10-701: Introduction to Machine Learning Lecture 10: Reinforcement Learning

Henry Chai & Zack Lipton

10/02/23

Front Matter

- Announcements
  - HW2 released 9/20, due 10/4 (Wednesday) at 11:59 PM
  - HW3 released 10/4 (Wednesday), due 10/11 at 11:59 PM
  - Project details will be released on 10/13
    - You will have a choice between a more research-based project and a more implementation-focused project
    - You must work on the project in groups of 3 or 4; **you may not work on the project alone.**
- Recommended Readings
  - Mitchell, Chapter 13

Learning Paradigms

- Supervised learning - $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(n)}, y^{(n)}\right)\right\}_{n=1}^{N}$
  - Regression - $y^{(n)} \in \mathbb{R}$
  - Classification - $y^{(n)} \in \{1, \dots, C\}$
- Reinforcement learning - $\mathcal{D} = \left\{\left(\boldsymbol{s}^{(n)}, \boldsymbol{a}^{(n)}, r^{(n)}\right)\right\}_{n=1}^{N}$

Source: https://techobserver.net/2019/06/argo-ai-self-driving-car-research-center/

Source: https://www.wired.com/2012/02/high-speed-trading/

Reinforcement Learning: Examples

Source: https://www.cnet.com/news/boston-dynamics-robot-dog-spot-finally-goes-on-sale-for-74500/

Source: https://twitter.com/alphagomovie

AlphaGo

Source: https://www.youtube.com/watch?v=WXuK6gekU1Y&ab_channel=DeepMind

Reinforcement Learning: Problem Formulation

- State space, $\mathcal{S}$
- Action space, $\mathcal{A}$
- Reward function
  - Stochastic, $p(r \mid s, a)$
  - Deterministic, $R \colon \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$
- Transition function
  - Stochastic, $p(s' \mid s, a)$
  - Deterministic, $\delta \colon \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$

Reinforcement Learning: Problem Formulation

- Policy, $\pi : \mathcal{S} \rightarrow \mathcal{A}$
  - Specifies an action to take in *every* state
- Value function, $V^{\pi}: \mathcal{S} \rightarrow \mathbb{R}$
  - Measures the expected total payoff of starting in some state s and executing policy π, i.e., in every state, taking the action that π returns

Toy Example

- $\mathcal{S}$ = all empty squares in the grid
- $\mathcal{A}$ = {up, down, left, right}
- Deterministic transitions
- Rewards of +1 and -1 for entering the labelled squares
- Terminate after receiving either reward

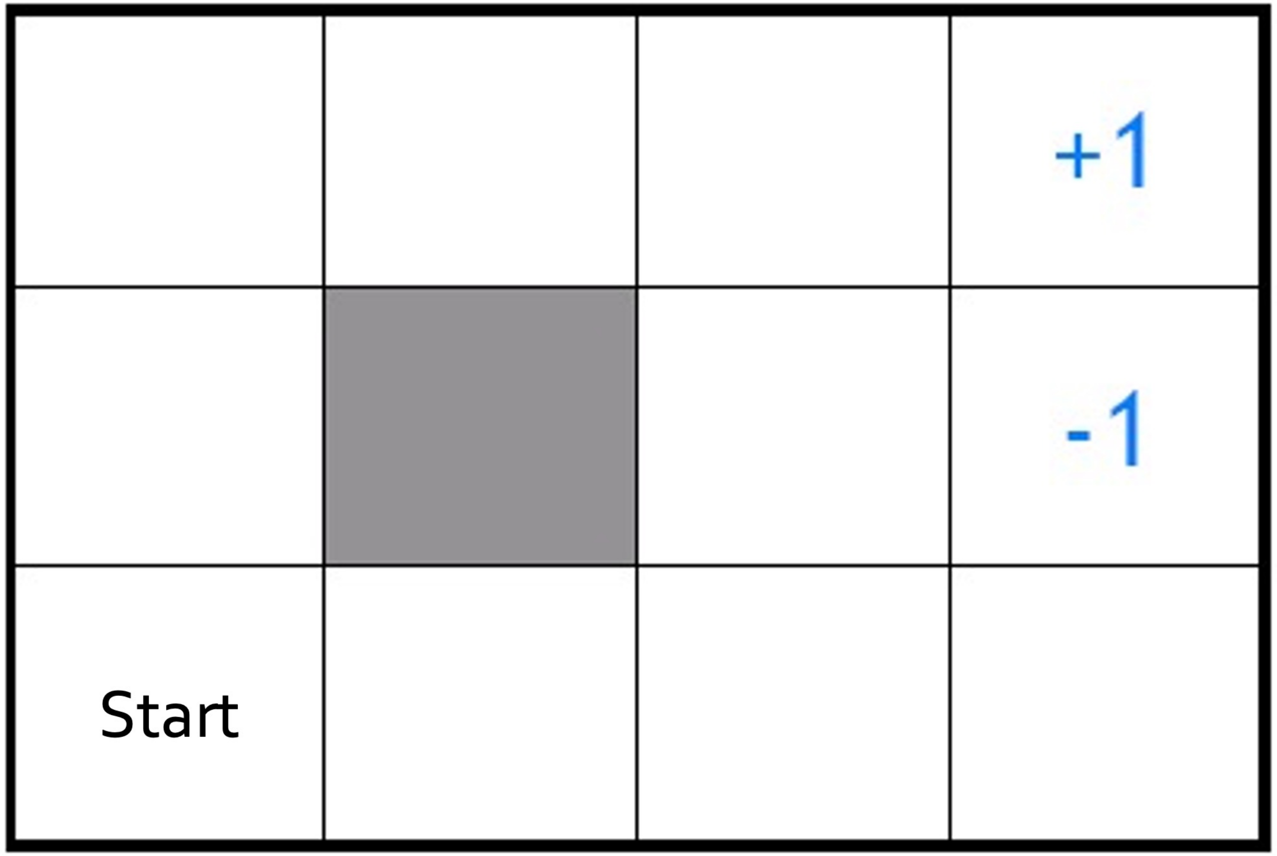


Figure courtesy of Eric Xing

Toy Example

Is this policy optimal?

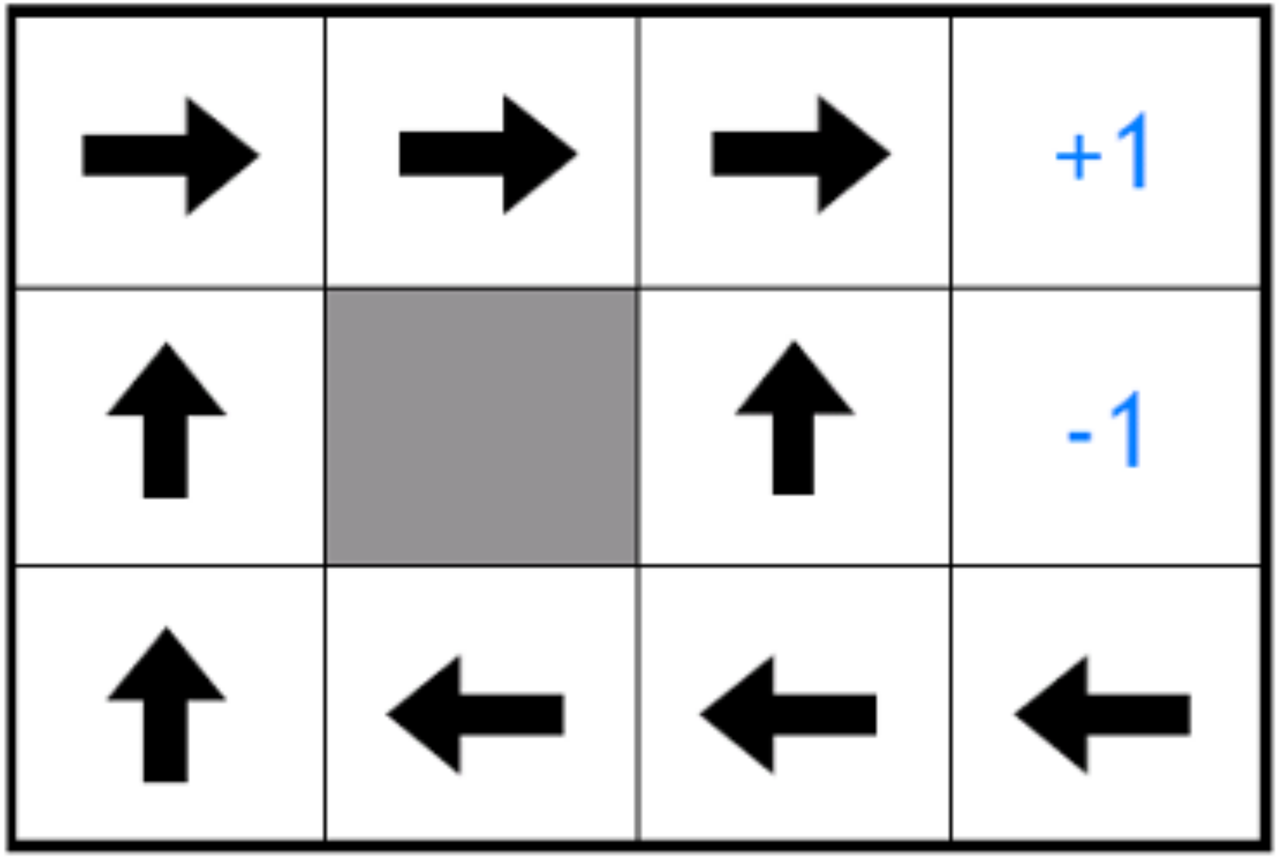


Figure courtesy of Eric Xing

Toy Example

Optimal policy given a reward of -2 per step

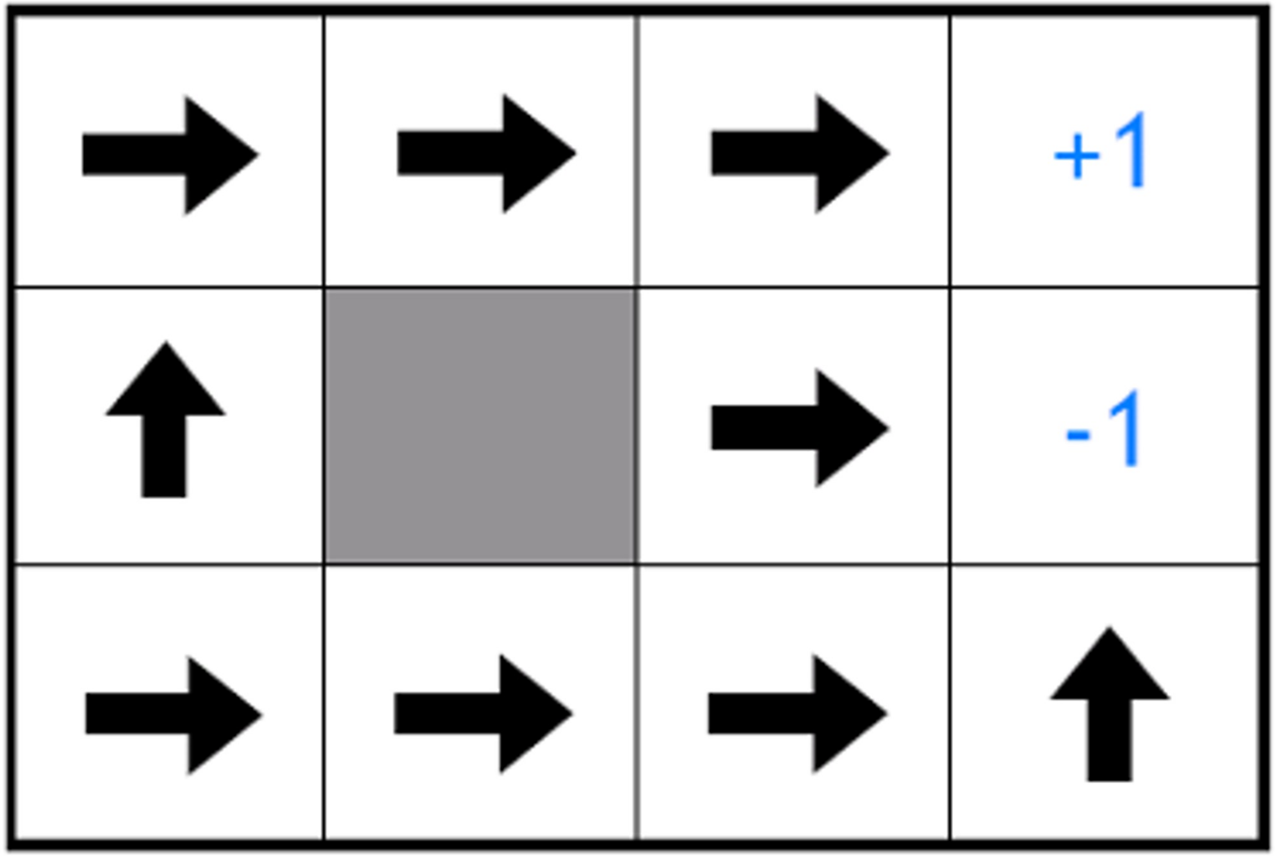


Figure courtesy of Eric Xing

Toy Example

Optimal policy given a reward of -0.1 per step

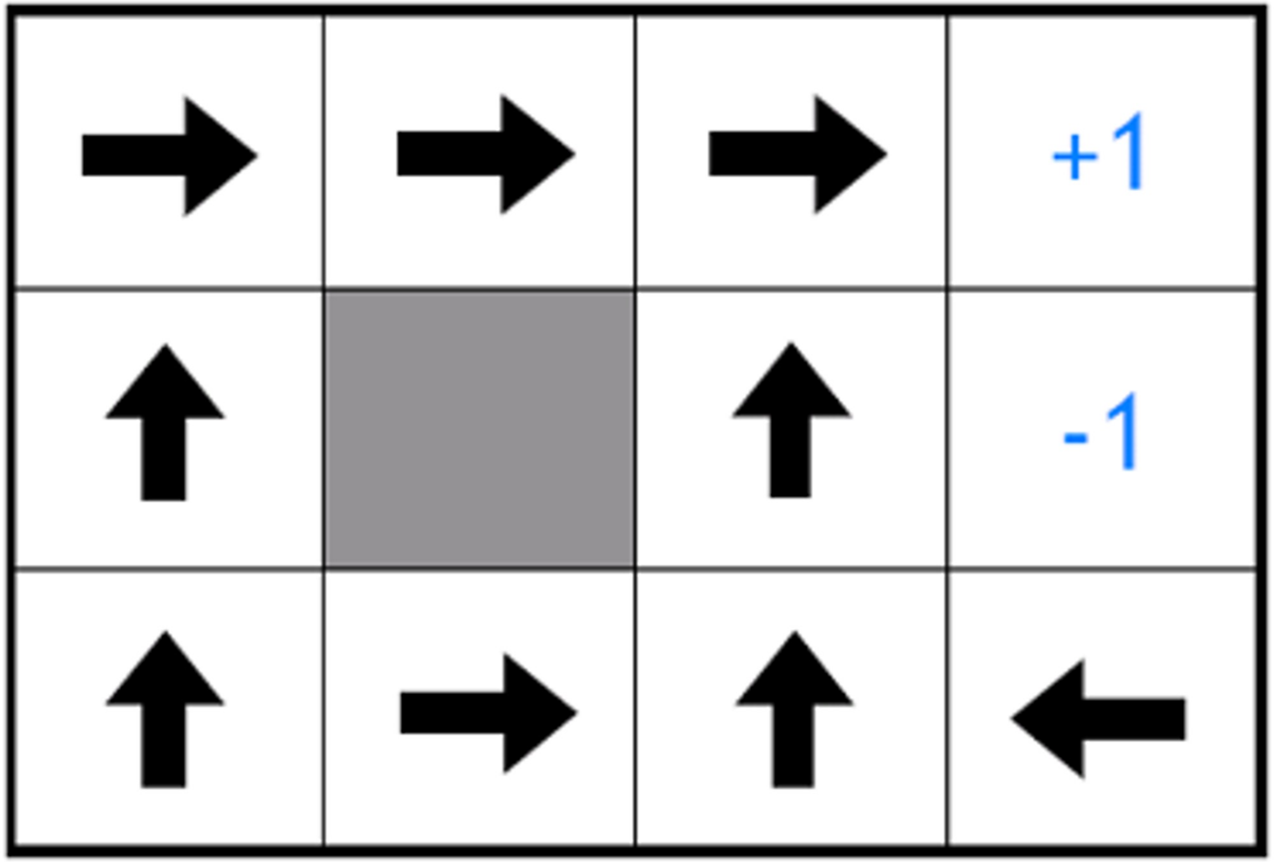


Figure courtesy of Eric Xing

Markov Decision Process (MDP)

- Assume the following model for our data:

1. Start in some initial state s_0
2. For time step t:
   1. Agent observes state s_t
   2. Agent takes action $a_t = \pi(s_t)$
   3. Agent receives reward $r_t \sim p(r \mid s_t, a_t)$
   4. Agent transitions to state $s_{t+1} \sim p(s' \mid s_t, a_t)$
3. Total reward is $\sum_{t=0}^{\infty} \gamma^t r_t$

γ = discount factor

- MDPs make the *Markov assumption*: the reward and next state only depend on the current state and action.

Reinforcement Learning: 3 Key Challenges

1. The algorithm has to gather its own training data
2. The outcome of taking some action is often stochastic or unknown until after the fact
3. Decisions can have a delayed effect on future outcomes (exploration-exploitation tradeoff)

MDP Example: Multi-armed bandit

- Single state: $|\mathcal{S}| = 1$
- Three actions: $\mathcal{A} = \{1, 2, 3\}$
- Deterministic transitions
- Rewards are stochastic

MDP Example: Multi-armed bandit

Bandit 1	Bandit 2	Bandit 3
1	2	1
1	0	0
1	0	3
1	0	2
0	0	4
1	2	2
0	0	1
1	2	4
1	0	0
1	2	3
1	0	3
0	0	1

Reinforcement Learning: Objective Function

Stochastic transitions
Deterministic Rewards

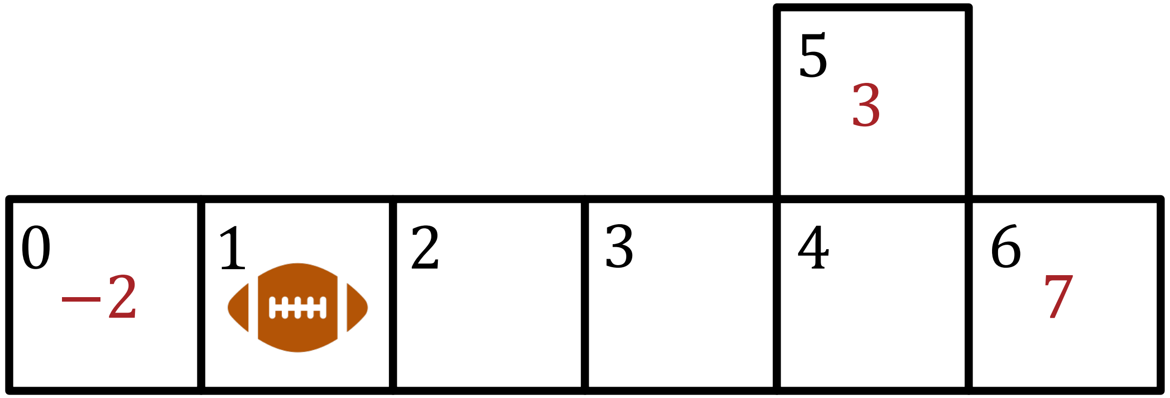
- Find a policy $\pi^* = \underset{\pi}{\text{argmax}}\ V^\pi(s)\ \forall\, s \in \mathcal{S}$
- $V^\pi(s) = \mathbb{E}[$*discounted* total reward of starting in state s and executing policy π forever$]$

$$= E_{p(s'|s,a)}\left[R(s_0, \pi(s_0)) + \gamma R(s_1, \pi(s_1)) + \gamma^2 R(s_2, \pi(s_2)) + \ldots\right]$$

$$= \sum_{t=0}^{\infty} \gamma^t E_{p(s'|s,a)}\left[R(s_t, \pi(s_t))\right]$$

where $0 \leq \gamma < 1$ is the discount factor

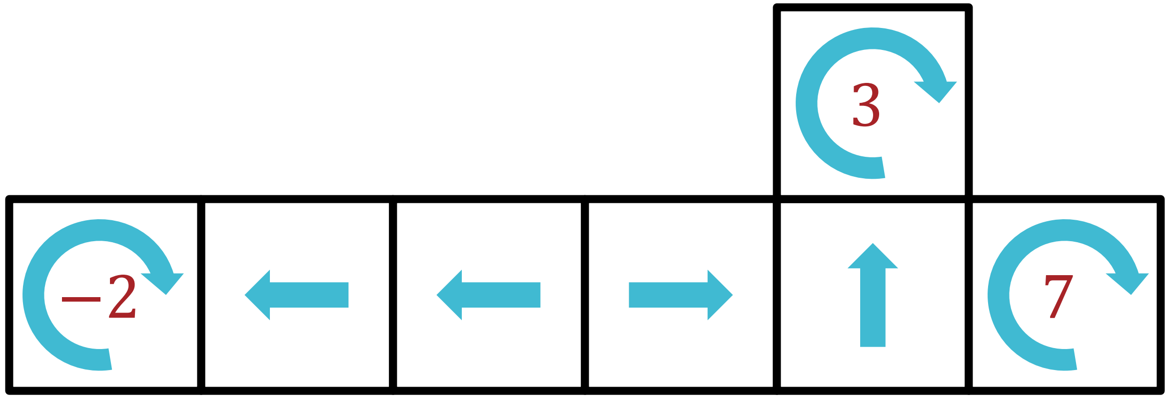
Value Function: Example



$$R(s,a) = \begin{cases} -2 \text{ if entering state 0 (safety)} \\ 3 \text{ if entering state 5 (field goal)} \\ 7 \text{ if entering state 6 (touch down)} \\ 0 \text{ otherwise} \end{cases}$$

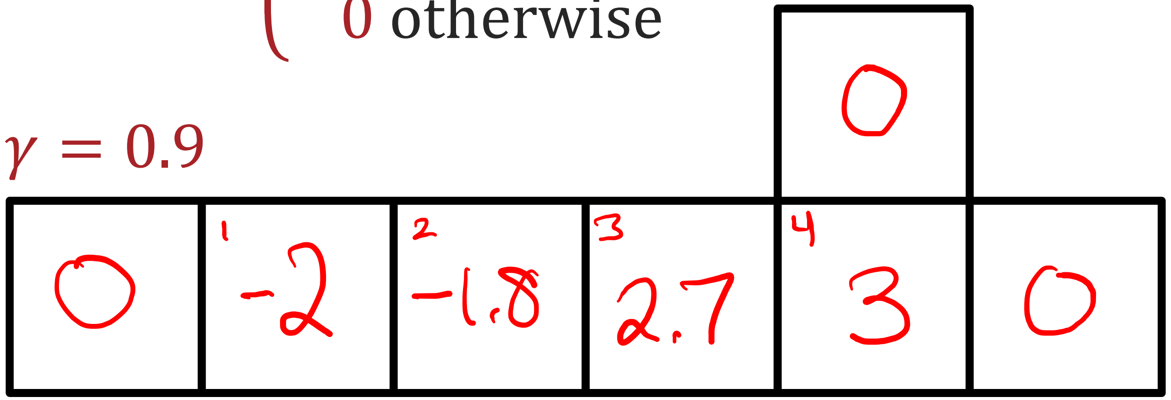
$$\gamma = 0.9$$

Value Function: Example

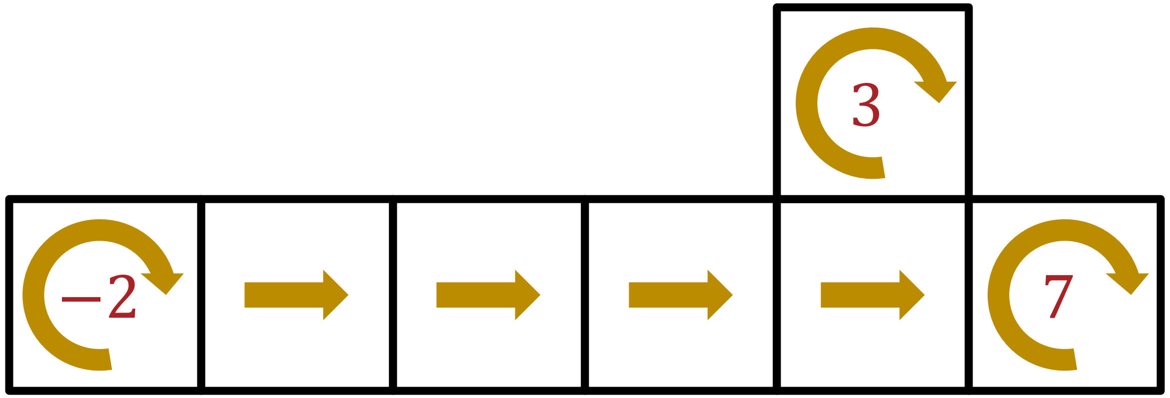


$$R(s,a) = \begin{cases} -2 \text{ if entering state 0 (safety)} \\ 3 \text{ if entering state 5 (field goal)} \\ 7 \text{ if entering state 6 (touch down)} \\ 0 \text{ otherwise} \end{cases}$$

$\gamma = 0.9$

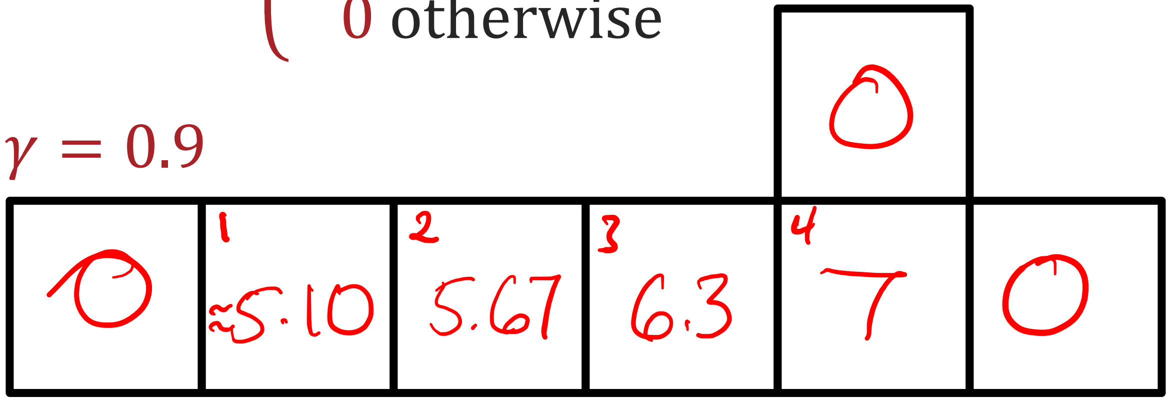


Value Function: Example



$$R(s,a) = \begin{cases} -2 \text{ if entering state 0 (safety)} \\ 3 \text{ if entering state 5 (field goal)} \\ 7 \text{ if entering state 6 (touch down)} \\ 0 \text{ otherwise} \end{cases}$$

$\gamma = 0.9$



stochastic transitions

Value Function

- $V^{\pi}(s) = \mathbb{E}[$discounted total reward of starting in state s and executing policy π forever$]$

$$= \mathbb{E}[R(s_0, \pi(s_0)) + \gamma R(s_1, \pi(s_1)) + \gamma^2 R(s_2, \pi(s_2)) + \cdots \mid s_0 = s]$$

$$= R(s, \pi(s)) + \gamma \mathbb{E}[R(s_1, \pi(s_1)) + \gamma R(s_2, \pi(s_2)) + \ldots \mid s_0 = s]$$

$$= R(s, \pi(s)) + \gamma \sum_{s_1 \in S} \underbrace{p(s_1 \mid s, \pi(s))} \big(R(s_1, \pi(s_1)) + \gamma \mathbb{E}[R(s_2, \pi(s_2)) + \cdots \mid s_1]\big)$$

Value Function

- $V^{\pi}(s) = \mathbb{E}[$discounted total reward of starting in state s and executing policy π forever$]$

$$= \mathbb{E}[R(s_0, \pi(s_0)) + \gamma R(s_1, \pi(s_1)) + \gamma^2 R(s_2, \pi(s_2)) + \cdots \mid s_0 = s]$$

$$= R(s, \pi(s)) + \gamma \mathbb{E}[R(s_1, \pi(s_1)) + \gamma R(s_2, \pi(s_2)) + \ldots \mid s_0 = s]$$

$$= R(s, \pi(s)) + \gamma \sum_{s_1 \in \mathcal{S}} p(s_1 \mid s, \pi(s)) \Big(R(s_1, \pi(s_1))$$

$$+ \gamma \underline{\mathbb{E}[R(s_2, \pi(s_2)) + \cdots \mid s_1]}\Big)$$

Value Function

- $V^{\pi}(s) = \mathbb{E}[$discounted total reward of starting in state s and executing policy π forever$]$

$$= \mathbb{E}[R(s_0, \pi(s_0)) + \gamma R(s_1, \pi(s_1)) + \gamma^2 R(s_2, \pi(s_2)) + \cdots \mid s_0 = s]$$

$$= R(s, \pi(s)) + \gamma \mathbb{E}[R(s_1, \pi(s_1)) + \gamma R(s_2, \pi(s_2)) + \ldots \mid s_0 = s]$$

$$= R(s, \pi(s)) + \gamma \sum_{s_1 \in \mathcal{S}} p(s_1 \mid s, \pi(s)) \big(R(s_1, \pi(s_1)) + \gamma \mathbb{E}[R(s_2, \pi(s_2)) + \cdots \mid s_1]\big)$$

Value Function

- $V^{\pi}(s) = \mathbb{E}[$discounted total reward of starting in state s and executing policy π forever$]$

$$= \mathbb{E}[R(s_0, \pi(s_0)) + \gamma R(s_1, \pi(s_1)) + \gamma^2 R(s_2, \pi(s_2)) + \cdots \mid s_0 = s]$$

$$= R(s, \pi(s)) + \gamma \mathbb{E}[R(s_1, \pi(s_1)) + \gamma R(s_2, \pi(s_2)) + \ \ldots \mid s_0 = s]$$

$$= R(s, \pi(s)) + \gamma \sum_{s_1 \in \mathcal{S}} p(s_1 \mid s, \pi(s)) \Big(R(s_1, \pi(s_1)) + \gamma \mathbb{E}[R(s_2, \pi(s_2)) + \cdots \mid s_1]\Big)$$

Value Function

- $V^\pi(s) = \mathbb{E}[$discounted total reward of starting in state s and executing policy π forever$]$

$$= \mathbb{E}[R(s_0, \pi(s_0)) + \gamma R(s_1, \pi(s_1)) + \gamma^2 R(s_2, \pi(s_2)) + \cdots \mid s_0 = s]$$

$$= R(s, \pi(s)) + \gamma \mathbb{E}[R(s_1, \pi(s_1)) + \gamma R(s_2, \pi(s_2)) + \ldots \mid s_0 = s]$$

$$= R(s, \pi(s)) + \gamma \textstyle\sum_{s_1 \in \mathcal{S}} p(s_1 \mid s, \pi(s))\big(R(s_1, \pi(s_1)) + \gamma \mathbb{E}\big[R(s_2, \pi(s_2)) + \cdots \mid s_1\big]\big)$$

$$\underbrace{V^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s_1 \in \mathcal{S}} p(s_1 \mid s, \pi(s)) V^\pi(s_1)}_{\text{Bellman equations}}$$

Optimality

- Optimal value function:

$$V^*(s) = \max_{a \in \mathcal{A}} R(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s, a) V^*(s')$$

  - System of $|\mathcal{S}|$ equations and $|\mathcal{S}|$ variables

- Optimal policy:

$$\pi^*(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \underbrace{R(s,a)}_{\text{Immediate reward}} + \underbrace{\gamma \sum_{s' \in \mathcal{S}} p(s' \mid s, a) V^*(s')}_{\text{(Discounted) Future reward}}$$

Fixed Point Iteration

- Iterative method for solving a system of equations
- Given some equations and initial values

$$x_1 = f_1(x_1, \dots, x_n)$$
$$\vdots$$
$$x_n = f_n(x_1, \dots, x_n)$$
$$x_1^{(0)}, \dots, x_n^{(0)}$$

- While not converged, do

$$x_1^{(t+1)} \leftarrow f_1\left(x_1^{(t)}, \dots, x_n^{(t)}\right)$$
$$\vdots$$
$$x_n^{(t+1)} \leftarrow f_n\left(x_1^{(t)}, \dots, x_n^{(t)}\right)$$

Fixed Point Iteration: Example

$\left(\frac{1}{3}\right)\left(\frac{-1}{2}\right)+\left(\frac{1}{2}\right)=\frac{2}{6}=\frac{1}{3}$

$$x_1 = x_1 x_2 + \frac{1}{2}$$

$x_1^{(1)} = 1/2$

$$x_2 = -\frac{3x_1}{2}$$

$-\frac{3}{2}\left(\frac{1}{3}\right) = -\frac{1}{2}$

$x_2^{(1)} = 0$

$$x_1^{(0)} = x_2^{(0)} = 0$$

$$\hat{x}_1 = \frac{1}{3}, \hat{x}_2 = -\frac{1}{2}$$

t	$x_1^{(t)}$	$x_2^{(t)}$
0	0	0
1	0.5	0
2	0.5	-0.75
3	0.125	-0.75
4	0.4063	-0.1875
5	0.4238	-0.6094
6	0.2417	-0.6357
7	0.3463	-0.3626
8	0.3744	-0.5195
9	0.3055	-0.5616
10	0.3284	-0.4582
11	0.3495	-0.4926
12	0.3278	-0.5243
13	0.3281	-0.4917
14	0.3386	-0.4922
15	0.3333	-0.5080

Value Iteration

- Inputs: $R(s, a)$, $p(s' \mid s, a)$
- Initialize $V^{(0)}(s) = 0 \ \forall \ s \in \mathcal{S}$ (or randomly) and set $t = 0$
- While not converged, do:
  - For $s \in \mathcal{S}$

$$V^{(t+1)}(s) \leftarrow \max_{a \in \mathcal{A}} \underbrace{R(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s, a) V^{(t)}(s')}_{Q(s,a)}$$

  - $t = t + 1$
- For $s \in \mathcal{S}$

$$\pi^*(s) \leftarrow \operatorname*{argmax}_{a \in \mathcal{A}} R(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s, a) V^{(t)}(s')$$

- Return π^*

Synchronous Value Iteration

- Inputs: $R(s, a)$, $p(s' \mid s, a)$
- Initialize $V^{(0)}(s) = 0 \ \forall \ s \in \mathcal{S}$ (or randomly) and set $t = 0$
- While not converged, do:
  - For $s \in \mathcal{S}$
    - For $a \in \mathcal{A}$

$$Q(s, a) = R(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s, a) V^{(t)}(s')$$

    - $V^{(t+1)}(s) \leftarrow \max_{a \in \mathcal{A}} Q(s, a)$
  - $t = t + 1$
- For $s \in \mathcal{S}$

$$\pi^*(s) \leftarrow \underset{a \in \mathcal{A}}{\operatorname{argmax}} \ R(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s, a) V^{(t)}(s')$$

- Return π^*

Asynchronous Value Iteration

- Inputs: $R(s, a)$, $p(s' \mid s, a)$
- Initialize $V^{(0)}(s) = 0 \; \forall \, s \in \mathcal{S}$ (or randomly)
- While not converged, do:
  - For $s \in \mathcal{S}$
    - For $a \in \mathcal{A}$

$$Q(s,a) = R(s,a) + \gamma \overset{V(\delta(s,a))}{\cancel{\sum_{s' \in \mathcal{S}} p(s' \mid s,a) V(s')}}$$

    - $V(s) \leftarrow \max_{a \in \mathcal{A}} Q(s,a)$

- For $s \in \mathcal{S}$

$$\pi^*(s) \leftarrow \underset{a \in \mathcal{A}}{\operatorname{argmax}} \; R(s,a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s,a) V(s')$$

- Return π^*

Value Iteration Theory

- **Theorem 1**: Value function convergence

V will converge to V^* if each state is "visited" infinitely often (Bertsekas, 1989)

- **Theorem 2**: Convergence criterion

$$\text{if } \max_{s \in \mathcal{S}} \left| V^{(t+1)}(s) - V^{(t)}(s) \right| < \epsilon,$$

$$\text{then } \max_{s \in \mathcal{S}} \left| V^{(t+1)}(s) - V^*(s) \right| < \underbrace{\frac{2\epsilon\gamma}{1-\gamma}}$$ (Williams & Baird, 1993)

- **Theorem 3**: Policy convergence

The "greedy" policy, $\pi(s) = \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ Q(s, a)$, converges to the optimal π^* in a finite number of iterations, often before the value function has converged! (Bertsekas, 1987)

Policy Iteration

- Inputs: $R(s, a)$, $p(s' \mid s, a)$
- Initialize π randomly
- While not converged, do:
  - Solve the Bellman equations defined by policy π

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

  - Update π

$$\pi(s) \leftarrow \underset{a \in \mathcal{A}}{\operatorname{argmax}}\ R(s, a) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid s, a) V^{\pi}(s')$$

- Return π

Policy Iteration Theory

- In policy iteration, the policy improves in each iteration.
- Given finite state and action spaces, there are finitely many possible policies
  - Thus, the number of iterations needed to converge is bounded!
- Value iteration takes $O(|\mathcal{S}|^2|\mathcal{A}|)$ time / iteration
- Policy iteration takes $O(|\mathcal{S}|^2|\mathcal{A}| + |\mathcal{S}|^3)$ time / iteration
  - However, empirically policy iteration requires fewer iterations to converge

Two big Q's

1. What can we do if the reward and/or transition functions/distributions are unknown?

2. How can we handle infinite (or just very large) state/action spaces?

Key Takeaways

- In reinforcement learning, we assume our data comes from a Markov decision process
- The goal is to compute an optimal policy or function that maps states to actions
- Value function can be defined in terms of values of all other states; this is called the Bellman equations
- If the reward and transition functions are known, we can solve for the optimal policy (and value function) using value or policy iteration
  - Both algorithms are instances of fixed point iteration and are guaranteed to converge (under some assumptions)