Sutton and Barto definition of RL

- “RL is learning how to map situations to actions so as to maximize a numerical reward signal. The learner is not told which actions to take, but instead must discover which actions yield the most reward by trying them. In the most challenging cases, actions effect not only the immediate reward but also the next situation, and through that, all subsequent rewards. Trial-and-error search and delayed reward are the two most important distinguishing features of RL.”

Chapter 3: Markov Decision Processes

- $S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, \ldots$
- $p(s', r|s,a) := ?$
- What conditions does $p(s', r|s,a)$ satisfy?
- What is $p(s'|s,a)$ in terms of $p(s', r|s,a)$?
- $r(s,a) := ?$
- What is $r(s,a)$ in terms of $p(s', r|s,a)$?
- In this class, we will mostly work with $p(s'|s,a)$ and $r(s,a)$ rather than with $p(s', r|s,a)$.

What are examples of MDPs in real life?

Example: Simplified Blackjack

- You are initially given one card. You may take more cards, one at a time, if you like.
- jack = queen = king = 10; ace = 11
- Your goal is to get as close to 21 without going over 21.
- You get 0 return if you go over 21; otherwise your return is the total of your cards.
- a = 0 “stick”; a = 1 “hit”
- state = s = total of your cards
- $r(s, 0) = s$ if $s \leq 21$; $r(s,0) = 0$ if $s > 21$; $r(s,1) = 0$.
- MDP terminates after you stick or go over 21.

Blackjack: Example Episodes

- $S_0 = 7, A_0 = 1, R_1 = 0, S_1 = 12, A_1 = 1, R_2 = 0, S_2 = 20, A_2 = 0, R_3 = 20$
- $S_0 = 11, A_0 = 1, R_1 = 0, S_1 = 15, A_1 = 1, R_2 = 0, S_2 = 25, A_2 = 0, R_3 = 0$

Blackjack: Some probabilities and rewards

- p(12|9,hit) =
- r(9,hit) =
- r(16,stick) =
- p(over|19,hit) =
- r(23, stick) =

Example 4.2: Jack's Car Rental Jack manages two locations for a nationwide car rental company. Each day, some number of customers arrive at each location to rent cars. If Jack has a car available, he rents it out and is credited \$10 by the national company. If he is out of cars at that location, then the business is lost. Cars become available for renting the day after they are returned. To help ensure that cars are available where they are needed, Jack can move them between the two locations overnight, at a cost of \$2 per car moved. We assume that the number of cars requested and returned at each location are Poisson random variables, meaning that the probability that the number is n is $\frac{\lambda^n}{n!}e^{-\lambda}$, where λ is the expected number. Suppose λ is 3 and 4 for rental requests at the first and second locations and 3 and 2 for returns. To simplify the problem slightly, we assume that there can be no more than 20 cars at each location (any additional cars are returned to the nationwide company, and thus disappear from the problem) and a maximum of five cars can be moved from one location to the other in one night. We take the discount rate to be $\gamma = 0.9$ and formulate this as a continuing finite MDP, where the time steps are days, the state is the number of cars at each location at the end of the day, and the actions are the net numbers of cars moved between the two locations

Jack's Car Rental

- $s = (s_1, s_2)$ where s_i is the # of cars in store i at end of day
- Action a is an integer, number of cars moved from store 1 to store 2
  - can be negative
- What values can s_1 and s_2 take on?
- Suppose s = (6,3). What possible values can action a take on?
- Suppose s = (6, 19). What possible values can action a take on?
- p((2,3)|(0,0),0) = ? (assume all rental requests are at opening)
- r((0,0), 0) = ?
- r ((1,0), 1) = ?
- For p((2,3)|(1,0), 1), how many returns and requests at each store?

Returns and Episodes

- $G_t := R_{t+1} + R_{t+2} + \ldots + R_T$
- $G_0 := R_1 + R_2 + \ldots + R_T$

Blackjack: Example returns

- $S_0 = 7, A_0 = 1, R_1 = 0, S_1 = 12, A_1 = 1, R_2 = 0, S_2 = 20, A_2 = 0, R_3 = 20$

  $G_0 = 0 + 0 + 20 = 20$

- $S_0 = 11, A_0 = 1, R_1 = 0, S_1 = 15, A_1 = 1, R_2 = 0, S_2 = 25, A_2 = 0, R_3 = 0$

  $G_0 = 0 + 0 + 0 = 0$ episodic task

- In rental car example, every day you get a reward: -2*transfers + 10*rentals

  $G_0 := R_1 + R_2 + R_3 + \ldots\ldots$ continuing task

Discounted Return

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

- What is a simplified expression for G_0 ?
- What is G_0 if $R_t = 8$ for all t ?

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

Policies

- (Stochastic) policy π: $\pi(a|s)$ = probability of choosing action a when in state s under policy π
- A policy is deterministic if it can be expresses as $a = \pi(s)$.
- Quiz: If the current state is $S_t=s$, and actions are selected according to the stochastic policy π, then what is $E_\pi[R_{t+1}|S_t=s]$ in terms of π and r(s,a)?

- Remark: Under fixed policy π, $S_0, S_1, S_2,\ldots$ is a Markov chain. What is $P_\pi(S_{t+1}=s' \mid S_t=s)$ in terms of π and $p(s'|s,a)$?

Value Function

- $v_\pi(s) = E_\pi[\, G_0 \mid S_0 = s \,] = E_\pi\left[\, \sum_{k=0}^{\infty} \gamma^k R_{k+1} \mid S_0 = s \right]$

$$= E_\pi\,[R_1 + \gamma R_2 + \gamma^2 R_3 + \ldots \mid S_0 = s\,]$$

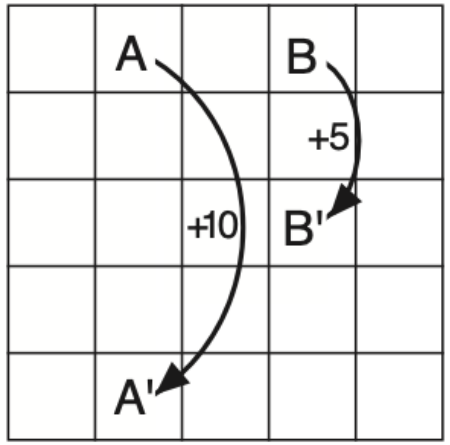
$$v_\pi(s) \doteq \mathbb{E}_\pi[G_t \mid S_t = s] = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,\middle|\, S_t = s\right]$$

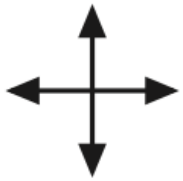
Recursive equation for value function

- $v_\pi(s) = \sum_a \pi(a|s)\,[\, r(s,a) + \gamma \sum_{s'} p(s'|s,a)\, v_\pi(s')\,], \quad s \in \mathcal{S}$
- "Bellman equation" **Quiz: Derive it!**
- $\pi(a|s)$ and $p(s'|s,a)$ are known
- Suppose $N = |\,\mathcal{S}\,|$ states. Then there are N equations with N unknowns: $v_\pi(s)$, $s \in \mathcal{S}$. Can be solved by standard techniques.

Gridworld

cause the agent to move one cell in the respective direction on the grid. 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′.





Actions

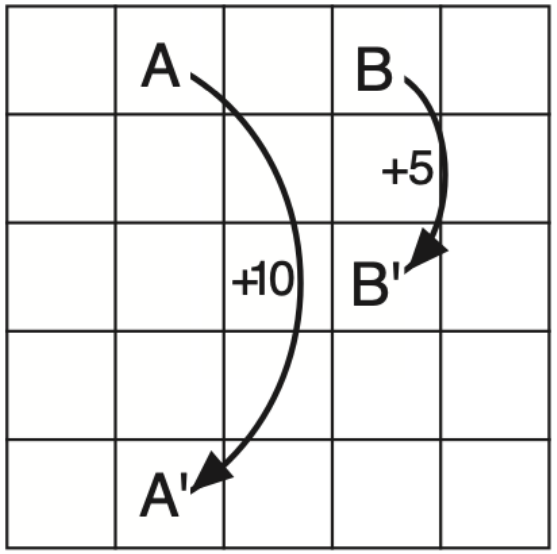
3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

Figure 3.2: Gridworld example: exceptional reward dynamics (left) and state-value function for the equiprobable random policy (right). $\gamma = 0.9$

Optimal Policies

- Say policy π^* is optimal if $v_{\pi^*}(s) \geq v_{\pi}(s)$ for all π and for all states $s \in \mathcal{S}$
- Fact: There exists an optimal policy that is deterministic.
- Denote all optimal policies by π^*
- Thus $v_{\pi^*}(s) = \max_{\pi} v_{\pi}(s)$ for all states $s \in \mathcal{S}$
- Let $v^*(s) := v_{\pi^*}(s)$

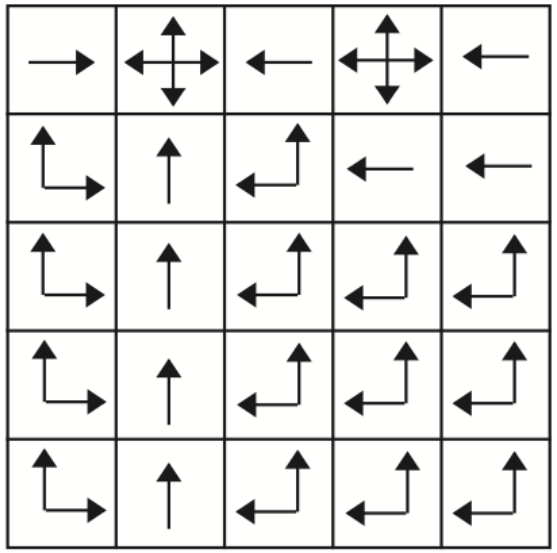
Optimal Policy for Gridworld



Gridworld

22.0	24.4	22.0	19.4	17.5
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7

v_*



π_*

Action-Value Function

- $q_\pi(s,a) := E_\pi[\, G_0 \mid S_0 = s, A_0 = a \,] = E_\pi\left[\, \sum_{k=0}^{\infty} \gamma^k R_{k+1} \mid S_0 = s, A_0 = a \right]$

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

Interpretation of $q_\pi(s,a)$: Expected return when first take action a in state s and then follow policy π

Exercise 3.12 Give an equation for v_π in terms of q_π and π.

Exercise 3.13 Give an equation for q_π in terms of v_π

Recursive equation for action-value function

- $v_\pi(s) = \sum_a \pi(a|s)\,[\, r(s,a) + \gamma \sum_{s'} p(s'|s,a)\, v_\pi(s')\,], \quad s \in \mathcal{S}$
- $q_\pi(s,a) = r(s,a) + \gamma \sum_{s'} p(s'|s,a) \sum_{a'} \pi(a'|s')\, q_\pi(s',a')\,], \quad s \in \mathcal{S}$
- Both are “Bellman equations”

Optimal policies and state-action value function

- $q^*(s,a) := \max_\pi q_\pi(s,a)$
- Policy that maximizes $q_\pi(s,a)$ will first obtain expected reward r(s,a), then move to state s', and then follow optimal policy:
- $q^*(s,a) = r(s,a) + \gamma \sum_{s'} p(s'|s,a) v^*(s')$