CS 486/686
Markov Decision Processes I

Yuntian Deng

Lecture 13

RN 17.1 · PM 9.5

Learning goals

- Motivate modeling a decision problem as a **Markov decision process**.
- Describe the **components** of a fully-observable MDP.
- Explain why we use a **discounted** reward function.
- Define the **policy** of an MDP.
- Give examples of how the **reward function** shapes the optimal policy.

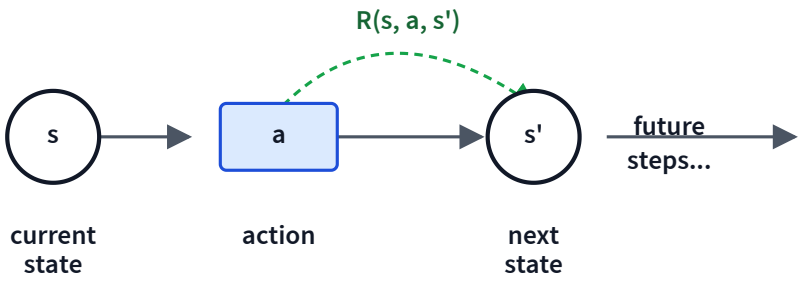
From one-shot decisions to ongoing ones

Two new things

- **Horizon:** the problem may never end.
  - *Infinite horizon:* the process may go on forever.
  - *Indefinite horizon:* the agent will stop eventually, but not at a known time.
- **Rewards along the way**, not just utility at the end.
  - The agent may never reach an "end".
  - Each step's reward includes action costs + bonuses/penalties.

A **Markov decision process (MDP)** formalizes this.

A Markov decision process



An MDP is a 4-tuple (S, A, P, R):

- S: set of states.
- A: set of actions.
- $P(s' \mid s, a)$: transition probabilities. *Stationary*: same for every t.
- $R(s, a, s')$: reward function.

For this lecture we'll use the simpler form $R(s)$ = reward for entering state s.

How do we aggregate rewards over time?

Total reward

$\sum_{t=0}^{\infty} R(S_t)$

May diverge $\Rightarrow$ policies are not comparable.

Average reward

$\lim_{n\to\infty} \frac{1}{n} \sum_{t=0}^{n} R(S_t)$

Zero whenever the total reward is finite.

Discounted reward

$\sum_{t=0}^{\infty} \gamma^t R(S_t), \quad \gamma \in [0, 1)$

Always finite; the standard choice.

The discounted sum is also called the **return** $G(S_0)$.

The discount factor γ

- **Time preference.** A dollar today is worth more than a dollar tomorrow.
- **Finiteness.** If $R(s) \in [-R_{\max}, R_{\max}]$ and $\gamma \in [0, 1)$:

$$\left|\sum_{t=0}^{\infty} \gamma^t R(S_t)\right| \leq \frac{R_{\max}}{1-\gamma}.$$

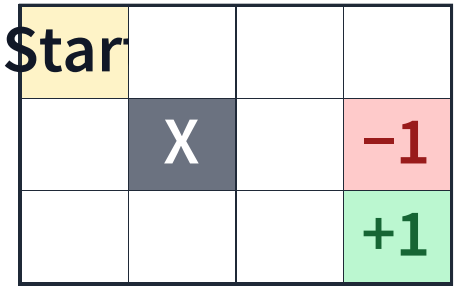
- **Effective horizon.** Roughly $\frac{1}{1-\gamma}$ steps matter (e.g. $\gamma = 0.9 \Rightarrow$ about 10 steps).

Variants

- **Fully-observable MDP** — the agent always knows its current state. (Today.)
- **Partially observable MDP (POMDP)** — combine MDP + HMM; the state is hidden behind observations.

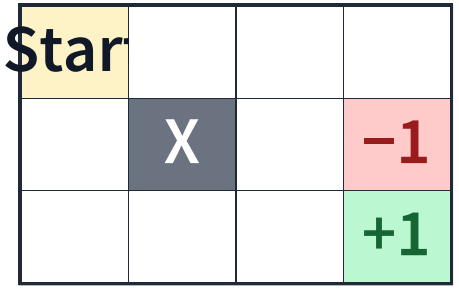
A 3 × 4 grid world

What should the robot do to maximize its rewards?



- Let s_{ij} be row i, column j. s_{11} is the start.
- s_{22} is a wall. s_{24} and s_{34} are goal states — the robot exits the world there.

The grid world as an MDP



Actions: up / down / left / right (in every state).

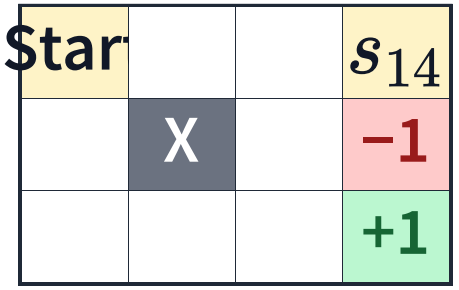
Transition $P(s' \mid s, a)$**:**

- 0.8 — action takes intended effect.
- 0.1 — 90° left of intended.
- 0.1 — 90° right of intended.
- If the robot bumps a wall, it stays put.

Reward $R(s)$**:** $R(s_{24}) = -1, R(s_{34}) = +1$; otherwise $R(s) = -0.04$.

Understanding the transition model

Q1. The robot is in s_{14} (top-right corner) and tries to move right. What is the probability it *stays* in s_{14}?



A. 0.1

B. 0.2

C. 0.8

D. **0.9**

E. 1.0

> **D — 0.9.** Trying right: 0.8 it bumps the wall (stays) + 0.1 it veers up (also bumps wall, stays) = 0.9. The remaining 0.1 veers down to s_{24}.

Why a fixed action sequence is not enough

Q2. In a *deterministic* env, the sequence *down, down, right, right, right* is optimal for the grid world.

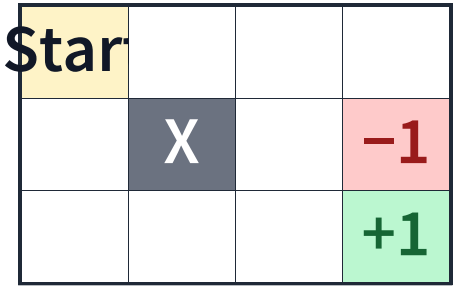
True. Five steps take us from s_{11} to s_{34}.

Q3. In the *stochastic* env, the same sequence may end at *more than one* square.

True. E.g. five right-veers in a row send us to s_{12}.

Q4. So in the stochastic env, a fixed sequence is still good enough.

False. Actions misfire — we need to *re-plan* from whatever state we end up in.



Need an action for **every state** — a **policy**.

A policy

> **Policy**
>
> A **policy** π specifies the agent's action as a function of the current state.

- **Stationary** — depends only on the state: $\pi : S \to A$.
- **Non-stationary** — also depends on the time step: $\pi : S \times \mathbb{N} \to A$.

For an infinite-horizon fully-observable MDP, the optimal policy is **stationary**.

The reward function reshapes the optimal policy

Same dynamics; different $R(s)$ for non-goal states — very different optimal behaviors.

We will look at five qualitatively different reward regions:

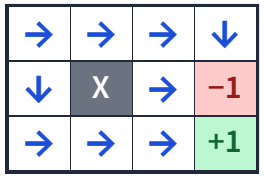
1. $R(s) < -1.6284$ — *life is painful*
2. $-0.4278 < R(s) < -0.0850$ — *life is quite unpleasant*
3. $R(s) = -0.04$ — *life is unpleasant* (the default)
4. $-0.0221 < R(s) \leq 0$ — *life is only slightly dreary*
5. $R(s) > 0$ — *life is GOOD*

Each region balances **risk** (falling into −1) against **reward** (reaching +1 quickly).

Five reward regions, five optimal policies

painful

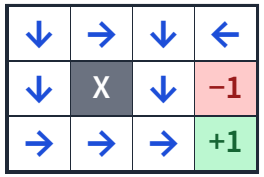
$R < -1.6284$



head to nearest exit, even −1

quite unpleasant

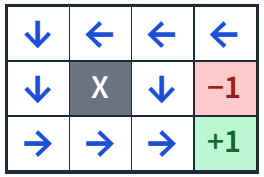
$-0.43 < R < -0.085$



shortcut through s_{13}

unpleasant

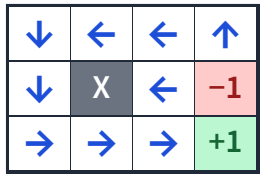
$R = -0.04$



long way around; avoid −1

slightly dreary

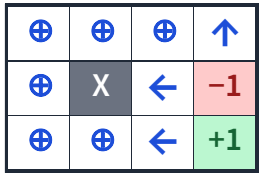
$-0.022 < R \leq 0$



no risk; bang against walls

GOOD

$R > 0$



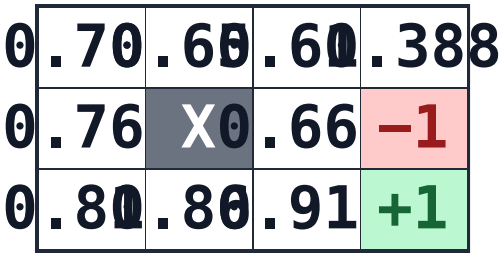
wander forever; avoid both goals

Watch the action at s_{13} (top, third column): it changes with every region.

The value function

Expected utility of a policy

- $V^{\pi}(s)$: expected utility of entering state s and following π thereafter.
- $V^{\star}(s)$: expected utility of entering s and following the **optimal** policy $\pi^{\star}$.



$V^{\star}(s)$ for $\gamma = 1, R(s) = -0.04$ on non-goal states.

The Q-function and the optimal policy

Q-function

$$Q^\star(s, a) = R(s) + \sum_{s'} P(s' \mid s, a)\, V^\star(s')$$

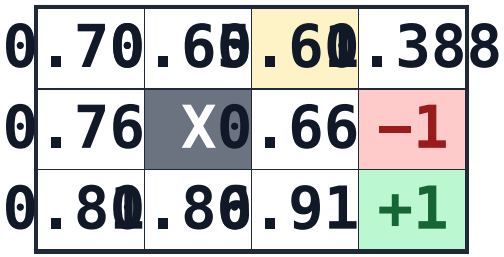
"If I were in s and took action a now, then followed $\pi^\star$ forever after — how good is my future?"

$$\pi^\star(s) = \arg\max_a Q^\star(s, a)$$

Given $V^\star$, the optimal policy is one greedy step away.

Optimal action at s_{13}?

Q5. Using $Q^\star(s,a) = R(s) + \sum_{s'} P(s' \mid s,a) V^\star(s')$, what is $\pi^\star(s_{13})$?



s_{13} is the yellow cell ($V^\star = 0.611$).

Computing each Q-value (4 actions):

```
Left:  -0.04 + 0.8·0.655 + 0.1·0.611 + 0.1·0.660 = 0.6111
Right: -0.04 + 0.8·0.388 + 0.1·0.611 + 0.1·0.660 = 0.3975
Down:  -0.04 + 0.8·0.660 + 0.1·0.388 + 0.1·0.655 = 0.5923
Up:    -0.04 + 0.8·0.611 + 0.1·0.655 + 0.1·0.388 = 0.5531
```

C — Left. Left has the highest $Q^\star$ (0.6111). It moves toward the safer states; the −1 to the right is too close.

Learning goals (recap)

✓ Motivate modeling a decision problem as a Markov decision process.

✓ Describe the **components** of a fully-observable MDP.

✓ Explain why we use a **discounted** reward function.

✓ Define the **policy** of an MDP.

✓ Give examples of how the **reward function** shapes the optimal policy.

Next: how do we compute $V^\star$?

Today: given $V^\star$, the optimal policy is just an argmax. But how do we *find* $V^\star$ in the first place?

L14: **Bellman equations** & **value iteration**.