

<https://introml.mit.edu/>

# 6.390 Intro to Machine Learning

## Lecture 10: Markov Decision Processes

Shen Shen

Apr 13, 2026

3pm, Room 10-250

[Slides and Lecture Recording](#)

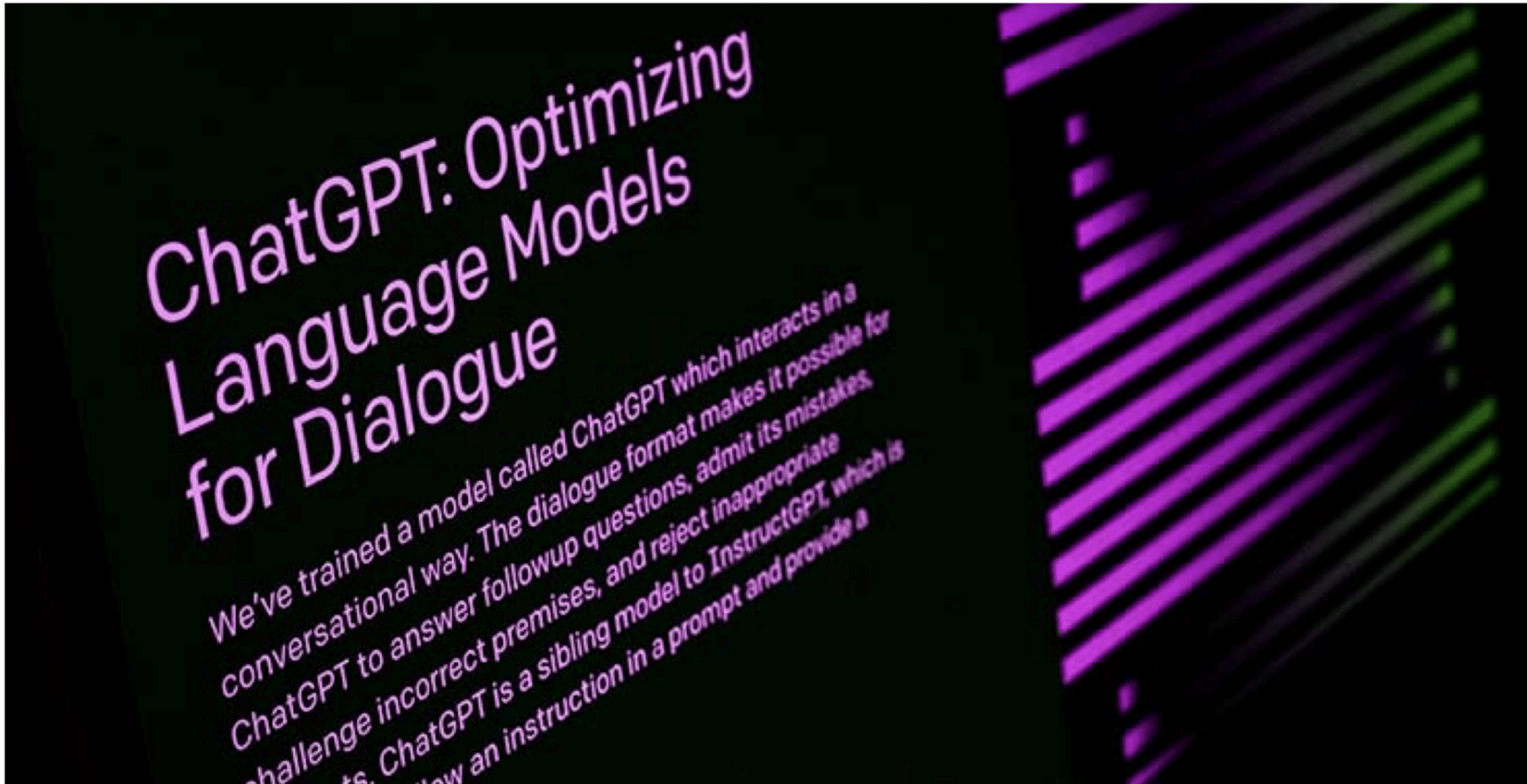
# Learning to Walk

Massachusetts Institute of  
Technology, 2004



Toddler demo, Russ Tedrake thesis, 2004  
uses vanilla policy gradient (actor-critic)

# Reinforcement Learning with Human Feedback



[Aligning language models to follow instructions. Ouyang et al. 2022]

ensor rank  
ar Standard

7

23

49

98

11

14

18

15

20

25

26

33

40

29

36

38

47

58

63

76

# Outline

- Markov Decision Processes Definition
- Policy Evaluation
  - State value functions:  $V^\pi$
  - Bellman recursions and Bellman equations
- Policy Optimization
  - Optimal policies  $\pi^*$
  - Optimal action value functions:  $Q^*$
  - Value iteration

# Markov Decision Processes

- Research area initiated in the 50s by Bellman, known under various names:
  - Stochastic optimal control (Control theory)
  - Stochastic shortest path (Operations research)
  - Sequential decision making under uncertainty (Economics)
  - Reinforcement learning (Artificial intelligence, Machine learning)
- A rich variety of elegant theory, mathematics, algorithms, and applications, but also considerable variation in notation.
- We will use the most RL-flavored notations.



## Running example: Mario in a grid-world

1	2	3
4	5	6
7	8	9

A 3x3 grid world with states 1-9. From state 6, an action 'up' leads to state 3 with 80% probability and state 2 with 20% probability.

- 9 possible **states**  $s$
- 4 possible **actions**  $a$ : {Up  $\uparrow$ , Down  $\downarrow$ , Left  $\leftarrow$ , Right  $\rightarrow$ }
- (state, action) results in a **transition**  $T$  into a next state:
  - Normally, we get to the “intended” state;
    - E.g., in state (7), action “ $\uparrow$ ” gets to state (4)
  - If an action would take Mario out of the grid world, stay put;
    - E.g., in state (9), “ $\rightarrow$ ” gets back to state (9)
  - In state (6), action “ $\uparrow$ ” leads to two possibilities:
    - 20% chance to (2)
    - 80% chance to (3).



## Mario in a grid-world, cont'd

- (state, action) pairs give **rewards**:



- in state 3, any action gives reward 1



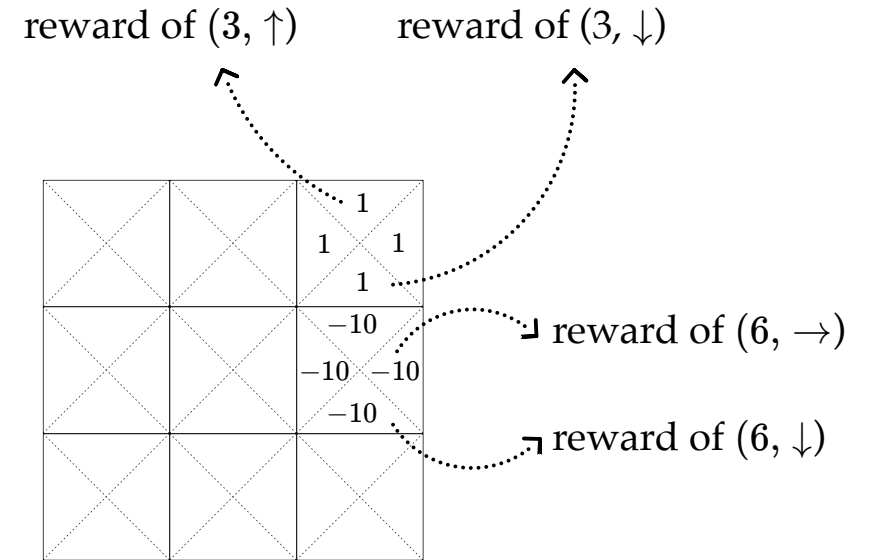
- in state 6, any action gives reward -10



- any other (state, action) pair gives reward 0

- **discount factor**: a scalar of 0.9 that reduces the 'worth' of future rewards depending on when Mario receives them.

- So, e.g., for  $(3, \leftarrow)$  pair, Mario gets
  - at the start of the game, a reward of 1
  - at the 2nd time step, a discounted reward of 0.9
  - at the 3rd time step, a further discounted reward of  $(0.9)^2$  ... and so on



# Markov Decision Processes - Definition and terminologies

- $\mathcal{S}$  : state space, contains all possible states  $s$ .
- $\mathcal{A}$  : action space, contains all possible actions  $a$ .
- $T(s, a, s')$  : the probability of transition from state  $s$  to  $s'$  when action  $a$  is taken.

1	2	3
4	5	6
7	8	9

A 3x3 grid of states. From state 6, a solid arrow points up to state 3 with a probability of 80%, and a dotted arrow points up-left to state 2 with a probability of 20%.

$$T(7, \uparrow, 4) = 1$$

$$T(9, \rightarrow, 9) = 1$$

$$T(6, \uparrow, 3) = 0.8$$

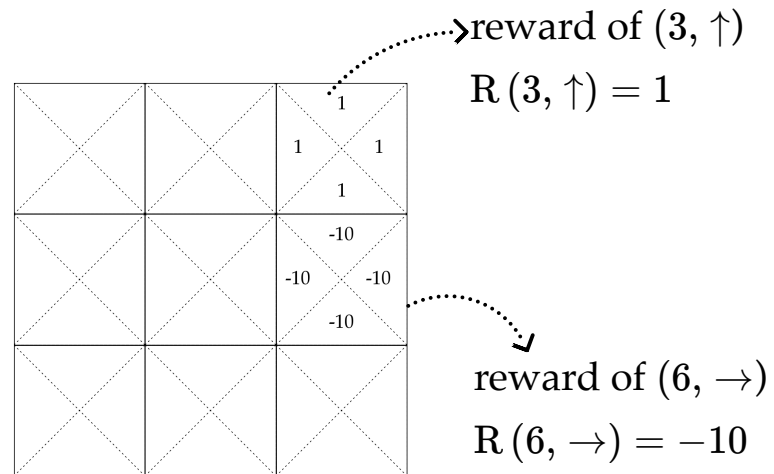
$$T(6, \uparrow, 2) = 0.2$$

In 6.390,

- $\mathcal{S}$  and  $\mathcal{A}$  are small discrete sets, unless otherwise specified.
- $s'$  and  $a'$  are short-hand for the next-timestep state and action.

# Markov Decision Processes - Definition and terminologies

- $\mathcal{S}$  : state space, contains all possible states  $s$ .
- $\mathcal{A}$  : action space, contains all possible actions  $a$ .
- $T(s, a, s')$  : the probability of transition from state  $s$  to  $s'$  when action  $a$  is taken.
- $R(s, a)$  : reward, takes in a (state, action) pair and returns a reward.



In 6.390,

- $\mathcal{S}$  and  $\mathcal{A}$  are small discrete sets, unless otherwise specified.
- $s'$  and  $a'$  are short-hand for the next-timestep state and action.
- $R(s, a)$  is deterministic and bounded.

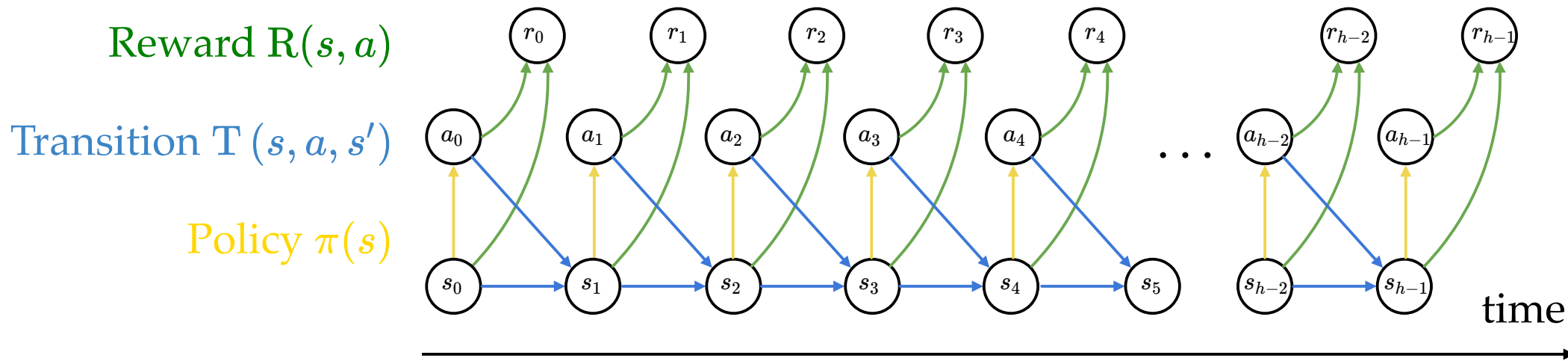
## Markov Decision Processes - Definition and terminologies

- $\mathcal{S}$  : state space, contains all possible states  $s$ .
- $\mathcal{A}$  : action space, contains all possible actions  $a$ .
- $T(s, a, s')$  : the probability of transition from state  $s$  to  $s'$  when action  $a$  is taken.
- $R(s, a)$  : reward, takes in a (state, action) pair and returns a reward.
- $\gamma \in [0, 1]$ : discount factor, a scalar.
- $\pi(s)$  : policy, takes in a state and returns an action.

The goal of an MDP is to find a good policy.

In 6.390,

- $\mathcal{S}$  and  $\mathcal{A}$  are small discrete sets, unless otherwise specified.
- $s'$  and  $a'$  are short-hand for the next-timestep state and action.
- $R(s, a)$  is deterministic and bounded.
- $\pi(s)$  is deterministic.



a trajectory (also called an experience or rollout) of horizon  $h$

$$\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \dots, s_{h-1}, a_{h-1}, r_{h-1})$$

initial state

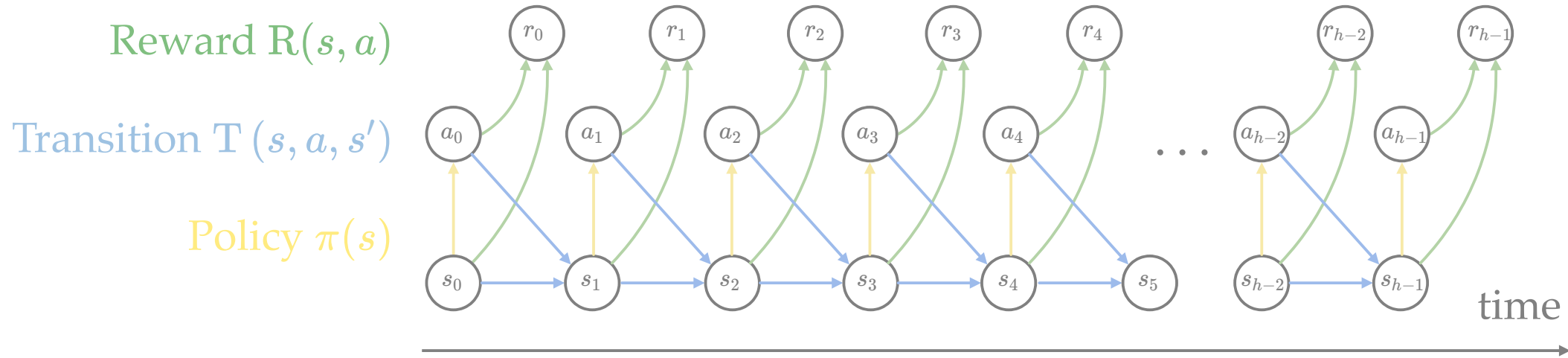
all depends on  $\pi$

- $a_t = \pi(s_t)$
- $r_t = R(s_t, a_t)$
- $T(s, a, s')$

# Outline

- Markov Decision Processes Definition
- Policy Evaluation
  - State value functions:  $V^\pi$
  - Bellman recursions and Bellman equations
- Policy Optimization
  - Optimal policies  $\pi^*$
  - Optimal action value functions:  $Q^*$
  - Value iteration

Starting in state  $s$ , how good is it to follow a *given* policy  $\pi$  for  $h$  time steps?



One idea:

$$R(s_0, \pi(s_0)) + \gamma R(s_1, \pi(s_1)) + \gamma^2 R(s_2, \pi(s_2)) + \gamma^3 R(s_3, \pi(s_3)) + \dots + \gamma^{h-1} R(s_{h-1}, \pi(s_{h-1}))$$

But if we start at  $s_0 = 6$  and follow the "always-up" policy:

states and one special transition:



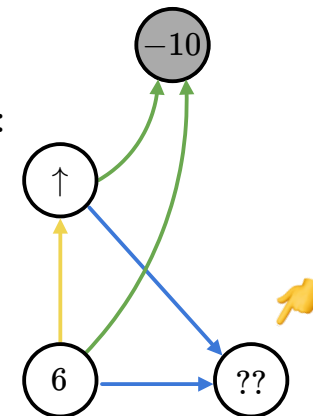
1	2	3
4	5	6
7	8	9

20% ↗      80% ↑

rewards:

0	0	1
0	0	1
0	0	1
0	0	-10
0	0	-10
0	0	-10
0	0	0
0	0	0
0	0	0

trajectory:



## Value functions:

$$\begin{aligned} V_h^\pi(s) &:= \mathbb{E} \left[ \mathbf{R}(s_0, \pi(s_0)) + \gamma \mathbf{R}(s_1, \pi(s_1)) + \gamma^2 \mathbf{R}(s_2, \pi(s_2)) + \gamma^3 \mathbf{R}(s_3, \pi(s_3)) + \dots + \gamma^{h-1} \mathbf{R}(s_{h-1}, \pi(s_{h-1})) \right] \\ &= \mathbb{E} \left[ \sum_{t=0}^{h-1} \gamma^t \mathbf{R}(s_t, \pi(s_t)) \mid s_0 = s, \pi \right] \quad (\text{eq. } \mathbf{1}) \end{aligned}$$

- $V_h^\pi(s)$  : expected sum of discounted rewards starting in state  $s$  and follow  $\pi$  for  $h$  steps
- Value is **long-term**; reward is **immediate** (one-time)
- Convention:  $V_0^\pi(s) = 0$  for all states
- The value expectation in 6.390 is only w.r.t. the transition probabilities  $\mathbf{T}(s, a, s')$



evaluate  $V_h^\pi(s)$  under the "always-up" policy

states and  
one special transition:

1	2	3
4	5	6
7	8	9

Diagram showing a 3x3 grid of states. A dashed arrow points from state 2 to state 3, labeled "20%". Another dashed arrow points from state 3 to state 6, labeled "80%".

rewards

0	0	1	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0
0	0	-10	0	0	-10	0	0	-10
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

- $\pi(s) = \text{"}\uparrow\text{"}, \forall s$
- $\gamma = 0.9$

$$V_h^\uparrow(s) = \mathbb{E} \left[ \sum_{t=0}^{h-1} \gamma^t \mathbf{R}(s_t, \uparrow) \mid s_0 = s \right]$$

$$= \mathbb{E} \left[ \underbrace{\mathbf{R}(s_0, \uparrow) + \gamma \mathbf{R}(s_1, \uparrow) + \dots + \gamma^{h-1} \mathbf{R}(s_{h-1}, \uparrow)}_{h \text{ terms}} \right]$$

horizon  $h = 0$ : no step left

$$V_0^\uparrow(s) = 0$$

0	0	0
0	0	0
0	0	0

horizon  $h = 1$ : receive the rewards

$$V_1^\uparrow(s) = \mathbf{R}(s, \uparrow)$$

0	0	1
0	0	-10
0	0	0



horizon  $h = 2$

states and  
one special transition:

1	2	3
4	5	6
7	8	9

Transitions from state 2: 20% to state 3, 80% to state 6.

rewards

0	0	1	0	0	1	0	0	1
0	0	0	0	0	0	-10	-10	-10
0	0	0	0	0	0	0	0	0

- $\pi(s) = \text{“}\uparrow\text{”}, \forall s$
- $\gamma = 0.9$

$$V_2^\uparrow(s) : \mathbb{E} \left[ \underbrace{R(s_0, \uparrow) + \gamma R(s_1, \uparrow)}_{2 \text{ terms}} \right]$$

$$R(1, \uparrow) + \gamma R(1, \uparrow)$$

$$R(2, \uparrow) + \gamma R(2, \uparrow)$$

$$R(3, \uparrow) + \gamma R(3, \uparrow)$$

$$= 1 + 0.9 * (1) \Rightarrow 1.9$$

$$R(4, \uparrow) + \gamma R(1, \uparrow)$$

0	0	1.9
0	0	

$$R(5, \uparrow) + \gamma R(2, \uparrow)$$



horizon  $h = 2$

states and  
one special transition:

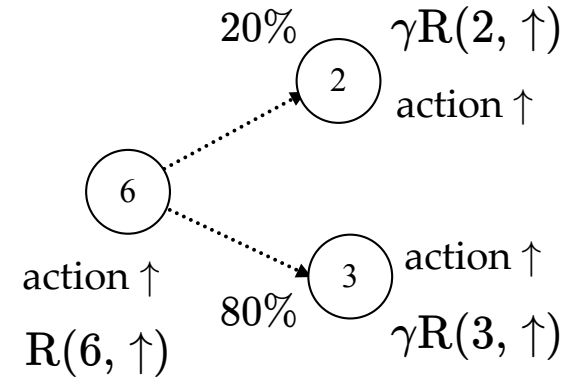
1	2	3
4	5	6
7	8	9

rewards

		1
		1 1
		1
		-10
		-10 -10
		-10

- $\pi(s) = \text{“}\uparrow\text{”}, \forall s$
- $\gamma = 0.9$

$$V_2^\uparrow(s) : \mathbb{E} \left[ \underbrace{R(s_0, \uparrow) + \gamma R(s_1, \uparrow)}_{2 \text{ terms}} \right]$$



0	0	1.9
0	0	-9.28
0	0	-9

$$\begin{aligned}
 & \rightarrow R(6, \uparrow) + \gamma[.2R(2, \uparrow) + .8R(3, \uparrow)] \\
 & = -10 + 0.9 * (0.2 * 0 + 0.8 * 1) \\
 & \Rightarrow -9.28
 \end{aligned}$$

$$R(7, \uparrow) + \gamma R(4, \uparrow)$$

$$R(8, \uparrow) + \gamma R(5, \uparrow)$$

$$\begin{aligned}
 & R(9, \uparrow) + \gamma R(6, \uparrow) \\
 & = 0 + 0.9 * (-10) \Rightarrow -9
 \end{aligned}$$



horizon  $h = 3$

states and  
one special transition:

1	2	3
4	5	6
7	8	9

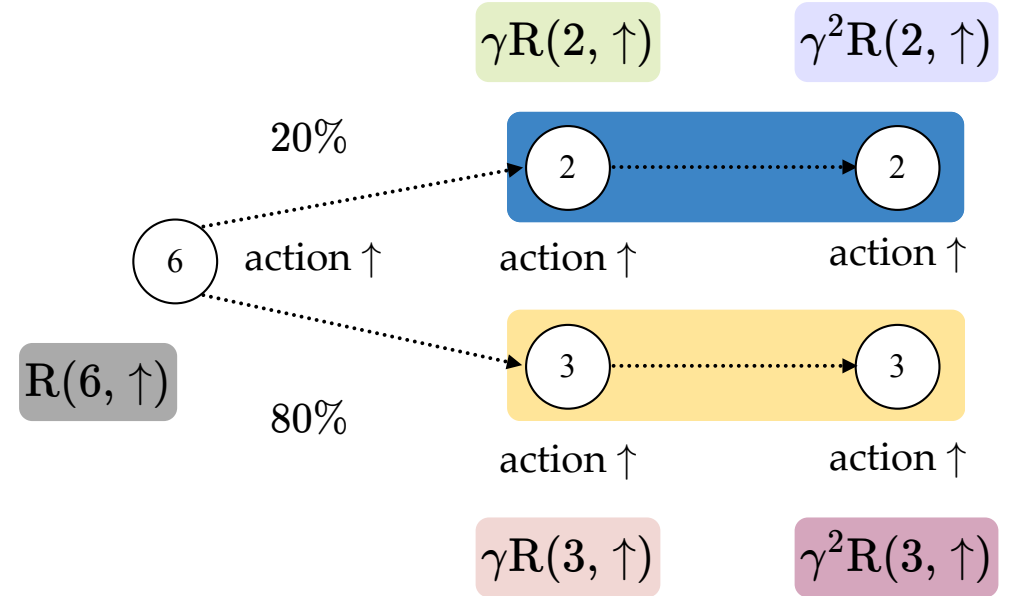
Transitions from state 6: 20% to state 2, 80% to state 3.

rewards

		1
		1 1
		1
		-10
		-10 -10
		-10

- $\pi(s) = \text{“}\uparrow\text{”}, \forall s$
- $\gamma = 0.9$

$$V_3^\uparrow(s) : \mathbb{E} \left[ \underbrace{R(s_0, \uparrow) + \gamma R(s_1, \uparrow) + \gamma^2 R(s_2, \uparrow)}_{3 \text{ terms}} \right]$$



$$\begin{aligned}
 V_3^\uparrow(6) &= R(6, \uparrow) + 20\% \gamma R(2, \uparrow) + 80\% \gamma R(3, \uparrow) + 20\% \gamma^2 R(2, \uparrow) + 80\% \gamma^2 R(3, \uparrow) \\
 &= R(6, \uparrow) + 20\% [ \gamma R(2, \uparrow) + \gamma^2 R(2, \uparrow) ] + 80\% [ \gamma R(3, \uparrow) + \gamma^2 R(3, \uparrow) ] \\
 &= R(6, \uparrow) + 20\% \gamma [ R(2, \uparrow) + \gamma R(2, \uparrow) ] + 80\% \gamma [ R(3, \uparrow) + \gamma R(3, \uparrow) ] \\
 &= R(6, \uparrow) + 20\% \gamma V_2^\uparrow(2) + 80\% \gamma V_2^\uparrow(3)
 \end{aligned}$$

$$V_3^\uparrow(6) = R(6, \uparrow) + 20\% \gamma V_2^\uparrow(2) + 80\% \gamma V_2^\uparrow(3)$$

horizon- $h$  value in state  $s$ : the expected sum of discounted rewards, starting in state  $s$  and following policy  $\pi$  for  $h$  steps.

$$\text{(eq. 2)} \quad V_h^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s, \pi(s), s') V_{h-1}^\pi(s')$$

the immediate reward for taking the policy-prescribed action  $\pi(s)$  in state  $s$ .

$(h - 1)$  horizon future value at a next state  $s'$

sum of future values weighted by the probability of reaching that next state  $s'$

discounted by  $\gamma$

# Bellman Recursion (eq. 2)

$$V_h^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s, \pi(s), s') V_{h-1}^\pi(s')$$

states and  
one special transition:

1	2	3
4	5	6
7	8	9

rewards

0	0	1
0	0	1
0	0	1
0	0	-10
0	0	-10
0	0	-10
0	0	0
0	0	0
0	0	0

- $\pi(s) = \text{“}\uparrow\text{”}, \forall s$
- $\gamma = 0.9$

$$V_1^\uparrow(s) = R(s, \uparrow)$$

0	0	1
0	0	-10
0	0	0

$$V_2^\uparrow(s)$$

0	0	1.9
0	0	-9.28
0	0	-9

$$V_2^\uparrow(9) = R(9, \uparrow) + \gamma[V_1^\uparrow(6)]$$

$$= 0 + 0.9 \times [-10]$$

$$= -9$$

# Bellman Recursion (eq. 2)

$$V_h^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s, \pi(s), s') V_{h-1}^\pi(s')$$

states and  
one special transition:

1	2	3
4	5	6
7	8	9

Diagram showing a 3x3 grid of states. A dashed arrow points from state 2 to state 3 with a label "20%". A solid arrow points from state 3 to state 6 with a label "80%".

rewards

0	0	1
0	0	1
0	0	1
0	0	-10
0	0	-10
0	0	-10
0	0	0
0	0	0
0	0	0

Diagram showing a 3x3 grid of rewards. The top row has values 0, 0, 1. The middle row has 0, 0, 1. The bottom row has 0, 0, 1. The rightmost column has values 1, 1, 1. The bottom-right cell (state 6) is shaded gray and contains -10. The cells below it (states 5 and 4) also contain -10.

- $\pi(s) = \text{“}\uparrow\text{”}, \forall s$
- $\gamma = 0.9$

$V_1^\uparrow(s) = R(s, \uparrow)$

0	0	1
0	0	-10
0	0	0

$V_2^\uparrow(s)$

0	0	1.9
0	0	-9.28
0	0	-9

$V_3^\uparrow(s)$

0	0	2.71
0	0	-8.63
0	0	-8.35

$$\begin{aligned} V_3^\uparrow(6) &= R(6, \uparrow) + \gamma[.2 \times V_2^\uparrow(2) + .8 \times V_2^\uparrow(3)] \\ &= -10 + .9[.2 \times 0 + 0.8 \times 1.9] \\ &= -8.632 \end{aligned}$$

# Bellman Recursion (eq. 2)

$$V_h^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s, \pi(s), s') V_{h-1}^\pi(s')$$

states and  
one special transition:

1	2	3
4	5	6
7	8	9

Diagram showing a 3x3 grid of states. A dashed arrow points from state 2 to state 3 with a label "20%". A solid arrow points from state 3 to state 6 with a label "80%".

rewards

0	0	1
0	0	1
0	0	1
0	0	-10
0	0	-10
0	0	-10
0	0	0
0	0	0
0	0	0

Diagram showing a 3x3 grid of rewards. The top row and middle-left cells have 0. The top-right cell has 1. The middle-right and bottom rows have -10. The bottom row has 0.

- $\pi(s) = \text{"}\uparrow\text{"}, \forall s$
- $\gamma = 0.9$

$V_4^\uparrow(s)$

0	0	3.44
0	0	-8.05
0	0	-7.77

$V_5^\uparrow(s)$

0	0	4.1
0	0	-7.52
0	0	-7.24

$V_6^\uparrow(s)$

0	0	4.69
0	0	-7.05
0	0	-6.77

...

$$\begin{aligned} V_6^\uparrow(6) &= R(6, \uparrow) + \gamma[.2 \times V_5^\uparrow(2) + .8 \times V_5^\uparrow(3)] \\ &= -10 + .9[.2 \times 0 + 0.8 \times 4.10] \\ &= -7.048 \end{aligned}$$

# Bellman Recursion (eq. 2)

$$V_h^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s, \pi(s), s') V_{h-1}^\pi(s')$$

states and  
one special transition:

1	2	3
4	5	6
7	8	9

Diagram showing a 3x3 grid of states. A dashed arrow points from state 2 to state 3 with a label "20%". A solid arrow points from state 3 to state 6 with a label "80%".

rewards

0	0	1
0	0	1
0	0	1
0	0	-10
0	0	-10
0	0	-10
0	0	0
0	0	0
0	0	0

Diagram showing a 3x3 grid of rewards. The top row has values 0, 0, 1. The middle row has values 0, 0, -10. The bottom row has values 0, 0, 0. The cell (3,3) with value -10 is shaded gray.

- $\pi(s) = \text{“}\uparrow\text{”}, \forall s$
- $\gamma = 0.9$

$V_{59}^\uparrow(s)$

0	0	9.98
0	0	-2.82
0	0	-2.54

$V_{60}^\uparrow(s)$

0	0	9.98
0	0	-2.81
0	0	-2.53

Note: The cell (1,2) with value 0 is highlighted in yellow, and the cell (1,3) with value 9.98 is highlighted in pink.

$V_{61}^\uparrow(s)$

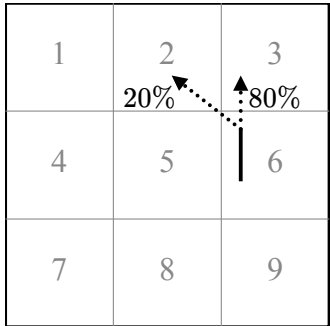
0	0	9.98
0	0	-2.81
0	0	-2.53

Note: The cell (2,3) with value -2.81 is highlighted in light blue. Ellipses follow the table.

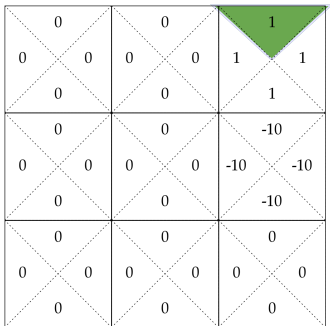
$$\begin{aligned}
 V_{61}^\uparrow(6) &= R(6, \uparrow) + \gamma[.2 \times V_{60}^\uparrow(2) + .8 \times V_{60}^\uparrow(3)] \\
 &= -10 + .9[.2 \times 0 + 0.8 \times 9.98] \\
 &= -2.8144
 \end{aligned}$$

# Value functions converge as $h \rightarrow \infty$

states and  
one special transition:



rewards



- $\pi(s) = \text{“}\uparrow\text{”}, \forall s$
- $\gamma = 0.9$

$$V_{60}^{\uparrow}(s)$$

0	0	9.98
0	0	-2.81
0	0	-2.53

$$V_{61}^{\uparrow}(s)$$

0	0	9.98
0	0	-2.81
0	0	-2.53

...

$$V_{\infty}^{\uparrow}(s)$$

0	0	10
0	0	-2.8
0	0	-2.52

- As we extend the horizon, value differences shrink
- because longer-term rewards are heavily discounted
- so, as  $h \rightarrow \infty$ , the value functions stop changing
- convergence can be seen, e.g., via  $V_{\infty}^{\uparrow}(3) = 1 + .9 + .9^2 + .9^3 + \dots = 10$

Typically,  $\gamma < 1$  to ensure  $V_{\infty}$  is finite.

As horizon  $h \rightarrow \infty$ , the Bellman recursion becomes the Bellman equation

states and  
one special transition:

1	2	3
4	5	6
7	8	9

Diagram showing transitions from state 2: a dotted arrow to state 3 labeled 20%, and a solid arrow to state 6 labeled 80%.

rewards

0	0	1
0	0	1
0	0	1
0	0	-10
0	0	-10
0	0	-10
0	0	0
0	0	0
0	0	0

- $\pi(s) = \text{“}\uparrow\text{”}, \forall s$
- $\gamma = 0.9$

Recursion (finite  $h$ ) **2**  $V_h^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s, \pi(s), s') V_{h-1}^\pi(s')$

Equation ( $h \rightarrow \infty$ ) **3**  $V_\infty^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s, \pi(s), s') V_\infty^\pi(s')$

A system of  $|\mathcal{S}|$  self-consistent linear equations, one for each state

$V_\infty^\uparrow(s)$

0	0	10
0	0	-2.8
0	0	-2.52

$$V_\infty^\uparrow(3) = R(3, \uparrow) + \gamma[V_\infty^\uparrow(3)]$$

$$= 1 + .9 \times 10 \Rightarrow 10$$

$$V_\infty^\uparrow(6) = R(6, \uparrow) + \gamma[.2 \times V_\infty^\uparrow(2) + .8 \times V_\infty^\uparrow(3)]$$

$$= -10 + .9[.2 \times 0 + 0.8 \times 10] \Rightarrow -2.8$$

Quick summary



Use the definition and sum up expected rewards:

1 
$$V_h^\pi(s) := \mathbb{E} \left[ \sum_{t=0}^{h-1} \gamma^t \mathbf{R}(s_t, \pi(s_t)) \mid s_0 = s, \pi \right]$$

Or, leverage the recursive structure:

2 **finite-horizon Bellman recursions**

$$V_h^\pi(s) = \mathbf{R}(s, \pi(s)) + \gamma \sum_{s'} \mathbf{T}(s, \pi(s), s') V_{h-1}^\pi(s')$$

3 **infinite-horizon Bellman equations**

$$V_\infty^\pi(s) = \mathbf{R}(s, \pi(s)) + \gamma \sum_{s'} \mathbf{T}(s, \pi(s), s') V_\infty^\pi(s')$$

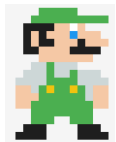
# Outline

- Markov Decision Processes Definition
- Policy Evaluation
  - State value functions:  $V^\pi$
  - Bellman recursions and Bellman equations
- Policy Optimization
  - Optimal policies  $\pi^*$
  - Optimal action value functions:  $Q^*$
  - Value iteration

# Optimal policy $\pi^*$

- Intuitively, an optimal policy  $\pi^*$  is a policy that yields the highest possible value  $V_h^*(s)$  from every state
- An MDP in 6.390 has a unique optimal value  $V_h^*(s)$
- Optimal policy  $\pi^*$  might not be unique

e.g. in the "Luigi game", any policy is an optimal policy



States and one special transition:

1	2	3
4	5	6
7	8	9

Diagram showing a 3x3 grid of states (1-9). A dashed arrow points from state 3 to state 2 with a label "20%". A solid arrow points from state 6 to state 3 with a label "80%".

Rewards:

1	1	1
1	1	1
1	1	1

Diagram showing a 3x3 grid of rewards. Each cell contains a '1' and is crossed by a dashed diagonal line from top-left to bottom-right.

$\gamma = 0.9$

## Optimal policy $\pi^*$

- Formally: an optimal policy  $\pi^*$  is such that:

$$V_h^{\pi^*}(s) = \max_{\pi} V_h^{\pi}(s) = V_h^*(s), \forall s \in \mathcal{S}$$

- How to search for an optimal policy  $\pi^*$ ?
- Even if we tediously enumerate over all  $\pi$ , do policy evaluation, compare values to get  $V_h^*(s)$ ...it's not yet clear how to choose actions.

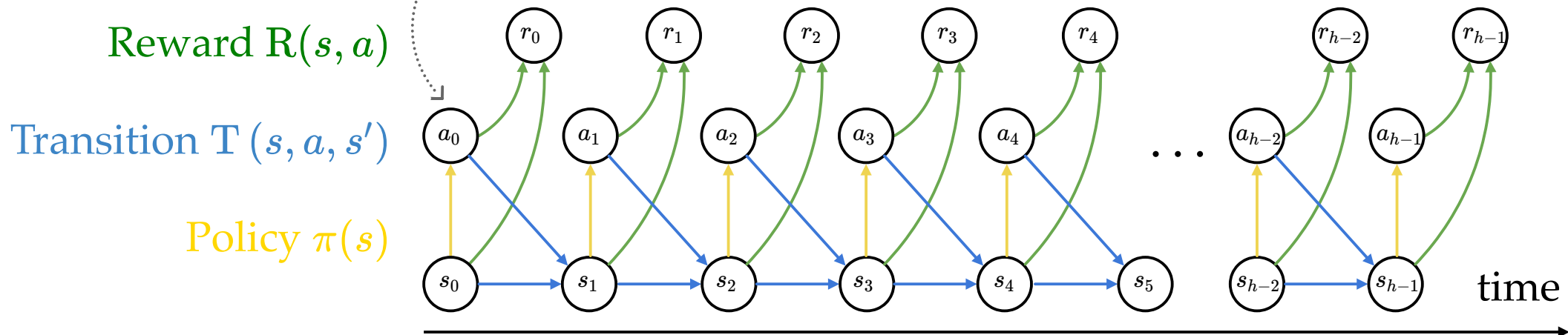
$V^*(s)$  is defined over states, not actions.

It tells us where we'd like to *be* — not what we should do to *get* there.

if we've acted optimally for  $h$  steps:  $V_h^*(s)$

with the first step action  
that led to the optimal future

we must have acted optimally from  
the first step onward  $V_{h-1}^*(s')$



$$V_h^\pi(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s, \pi(s), s') V_{h-1}^\pi(s') \quad (\text{recall, eq. 2, for any policy})$$

$$V_h^*(s) = \max_a [R(s, a) + \gamma \sum_{s'} T(s, a, s') V_{h-1}^*(s')] \quad (\text{new, eq. 4, for an optimal policy})$$

$$V_h^*(s) = \max_a \left[ \underbrace{R(s, a) + \gamma \sum_{s'} T(s, a, s') V_{h-1}^*(s')}_{Q_h^*(s, a)} \right] \quad (\text{eq. 4})$$

Define the optimal state-action value functions  $Q_h^*(s, a)$  :

the expected sum of discounted rewards, obtained by

- starting in state  $s$
- take action  $a$ , for one step
- act **optimally** thereafter for the remaining  $(h - 1)$  steps

$Q^*$  satisfies:

$$Q_h^*(s, a) = R(s, a) + \gamma \sum_{s'} T(s, a, s') \max_{a'} Q_{h-1}^*(s', a') \quad (\text{eq. 5})$$



$Q_h^*(s, a)$ : the value for

- starting in state  $s$ ,
- take action  $a$ , for one step
- act **optimally** thereafter for the remaining  $(h - 1)$  steps

States and one special transition:

transition:

1	2	3
4	5	6
7	8	9

20% (dotted arrow from 2 to 3)  
80% (solid arrow from 2 to 6)

$\gamma = 0.9$

Rewards:

0	0	1	0	0	1	0	0	1
0	0	0	0	0	1	0	0	1
0	0	0	0	0	-10	0	0	-10
0	0	0	0	0	-10	0	0	-10
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

$$Q_0^*(s, a)$$

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

$$Q_1^*(s, a) = R(s, a)$$

0	0	1	0	0	1	0	0	1
0	0	1	0	0	1	0	0	1
0	0	0	0	0	-10	0	0	-10
0	0	0	0	0	-10	0	0	-10
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0



$Q_h^*(s, a)$ : the value for

- starting in state  $s$ ,
- take action  $a$ , for one step
- act **optimally** thereafter for the remaining  $(h - 1)$  steps

States and one special transition:

1	2	3
4	5	6
7	8	9

$\gamma = 0.9$

20% (dotted arrow from 2 to 3)  
80% (solid arrow from 2 to 6)

Rewards:

0	0	1
0	0	1
0	0	-10
0	0	-10
0	0	-10
0	0	0
0	0	0
0	0	0

$Q_1^*(s, a)$

0	0	1
0	0	1
0	0	-10
0	0	-10
0	0	-10
0	0	0
0	0	0
0	0	0

$Q_2^*(s, a)$

		-8

Consider  $Q_2^*(3, \downarrow)$

- receive  $R(3, \downarrow)$

- next state  $s' = 6$ , act **optimally** for the remaining one timestep
  - receive  $\max_{a'} Q_1^*(6, a')$

$$\begin{aligned}
 Q_2^*(3, \downarrow) &= R(3, \downarrow) + \gamma \max_{a'} Q_1^*(6, a') \\
 &= 1 + .9 \times -10 \\
 &= -8
 \end{aligned}$$



$Q_h^*(s, a)$ : the value for

- starting in state  $s$ ,
- take action  $a$ , for one step
- act **optimally** thereafter for the remaining  $(h - 1)$  steps

States and one special transition:

1	2	3
4	5	6
7	8	9

20% (dotted arrow from 2 to 3)  
80% (solid arrow from 2 to 6)

Rewards:

0	0	1	0	0	1
0	0	0	0	0	1
0	0	0	-10	-10	-10
0	0	0	0	0	0
0	0	0	0	0	0

$\gamma = 0.9$

$Q_1^*(s, a)$

0	0	1	0	0	1
0	0	0	0	0	1
0	0	0	-10	-10	-10
0	0	0	0	0	0
0	0	0	0	0	0

$Q_2^*(s, a)$

		1			-8

Let's consider  $Q_2^*(3, \leftarrow)$

- receive  $R(3, \leftarrow)$
- next state  $s' = 2$ , act **optimally** for the remaining one timestep
  - receive  $\max_{a'} Q_1^*(2, a')$

$$\begin{aligned}
 Q_2^*(3, \leftarrow) &= R(3, \leftarrow) + \gamma \max_{a'} Q_1^*(2, a') \\
 &= 1 + .9 \times 0 \\
 &= 1
 \end{aligned}$$



$Q_h^*(s, a)$ : the value for

- starting in state  $s$ ,
- take action  $a$ , for one step
- act **optimally** thereafter for the remaining  $(h - 1)$  steps

States and one special transition:

1	2	3
	20%	80%
4	5	6
7	8	9

$\gamma = 0.9$

Rewards:

0	0	0	0	0	1	1
0	0	0	0	0	1	1
0	0	0	0	0	-10	-10
0	0	0	0	0	-10	-10
0	0	0	0	0	0	0
0	0	0	0	0	0	0

$Q_1^*(s, a)$

0	0	0	0	1	1
0	0	0	0	1	1
0	0	0	0	-10	-10
0	0	0	0	-10	-10
0	0	0	0	0	0
0	0	0	0	0	0

$Q_2^*(s, a)$

					1.9
				1	-8

Let's consider  $Q_2^*(3, \uparrow)$

- receive  $R(3, \uparrow)$
- next state  $s' = 3$ , act **optimally** for the remaining one timestep
  - receive  $\max_{a'} Q_1^*(3, a')$

$$\begin{aligned}
 Q_2^*(3, \uparrow) &= R(3, \uparrow) + \gamma \max_{a'} Q_1^*(3, a') \\
 &= 1 + .9 \times 1 \\
 &= 1.9
 \end{aligned}$$



$Q_h^*(s, a)$ : the value for

- starting in state  $s$ ,
- take action  $a$ , for one step
- act **optimally** thereafter for the remaining  $(h - 1)$  steps

States and one special transition:

1	2	3
4	5	6
7	8	9

$\gamma = 0.9$

Transitions: 2 to 3 (80%), 2 to 5 (20%)

Rewards:

0	0	1	0	0	1
0	0	0	0	0	1
0	0	0	-10	-10	-10
0	0	0	0	0	0
0	0	0	0	0	0

$Q_1^*(s, a)$

0	0	1	0	0	1
0	0	0	0	0	1
0	0	0	-10	-10	-10
0	0	0	0	0	0
0	0	0	0	0	0

$Q_2^*(s, a)$

		1.9			1.9
		1			-8

Let's consider  $Q_2^*(3, \rightarrow)$

- receive  $R(3, \rightarrow)$
- next state  $s' = 3$ , act **optimally** for the remaining one timestep
  - receive  $\max_{a'} Q_1^*(3, a')$

$$\begin{aligned}
 Q_2^*(3, \rightarrow) &= R(3, \rightarrow) + \gamma \max_{a'} Q_1^*(3, a') \\
 &= 1 + .9 \times 1 \\
 &= 1.9
 \end{aligned}$$



$Q_h^*(s, a)$ : the value for

- starting in state  $s$ ,
- take action  $a$ , for one step
- act **optimally** thereafter for the remaining  $(h - 1)$  steps

States and one special transition:

transition:

$\gamma = 0.9$

1	2	3
4	5	6
7	8	9

Transitions: 2 to 3 (80%), 2 to 5 (20%)

Rewards:

0	0	1	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	-10	-10	-10
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

$Q_1^*(s, a)$

0	0	1	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	-10	-10	-10
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

$Q_2^*(s, a)$

		1.9			1.9			-8
		1			1.9			-8
								-19

Let's consider  $Q_2^*(6, \rightarrow)$

- receive  $R(6, \rightarrow)$

- act optimally at the next state  $s' = 6$   
receive  $\max_{a'} Q_1^*(6, a')$

$$\begin{aligned}
 Q_2^*(6, \rightarrow) &= R(6, \rightarrow) + \gamma [\max_{a'} Q_1^*(6, a')] \\
 &= -10 + .9 \times -10 \Rightarrow -19
 \end{aligned}$$



$Q_h^*(s, a)$ : the value for

- starting in state  $s$ ,
- take action  $a$ , for one step
- act **optimally** thereafter for the remaining  $(h - 1)$  steps

States and one special transition:

1	2	3
4	5	6
7	8	9

$\gamma = 0.9$

Rewards:

0	0	1
0	0	1
0	0	-10
0	0	-10
0	0	-10
0	0	0
0	0	0
0	0	0

$Q_1^*(s, a)$

0	0	1
0	0	1
0	0	-10
0	0	-10
0	0	-10
0	0	0
0	0	0
0	0	0

$Q_2^*(s, a)$

		1.9
		1 1.9
		-8
		-9.28
		-19

Let's consider  $Q_2^*(6, \uparrow)$

- receive  $R(6, \uparrow)$
- act **optimally** at the next state  $s'$ 
  - 20% chance,  $s' = 2$ , act optimally, get  $\max_{a'} Q_1^*(2, a')$
  - 80% chance,  $s' = 3$ , act optimally, get  $\max_{a'} Q_1^*(3, a')$

$$\begin{aligned}
 Q_2^*(6, \uparrow) &= R(6, \uparrow) + \gamma[.2 \max_{a'} Q_1^*(2, a') + .8 \max_{a'} Q_1^*(3, a')] \\
 &= -10 + .9[.2 \times 0 + .8 \times 1] \Rightarrow -9.28
 \end{aligned}$$



$Q_h^*(s, a)$ : the value for

- starting in state  $s$ ,
- take action  $a$ , for one step
- act **optimally** thereafter for the remaining  $(h - 1)$  steps

States and one special transition:

1	2	3
4	5	6
7	8	9

$\gamma = 0.9$

Rewards:

0	0	1
0	0	1
0	0	-10
0	0	-10
0	0	-10
0	0	0
0	0	0
0	0	0

$Q_2^*(s, a)$

0	0	1.9
0	0.9	1.9
0	0	-8
0	0	-9.28
0	0	-9
0	0	-10
0	0	-10
0	0	-19
0	0	-10
0	0	-9
0	0	0
0	0	0

$Q_3^*(s, a)$

0	0.81	2.71
0	0.81	1.71
0	0	1.81
0	0	-7.35
0	0.81	-8.47
0	0	-8.35
0	0	-10
0	0	-10
0	0	-18.35
0	0	-10
0	0	-8.35
0	0	0
0	0	0

Let's consider  $Q_3^*(6, \uparrow)$

- receive  $R(6, \uparrow)$
- act **optimally** at the next state  $s'$ 
  - 20% chance,  $s' = 2$ , act optimally, get  $\max_{a'} Q_2^*(2, a')$
  - 80% chance,  $s' = 3$ , act optimally, get  $\max_{a'} Q_2^*(3, a')$

$$\begin{aligned}
 Q_3^*(6, \uparrow) &= R(6, \uparrow) + \gamma[.2 \max_{a'} Q_2^*(2, a') + .8 \max_{a'} Q_2^*(3, a')] \\
 &= -10 + .9[.2 \times 0.9 + .8 \times 1.9] \Rightarrow -8.47
 \end{aligned}$$

Value iteration: what we just did, iteratively invoke (eq. 5):

$$Q_h^*(s, a) = R(s, a) + \gamma \sum_{s'} T(s, a, s') \max_{a'} Q_{h-1}^*(s', a')$$

### Value Iteration

1. **for**  $s \in \mathcal{S}, a \in \mathcal{A}$ :

2.  $Q_{\text{old}}(s, a) = 0$

3. **while** True:

4. **for**  $s \in \mathcal{S}, a \in \mathcal{A}$ :

5.  $Q_{\text{new}}(s, a) \leftarrow R(s, a) + \gamma \sum_{s'} T(s, a, s') \max_{a'} Q_{\text{old}}(s', a')$

6. **if**  $\max_{s,a} |Q_{\text{old}}(s, a) - Q_{\text{new}}(s, a)| < \epsilon$ :

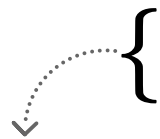
7. **return**  $Q_{\text{new}}$

8.  $Q_{\text{old}} \leftarrow Q_{\text{new}}$

if run this block  $h$  times

and break, then the  
returns are exactly  $Q_h^*$

$Q_{\infty}^*(s, a)$



Optimal policy easily extracted: **6**  $\pi_h^*(s) = \arg \max_a Q_h^*(s, a)$

e.g. the best actions to take in state 5

$Q_1^*(s, a)$

$Q_2^*(s, a)$

$Q_3^*(s, a)$

...

$Q_\infty^*(s, a)$

0	0	1
0	0	1
0	0	1
0	0	-10
0	0	-10
0	0	-10
0	0	0
0	0	0
0	0	0

0	0	1.9
0	0	1.9
0	0	-8
0	0	-9.28
0	0	-9
0	0	-10
0	0	-10
0	0	-9
0	0	0
0	0	0

0	0.81	2.71
0	0.81	1.71
0	0	-7.35
0	0.81	-8.47
0	0	-8.35
0	0	-10
0	0	-10
0	0	-8.35
0	0	0
0	0	0

7.29	8.1	10
7.29	8.1	9.1
6.56	7.29	-0.07
7.29	8.1	-1.18
6.56	7.29	-2.71
5.9	6.56	-4.1
6.56	7.29	-1.07
5.9	6.56	5.9
5.9	6.56	5.9

- For finite  $h$ , optimal policy  $\pi_h^*$  depends on how many time steps are left
- When  $h \rightarrow \infty$ , time no longer matters, i.e., there exists a stationary  $\pi^*$

# Summary

- A Markov decision process  $(\mathcal{S}, \mathcal{A}, T, R, \gamma)$  is the mathematical framework for sequential decision-making and the foundation of reinforcement learning.
- To evaluate a *given* policy  $\pi$ , we compute state value functions  $V^\pi(s)$  via the Bellman recursion (finite horizon) or the Bellman equation (infinite horizon).
- To *find* an optimal policy, we compute  $Q^*(s, a)$  via the value iteration algorithm, then act greedily:  $\pi^*(s) = \arg \max_a Q^*(s, a)$ .