Chapter 3
Finite Markov Decision Processes

- MDP
  - A classical formalization of sequential decision making
  - Actions influence not just immediate rewards, but also subsequent situations, or states, and through those future rewards.

3.1 The Agent–Environment Interface

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

- Time step
  - $t = 0, 1, 2, \ldots$
- States
  - $\mathcal{S}$: set of all states
  - $S_t \in \mathcal{S}$
- Actions
  - $\mathcal{A}(s)$: set of all actions for state s
  - $A_t \in \mathcal{A}(s)$
- Rewards
  - $\mathcal{R}$: set of all rewards
  - $R_{t+1} \in \mathcal{R}$
- Trajectory
  - $S_0,\ A_0,\ R_1,\ S_1,\ A_1,\ R_2, \ldots$

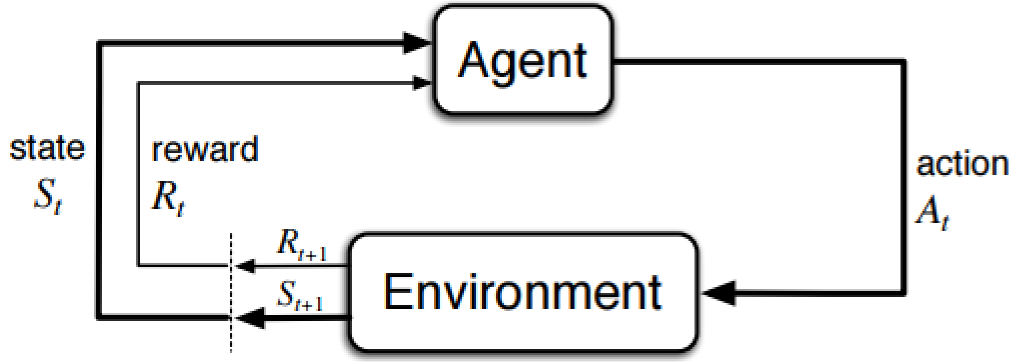


Figure 3.1: The agent–environment interaction in a Markov decision process.

The dynamics of the MDP

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

for all $s', s \in \mathcal{S}$, $r \in \mathcal{R}$, and $a \in \mathcal{A}(s)$.

$$\sum_{s'} \sum_{r} p(s', r \mid s, a) = 1$$

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

$$r(s, a) = \mathbb{E}[R_t \mid S_{t-1} = s, A_{t-1} = a]$$

$$r(s, a, s') = \mathbb{E}[R_t \mid S_{t-1} = s, A_{t-1} = a, S_t = s']$$

3.2 Goals and Rewards

- Goal
  - To maximize the total amount of reward

3.3 Returns and Episodes

- Return
  - $G_t = R_{t+1} + R_{t+2} + \cdots + R_T$ where T is a final time step
- Episode
  - The agent–environment interactions breaks naturally with final time step
- Episodic task
  - Task with episode
- Continuing task
  - The agent–environment interaction does not break naturally
- Discounted return
  - $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} R_{t+k+1}$
  - Discounted rate γ, $0 \le \gamma \le 1$

3.4 Unified Notation for Episodic and Continuing Tasks

3.5 Policies and Value Functions

- Policy π
  - Probabilities of selecting an action
  - $\pi(a|s) = \Pr\{A_t = a \mid S_t = s\}$

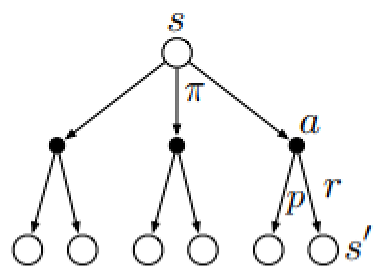
3.5 Policies and Value Functions

- The state-value function v_π for policy π
  - The value function of a state s under a policy π
  - the expected return when starting in s
  - $v_\pi(s) = \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k G_{t+k+1} \,\middle|\, S_t = s]$

- The action-value function q_π for policy π
  - the value of an action a in state s under a policy π
  - $q_\pi(s, a) = \mathbb{E}_\pi[G_t \mid S_t = s, A_t = a] = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k G_{t+k+1} \,\middle|\, S_t = s, A_t = a]$

- Bellman equation for v_π

$$v_\pi(s) = \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)[r + \gamma v_\pi(s')]$$



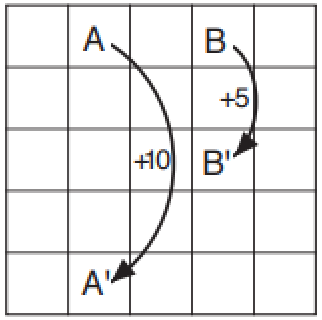
Backup diagram for v_π

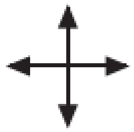
  - Proof

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi[G_t \mid S_t = s] \\ &= \mathbb{E}_\pi[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\ &= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)\big[r + \gamma \mathbb{E}[G_{t+1}|S_{t+1} = s']\big] \\ &= \sum_a \pi(a|s) \sum_{s'} \sum_r p(s', r|s, a)[r + \gamma v_\pi(s')] \end{aligned}$$

Example 3.5: Gridworld

- The cells of the grid correspond to the states of the environment
- At each cell, four actions are possible: `north`, `south`, `east`, and `west`
    - Actions cause the agent to move one cell in the respective direction
    - Actions that would take the agent off the grid leave its location unchanged, but also result in a reward of -1
    - Other actions result in a reward of 0, except those that move the agent out of the special states A and B
    - From state A, all four actions yield a reward of $+10$ and take the agent to A'
    - From state B, all actions yield a reward of $+5$ and take the agent to B'
- The agent selects all four actions with equal probability
- $\gamma = 0.9$





Actions

3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

$$v_\pi\big((1,1)\big) = \frac{1}{4}\Big(-1 + \gamma v_\pi\big((1,1)\big)\Big) + \frac{1}{4}\Big(0 + \gamma v_\pi\big((1,2)\big)\Big) + \frac{1}{4}\Big(-1 + \gamma v_\pi\big((1,1)\big)\Big) + \frac{1}{4}\Big(0 + \gamma v_\pi\big((2,1)\big)\Big)$$
$$= -\frac{1}{2} + \frac{1}{2}\gamma v_\pi\big((1,1)\big) + \frac{1}{4}\gamma\left(v_\pi\big((1,2)\big) + v_\pi\big((2,1)\big)\right)$$
$$0.55 v_\pi\big((1,1)\big) = -0.5 + 0.225(8.8 + 1.5)$$
$$v_\pi\big((1,1)\big) = 3.3$$

$$v_\pi\big((1,1)\big) = \frac{1}{4}\Big(-1 + \gamma v_\pi\big((1,1)\big)\Big) + \frac{1}{4}\Big(0 + \gamma v_\pi\big((1,2)\big)\Big) + \frac{1}{4}\Big(-1 + \gamma v_\pi\big((1,1)\big)\Big) + \frac{1}{4}\Big(0 + \gamma v_\pi\big((2,1)\big)\Big)$$
$$= -\frac{1}{2} + \frac{1}{2}\gamma v_\pi\big((1,1)\big) + \frac{1}{4}\gamma\left(v_\pi\big((1,2)\big) + v_\pi\big((2,1)\big)\right)$$
$$0.55 v_\pi\big((1,1)\big) = -0.5 + 0.225(8.8 + 1.5)$$
$$v_\pi\big((1,1)\big) = 3.3$$

3.6 Optimal Policies and Optimal Value Functions

- Optimal policy
  - One policy that is better than or equal to all other policies
  - $v_{\pi_*}(s) \geq v_\pi(s)$ for all policy π
- Optimal state-value function v_*

$$v_*(s) = \max_\pi v_\pi(s)$$

- Optimal action-value function q_*

$$q_*(s,a) = \max_\pi q_\pi(s,a)$$

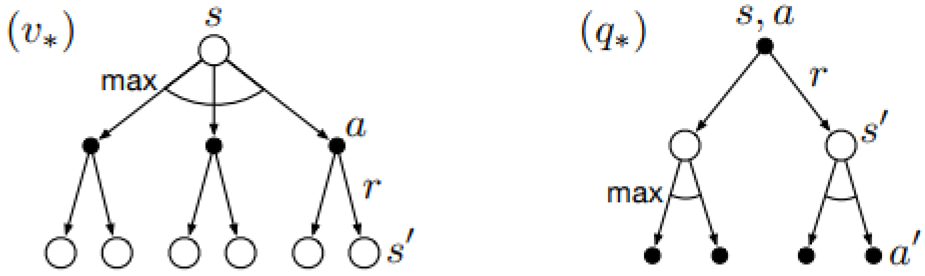


Figure 3.4: Backup diagrams for v_* and q_*

Example 3.8: Solving the Gridworld

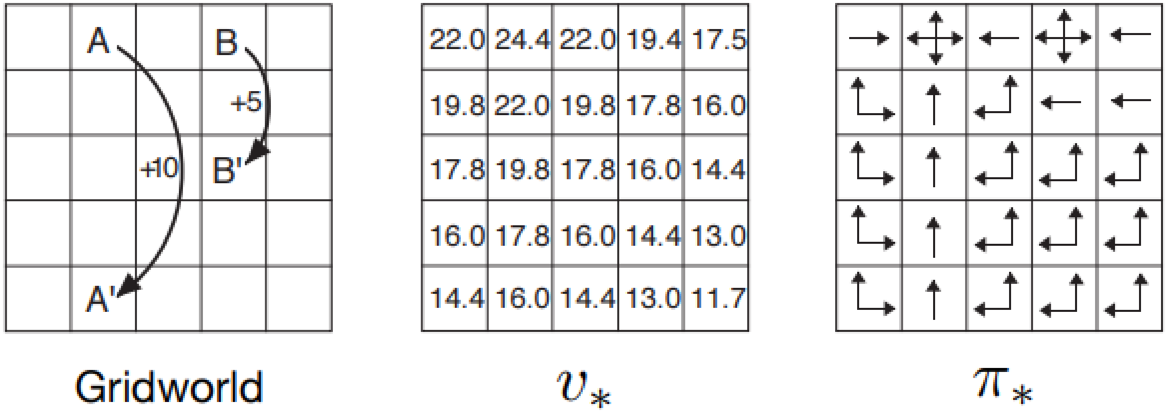


Figure 3.5: Optimal solutions to the gridworld example.

3.7 Optimality and Approximation

- Optimal policies can be generated only with extreme computational cost
- It is an ideal that agents can only approximate