, a) \tag{1.2}$$

$$\begin{aligned} r(s, a) = \mathbb{E}[R_t \mid S_{t-1=} s, A_{t-1} = a] = \sum_{r \in \mathbb{R}} r p(r \mid s, a) \\ = \sum_{r \in \mathbb{R}, s' \in \mathcal{S}} r p(s', r \mid s, a) \end{aligned} \tag{1.3}$$

$$\begin{aligned} r(s', s, a) = \mathbb{E}\{r \mid s', s, a\} = \sum_{r \in \mathbb{R}} r p(r \mid s', s, a) \\ = \sum_{r \in \mathbb{R}} r \frac{p(s', r \mid s, a)}{p(s' \mid s, a)} \end{aligned} \tag{1.4}$$

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} = R_{t+1} + \gamma G_{t+1} \tag{1.5}$$

State-value and state-action function

$$v_\pi(s) = \mathbb{E}[G_t \mid S_t = s] = \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s] \tag{1.6}$$

$$\begin{aligned} q_\pi(s, a) &= \mathbb{E}[G_t \mid S_t = s, A_t = a] \\ &= \mathbb{E}[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid S_t = s, A_t = a] \end{aligned} \tag{1.7}$$

Recursive formula of state-value and state-action function

Due to **Markov Property**, the most important step in the derivation as follows:

$$\begin{aligned} \mathbb{E}[G_{t+1} \mid S_t = s] &= \sum_{s' \in \mathcal{S}} p(s' \mid s) \mathbb{E}[G_{t+1} \mid S_{t+1} = s', S_t = s] \\ &= \sum_{s' \in \mathcal{S}, a \in \mathcal{A}} \pi(a \mid s) p(s' \mid s, a) \mathbb{E}[G_{t+1} \mid S_{t+1} = s'] \\ &= \sum_{s' \in \mathcal{S}, a \in \mathcal{A}, r \in \mathcal{R}} \pi(a \mid s) p(s', r \mid s, a) v_\pi(s') \end{aligned}$$

$$\begin{aligned} \mathbb{E}[G_{t+1} \mid S_t = s, A_t = a] &= \sum_{s' \in \mathcal{S}} p(s' \mid s, a) \mathbb{E}[G_{t+1} \mid S_{t+1} = s', S_t = s, A_t = a] \\ &= \sum_{s' \in \mathcal{S}} p(s' \mid s, a) \mathbb{E}[G_{t+1} \mid S_{t+1} = s'] \\ &= \sum_{s' \in \mathcal{S}} p(s' \mid s, a) \sum_{a' \in \mathbb{A}} \pi(a' \mid s') \mathbb{E}[G_{t+1} \mid S_{t+1} = s', A_{t+1} = a'] \\ &= \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r \mid s, a) \sum_{a' \in \mathbb{A}} \pi(a' \mid s') q_\pi(s', a') \end{aligned}$$

Among these equations,we make full use of **Total Probability Rule**.Take above equations and (1.5) into (1.6) and (1.7),for all $s \in \mathcal{S}, a \in \mathcal{A}$,we obtain:

$$v_\pi(s) = \sum_{s' \in \mathcal{S}, a \in \mathcal{A}, r \in \mathcal{R}} \pi(a \mid s) p(s', r \mid s, a)(r + \gamma v_\pi(s')) \tag{1.8}$$

$$q_\pi(s,a) = \sum_{s'\in\mathcal{S},r\in\mathcal{R}} p(s',r \mid s,a)(r+\gamma\sum_{a'\in\mathbb{A}}\pi(a' \mid s')q_\pi(s',a')) \tag{1.9}$$

From(1.8) and (1.9),we can get the state and state-action value function under deterministic strategy with following fomula with initial $v_\pi^0(s)$ and $q_\pi^0(s,a)$: for all $s\in\mathcal{S}, a\in\mathcal{A}$

$$v_\pi^{n+1}(s) = \sum_{s'\in\mathcal{S},a\in\mathcal{A},r\in\mathcal{R}} \pi(a \mid s)p(s',r \mid s,a)(r+\gamma v_\pi^n(s')) \tag{1.10}$$

$$q_\pi^{n+1}(s,a) = \sum_{s'\in\mathcal{S},r\in\mathcal{R}} p(s',r \mid s,a)(r+\gamma\sum_{a'\in\mathbb{A}}\pi(a' \mid s')q_\pi^n(s',a')) \tag{1.11}$$

As we want to get a strategy in a MDP problem,that is,when we are in a states,which action shall we choose can make the largest long-term reward,we can select action as followed:$s\in\mathcal{S}, a\in\mathcal{A}$:

$$v_\pi(s) = \max_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S},r\in\mathcal{R}} p(s',r \mid s,a)(r+\gamma v_\pi(s')) \tag{1.12}$$

$$q_\pi(s,a) = \sum_{s'\in\mathcal{S},r\in\mathcal{R}} p(s',r \mid s,a)(r+\gamma\max_{a'\in\mathcal{A}} q_\pi(s',a')) \tag{1.13}$$

Intuitively, optimal value: $v^*(s)$ and $q^*(s,a)$ also confirm (1.12) and (1.13),which called ***Bellman optimality equation*** . Actually, we can have an intuition that $q^*(s,a)$ is superior to $v^*(s)$ during we get the optimal strategy according to the position of the max operator in (1.12) and (1.13).

Realtions between $v(s)$ and $q(s,a)$

Also by means of **Markov Property** and **Total Probability Rule**, we can derive some relationships as followed:

$$v_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a \mid s)q_\pi(s,a) \tag{1.14}$$

$$q_\pi(s,a) = \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} p(s', r \mid s, a)(r + \gamma v_\pi(s')) \tag{1.15}$$

Above all, everything about mathematics of this chapter is concluded, we will talk about some conceptual details later.

Details

Some professional concept

Episode and Episodic Task

During the agent-environment interaction, there is a nature notion of final time step and **we can recognize the terminal state** clearly. Intuitively, we can break an interaction into several sequences on according to the terminal state, each sequence is called an episode or a trail. Tasks with episodes of this kind are called episodic task.

Continuing Tasks

The agent-environment interaction **does not break naturally into identifiable episodes**, but goes on continually without limit. We call these continuing tasks.

Backup Diagram

A diagram is a symbolic representation of information using visualization techniques. **Backup diagram describes relationships that form the basis of the update or backup operations** that are at the heart of reinforcement learning methods.(see page 59 of book of Sutton.)

Tabular Method

In tasks with small, finite state sets, it is possible to form these approximations of value funciton using arrays or tables with one entry for each state(or state-action pair) and the corresponding methods we call tabular methods.

We should care about...

The boundary between agent and environment

The generally rule is that anything can't be changed by agent is considered to be outside of it and thus part of its environment. The agent-environment boundary represents the limit of the agent's absolutely contuol not of its knowledge.

Design of reward

Maximizing reward must be able to lead to achieve our final goals. If achieving subgoals were rewarded, then the agent might find a way to achieve subgoals without achieving the real goal. The reward signal is the way of communicating with the robot **what you want to achieve**, not **how you want to achieve**.

Why optimal function is important?

Using greedy strategy, we may not get the optimal rewards in the end. But if we use optimal value function to evalute the short-term consequences of actions, then a greedy policy is actually optimal in the long-term sense in all possible future because the long-term return is turned into a quantity that is locally and immediately available for each state.

v_π^* and q_π^*, which is better?

Absolutely q_π^* is better. Think about what we want to do in the beginning: make a choice(choose an action) when we face a state,that is, a policy. Having v_π^* is good, we can get a lot of information from it, but we don't want that much, we can get the policy from q_π^* directly, why we spend more resources to compute v_π^*?