Học phần RBE3043: Các thuật toán thích nghi

Buổi 4: Quá trình ra quyết định Markov hữu hạn

Giảng viên: TS. Nguyễn Thế Hoàng Anh

Hà Nội, ngày 4 tháng 10 năm 2023

The Agent-Environment Interface

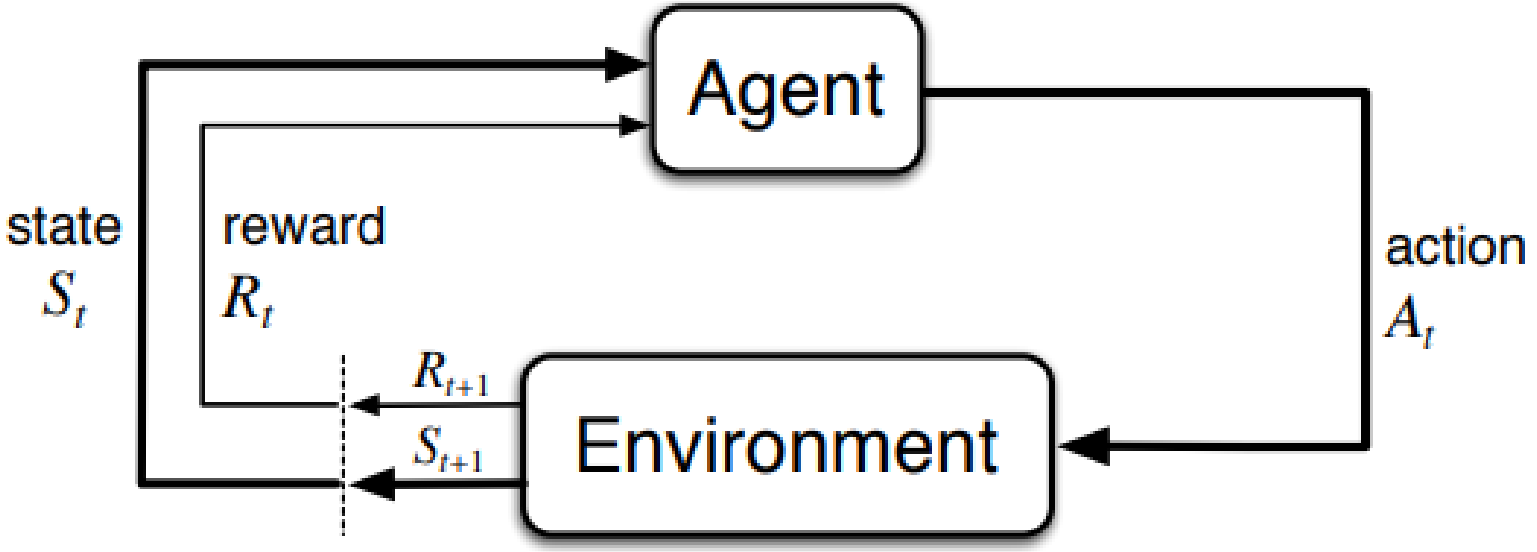


Figure 3.1: The agent–environment interaction in a Markov decision process.

MDPs are meant to be a straightforward framing of the problem of learning from interaction to achieve a goal

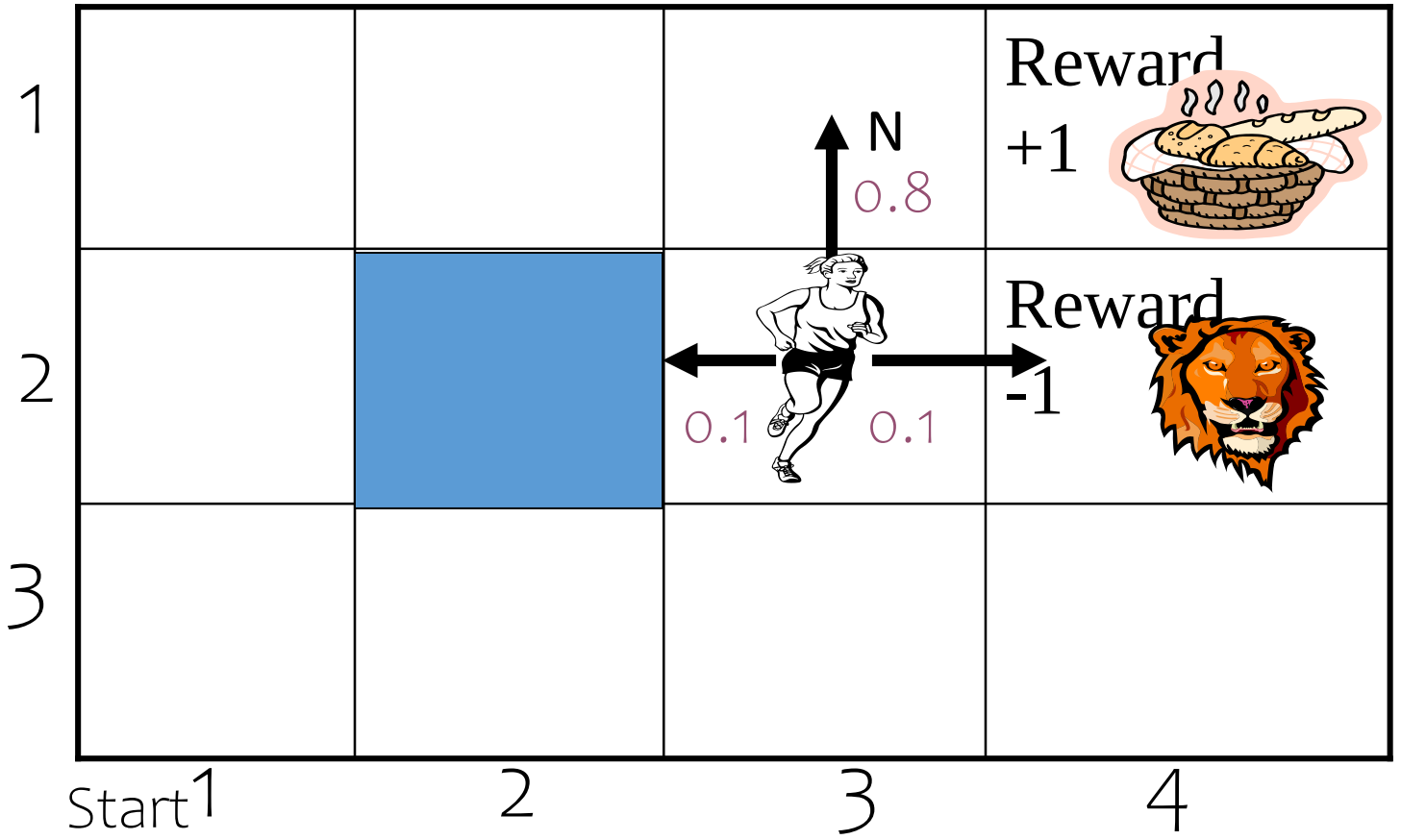
Markov Decision Process (MDP)

- Defined as a tuple: <S, A, P, R>
  - *S: State*
  - *A: Action*
  - *P: Transition function*
    - Table P(s’| s, a), prob of s’ given action “a” in state “s”
  - *R: Reward*
    - *R(s, a) = cost or reward of taking action a in state s*

- Choose a sequence of actions (not just one decision or one action)
  - *Utility based on a sequence of decisions*

Example: What SEQUENCE of actions should our agent take?

- Each action costs –1/25
- Agent can take action N, E, S, W
- Faces uncertainty in every state



MDP Tuple: <S, A, P, R>

- *S: State of the agent on the grid (4,3)*
  - Note that cell denoted by (x,y)
- *A: Actions of the agent, i.e., N, E, S, W*

- *P: Transition function*
  - Table P(s'| s, a), prob of s' given action "a" in state "s"
  - E.g., P((4,3) | (3,3), N) = 0.1
  - E.g., P((3, 2) | (3,3), N) = 0.8
  - (Robot movement, uncertainty of another agent's actions,…)

- *R: Reward (more comments on the reward function later)*
  - *R((3, 3), N) =* -1/25
  - R (4,1) = +1

MDP

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

$$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \ldots \tag{3.1}$$

Probability of the occurrence of the next state with reward given the previous state

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

The function $p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$ is an ordinary deterministic function of four arguments.

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

MDP

state-transition probabilities $p : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$

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

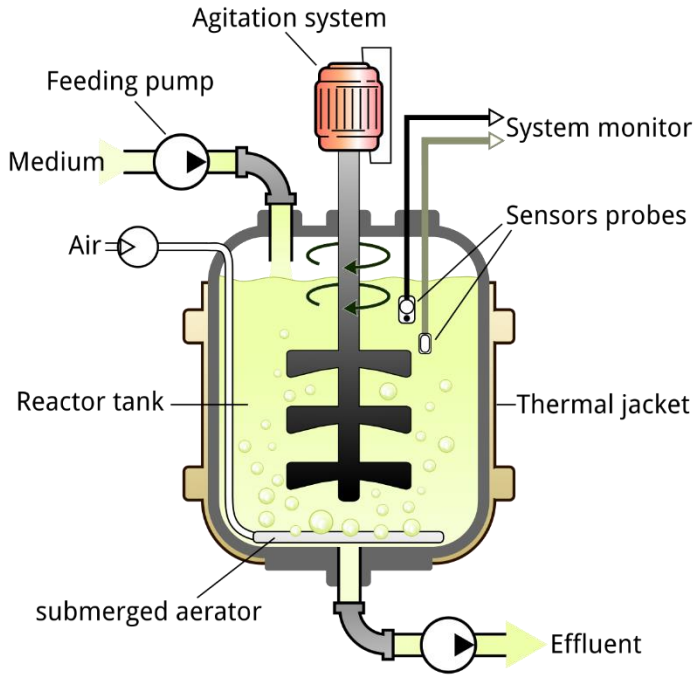
the expected rewards for state–action pairs

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

the expected rewards for state–action–next-state triples

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

Examples



Example 3.1: Bioreactor Suppose reinforcement learning is being applied to determine moment-by-moment temperatures and stirring rates for a bioreactor (a large vat of nutrients and bacteria used to produce useful chemicals). The actions in such an application might be target temperatures and target stirring rates that are passed to lower-level control systems that, in turn, directly activate heating elements and motors to attain the targets. The states are likely to be thermocouple and other sensory readings, perhaps filtered and delayed, plus symbolic inputs representing the ingredients in the vat and the target chemical. The rewards might be moment-by-moment measures of the rate at which the useful chemical is produced by the bioreactor. Notice that here each state is a list, or vector, of sensor readings and symbolic inputs, and each action is a vector consisting of a target temperature and a stirring rate. It is typical of reinforcement learning tasks to have states and actions with such structured representations. Rewards, on the other hand, are always single numbers. ■

Examples

Example 3.2: Pick-and-Place Robot Consider using reinforcement learning to control the motion of a robot arm in a repetitive pick-and-place task. If we want to learn movements that are fast and smooth, the learning agent will have to control the motors directly and have low-latency information about the current positions and velocities of the mechanical linkages. The actions in this case might be the voltages applied to each motor at each joint, and the states might be the latest readings of joint angles and velocities. The reward might be +1 for each object successfully picked up and placed. To encourage smooth movements, on each time step a small, negative reward can be given as a function of the moment-to-moment "jerkiness" of the motion. ■

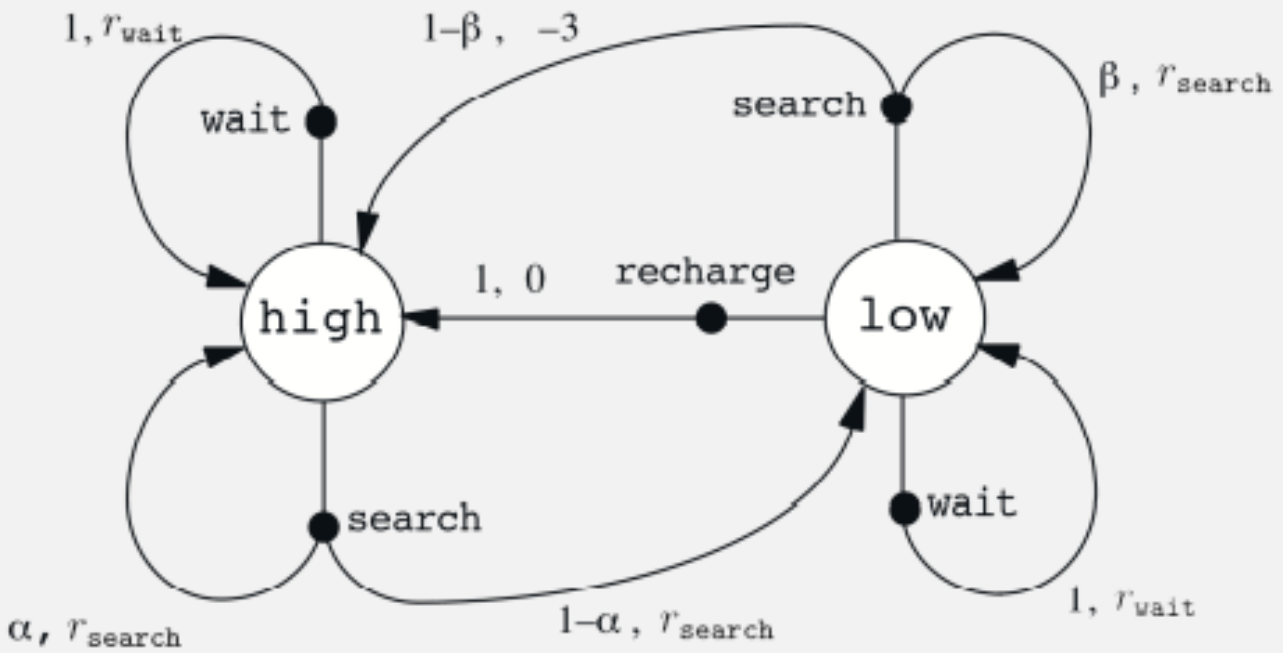
Example

Example 3.3 Recycling Robot

A mobile robot has the job of collecting empty soda cans in an office environment. It has sensors for detecting cans, and an arm and gripper that can pick them up and place them in an onboard bin; it runs on a rechargeable battery. The robot's control system has components for interpreting sensory information, for navigating, and for controlling the arm and gripper. High-level decisions about how to search for cans are made by a reinforcement learning agent based on the current charge level of the battery. To make a simple example, we assume that only two charge levels can be distinguished, comprising a small state set $\mathcal{S} = \{\texttt{high}, \texttt{low}\}$. In each state, the agent can decide whether to (1) actively **search** for a can for a certain period of time, (2) remain stationary and **wait** for someone to bring it a can, or (3) head back to its home base to **recharge** its battery. When the energy level is **high**, recharging would always be foolish, so we do not include it in the action set for this state. The action sets are then $\mathcal{A}(\texttt{high}) = \{\texttt{search}, \texttt{wait}\}$ and $\mathcal{A}(\texttt{low}) = \{\texttt{search}, \texttt{wait}, \texttt{recharge}\}$.

A finite MDP system representation

s	a	s'	$p(s' \mid s, a)$	$r(s, a, s')$
high	search	high	α	$r_{\texttt{search}}$
high	search	low	$1-\alpha$	$r_{\texttt{search}}$
low	search	high	$1-\beta$	-3
low	search	low	β	$r_{\texttt{search}}$
high	wait	high	1	$r_{\texttt{wait}}$
high	wait	low	0	$r_{\texttt{wait}}$
low	wait	high	0	$r_{\texttt{wait}}$
low	wait	low	1	$r_{\texttt{wait}}$
low	recharge	high	1	0
low	recharge	low	0	0



Other examples







- Finance: deciding how much to invest in stock.
- Robotics:
  - A dialogue system to interact with people.
  - Robot bartender.
  - Robot exploration for navigation.
- Agriculture: how much to plant based on weather and soil state.
- Water resources: keep the correct water level at reservoirs.
- Inspection, maintenance and repair: when to replace/inspect based on age, condition, etc.
- Purchase and production: how much to produce based on demand.
- Queues: reduce waiting time.

Goals and Rewards

Reward hypothesis

That all of what we mean by goals and purposes can be well thought of as the maximization of the expected value of the cumulative sum of a received scalar signal (called reward).

Returns and Episodes

Expected return ~ the sum of the rewards

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

T is final time step

Each episode ends in a special state called the terminal state, followed by a reset to a standard starting state or to a sample from a standard distribution of starting states. Even if you think of episodes as ending in different ways, such as winning and losing a game, the next episode begins independently of how the previous one ended. Thus the episodes can all be considered to end in the same terminal state, with different rewards for the different outcomes.

Tasks with episodes of this kind are called episodic tasks. In episodic tasks we sometimes need to distinguish the set of all nonterminal states, denoted *S*, from the set of all states plus the terminal state, denoted S^+. The time of termination, *T*, is a random variable that normally varies from episode to episode.

Discounted return

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

where γ is a parameter, $0 \leq \gamma \leq 1$, called the *discount rate*.

Iteratively or in reinforcement learning way

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

If the reward is a constant of +1

$$G_t = \sum_{k=0}^{\infty} \gamma^k = \frac{1}{1-\gamma}$$

Nhớ giữ sức khỏe! Take care!

Thank you!
Q&A