10-301/601: Introduction to Machine Learning Lecture 24: Markov Decision Processes

Henry Chai

7/24/23

Front Matter

- Announcements
  - PA6 released 7/20, due 7/27 at 11:59 PM
    - Please be mindful of your grace day usage (see the course syllabus for the policy)
  - Quiz 8: Graphical Models on 7/21 (tomorrow!)
  - Wellness day on 7/31 (next Monday): no lecture or OH
- 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 - $y^{(n)} \in \{1, \ldots, C\}$
- Unsupervised learning - $\mathcal{D} = \left\{\boldsymbol{x}^{(n)}\right\}_{n=1}^{N}$
  - Clustering
  - Dimensionality reduction
- Reinforcement learning - $\mathcal{D} = \left\{\boldsymbol{s}^{(n)}, \boldsymbol{a}^{(n)}, r^{(n)}\right\}_{n=1}^{N}$

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

Outline

- Problem formulation
  - Time discounted cumulative reward
  - Markov decision processes (MDPs)
- Algorithms:
  - Value & policy iteration (dynamic programming) (tomorrow)
  - (Deep) Q-learning (temporal difference learning) (Wednesday)

Reinforcement Learning: Problem Formulation

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

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

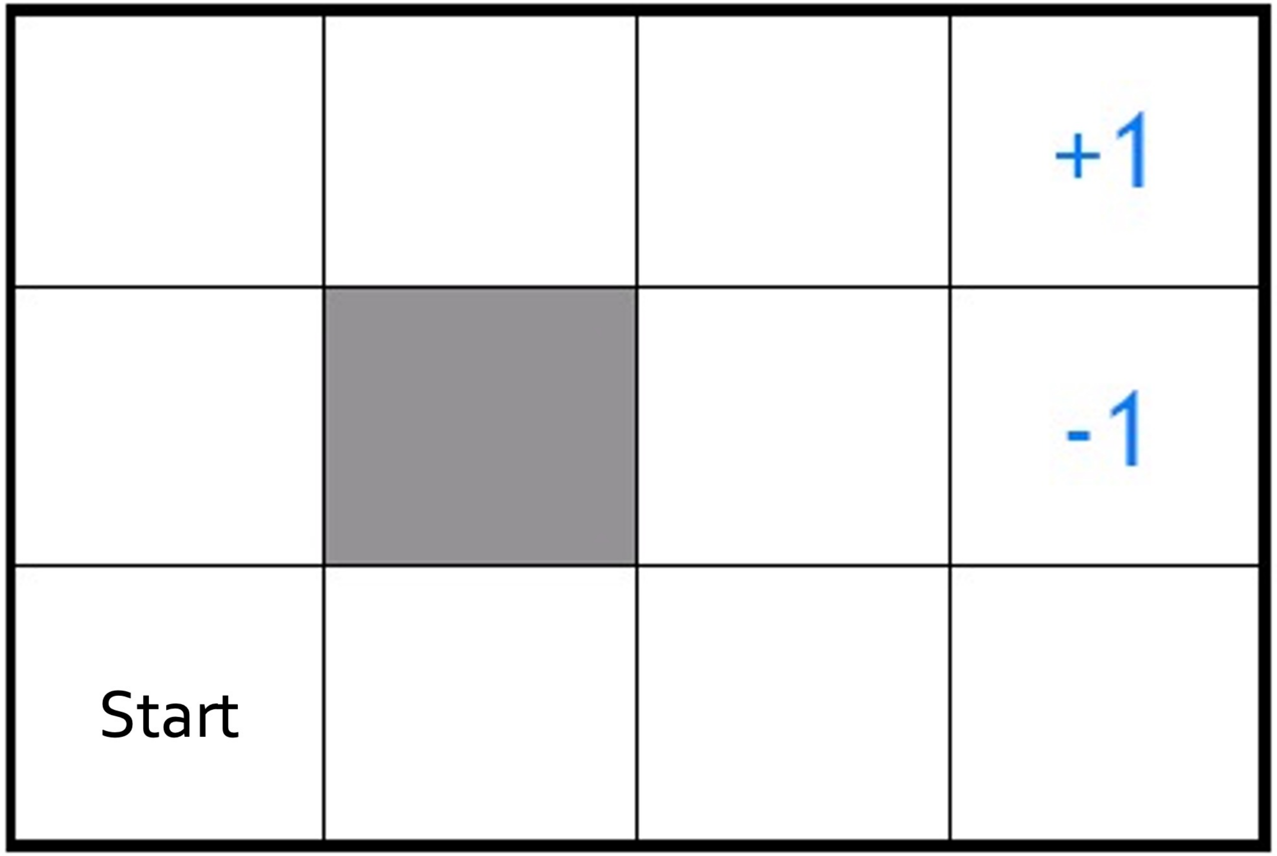


Figure courtesy of Eric Xing

Toy Example

Is this policy optimal?

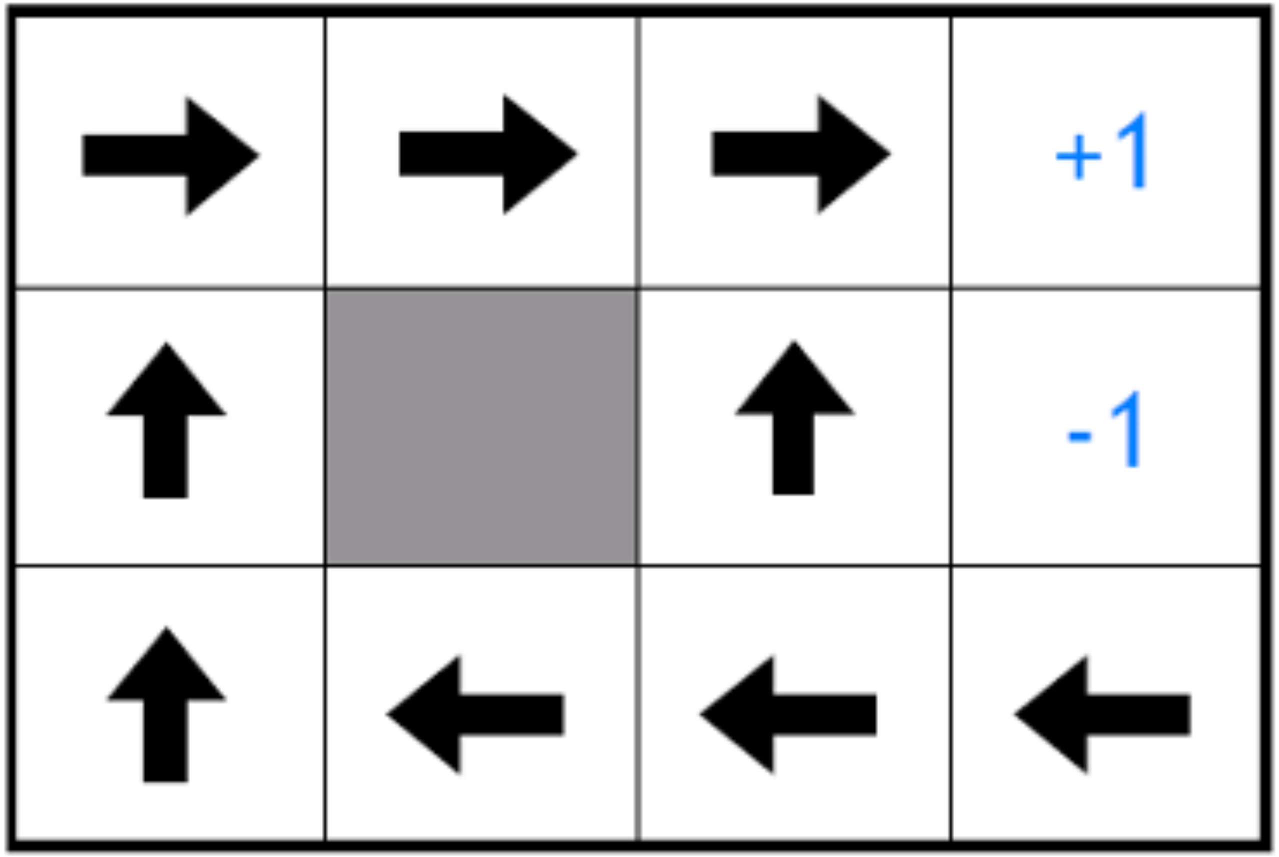


Figure courtesy of Eric Xing

When survey is active, respond at **pollev.com/301601polls**

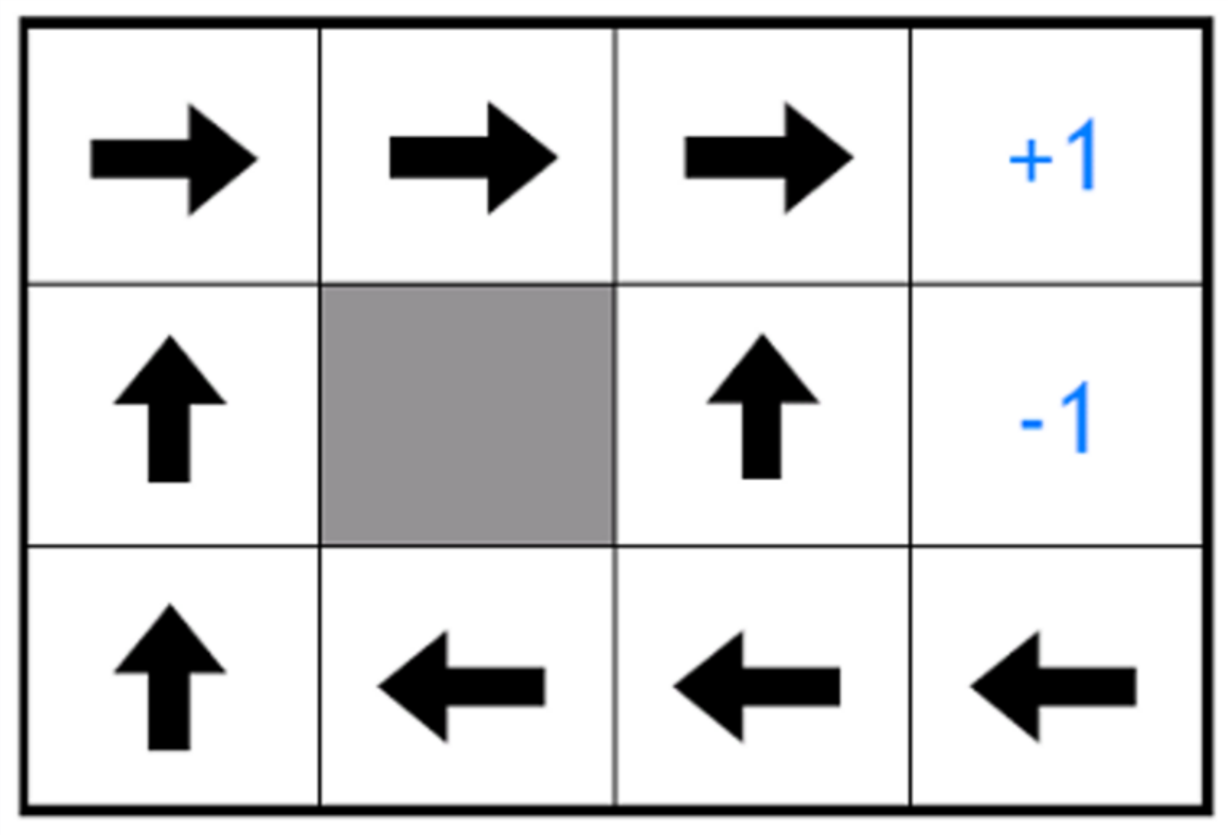
Lecture 24 Polls

0 done

0 underway

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Is this policy optimal?



- Yes
- No

Briefly justify your answer to the previous question

Join by Web

1. Go to **PollEv.com**
2. Enter **301601POLLS**
3. Respond to activity

Instructions not active. **Log in** to activate

Toy Example

Optimal policy given a reward of -2 per step

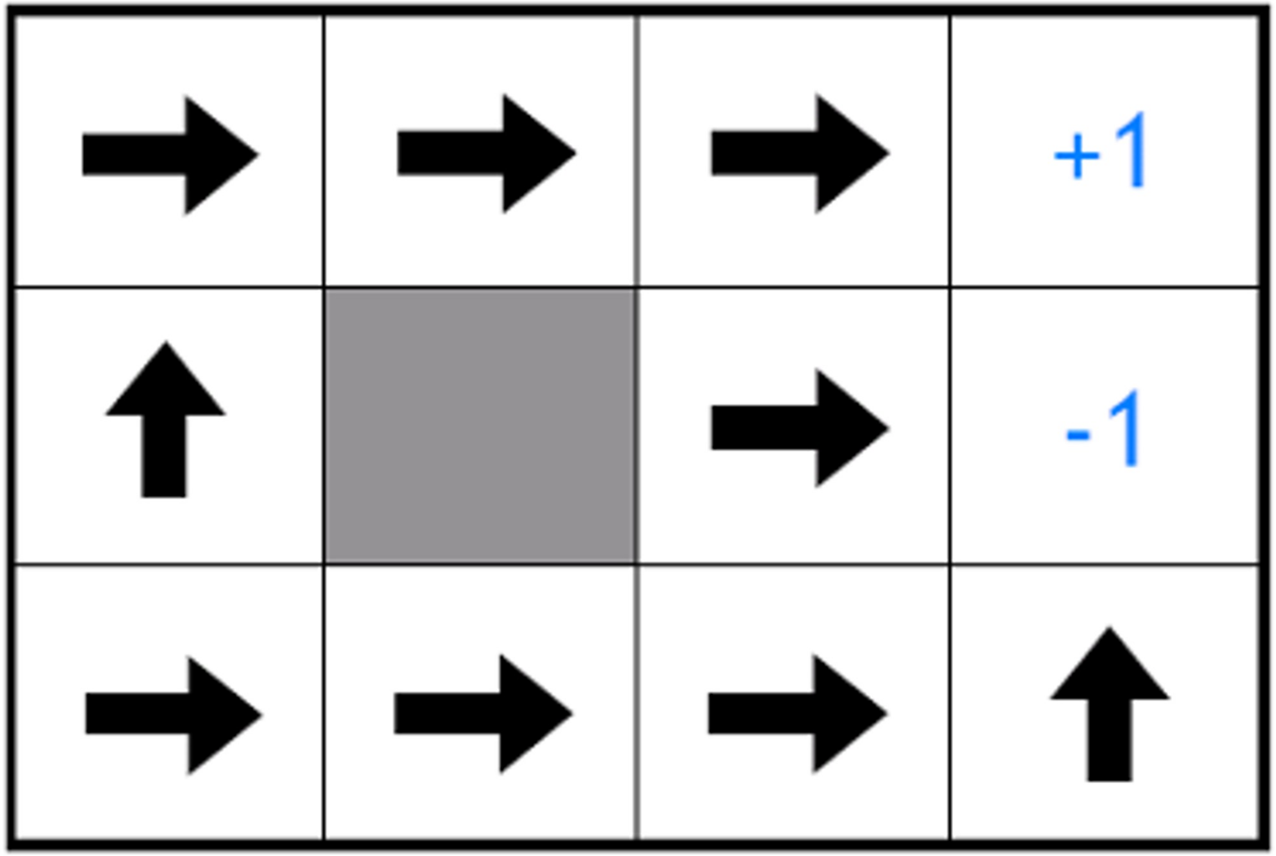


Figure courtesy of Eric Xing

Toy Example

Optimal policy given a reward of -0.1 per step

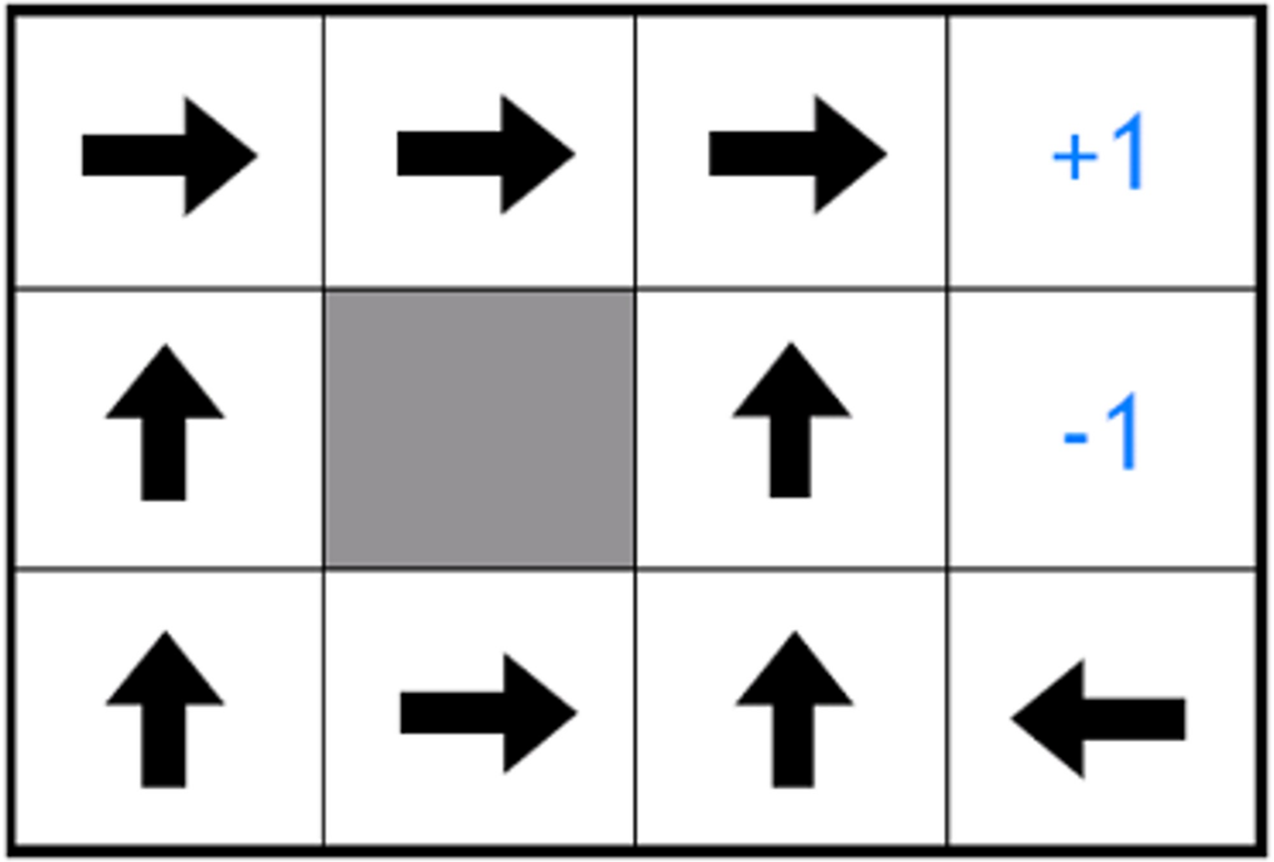


Figure courtesy of Eric Xing

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 $\sum_{t=0}^{\infty} \gamma^t r_t$

- 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	???	???
1	???	???
1	???	???
???	???	???
???	???	???
???	???	???
???	???	???
???	???	???
???	???	???
???	???	???
???	???	???
???	???	???

Reinforcement Learning: Objective Function

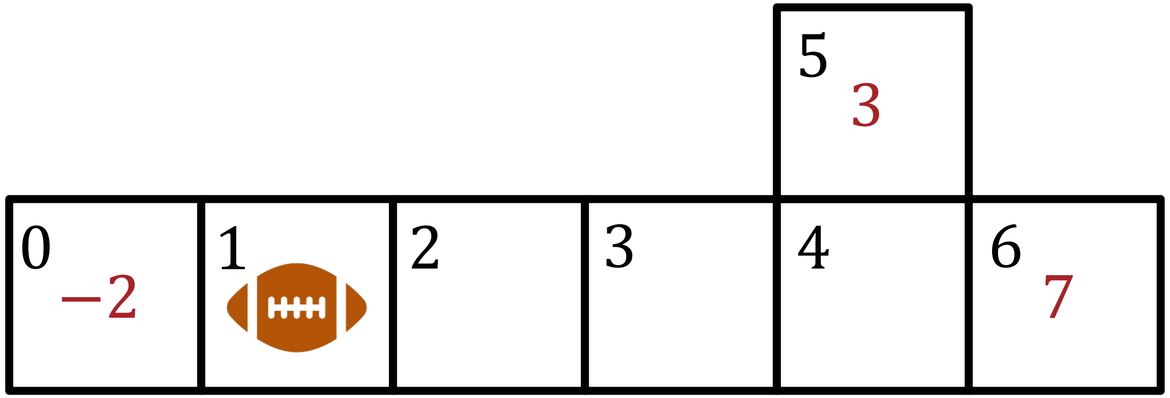
- Find a policy $\pi^* = \underset{\pi}{\operatorname{argmax}} \; V^{\pi}(s) \;\; \forall \, s \in \mathcal{S}$
- $V^{\pi}(s) = \mathbb{E}[$*discounted* total reward of starting in state s and executing policy π forever$]$

$$= \mathbb{E}_{p(s' \mid s, a)}\left[R\big(s_0 = s, \pi(s_0)\big) + \gamma R\big(s_1, \pi(s_1)\big) + \gamma^2 R\big(s_2, \pi(s_2)\big) + \cdots\right]$$

$$= \sum_{t=0}^{\infty} \gamma^t \mathbb{E}_{p(s' \mid s, a)}\left[R\big(s_t, \pi(s_t)\big)\right]$$

where $0 < \gamma < 1$ is some discount factor for future rewards

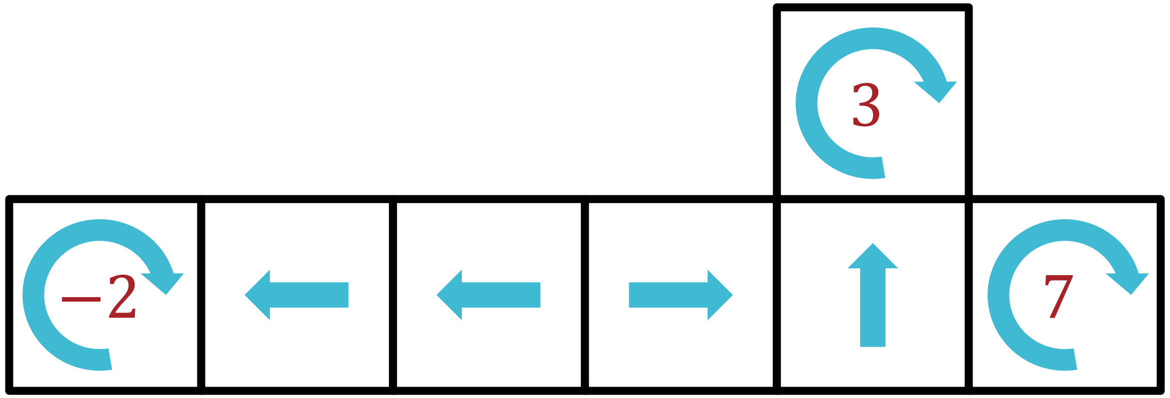
Value Function: Example



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

$$\gamma = 0.9$$

Value Function: Example

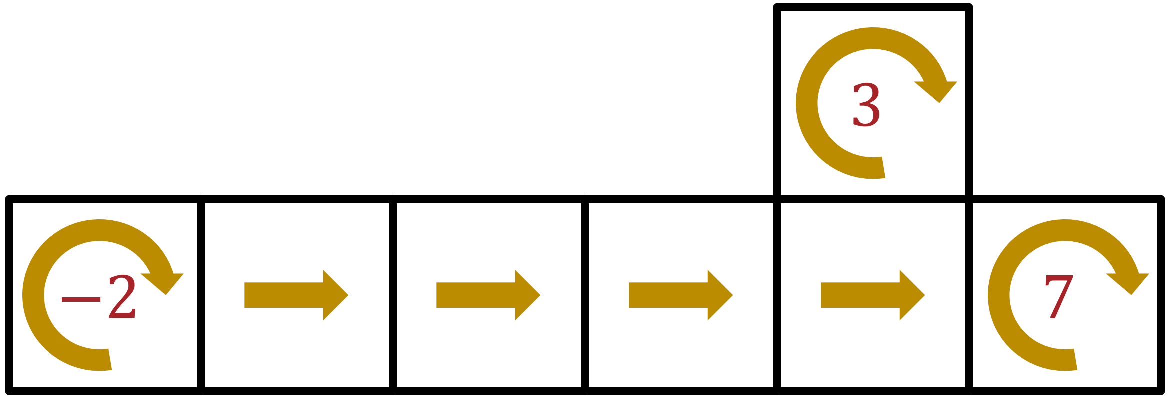


$$R(s,a) = \begin{cases} -2 \text{ if entering state 0 (safety)} \\ 3 \text{ if entering state 5 (field goal)} \\ 7 \text{ if entering state 6 (touch down)} \\ 0 \text{ otherwise} \end{cases}$$

$\gamma = 0.9$

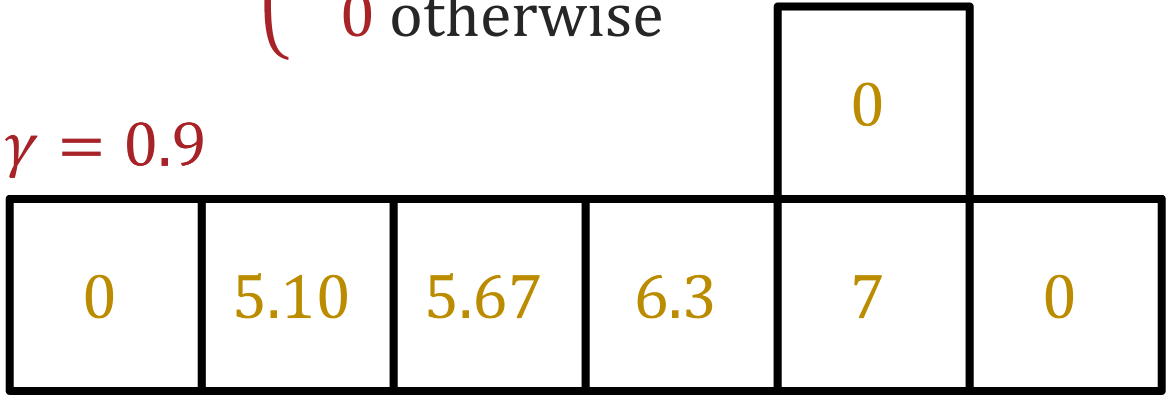
				0	
0	-2	-1.8	2.7	3	0

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

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

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

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

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

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$]$

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

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

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

Value Function

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

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

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

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

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

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