10-701: Introduction to Machine Learning

Lecture 16 – Reinforcement Learning: Value & Policy Iteration

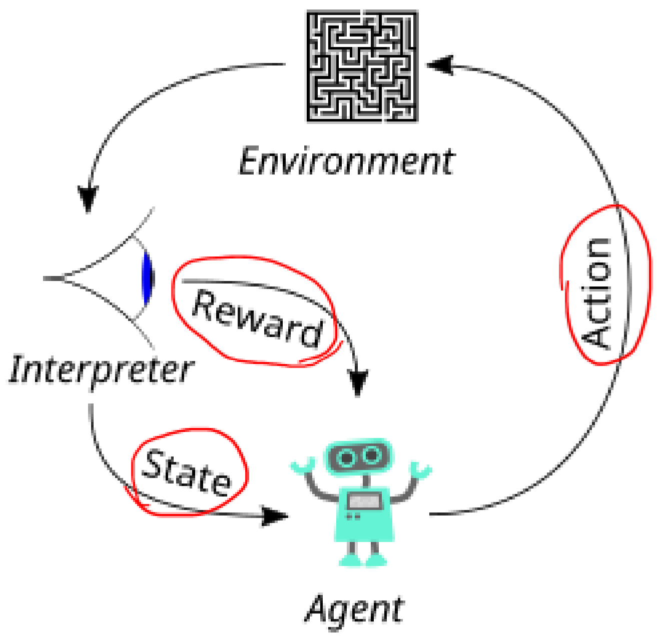
Hoda Heidari

* Slides adopted from F24 offering of 10701 by Henry Chai.

Learning Paradigms

- Supervised learning - $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(i)}, y^{(i)}\right)\right\}_{i=1}^{N}$ i.i.d
  - Regression - $y^{(i)} \in \mathbb{R}$
  - Classification - $y^{(i)} \in \{1, \ldots, C\}$
- Unsupervised learning - $\mathcal{D} = \left\{\boldsymbol{x}^{(i)}\right\}_{i=1}^{N}$ i.i.d
  - Clustering
  - Dimensionality reduction
- Reinforcement learning - $\mathcal{D} = \left\{\left(\underbrace{\boldsymbol{s}^{(n)}}_{\text{state}}, \underbrace{\boldsymbol{a}^{(n)}}_{\text{action}}, \underbrace{r^{(n)}}_{\text{reward}}\right)\right\}_{n=1}^{N}$ ~~i.i.d~~
- Active learning
- Semi-supervised learning
- Online learning

Reinforcement Learning (RL)



The typical framing of a reinforcement learning (RL) scenario: an agent takes actions in an environment, which is interpreted into a reward and a state representation, which are fed back to the agent.
From https://en.wikipedia.org/wiki/Reinforcement_learning

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

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 $a_t = \pi(s_t)$ policy
    3. Agent receives reward $r_t \sim p(r \mid s_t, a_t)$ ← $r_t \sim p(r \mid s_0, s_1, \ldots s_t, a_0, a_1, \ldots a_t)$
    4. Agent transitions to state $s_{t+1} \sim p(s' \mid s_t, a_t)$

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

Formalization

- **State** space, $\mathcal{S}$ $s_0, s_1, s_2, \ldots \in \mathcal{S}$ $|\mathcal{S}|, |\mathcal{A}| < \infty$
- **Action** space, $\mathcal{A}$ $a_0, a_1, \ldots \in \mathcal{A}$
- **Reward** function
  - Stochastic, $p(r \mid s, a)$
  - Deterministic, $R: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ $R: \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$
- **Transition** function
  - Stochastic, $p(s' \mid s, a)$
  - Deterministic, $\delta: \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$

Formalization

- **Policy,** $\pi : \underbrace{\mathcal{S}} \to \underbrace{\mathcal{A}}$
  - Specifies an action to take in *every* state
- **Value function,** $V^{\pi} : \mathcal{S} \to \mathbb{R}$ $\quad V^{\pi}(s) \quad \forall s \in S$
  - 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

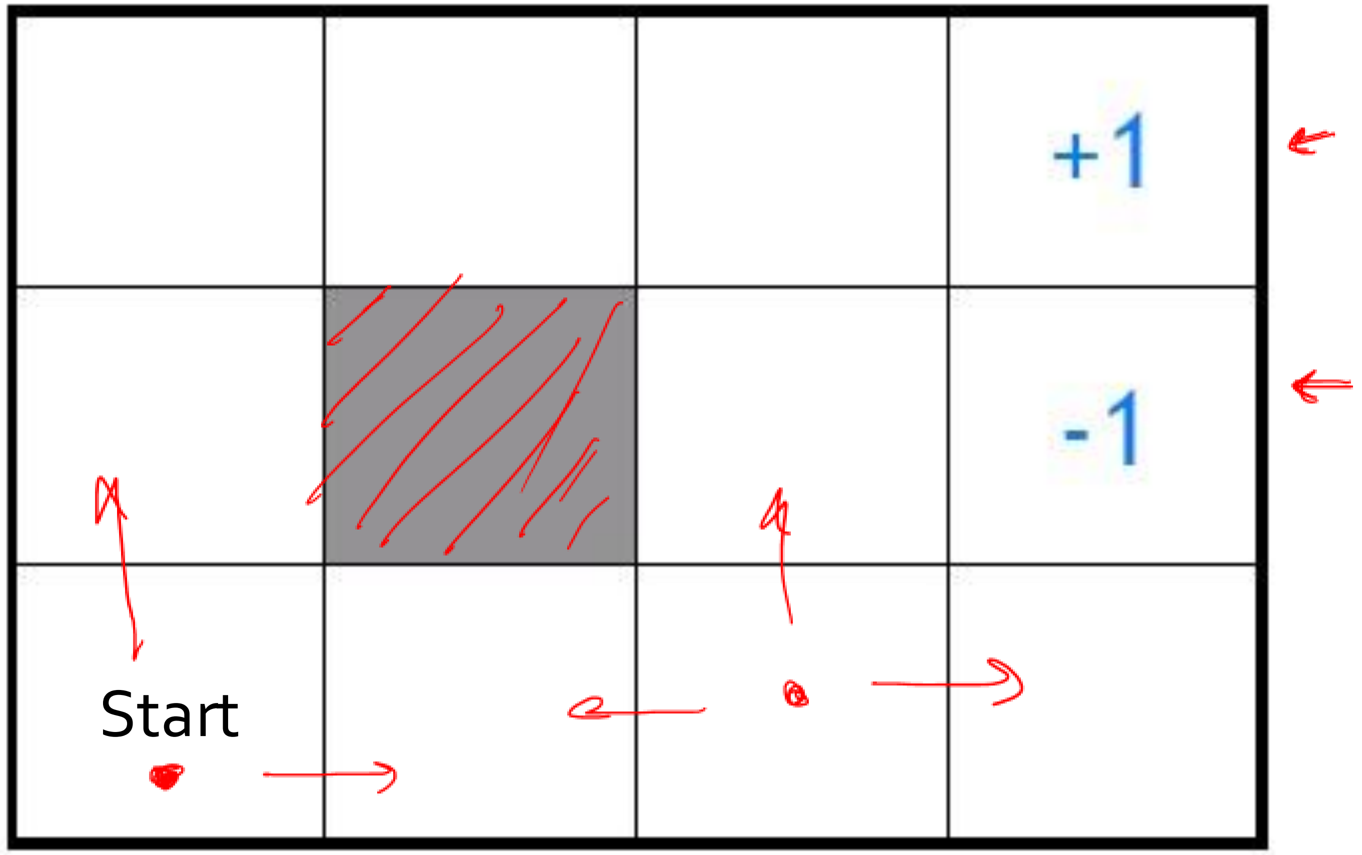


Figure courtesy of Eric Xing

Toy Example

Is this policy optimal?

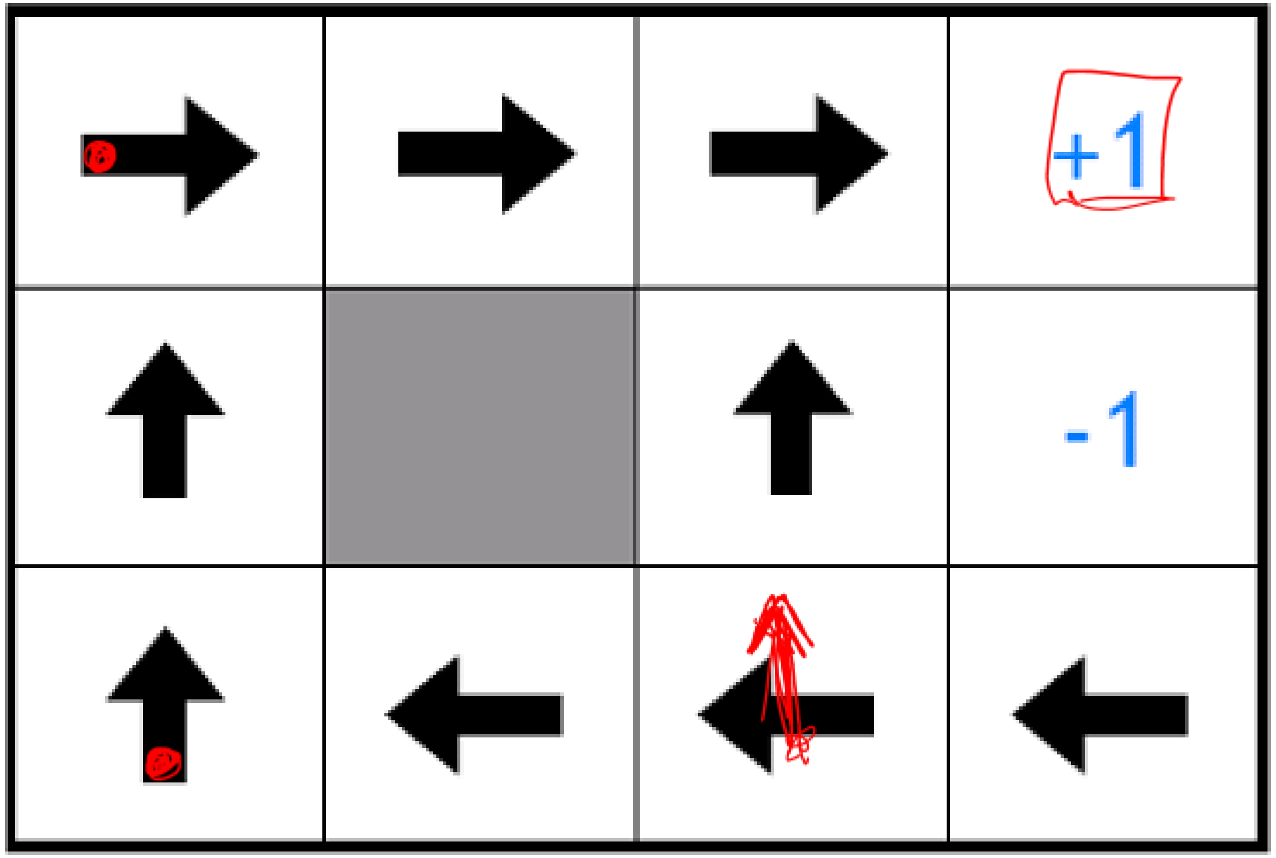


Figure courtesy of Eric Xing

Toy Example

Optimal policy given a reward of -2 per step

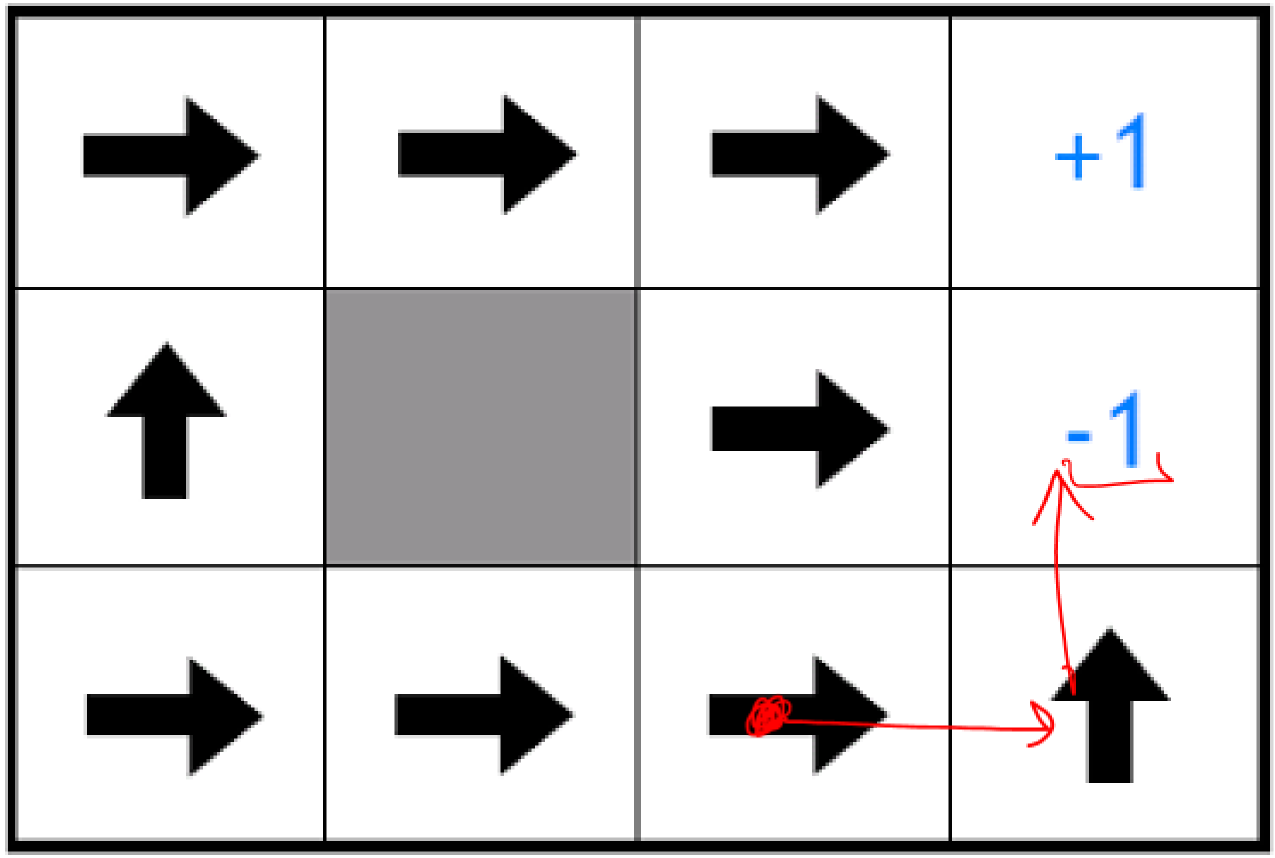
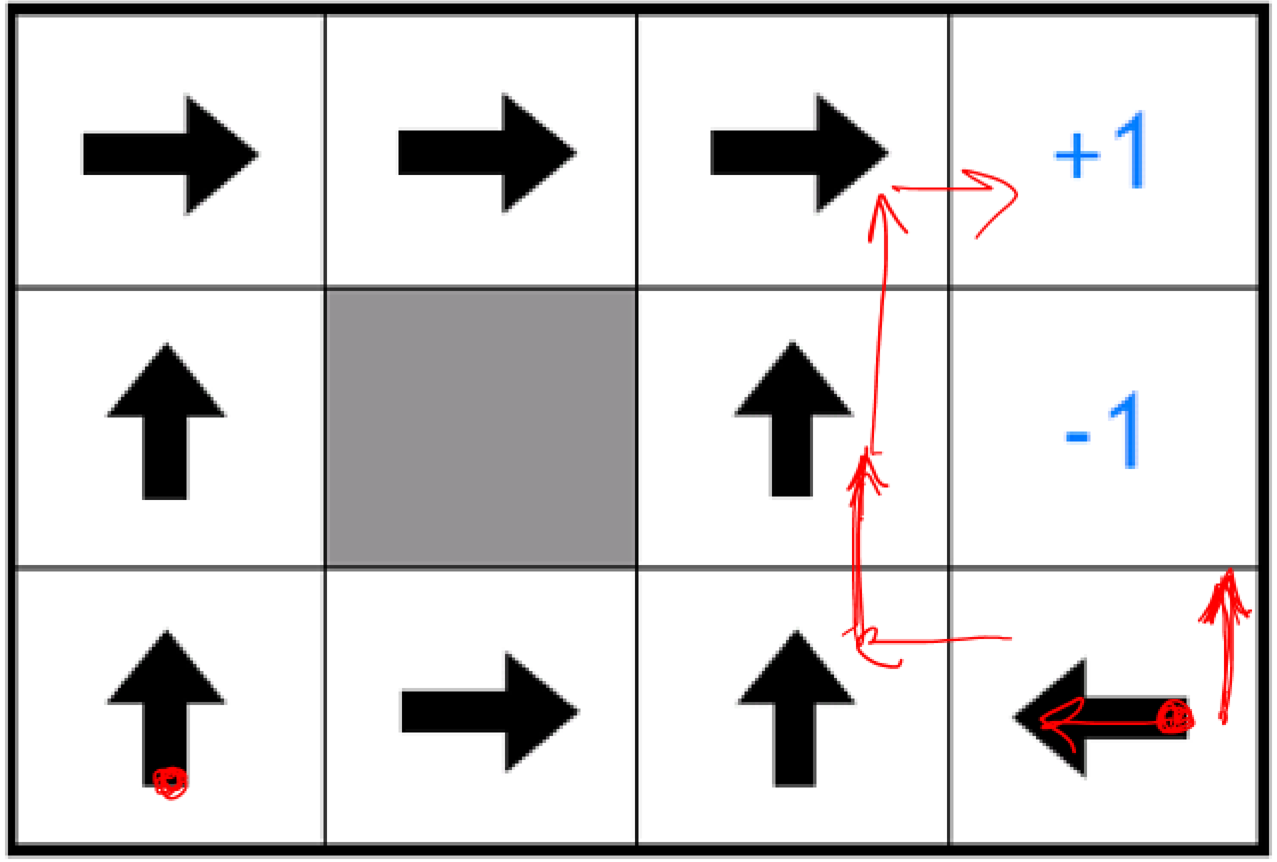


Figure courtesy of Eric Xing

Toy Example

Optimal policy given a reward of -0.1 per step



-1

$3 \times (-0.1) + 1$

Figure courtesy of Eric Xing

The Objective Function

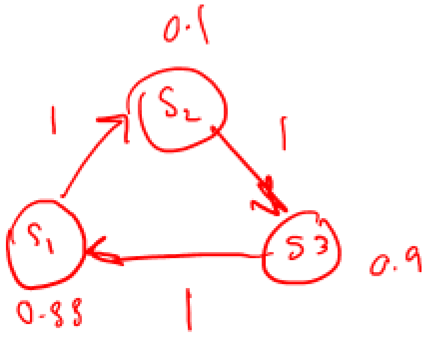
- Agent receives reward $r_t \sim p(r \mid s_t, a_t)$ at time t.
- The cumulative reward can be defined as
  - Finite time-horizon

$$\underbrace{\sum_{t=0}^{T} r_t}$$

  - Infinite time-horizon

$0 \leq \gamma < 1$ discount factor

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



- The optimal policy π^* on an MDP is the one yielding the highest possible expected cumulative reward among all allowable policies.

MDP with $|S|$ states $|A|$ actions at each state

$\pi : S \to A$ $|A|^{|S|}$ many policies

Planning Challenges

The MDP is fully specified, i.e, we know $\mathcal{A}, \mathcal{S}, P, R$

Known environment:

1. The outcome of taking some action is often stochastic or unknown until after the fact
2. Decisions can have a delayed effect on future outcomes (exploration-exploitation tradeoff)

Assumption: $R(s,a)$ is deterministic.

Value Function

- Find a policy $\pi^* = \underset{\pi}{\operatorname{argmax}}\ V^\pi(s)$ for $s \in \mathcal{S}$

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

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

$$= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)) \,\middle|\, s_0 = s\right]$$

linearity of Exp

$$= \sum_{t=0} \gamma^t\, \mathbb{E}\left[R(s_t, \pi(s_t)) \,\middle|\, s_0 = s\right]$$

Value Function

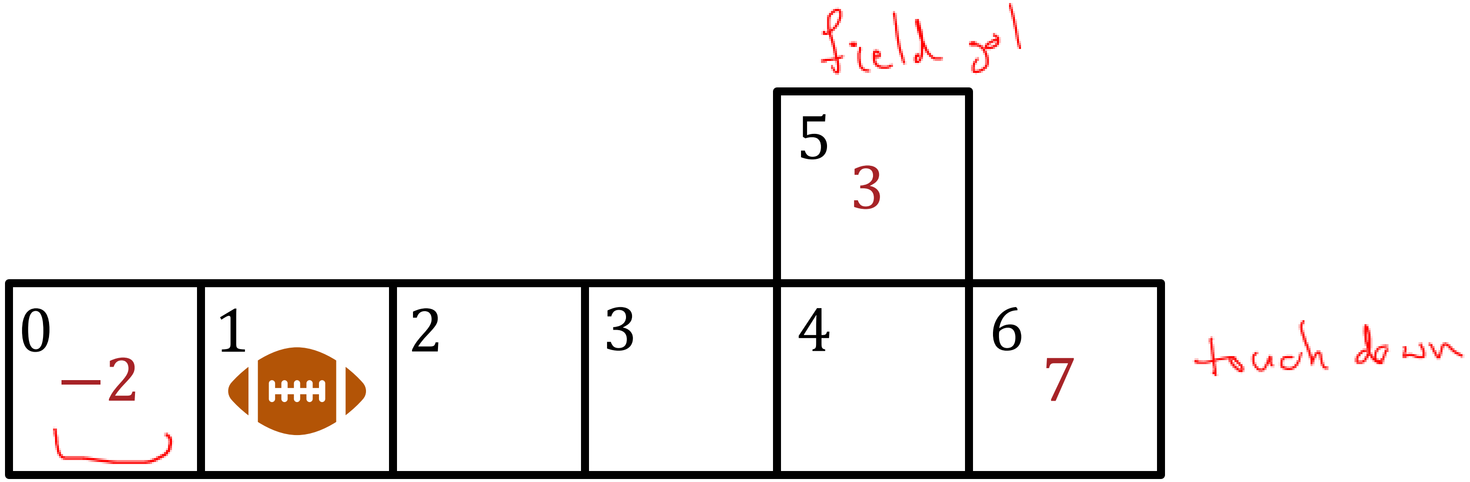
- Find a policy $\pi^* = \underset{\pi}{\operatorname{argmax}}\ V^{\pi}(s)$ for $s \in \mathcal{S}$
- $V^{\pi}(s) = \mathbb{E}[$*discounted* total reward of starting in state s and executing policy π forever$]$

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

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

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

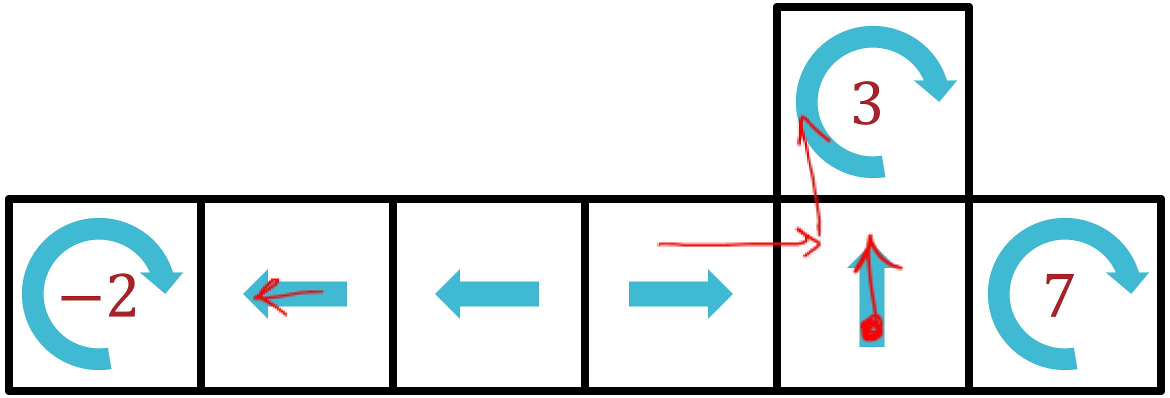
Value Function: Example



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

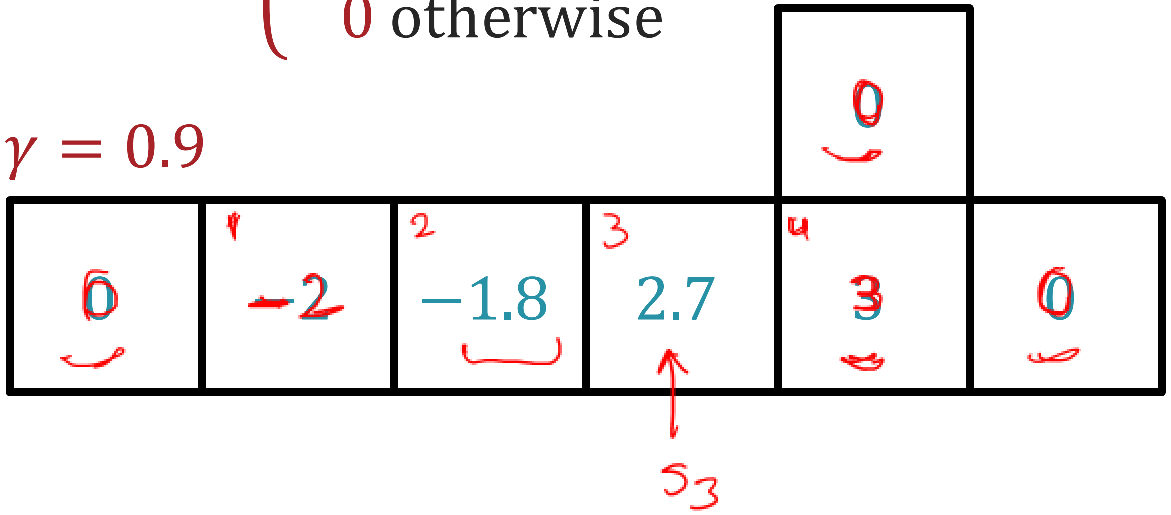
$$\gamma = 0.9$$

Value Function: Example



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

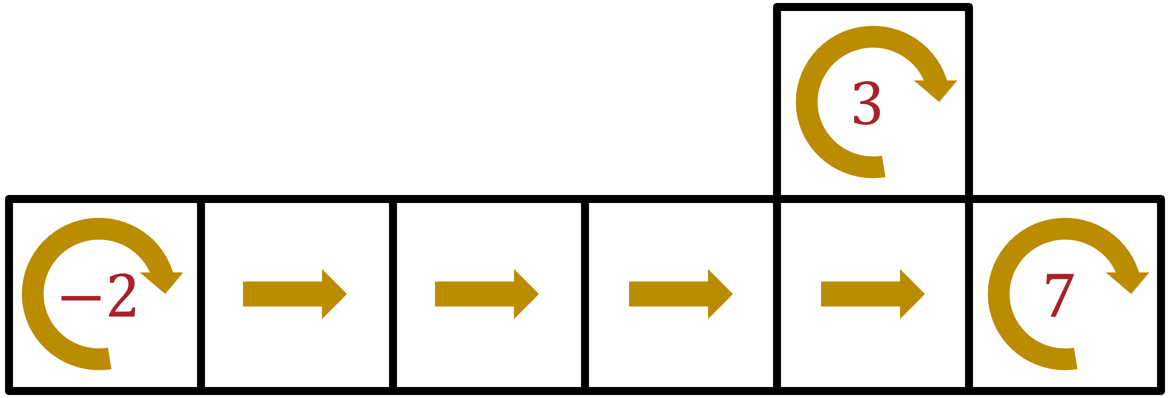
$\gamma = 0.9$



$V^{\pi}(s)$

$0 + \gamma(3) + 0$

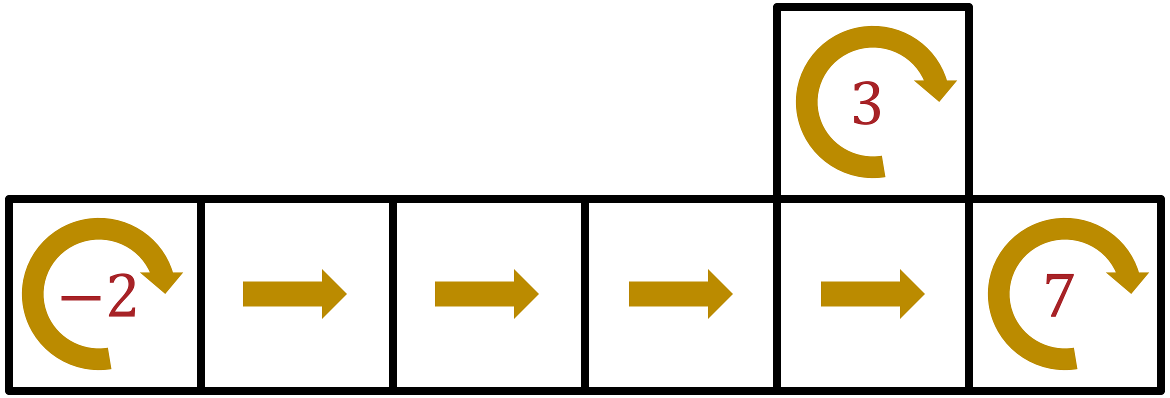
Value Function: Example



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

$\gamma = 0.9$

How can we learn this optimal policy?



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

$\gamma = 0.9$

Assumption: R is determ

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$]$

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

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

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

Value Function

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

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

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

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

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

system of linear equation
with variable $V^{\pi}(s_0), V^{\pi}(s_1), \ldots$ $s_i \in S$

Bellman equations

Optimality

- Optimal value function:

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

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

- Optimal policy:

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

fixed point of a fn g is $g(\vec{x}) = \vec{x}$

Fixed Point Iteration

Iterative method for solving a system of equations

- Given some equations and initial values

$$\begin{cases} x_1 = f_1(x_1, \dots, x_n) \\ \vdots \\ x_n = f_n(x_1, \dots, x_n) \end{cases}$$

fixed point

$$x_1^{(0)}, \dots, x_n^{(0)}$$

- 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

$$\begin{cases} x_1 = x_1 x_2 + \dfrac{1}{2} \\[2ex] x_2 = -\dfrac{3x_1}{2} \end{cases}$$

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

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

$x_1 \leftarrow 0 \cdot 0 + \frac{1}{2}$

$x_2 \leftarrow -\frac{3}{2}\, 0$

$$x_1^{(0)} = x_2^{(0)} = 0 \longrightarrow \ldots$$

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

solutions

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

$\frac{1}{3}$ $\quad -\frac{1}{2}$

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 $s \in \mathcal{S}$

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

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

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

- Return π^*

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 $a \in \mathcal{A}$

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

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

- 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

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

then $\max_{s \in \mathcal{S}} \left|V^{(t+1)}(s) - V^{*}(s)\right| < \frac{2\epsilon\gamma}{1-\gamma}$ (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)

Bellman Optimality Characterization

- A policy π is optimal if and only if it is greedy (optimal) w.r.t. its own value function V^{π}.
- Proof:
  - **(⇒) If π is optimal, then it must be greedy w.r.t V^{π}.** If π were not greedy at some state, there would exist an action with strictly higher expected return ⇒ we could improve the policy ⇒ π was not optimal. Contradiction.
  - **(⇐) If π is greedy w.r.t V^{π} ,then π is optimal.**
    Greedy w.r.t its own value solves the Bellman *optimality* fixed point, which is known to have a unique solution. So $V^{\pi} = V^{*}$and π is optimal.

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

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

  - 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)