10-301/601: Introduction to Machine Learning Lecture 21: Markov Decision Processes

Henry Chai & Matt Gormley

11/14/22

Front Matter

- Announcements
  - HW7 released 11/11, due 11/21 at 11:59 PM
    - Please be mindful of your grace day usage

Q & A:
I've had such a great experience with this class, especially with your excellent TAs; how can I be more like them and contribute to future iterations of this class?

- You can apply to be a TA for this course next semester (S23)!
- Applications are due by Thursday, November 17th
- For more information and the application, see https://www.ml.cmu.edu/academics/ta.html

Learning Paradigms

- Supervised learning - $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$
  - Regression - $y^{(i)} \in \mathbb{R}$
  - Classification - $y^{(i)} \in \{1, \ldots, C\}$
- Unsupervised learning - $\mathcal{D} = \left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^{N}$
  - Clustering
  - Dimensionality reduction
- Reinforcement learning - $\mathcal{D} = \left\{\boldsymbol{s}^{(t)}, \boldsymbol{a}^{(t)}, r^{(t)}\right\}_{t=1}^{T}$

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)
  - (Deep) Q-learning (temporal difference learning)

Reinforcement Learning: Problem Formulation

- State space, $\mathcal{S}$
- Action space, $\mathcal{A}$
- Reward function
  - Stochastic, $p(r \mid s, a)$
  - Deterministic, $R\colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$
- Transition function
  - Stochastic, $p(s' \mid s, a)$
  - Deterministic, $\delta\colon \mathcal{S} \times \mathcal{A} \to \mathcal{S}$
- Reward and transition functions can be known or unknown

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

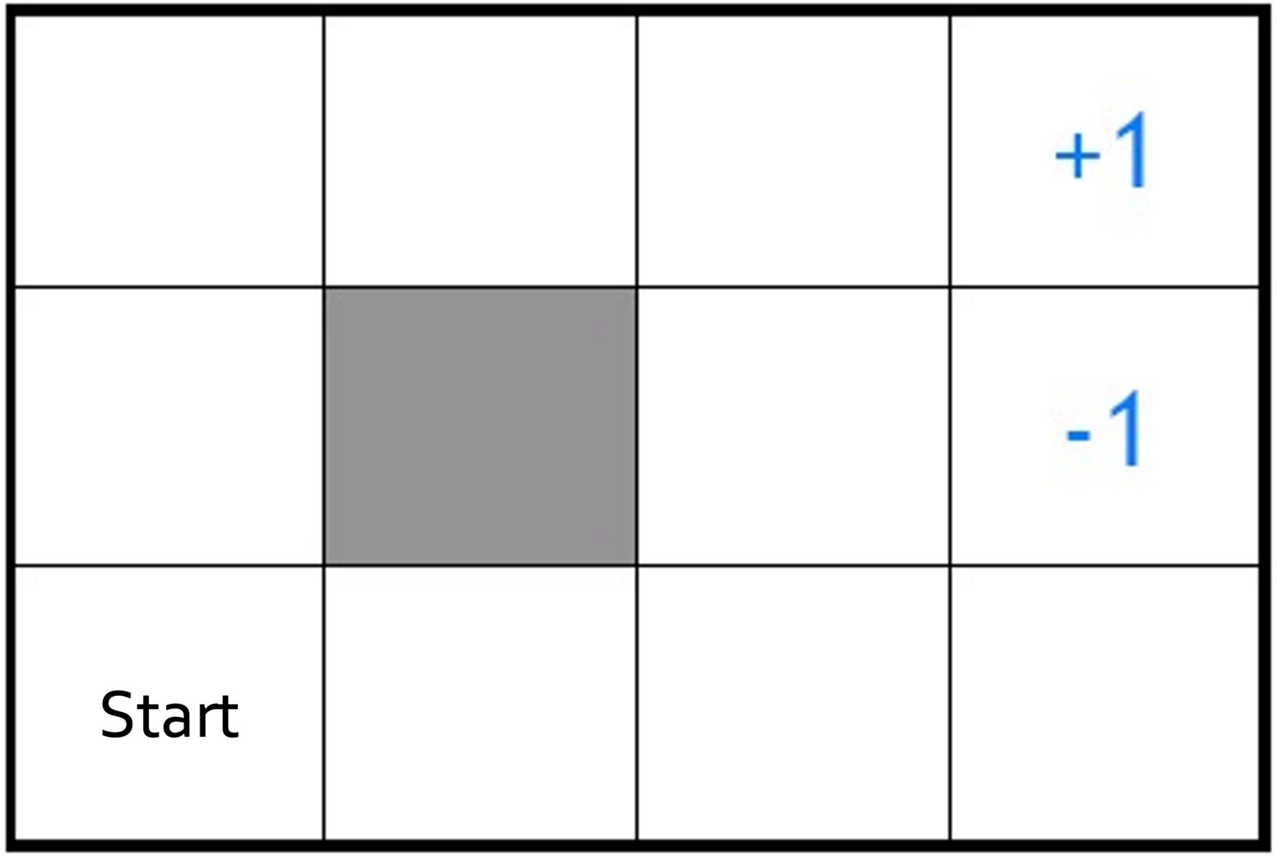


Figure courtesy of Eric Xing

Toy Example Policy

- $\mathcal{S}$ = 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

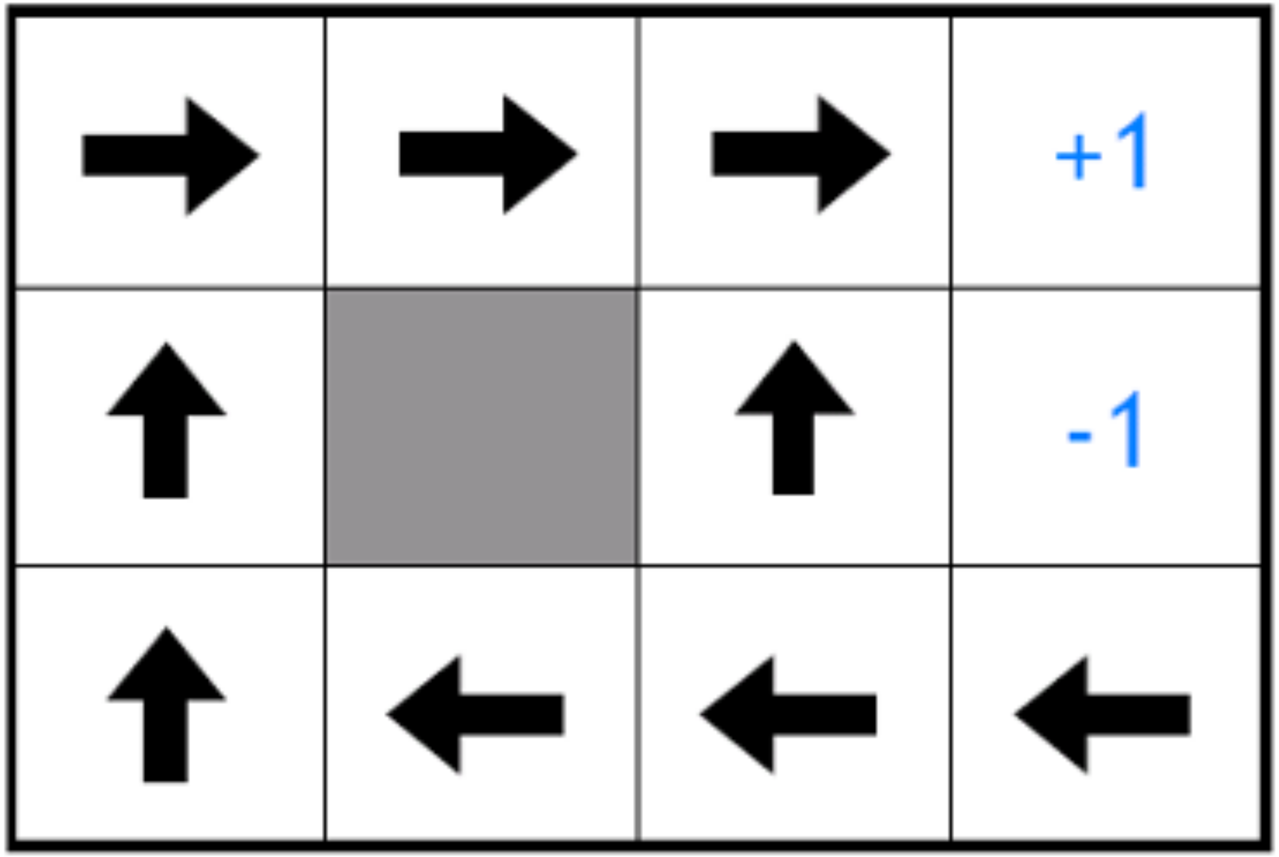


Figure courtesy of Eric Xing

Poll Question 1: Is this policy optimal?

A. Yes

B. No

C. **TOXIC**

Poll Question 2:

Briefly justify your answer to the previous question

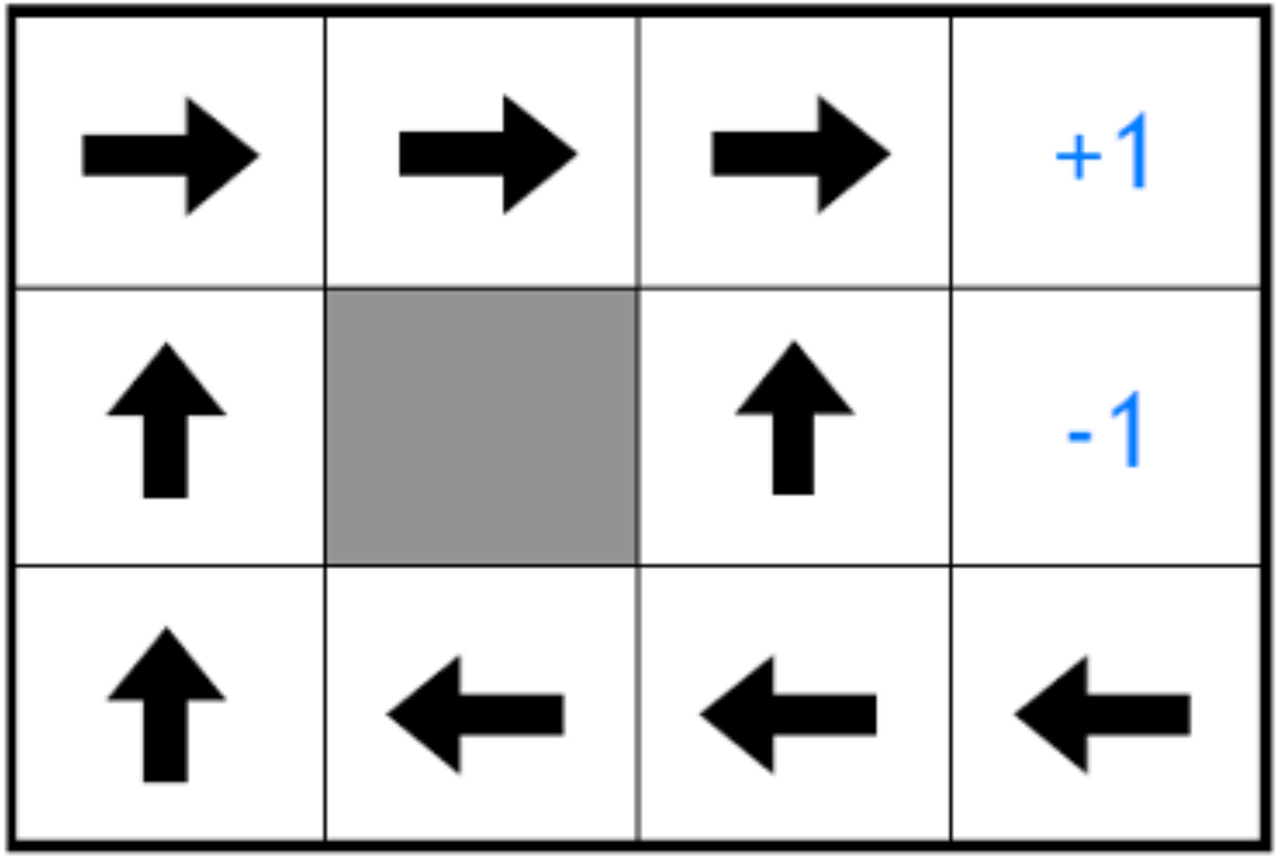


Figure courtesy of Eric Xing

Toy Example

Optimal policy given a reward of -2 per step

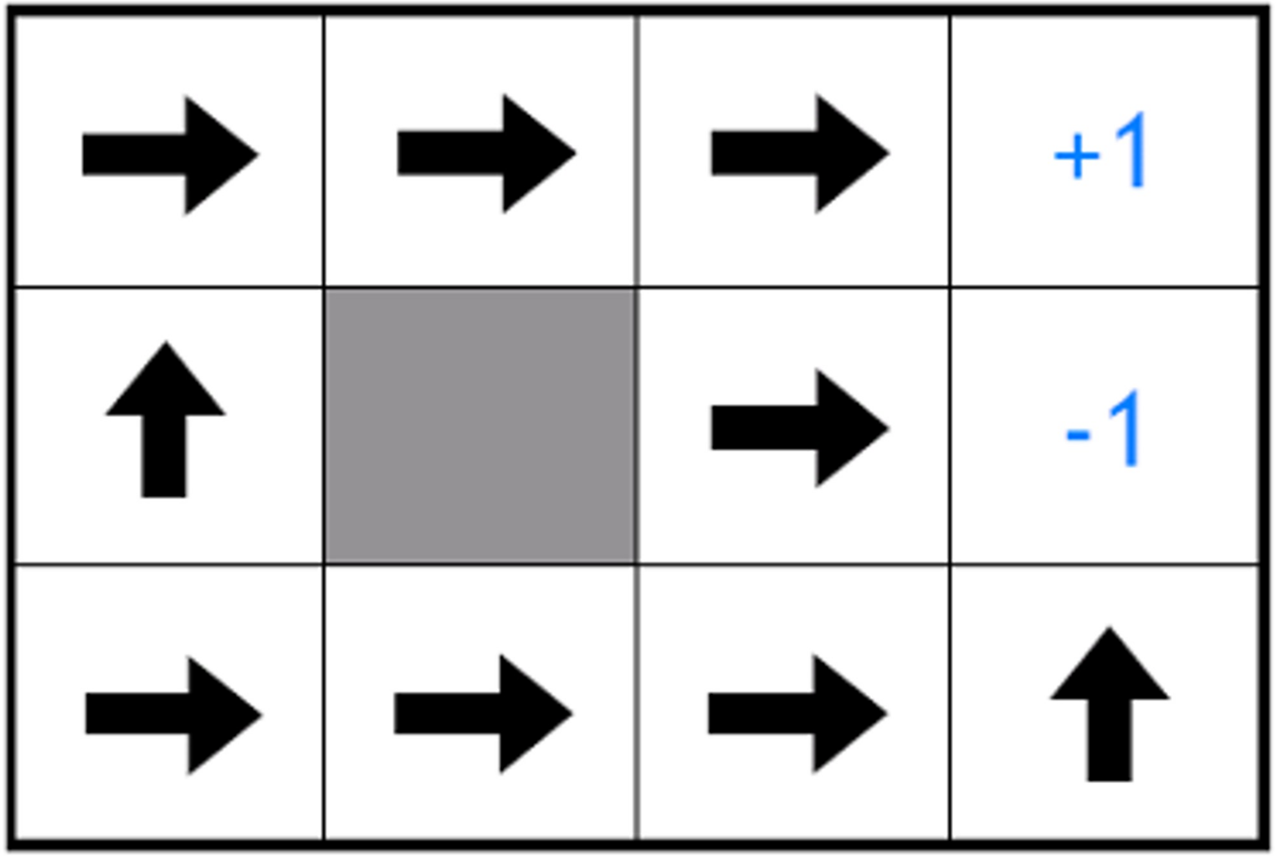


Figure courtesy of Eric Xing

Toy Example

Optimal policy given a reward of -0.1 per step

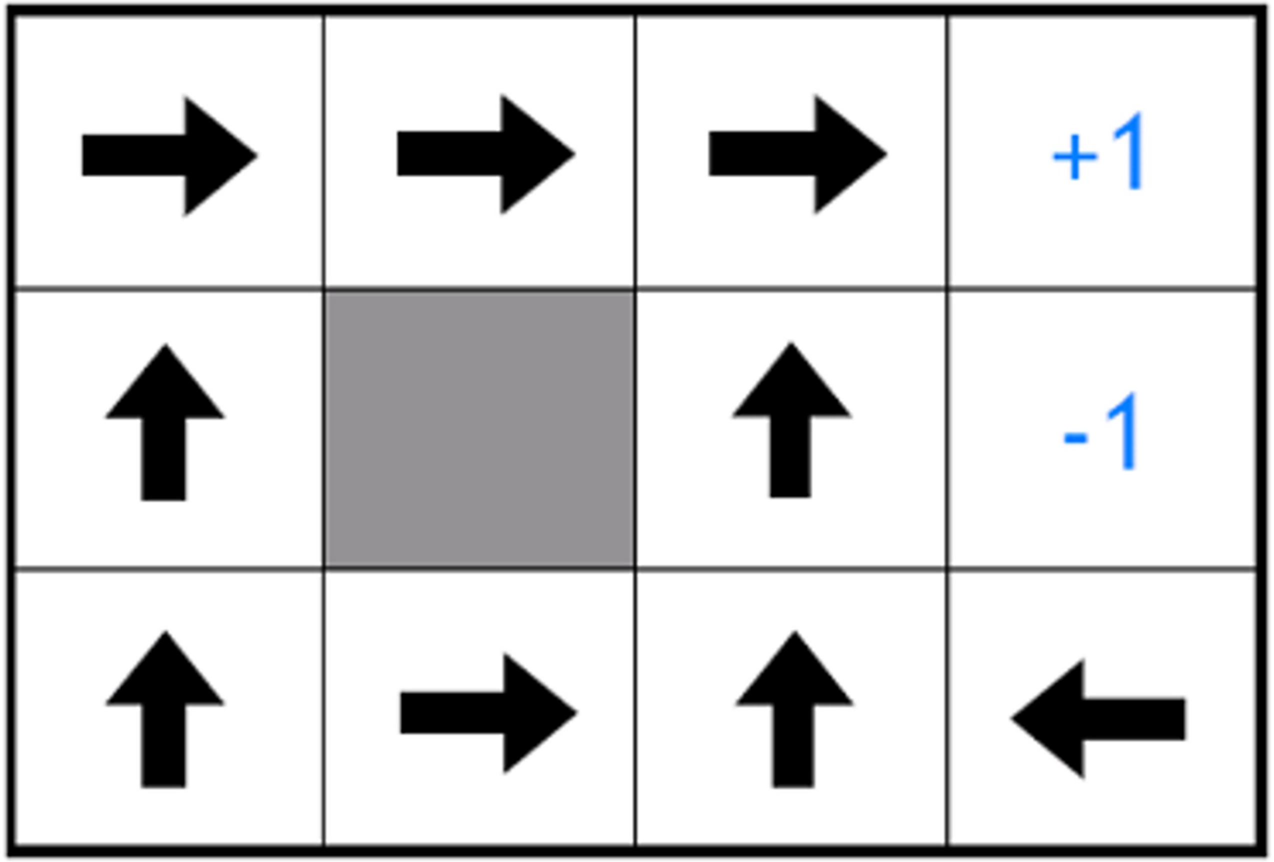


Figure courtesy of Eric Xing

Markov Decision Process (MDP)

- Assume the following model for our data:

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

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

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

Reinforcement Learning: Key Challenges

- The algorithm has to gather its own training data
- The outcome of taking some action is often stochastic or unknown until after the fact
- 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

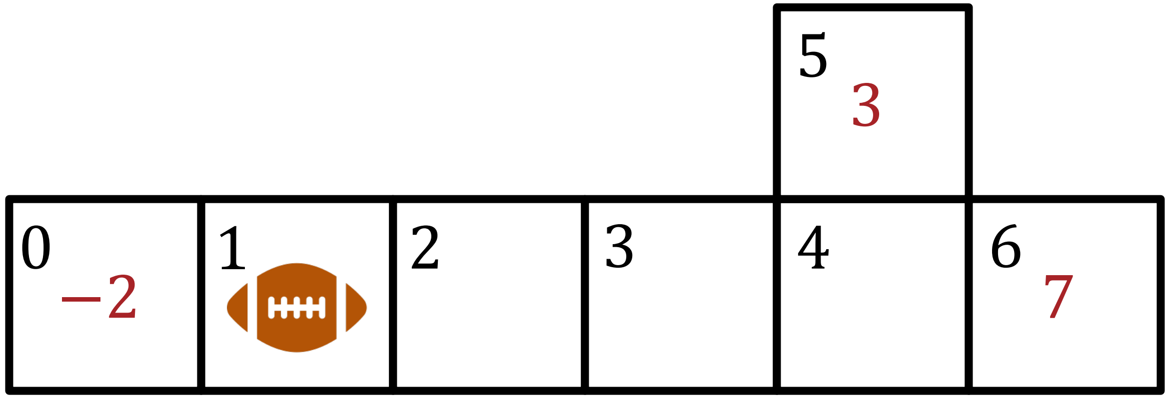
- Find a policy $\pi^* = \underset{\pi}{\operatorname{argmax}}\ V^\pi(s)\ \forall\, s \in \mathcal{S}$
- $V^\pi(s) = \mathbb{E}[$*discounted* total reward of starting in state s and executing policy π forever$]$

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

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

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

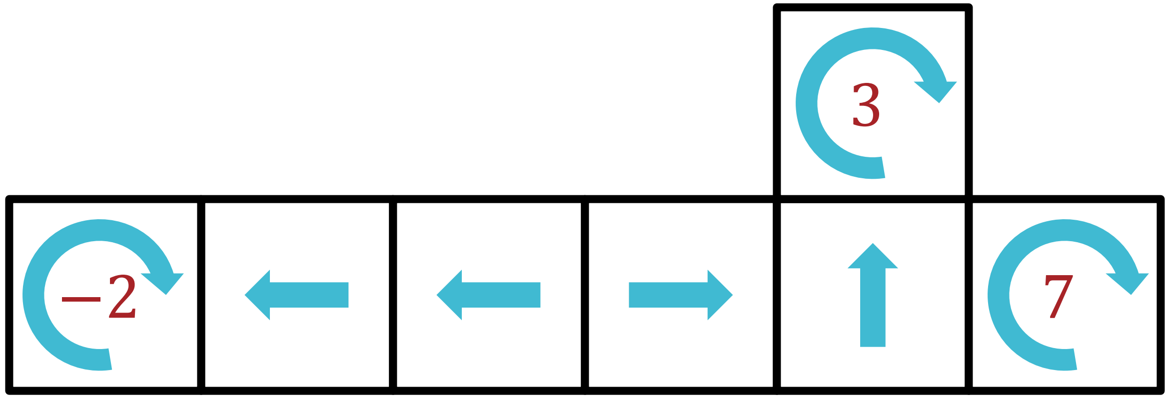
Value Function: Example



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

$$\gamma = 0.9$$

Value Function: Example

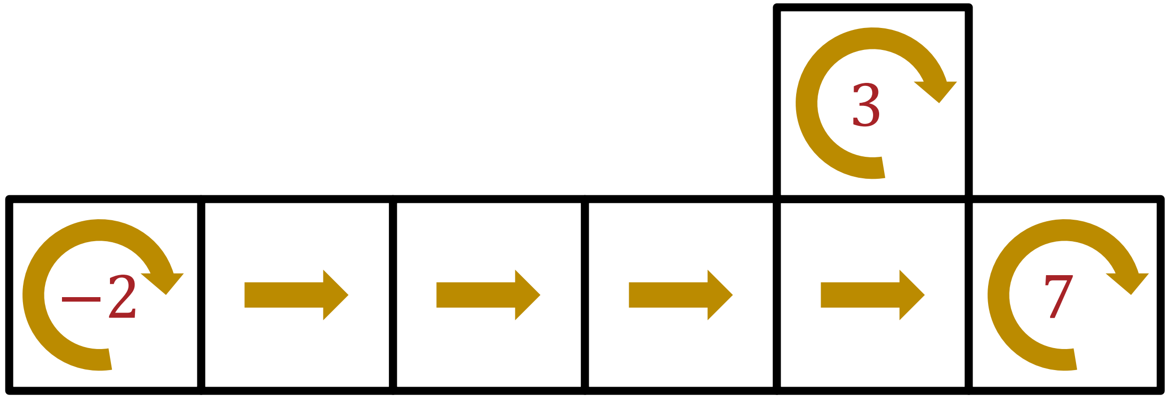


$$R(s,a) = \begin{cases} -2 \text{ if entering state 0 (safety)} \\ 3 \text{ if entering state 5 (field goal)} \\ 7 \text{ if entering state 6 (touch down)} \\ 0 \text{ otherwise} \end{cases}$$

$\gamma = 0.9$

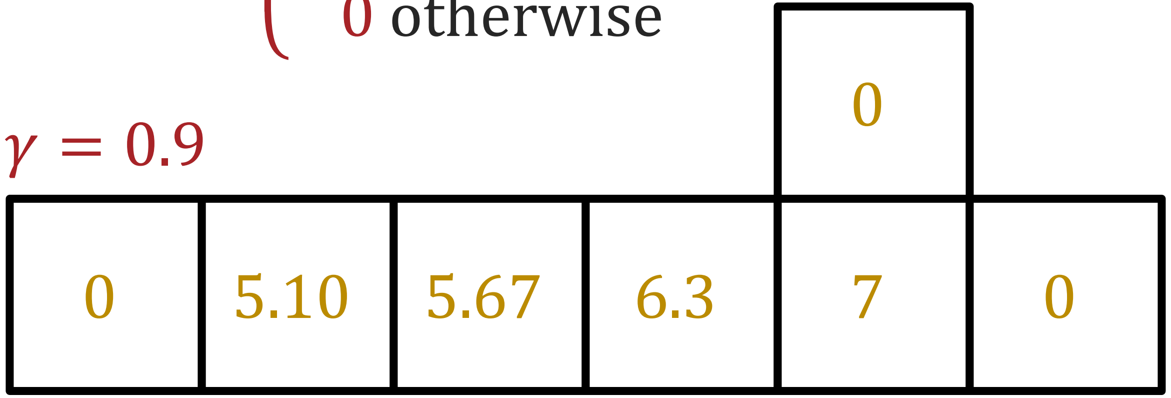
				0	
0	−2	−1.8	2.7	3	0

Value Function: Example

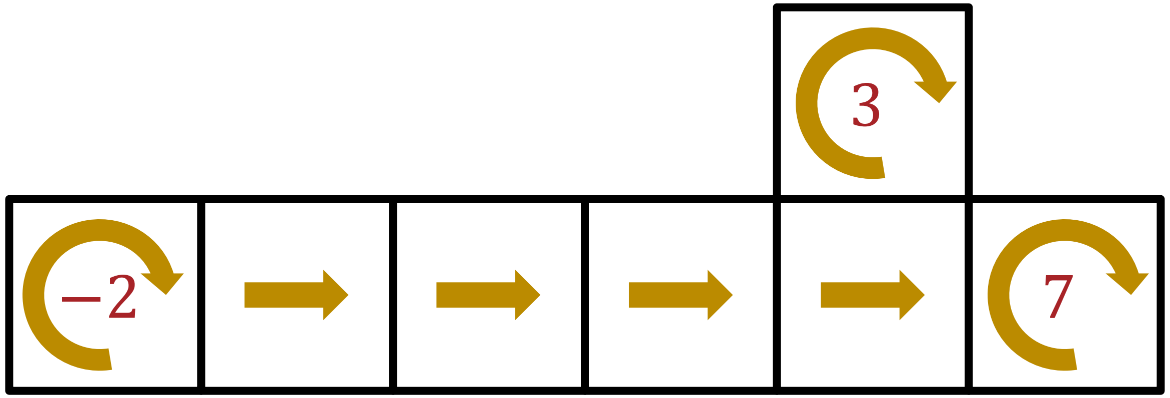


$$R(s,a) = \begin{cases} -2 \text{ if entering state 0 (safety)} \\ 3 \text{ if entering state 5 (field goal)} \\ 7 \text{ if entering state 6 (touch down)} \\ 0 \text{ otherwise} \end{cases}$$

$\gamma = 0.9$



How can we learn this optimal policy?



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

$\gamma = 0.9$