Markov Decision Processes

Infinite-horizon discounted MDPs

An MDP $M = (S, A, P, R, \gamma)$

- State space S.
- Action space A.

We will only consider discrete and finite spaces in this course (w/o losing much).

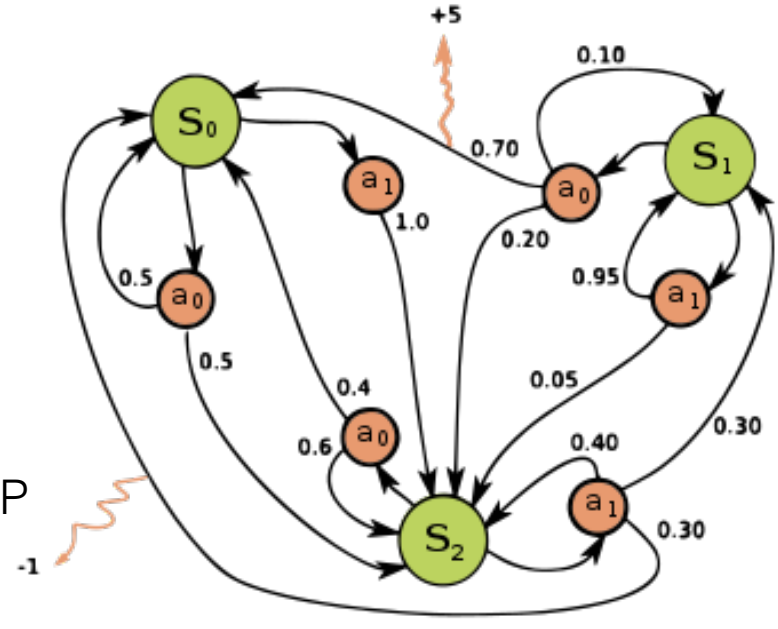
- Transition function $P : S \times A \to \Delta(S)$. $\Delta(S)$ is the probability simplex over S, i.e., all non-negative vectors of length $|S|$ that sums up to 1
- Reward function R: $S \times A \to \mathbb{R}$. (deterministic reward function)
- Discount factor $\gamma \in [0,1)$
- The agent starts in some state s_1, takes action a_1, receives reward $r_1 = R(s_1, a_1)$, transitions to $s_2 \sim P(s_1, a_1)$, takes action a_2, so on so forth — the process continues **forever**
- Objective: (discounted) expected total reward
  - other terms used: return, value, utility, long-term reward, etc

Additional/alternative notations

- The probability of transitioning to a particular state: $P(s' \mid s, a)$
- Sometimes reward is random and/or depends on the next-state, e.g., $R(s, a, s')$, or $R(s, a)$ is a random variable
  - The most general case: given (s, a), (r, s') is drawn from some joint distribution (Sutton & Barto 2nd ed; see link on course website)
  - When we consider random rewards we will assume a simplified case: r and s' are independent conditioned on (s, a)
  - Such differences usually don't matter—we will revisit later
- Sometimes the available actions depend on the state, A_s
  - Again, the theory/algs developed for fixed action spaces usually extend to state-dependent action spaces

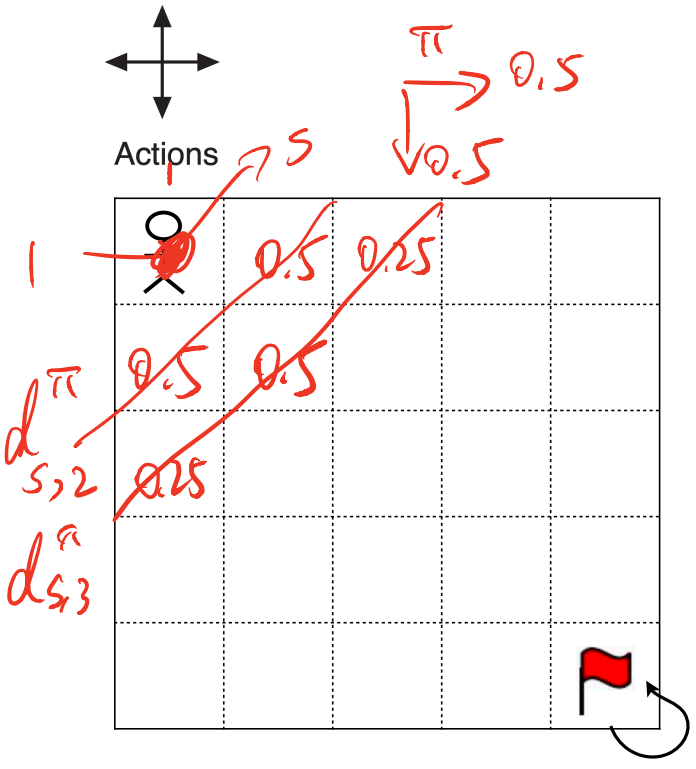
Toy Example

- $P(s_0 \mid s_2, a_0) = 0.4$
- $P(s_1 \mid s_2, a_0) = 0$
- $P(s_2 \mid s_2, a_0) = 0.6$
- …



- If there is only 1 action, the MDP becomes a Markov chain (associated with a reward function)
- As another special case, if there are multiple actions but transition is deterministic, the MDP becomes a directed graph

Example: Gridworld



- State: grid No. (or integer x, y coordinates)
- Action: N, S, E, W
- Dynamics:
  - Deterministic transitions to the adjacent grid in the direction of action in most cases.
  - Keep in the current state if moving towards wall or having reached goal
- Reward: 0 in the goal state and -1 everywhere else
- Discount factor γ: 0.99

"Absorbing state": emulate the termination of a task in infinite-horizon MDPs

Why discounting?

- When defining the MDP (esp. the reward function & the discount factor/horizon), you should make sure that the total expected reward of a policy precisely reflects how you like that policy. **Don't worry about learning yet.**
  - e.g., in the grid world example, if there is no discounting (γ = 1), the total expected reward is the negative expected number of steps before reaching the goal
  - After you've learned a policy, **you should evaluate the policy using this reward function / horizon**
- So why discounting?
  - In the previous example, $\gamma = 1$ allows some strategies to obtain -∞ expected return—we don't like infinities

Why discounting? (cont.)

As a mathematical convenience…

- We introduce a discount factor close to (but smaller than) 1, so that even the agent moves in circles, the expected total reward is still finite.
- On the other hand, for most reasonable policies (esp. the near-optimal ones), the total reward is still approximately the negative total number of steps—**our training objective** closely tracks **what we really care about**
- Reason 2: discounting + infinite horizon = stationary optimal policy & value functions, again a mathematical convenience
- Reason 3: heavy discounting (small γ) yields faster planning / learning (we will see)
- Discounting does have economic interpretations, but they are seldom relevant in RL

Homework

- Homework 0 (handled via Gradescope; more info to come on Campuswire)
  - Due 1 week from now, before class starts
  - Main purpose: remind you some maths & set the expectation about how mathy this course will be
  - Does not count towards final grades but (1) we will look at it, and (2) you have to submit in time otherwise you **use up** your change to drop the lowest score for homework (see late policy)
- Reading: Sutton & Barto, Chap 3
  - Note difference in notations
  - Focus on understanding how to translate between Table 3.1 and Fig 3.2. Also Chap 3.4; Example 3.6; Example 3.10

Value and policy

- Want to take actions in a way that maximizes value (or return):

$$\mathbb{E}\left[\sum_{t=1}^{\infty}\gamma^{t-1}r_t\right]$$

  - This value depends on where you start and how you act
- Often assume boundedness of rewards: $r_t \in [0, R_{\max}]$
  - What's the range of $\mathbb{E}\left[\sum_{t=1}^{\infty}\gamma^{t-1}r_t\right]$? $\left[0, \frac{R_{\max}}{1-\gamma}\right]$
- A (deterministic) policy π: $S \to A$ describes how the agent acts: at state s_t, chooses action $a_t = \pi(s_t)$.
- More generally, the agent may choose actions randomly (π: $S \to \Delta(A)$), or even in a way that varies across time steps ("non-stationary policies")
- Define $V^{\pi}(s) = \mathbb{E}\left[\sum_{t=1}^{\infty}\gamma^{t-1}r_t \,\middle|\, s_1 = s, \pi\right]$

$r_t \in [0, R_{max}]$.

Claim: $\forall \gamma \in [0,1)$. $\sum_{t=1}^{\infty} \gamma^{t-1} r_t \in [0, \frac{R_{max}}{1-\gamma}]$.

$\sum_{t=1}^{\infty} \gamma^{t-1} r_t \geq 0$. ✓

$\sum_{t=1}^{\infty} \gamma^{t-1} r_t \leq \sum_{t=1}^{\infty} \gamma^{t-1} R_{max}$

$= R_{max} \cdot \sum_{t=1}^{\infty} \gamma^{t-1}$.

$1 + \gamma + \gamma^2 + \gamma^3 + \gamma^4 + \dots$

$= R_{max} \cdot \frac{1}{1-\gamma}$.

$\lim_{T \to \infty} \sum_{t=1}^{T} \gamma^{t-1} = \lim_{T \to \infty} \frac{1-\gamma^T}{1-\gamma}$.

$= \frac{1}{1-\gamma}$.

$\mathbb{E}[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \mid s_1 = s, \pi] =: V^{\pi}(s)$.

$s_1, a_1, r_1, s_2, a_2, r_2, s_3, a_3, r_3, \dots$

s $\pi(s)$ $R(s, a)$ $\sim P(\cdot \mid s, a)$ $\pi(s_2)$

$s_1, s_2, \dots, s_t$: r.v.

s, s': realization.

Bellman equation for policy evaluation

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \;\middle|\; s_1 = s, \pi\right]$$

$$= \mathbb{E}\left[r_1 + \sum_{t=2}^{\infty} \gamma^{t-1} r_t \;\middle|\; s_1 = s, \pi\right]$$

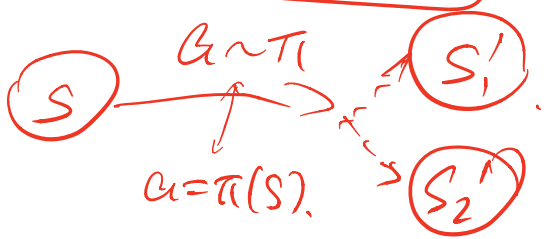
$$= R(s, \pi(s)) + \sum_{s' \in \mathcal{S}} P(s'|s, \pi(s))\, \mathbb{E}\left[\gamma \sum_{t=2}^{\infty} \gamma^{t-2} r_t \;\middle|\; s_1 = s, s_2 = s', \pi\right]$$

$$= R(s, \pi(s)) + \sum_{s' \in \mathcal{S}} P(s'|s, \pi(s))\, \mathbb{E}\left[\gamma \sum_{t=2}^{\infty} \gamma^{t-2} r_t \;\middle|\; s_2 = s', \pi\right]$$

$$= R(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, \pi(s))\, \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \;\middle|\; s_1 = s', \pi\right]$$

$$= R(s, \pi(s)) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s, \pi(s))\, V^\pi(s')$$

$$= R(s, \pi(s)) + \gamma \langle P(\cdot|s, \pi(s)),\, V^\pi(\cdot) \rangle$$



$V^\pi(s) = \$100$. $r = R(s, \pi(s)) = \$20$.

$\gamma \cdot \mathbb{E}_{s' \sim P(\cdot|s,\pi(s))}[V^\pi(s')] = \$80$.

$V^\pi(s) = \mathbb{E}[r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots \mid s_1 = s, \pi]$.

$= \mathbb{E}[r_1 \mid s_1 = s, \pi] + \mathbb{E}[\gamma r_2 + \gamma^2 r_3 + \cdots \mid s_1 = s, \pi]$

$= R(s, \pi(s)) + \gamma \mathbb{E}[r_2 + \gamma r_3 + \gamma^2 r_4 + \cdots \mid s_1 = s, \pi]$

integrate out s_2

$= R(s, \pi(s)) + \gamma \mathbb{E}\big[\mathbb{E}[r_2 + \gamma r_3 + \gamma^2 r_4 + \cdots \mid s_1 = s, s_2, \pi]\big]$

Markov property.

$\mathbb{E}[r_2 + \gamma r_3 + \gamma^2 r_4 + \cdots \mid s_1, \pi]$

$= V^\pi(s_2)$

$= R(s, \pi(s)) + \gamma \mathbb{E}[V^\pi(s_2) \mid s_1 = s, \pi]$.

$= R(s, \pi(s)) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,\pi(s))}[V^\pi(s')]$.

$$\mathbb{E}_{s' \sim P(\cdot|s,\pi(s))}[V^\pi(s')] = \sum_{s' \in S} P(s'|s,\pi(s)) \cdot V^\pi(s')$$

Bellman equation for policy evaluation

$$V^\pi(s) = R(s,\pi(s)) + \gamma \langle P(\cdot \mid s,\pi(s)), V^\pi(\cdot) \rangle$$

Matrix form: define

- V^π as the $|S| \times 1$ vector $[V^\pi(s)]_{s \in S}$
- R^π as the vector $[R(s,\pi(s))]_{s \in S}$
- P^π as the matrix $[P(s' \mid s,\pi(s))]_{s \in S,\, s' \in S}$

$$V^\pi = R^\pi + \gamma P^\pi V^\pi$$

$$(I - \gamma P^\pi) V^\pi = R^\pi$$

$$V^\pi = (I - \gamma P^\pi)^{-1} R^\pi$$

This is always invertible (Proof?), i.e., solution to Bellman equation is always unique

W.t.s. $I-\gamma P^\pi$ is invertible.

$\Leftrightarrow$ $\forall x \neq \vec{0}$. $(I-\gamma P^\pi)x \neq \vec{0}$.

$\forall x \in \mathbb{R}^d$.

$x = (x_1, x_2, x_3, \cdots, x_d)$

$\|x\|_\infty = \max_i |x_i|$.

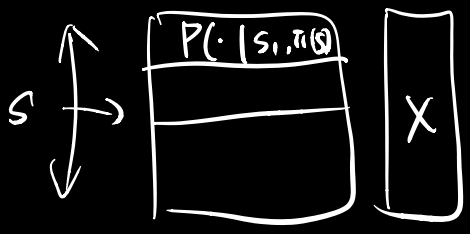
$\rightarrow \|(I-\gamma P)x\|_\infty = \|x - \gamma P^\pi x\|_\infty$.

$\geq \|x\|_\infty - \|\gamma P^\pi x\|_\infty$.

$= \|x\|_\infty - \gamma \|P^\pi x\|_\infty$. $\geq \|x\|_\infty - \gamma \cdot \|x\|_\infty$

$= (1-\gamma) \cdot \|x\|_\infty$

$\|P^\pi x\|_\infty \leq \|x\|_\infty$.



$[P^\pi x]_s = \mathop{\mathbb{E}}_{i \sim P(\cdot|s,\pi(s))} [x_i]$

Some remarks about rewards

- For more general formulations of rewards (e.g., random rewards, rewards that depend on s', etc)

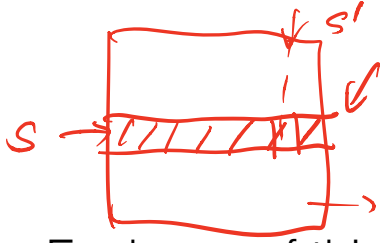
$$V^\pi(s) = \mathbb{E}_{r,s'|s,\pi(s)}[r + \gamma V^\pi(s')]$$

  - Can also be translated to an equivalent problem where rewards deterministically depends on (s,a): $R(s,a) := \mathbb{E}_{r|s,a}[r]$

- In infinite-horizon discounted MDPs, shifting the reward function by a constant for every state-action pair does not change anything

  - If we increase the reward function by c, you are just earning an extra c units of "background" rewards however you behave

  - All value functions increase by $c/(1-\gamma)$



Interpretation of $(I - \gamma P^\pi)^{-1}$

$V^\pi = (I-\gamma P^\pi)^{-1} R^\pi$

$\in \mathbb{R}^{|S| \times |S|}$.

Each row of this matrix (indexed by s) is the unnormalized **discounted state occupancy**, whose (s')-th entry is $d_s^\pi(s')/(1-\gamma)$, where

$$d_s^\pi(s') := (1-\gamma) \cdot \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} \mathbb{I}[s_t = s'] \,\middle|\, s_1 = s, \pi\right]$$

$\sum_{t=1}^{\infty} \gamma^{t-1} = \frac{1}{1-\gamma}$

- $\mathbb{I}[\cdot]$ is the indicator function: =1 if (.) is true, and =0 if false.
- Each row of $(I - \gamma P^\pi)^{-1}$ is like a distribution vector—except that the entries sum up to $1/(1-\gamma)$. Multiplying by $1-\gamma$ normalizes it
- Can also be interpreted as the value function of indicator reward function: $d_s^\pi(s') = (1-\gamma)V^\pi(s)$, where V^π is defined wrt the reward function that is equal to 1 at state s' and 0 elsewhere
- Can similarly define state-action occupancy

Optional: matrix expression of $d_s^\pi(s',a')$

$$d_s^\pi(s',a') = (1-\gamma) \cdot \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} \mathbb{I}[s_t = s', a_t = a'] \,\middle|\, s_1 = s, \pi\right]$$

More on State occupancy

- Alternative way of defining state occupancy: let $d^{\pi}_{s,t}$ be a (distribution) vector such that $d^{\pi}_{s,t}(s') = \Pr[s_t = s' \mid s_1 = s, \pi]$
  - It's the t-th step state distribution induced by starting at state s and following policy π
- The discounted occupancy is $d^{\pi}_{s} = (1-\gamma)\sum_{t=1}^{\infty} \gamma^{t-1} d^{\pi}_{s,t}$
- The s-th row of $(I - \gamma P^{\pi})^{-1}$ is $d^{\pi}_{s}/(1-\gamma)$, because:
  - Let $e_s = [0 \quad \cdots \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0]^{\top}$, where the 1 is at the s-th entry
  - s-th row of $(I - \gamma P^{\pi})^{-1}$ is

$$e_s^{\top}(I - \gamma P^{\pi})^{-1} = e_s^{\top}\left(\sum_{t=1}^{\infty}(\gamma P^{\pi})^{t-1}\right)$$

$$= \sum_{t=1}^{\infty} \gamma^{t-1} e_s^{\top}(P^{\pi})^{t-1} = \sum_{t=1}^{\infty} \gamma^{t-1}(d^{\pi}_{s,t})^{\top}$$

Alternative formula for value function

- Some further exercise that helps understanding
- $V^\pi(s) = e_s^\top (I - \gamma P^\pi)^{-1} R^\pi$

$$= e_s^\top \Big(\sum_{t=1}^{\infty} \gamma^{t-1} (P^\pi)^{t-1}\Big) R^\pi$$

$$= \sum_{t=1}^{\infty} \gamma^{t-1} \langle d_{s,t}^\pi, R^\pi \rangle$$

$\mathbb{E}[r_t \mid s_1 = s, \pi]$

$V^\pi = (I - \gamma P^\pi)^{-1} R^\pi$.

$V^\pi(s)$.

$= e_s^\top V^\pi$

$V^\pi(s) = \frac{1}{1-\gamma} \langle d_s^\pi, R^\pi \rangle$.

$[1]$? $= [0, 0, \cdots, \underset{\downarrow}{1}, 0, \cdots 0]$ ·

$$V^\pi(s) = R(s, \pi(s)) + \gamma \mathbb{E}_{s' \sim P(s, \pi(s))}[V^\pi(s')]$$

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \mid s_1 = s, \pi\right]$$

$$= \mathbb{E}[\quad \cdot \quad]$$

$r_t \in [0, \frac{R_{max}}{1-\gamma}]$

Monter Carlo : w.t. know $\mathbb{E}_{x \sim p}[f(x)]$.

$x_1, x_2, x_3, \cdots, x_n \overset{i.i.d.}{\sim} p$.

$$\frac{1}{n} \sum_{i=1}^{n} f(x_i) \approx \mathbb{E}_{x \sim p}[f(x)]$$

---

start from s, take action according to π.

generate traj. $s_1, a_1, r_1, s_2, a_2, r_2, \cdots,$

$$\sum_{t=1}^{\infty} \gamma^{t-1} r_t$$ or

Optimality

- For infinite-horizon discounted MDPs, there always exists a stationary and deterministic policy that is optimal for all starting states simultaneously
  - Proof: Puterman'94, Thm 6.2.7 (reference due to Shipra Agrawal)
- Let π^* denote this optimal policy, and $V^* := V^{\pi^*}$
- Bellman Optimality Equation:

$$V^\star(s) = \max_{a \in A} \left(R(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)} \left[V^\star(s') \right] \right)$$

- If we know V^*, how to get π^* ?
- Easier to work with Q-values: $Q^*(s, a)$, as $\pi^\star(s) = \arg\max_{a \in A} Q^\star(s,a)$

$$Q^\star(s,a) = R(s,a) + \gamma \mathbb{E}_{s' \sim P(s,a)} \left[\max_{a' \in A} Q^\star(s',a') \right]$$

- Note that $V^\star(s) = \max_{a \in A} Q^\star(s,a) = Q^\star(s, \pi^\star(s))$

$V^* = V^{\pi^*}$.

$\Rightarrow V^*(s) = R(s, \pi^*(s)) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,\pi^*(s))}[V^*(s')]$

$\exists \pi^* : S \to A$. s.t. $\forall s$. $\forall \pi$

$V^{\pi^*}(s) \geq V^{\pi}(s)$.

$d_0 \in \Delta(S)$.

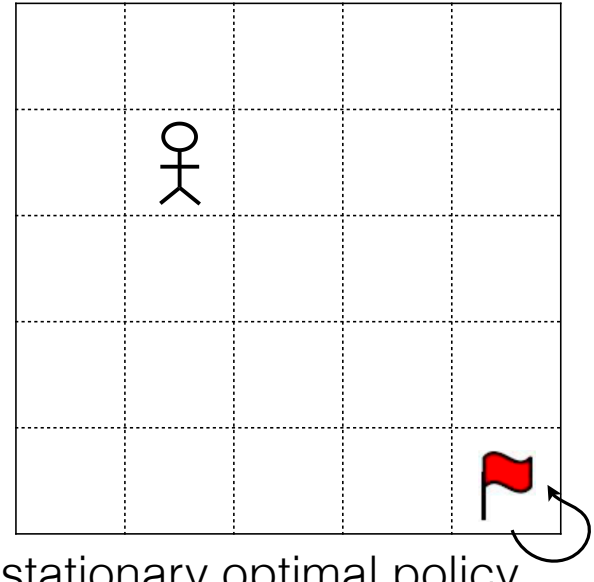
maximize $\mathbb{E}\left[\sum_{t=1}^{\infty} \gamma^{t-1} r_t \,|\, s_1 \sim d_0, \pi\right]$.

$\forall d_0, \quad \forall \pi$.

$\mathbb{E}_{s_1 \sim d_0}[V^{\pi^*}(s_1)] \geq \mathbb{E}_{s_1 \sim d_0}[V^{\pi}(s_1)]$.

Optimality

- Similar to the Bellman equations for policy evaluation, the Bellman optimality equations also have unique solutions V* and Q*
- But π^* may not be unique
- e.g., in the state shown in the figure, the optimal policy can take any of the following
  - Go right
  - Go down
  - Any probability distribution over right and down (stochastic policy)



- Remark: the fact that a deterministic & stationary optimal policy exists does NOT mean that all optimal policies have to be this way (they can be stochastic and/or non-stationary)

(Policy-specific) Q-functions

$$Q^{\pi}(s,a) := \mathbb{E}\left[\sum_{t=1}^{\infty}\gamma^{t-1}r_t \,\middle|\, s_1 = s, a_1 = a; \pi\right]$$

- $V^{\pi}(s) = Q^{\pi}(s, \pi(s))$
  - When π is stochastic, RHS becomes $\mathrm{E}_{a\sim\pi(\cdot\,|\,s)}[Q^{\pi}(s, a)]$
  - We will often abbreviate $Q^{\pi}(s, \pi(s))$ as $Q^{\pi}(s, \pi)$
- Also satisfies a similar Bellman equation
  $Q^{\pi}(s,a) = R(s,a) + \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}\left[Q^{\pi}(s',\pi)\right]$
- Compare to the optimality equation

  $$Q^{\star}(s,a) = R(s,a) + \gamma\mathbb{E}_{s'\sim P(\cdot|s,a)}\left[\max_{a'\in A} Q^{\star}(s',a')\right]$$

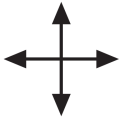
  - The only difference is in the action-selection operator at the next-state s': if you choose a' greedily (max operator), you get Q^{*}; if you choose a' according to π, you get Q^{π}

Clarification on definition of value functions

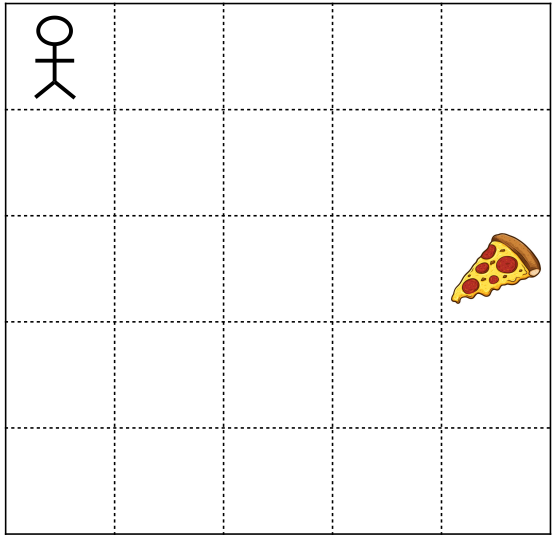
- $V^{\pi}(s)$, $Q^{\pi}(s, a)$, $V^{*}(s)$, $Q^{*}(s, a)$
- they are the expected return when starting from of $s_1{=}s$ and following the behaviors below:
  - $V^{\pi}(s)$: $a_1{\sim}\pi(s_1)$, $a_2{\sim}\pi(s_2)$, $a_3{\sim}\pi(s_3)$, …
  - $Q^{\pi}(s, a)$: $a_1 = a$, $a_2{\sim}\pi(s_2)$, $a_3{\sim}\pi(s_3)$, …
  - $V^{*}(s) = V^{\pi^{*}}(s)$: $a_1{\sim}\pi^{*}(s_1)$, $a_2{\sim}\pi^{*}(s_2)$, $a_3{\sim}\pi^{*}(s_3)$, …
  - $Q^{*}(s, a) = Q^{\pi^{*}}(s, a)$: $a_1 = a$, $a_2{\sim}\pi^{*}(s_2)$, $a_3{\sim}\pi^{*}(s_3)$, …
- The policy (or *) on the superscript denotes the long-term behavior
- For Q values, the second argument of the function specifies the first action; all future actions are according to the policy on the superscript

Some Subtleties in Formulating Problems as MDPs

Defining the state (in toy problems)



Actions

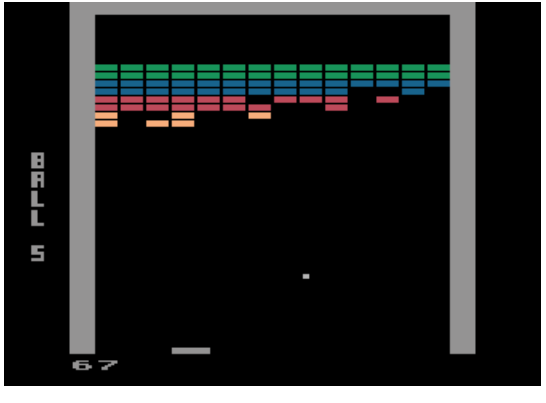


- The agent can move around just as in the previous example
- Reward: +1 when reaching food for the first time, and 0 otherwise
  - i.e., food is consumable
- How to define state?
  - The status of the food (consumed or not) also needs to be part of the state representation

- In general, make sure that you can fully determine reward and (the distribution of) next-state using (s, a) alone without other info
- The tricky part of this example is that dynamics are Markovian in the coordinates, but reward function is not

Defining state beyond toy problems

- In toy problems, we can give any information about the system to the agent, so that the agent always has a (Markov) state
- In reality, agent receives limited sensory inputs every time step
- i.e., a trajectory looks like $o_1, a_1, r_1, o_2, a_2, r_2, o_3 \ldots$
- We say that the problem is Markov (in *o*), if the future observations and rewards are independent of the past conditioned on the current observation (and *o* is state)
- But this is generally not true…
- Need to construct an (approximate) state from **history**
- Will come back to this when we discuss POMDPs later in this course



Alternative notions of horizons

- It is often more natural to model the problem as **indefinite-horizon** undiscounted MDPs
  - Some states are considered terminal
  - Trajectory does not go on forever—it stops at terminal states
  - The length of a trajectory is not fixed—it's random and/or may depend on the policy
  - e.g., navigation task: goal state is terminal, -1 per step before you terminate
  - Natural to model ***episodic*** tasks in this way
- Bellman equations remain the same for non-terminal states
- For terminal states, $V(s) = 0$ (regardless of V^{π} or V^{*})

Fixed-horizon MDPs

- Specified by (S, A, R, P, H)
- All trajectories end in precisely H steps
- No terminal states; termination is enforced "externally"
- Optimal policy may additionally depend on the time step
  - When selling goods, may price things differently near the end of the season, even if we assume the demand doesn't change
  - In the navigation example, when you are too far away from goal to reach it within the remaining steps, any action is optimal (there is nothing you can do)
- So do (policy-specific/optimal) value functions: $V^{\pi}_{H+1}(s) \equiv 0$

$$V^{\pi}_{h}(s) = R(s, \pi(s)) + \mathbb{E}_{s' \sim P(s,a)}[V^{\pi}_{h+1}(s')]$$

- Similar to the discounted setting, shifting rewards by constant doesn't change the relative preference over policies (all value functions increase by cH)

Translating between indefinite-horizon and fixed horizon MDPs

- "=>": If trajectory length has an upper bound, set H to be this upper bound and add a dummy absorbing state; if a trajectory terminates early, loop in the dummy state for the remaining steps
- "<=": Augment state space: include time step as part of the state representation, and define all states at time H+1 as terminal, i.e., If P is the dynamics of the fixed horizon MDP, then define a new transition function

$$P'((s', h') \mid (s, h), a) = \mathbb{I}[h' = h + 1] \cdot P(s' \mid s, a)$$

  - State augmentation is a common trick when we convert between different formulations in RL (e.g., non-Markov problems become Markov if we view history as state)

Translating between different formulations

- For discounted & fixed horizon, shifting rewards don't matter
- What about indefinite-horizon?
  - Think about what happens with the navigation example
  - Shift -1 to 1: you are rewarded if you don't reach the goal!
- Why is that?
  - For indefinite-horizon problems, since trajectory length is a variable, the positivity of reward actually carries a meaning: "+" means "live longer", and "-" means "finish faster"
  - For the other two formulations, such meaning doesn't exist
  - In-class exercise: formulate the navigation problem as a fixed horizon problem, then shift the reward. Explain why shifting rewards is fine in this case.

The navigation example

- Original formulation: indef-horizon
  - -1 reward per step everywhere
  - Goal is terminal
  - Cannot shift reward
- Translate into fixed-horizon
  - Make goal state absorbing
  - 0 reward in goal and -1 everywhere else
  - Terminate in H steps
- Shift reward?
  - Have to shift everywhere, **including the goal**
  - e.g., if shift by 1, then +1 in goal and 0 everywhere else
  - No penalty, but being rewarded when looping in goal