Reinforcement Learning

- 3. Finite Markov Decision Processes -

Lee Tae Min
koalakid154@gmail.com

1

Chapter 3 : Finite Markov Decision Process

Chapter 3 Finite Markov Decision Processes

Introduction

In this chapter we introduce the formal problem of **finite Markov decision processes**, or **finite MDPs.**

This problem involves **evaluative feedback**, as in bandits,
but also an **associative aspect**(choosing different actions in different situations).

Estimate

- the value $q_*(s, a)$ of each action a in each state s
- the value $v_*(s)$ of each state given optimal action selections.

Chapter 3 Finite Markov Decision Processes

Introduction

MDPs are a **mathematically idealized form** of the reinforcement learning problem for which precise theoretical statements can be made.

We introduce key elements of the problem's mathematical structure, such as **returns, value functions**, and **Bellman equations**.

Chapter 3 Finite Markov Decision Processes

3.1 The Agent-Environment Interface

Agent : The **learner** and **decision maker**
Environment : The thing it **interacts** with, comprising everything outside the agent

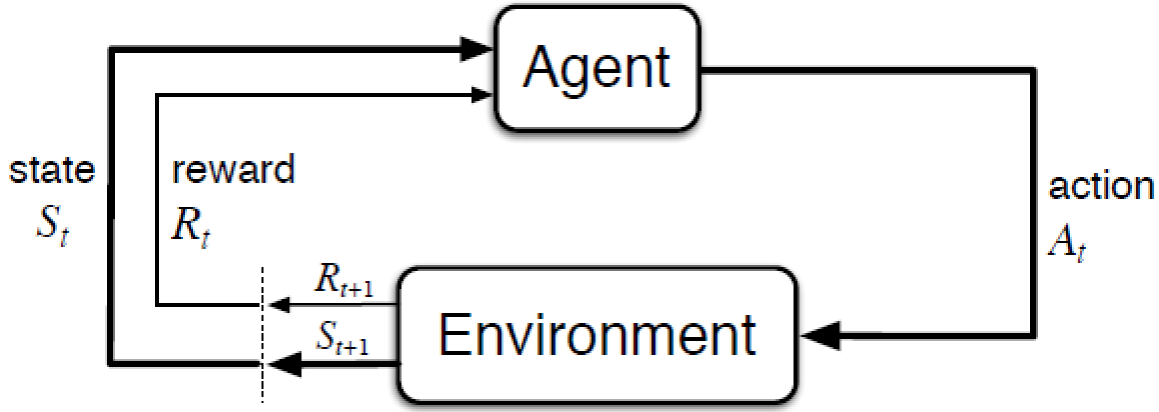


Figure 3.1: The agent–environment interaction in a Markov decision process.

Chapter 3 Finite Markov Decision Processes

3.1 The Agent-Environment Interface

The MDP and agent together thereby give rise to a sequence or **trajectory** that begins like this:

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \ldots$$

That is, for particular values of these random variables, $s' \in S \ and \ r \in R$, there is a probability of those values occurring at time t, given particular values of the preceding state and action:

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

Chapter 3 Finite Markov Decision Processes

3.1 The Agent-Environment Interface

p specifies a probability distribution for each choice of **s** and **a**, that is, that

$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) = 1, \text{ for all } s \in \mathcal{S}, a \in \mathcal{A}(s).$$

- **state-transition probabilities**

$$p(s' \mid s, a) \doteq \Pr\{S_t = s' \mid S_{t-1} = s, A_{t-1} = a\} = \sum_{r \in \mathcal{R}} p(s', r \mid s, a).$$

- **expected rewards for state-action pairs**

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

- **expected reward for state-action-next-state triples**

$$r(s, a, s') \doteq \mathbb{E}[R_t \mid S_{t-1} = s, A_{t-1} = a, S_t = s'] = \sum_{r \in \mathcal{R}} r \frac{p(s', r \mid s, a)}{p(s' \mid s, a)}$$

3.2 Goals and Rewards

The purpose or goal of the agent:

To maximize the total amount of **reward** it receives

The agent **always** learns to maximize its reward.

The reward signal is **not** the place to impart to the agent prior knowledge about **how** to achieve what we want it to do.
what you want it to achieve, **not how** you want it achieved.

3.3 Returns and Episodes

How to be defined formally the agent's goal?
In general, we **seek** to maximize **the expected return**, where the return, denoted **Gt**, is defined as some specific function of the reward sequence.

$$G_t \doteq R_{t+1} + R_{t+2} + R_{t+3} + \cdots + R_T$$

he agent-environment interaction breaks naturally into subsequences, which we call **episodes.**
Each episode ends in a special state called the **terminal state**

3.3 Returns and Episodes

Episodic tasks	Continuing tasks
• tasks with **episodes** • the set of nonterminal states, S, the set of all sates plus, S^+	• agent-environment interaction **can not be expressed in** episodes • T=∞, discounting

The expected discounted return:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1},$$

3.3 Returns and Episodes

- If **r<1**, the infinite sum has a finite value as long as the reward sequence is **bounded**. As r **approaches 1**, the return objective takes **future rewards** into account more strong.

- If **r=0**, the agent is "myopic" in being concerned only with **maximizing immediate rewards**

Returns at successive time steps are related to each other in a way that is important for the theory and algorithms of reinforcement learning:

$$\begin{aligned} G_t &\doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \cdots \\ &= R_{t+1} + \gamma \left(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \cdots \right) \\ &= R_{t+1} + \gamma G_{t+1} \end{aligned}$$

3.4 Unified Notation for Episodic and Continuing Tasks

we consider sometimes one kind of problem and sometimes the other, but often both.

useful to establish one notation that enables us to talk precisely about both cases simultaneously.

$$G_t \doteq \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k,$$

$T=\infty$ or $\gamma = 1$ (but not both).

3.5 Policies and Value Functions

value function : estimate **how good** it for the agent to be in a given state ("how good" is defined in terms of future rewards that can be expected or in terms of expected return.)

policy : value functions are defined with respect to **particular ways of acting**

policy is a mapping from states to probabilities of selecting each possible action.

$$\pi(a|s)$$

3.5 Policies and Value Functions

The value of a state s under a policy π:

$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right], \text{ for all } s \in \mathcal{S},$$

We call the **sate-value** function for policy π

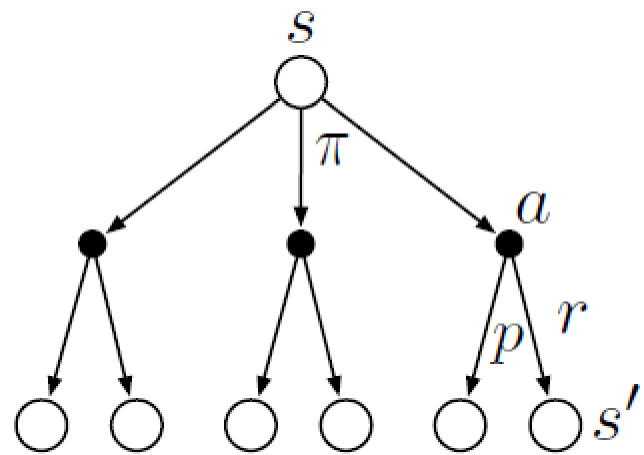
The value of taking action a in state s under a policy π:

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

We call the **action-value** function for policy π

Chapter 3 Finite Markov Decision Processes

3.5 Policies and Value Functions

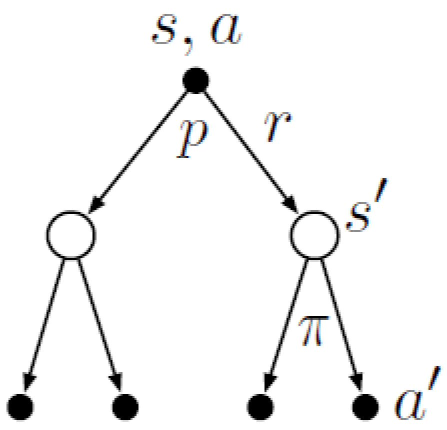


the **sate-value** function for policy π

$$
\begin{aligned}
v_\pi(s) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\
&= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r | s, a)\Big[r + \gamma \mathbb{E}_\pi[G_{t+1} | S_{t+1} = s']\Big] \\
&= \sum_a \pi(a|s) \sum_{s',r} p(s', r | s, a)\Big[r + \gamma v_\pi(s')\Big], \quad \text{for all } s \in \mathcal{S},
\end{aligned}
$$

Chapter 3 Finite Markov Decision Processes

3.5 Policies and Value Functions



the **action-value** function for policy π

$$
\begin{aligned}
q_\pi(s, a) &\doteq \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] \\
&= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a] \\
&= \sum_{s'} \sum_r p(s', r \mid s, a)\big[r + \gamma \mathbb{E}_\pi[G_{t+1} \mid S_{t+1} = s']\big] \\
&= \sum_{s'} \sum_r p(s', r \mid s, a)\big[r + \gamma v_\pi(s')\big], \text{ for all } s \in \mathcal{S},\ a \in \mathcal{A}(s).
\end{aligned}
$$

Chapter 3 Finite Markov Decision Processes

3.6 Optimal Policies and Optimal Value Functions

Solving a reinforcement learning task means, roughly, **finding a policy** that achieves a lot of reward over the long run.

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.

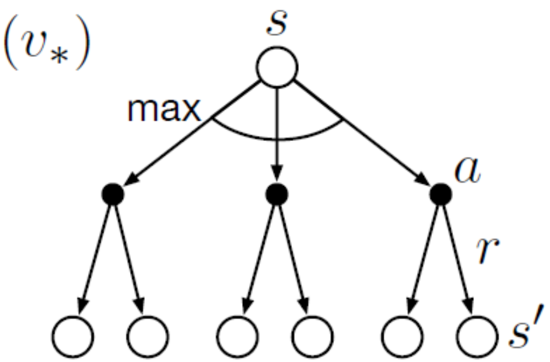
optimal state-value function: $v_*(s) \doteq \max_\pi v_\pi(s)$

optimal action-value function: $q_*(s, a) \doteq \max_\pi q_\pi(s, a)$

$$= \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a]$$

3.6 Optimal Policies and Optimal Value Functions

Bellman optimality equation for v_*

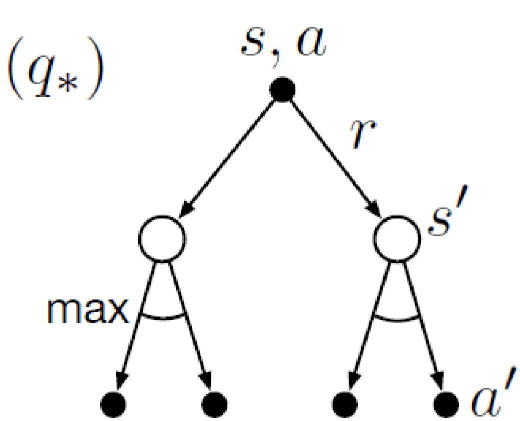


$$
\begin{aligned}
v_*(s) &= \max_{a \in \mathcal{A}(s)} q_{\pi_*}(s,a) \\
&= \max_a \mathbb{E}_{\pi_*}[G_t \mid S_t = s, A_t = a] \\
&= \max_a \mathbb{E}_{\pi_*}[R_{t+1} + \gamma G_{t+1} \mid S_t = s, A_t = a] \\
&= \max_a \mathbb{E}[R_{t+1} + \gamma v_*(S_{t+1}) \mid S_t = s, A_t = a] \\
&= \max_a \sum_{s',r} p(s',r|s,a)\big[r + \gamma v_*(s')\big].
\end{aligned}
$$

3.6 Optimal Policies and Optimal Value Functions

Bellman optimality equation for q_*

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



3.7 Optimality and Approximation

This solution relies on at least three assumptions that are **rarely true in practice**:

1. we accurately know **the dynamics of the environment**
2. we have **enough computational resources** to complete the computation of the solution
3. the **Markov property.**

In reinforcement learning one typically has to settle for **approximate solutions.**

3.7 Optimality and Approximation

The on-line nature of reinforcement learning makes it possible to approximate optimal policies in ways that put more effort into learning to make good decisions for frequently encountered states, at the expense of less effort for infrequently encountered states.

This is one key property that distinguishes reinforcement learning from other approaches to approximately solving MDPs.

Q & A