MDPs: reinforcement learning

Unknown transitions and rewards

Definition: Markov decision process

States: the set of states

$s_{\text{start}} \in$ States: starting state

Actions(s): possible actions from state s

IsEnd(s): whether at end of game

$0 \leq \gamma \leq 1$: discount factor (default: 1)

reinforcement learning!

- In this lecture, we assume that we have an MDP where we neither know the transitions nor the reward functions. We are still trying to maximize expected utility, but we are in a much more difficult setting called **reinforcement learning**.

Mystery game

Example: mystery buttons

For each round $r = 1, 2, \ldots$

- You choose A or B.
- You move to a new state and get some rewards.

Start A B

State: 5,0 Rewards: 0

- To put yourselves in the shoes of a reinforcement learner, try playing the game. You can either push the A button or the B button. Each of the two actions will take you to a new state and give you some reward.
- This simple game illustrates some of the challenges of reinforcement learning: we should take good actions to get rewards, but in order to know which actions are good, we need to explore and try different actions.

From MDPs to reinforcement learning

Markov decision process (offline)

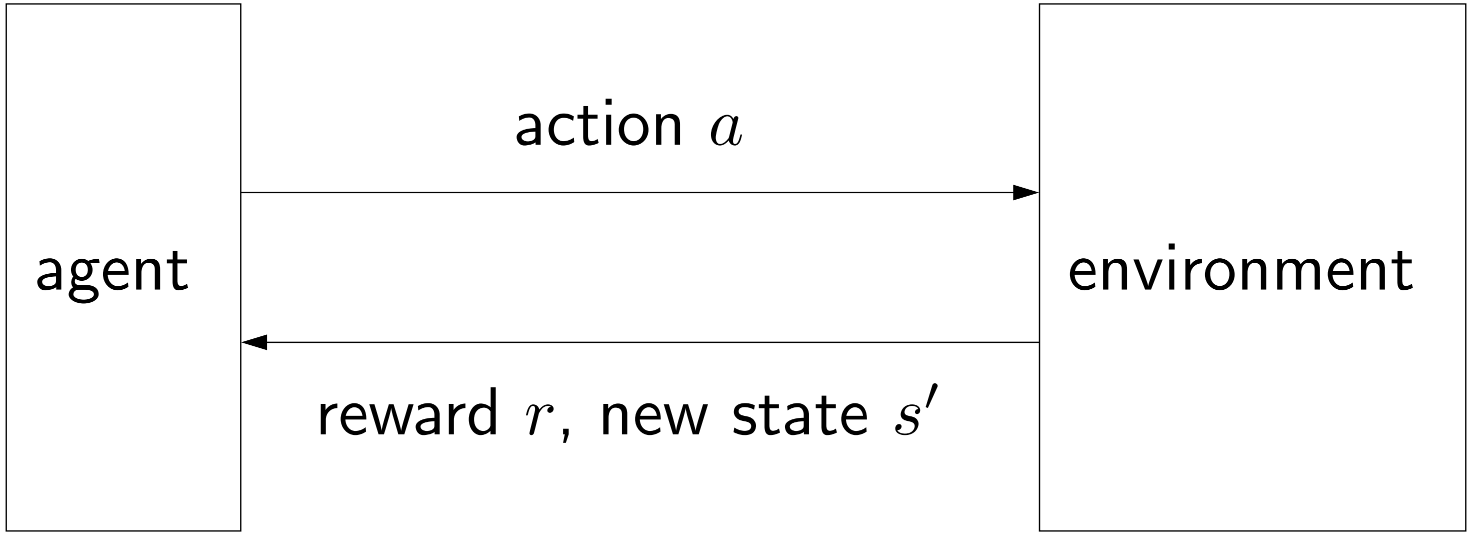
- Have mental model of how the world works.
- Find policy to collect maximum rewards.

Reinforcement learning (online)

- Don't know how the world works.
- Perform actions in the world to find out and collect rewards.

- An important distinction between solving MDPs (what we did before) and reinforcement learning (what we will do now) is that the former is **offline** and the latter is **online**.
- In the former case, you have a mental model of how the world works. You go lock yourself in a room, think really hard, and come up with a policy. Then you come out and use it to act in the real world.
- In the latter case, you don't know how the world works, but you only have one life, so you just have to go out into the real world and learn how it works from experiencing it and trying to take actions that yield high rewards.
- At some level, reinforcement learning is really the way humans work: we go through life, taking various actions, getting feedback. We get rewarded for doing well and learn along the way.

Reinforcement learning framework



Algorithm: reinforcement learning template

```
For $t = 1, 2, 3, \dots$
    Choose action $a_t = \pi_{\text{act}}(s_{t-1})$ (**how?**)
    Receive reward $r_t$ and observe new state $s_t$
    Update parameters (**how?**)
```

- To make the framework clearer, we can think of an **agent** (the reinforcement learning algorithm) that repeatedly chooses an action a_t to perform in the environment, and receives some reward r_t, and information about the new state s_t.
- There are two questions here: how to choose actions (what is π_{act}) and how to update the parameters. We will first talk about updating parameters (the learning part), and then come back to action selection later.

Volcano crossing

Run (or press ctrl-enter)

		-50	20
		-50	
2			

Utility: 2

a	r	s
		(2,1)
W	0	(2,1)
W	0	(2,1)
N	0	(1,1)
W	0	(1,1)
N	0	(1,1)
E	0	(1,2)
S	0	(2,2)
W	0	(2,1)
N	0	(2,2)
N	0	(3,2)
S	0	(3,2)
W	2	(3,1)

- Recall the volcano crossing example from the previous lecture. Each square is a state. From each state, you can take one of four actions to move to an adjacent state: north (N), east (E), south (S), or west (W). If you try to move off the grid, you remain in the same state. The starting state is (2,1), and the end states are the four marked with red or green rewards. Transitions from (s, a) lead where you expect with probability `1-slipProb` and to a random adjacent square with probability `slipProb`.
- If we solve the MDP using value iteration (by setting `numIters` to 10), we will find the best policy (which is to head for the 20). Of course, we can't solve the MDP if we don't know the transitions or rewards.
- If you set `numIters` to zero, we start off with a random policy. Try pressing the Run button to generate fresh episodes. How can we learn from this data and improve our policy?