https://introml.mit.edu/

6.390 Intro to Machine Learning

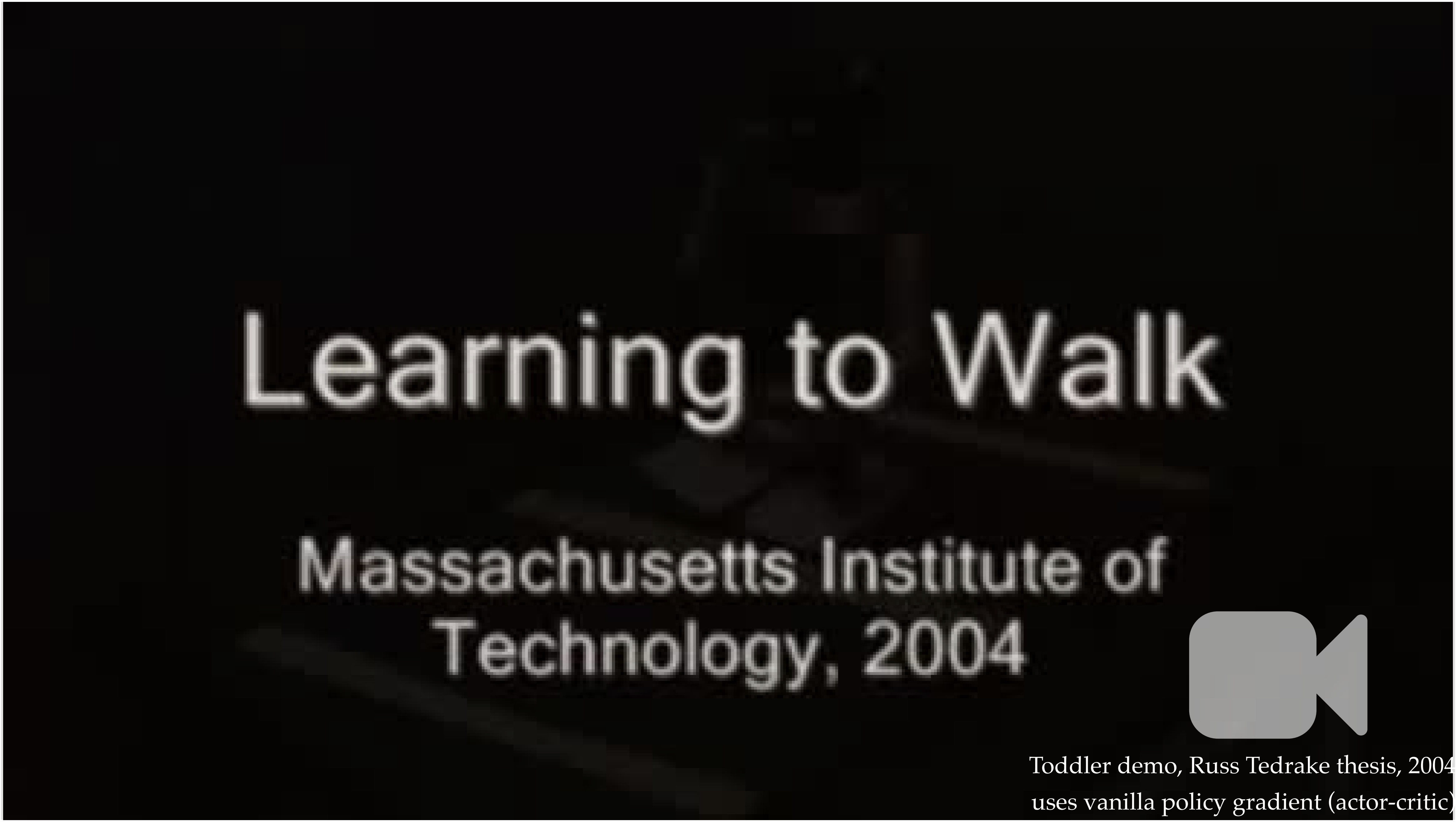
Lecture 11: Markov Decision Processes

Shen Shen

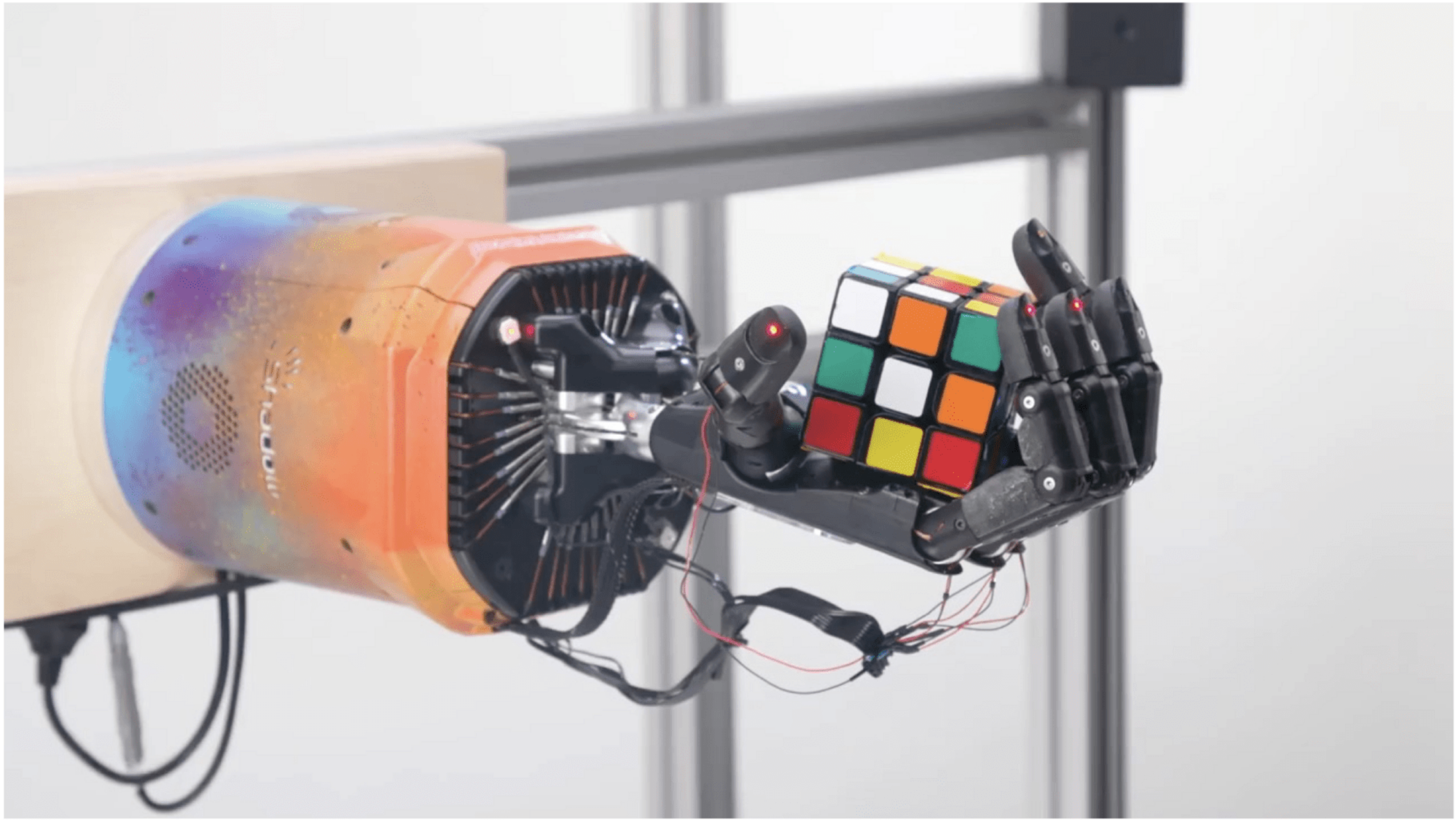
Nov 13, 2025

11am, Room 10-250

Interactive Slides and Lecture Recording



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



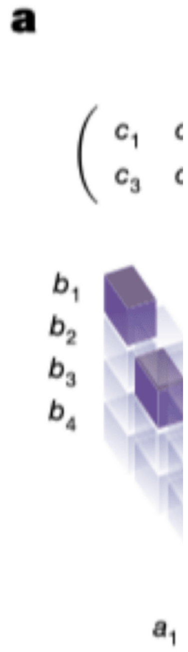
[Hu 15]

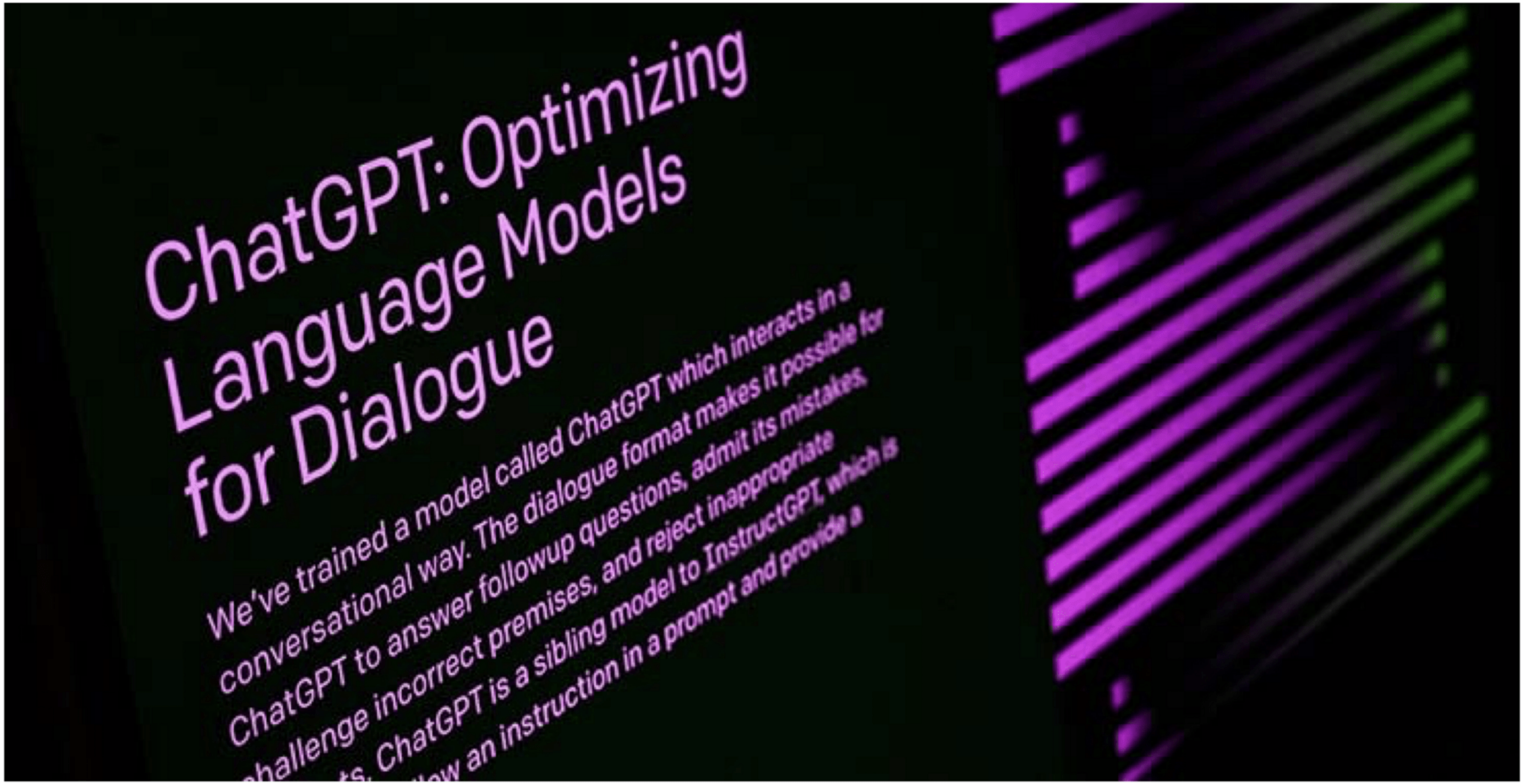
[Solving Rubik's cube with a robot hand. OpenAI. 2019]

Reinforcement Learning with Human Feedback

Discovering faster matrix multiplication algo

https://doi.org/
Received: 2 Oc
Accepted: 2 Au
Published onlir





[Aligning language models to follow instructions. Ouyang et al. 2022] https://say-can.github.io/img/demo_sequence_compressed.mp4

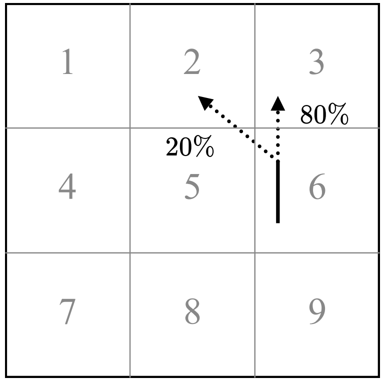
Outline

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

Markov Decision Processes

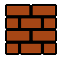
- Research area initiated in the 50s by Bellman, known under various names:
  - Stochastic optimal control (Control theory)
  - Stochastic shortest path (Operations research)
  - Sequential decision making under uncertainty (Economics)
  - Reinforcement learning (Artificial intelligence, Machine learning)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

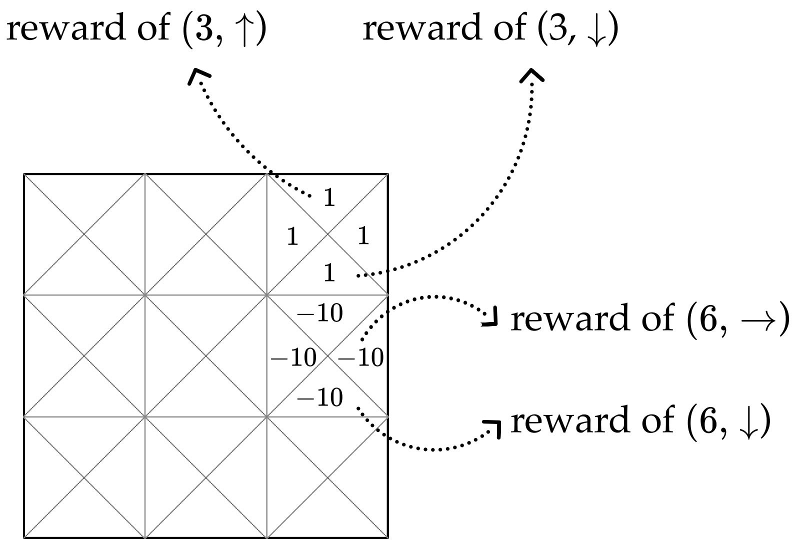
Running example: Mario in a grid-world



- 9 possible **states** $\boldsymbol{s}$
- 4 possible **actions** $\boldsymbol{a}$: {Up ↑, Down ↓, Left ←, Right →}
- (state, action) results in a **transition** T into a next state:
  - Normally, we get to the "intended" state;
    - E.g., in state (7), action "↑" gets to state (4)
  - If an action would take Mario out of the grid world, stay put;
    - E.g., in state (9), "→" gets back to state (9)
  - In state (6), action "↑" 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, ←) 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

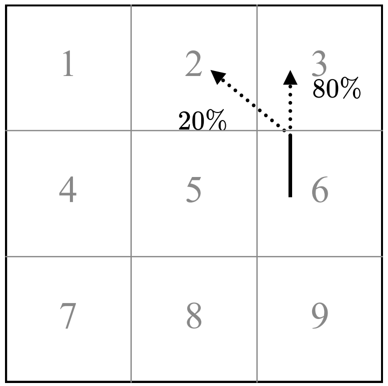
- $\mathcal{S}$: state space, contains all possible states s.
- $\mathcal{A}$: action space, contains all possible actions a.

In 6.390,

- $\mathcal{S}$ and $\mathcal{A}$ are small discrete sets, unless otherwise specified.

Markov Decision Processes - Definition and terminologies

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



$\mathrm{T}(7, \uparrow, 4) = 1$

$\mathrm{T}(9, \rightarrow, 9) = 1$

$\mathrm{T}(6, \uparrow, 3) = 0.8$

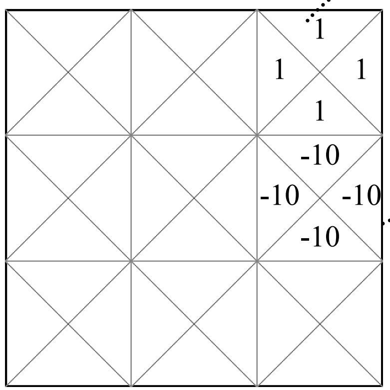
$\mathrm{T}(6, \uparrow, 2) = 0.2$

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.
- $\mathrm{T}\left(s, a, s'\right)$: the probability of transition from state s to s' when action a is taken.
- $\mathrm{R}(s, a)$: reward, takes in a (state, action) pair and returns a reward.



reward of $(3, \uparrow)$

$\mathrm{R}\left(3, \uparrow\right) = 1$

reward of $(6, \rightarrow)$

$\mathrm{R}\left(6, \rightarrow\right) = -10$

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.
- $\mathrm{R}(s, a)$ is deterministic and bounded.

Markov Decision Processes - Definition and terminologies

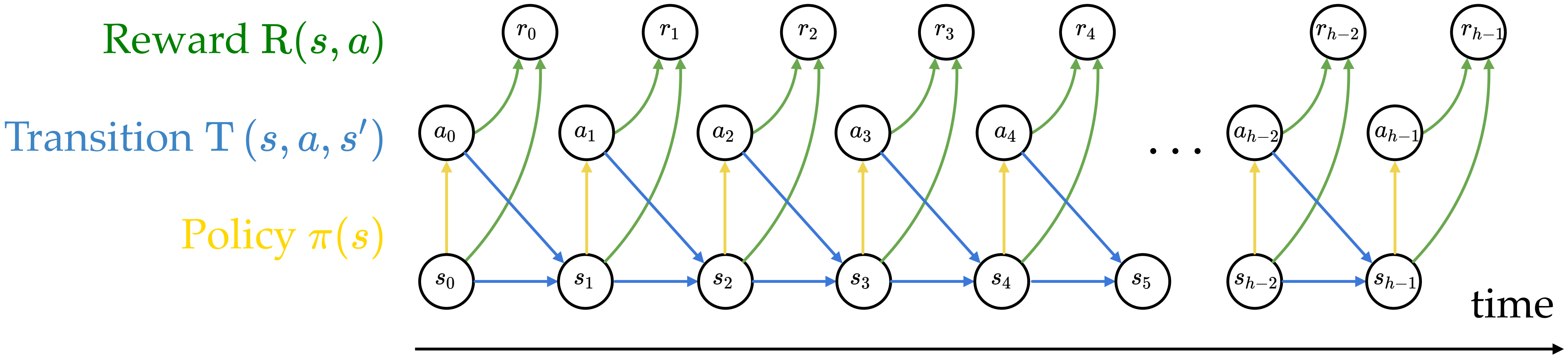
- $\mathcal{S}$: state space, contains all possible states s.
- $\mathcal{A}$: action space, contains all possible actions a.
- $\mathrm{T}\left(s, a, s'\right)$: the probability of transition from state s to s' when action a is taken.
- $\mathrm{R}(s, a)$: reward, takes in a (state, action) pair and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



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

$$\tau = (s_0, \underbrace{a_0, r_0, s_1, a_1, r_1, \dots s_{h-1}, a_{h-1}, r_{h-1}}_{\text{all depends on } \pi})$$

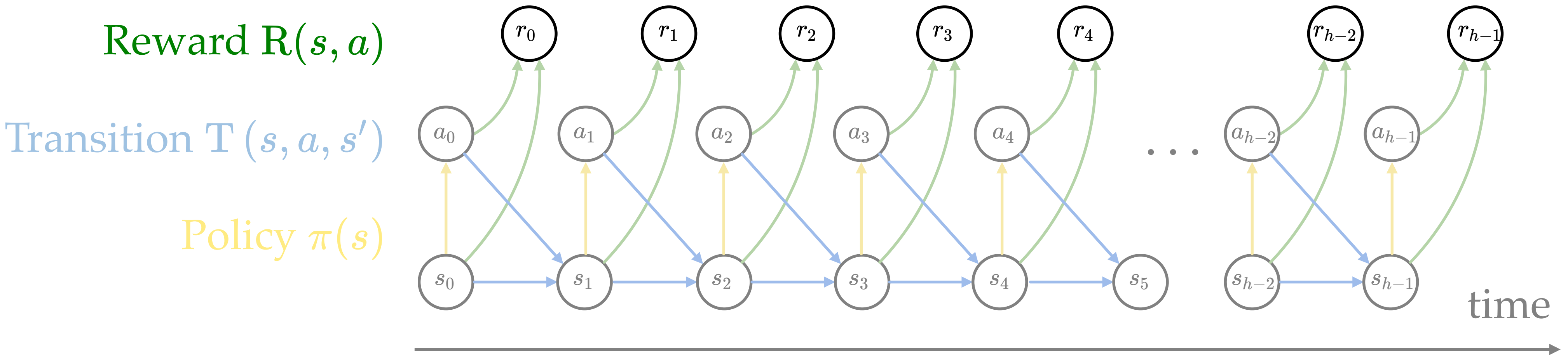
initial state

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

Outline

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

Starting in a given s_0, how good is it to follow a *given* policy π for h time steps?

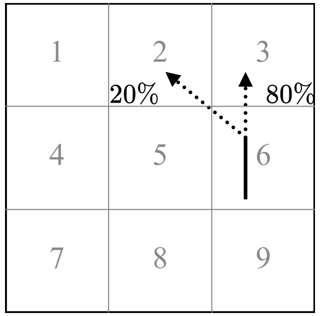


One idea:

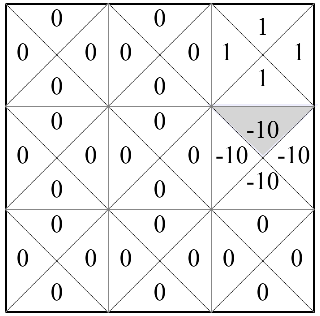
$$\mathrm{R}(s_0, \pi(s_0)) + \gamma \mathrm{R}(s_1, \pi(s_1)) + \gamma^2 \mathrm{R}(s_2, \pi(s_2)) + \gamma^3 \mathrm{R}(s_3, \pi(s_3)) + \ldots + \gamma^{h-1} \mathrm{R}(s_{h-1}, \pi(s_{h-1}))$$

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

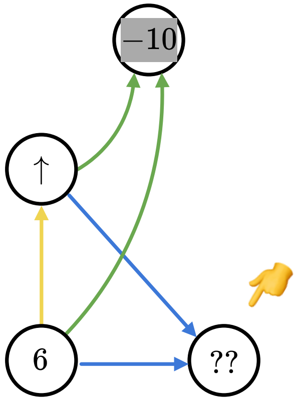
states and one special transition:



rewards:



trajectory:



Starting in a given s_0, how good is it to follow a *given* policy π for h time steps?

Value functions:

$$\mathrm{V}_h^\pi(s) := \mathbb{E}\Big[\mathrm{R}(s_0, \pi(s_0)) \ + \ \gamma \mathrm{R}(s_1, \pi(s_1)) \ + \ \gamma^2 \mathrm{R}(s_2, \pi(s_2)) \ + \ \gamma^3 \mathrm{R}(s_3, \pi(s_3)) \ + \ \ldots \ + \ \gamma^{h-1} \mathrm{R}(s_{h-1}, \pi(s_{h-1}))\Big]$$

$$= \mathbb{E}\left[\sum_{t=0}^{h-1} \gamma^t \mathrm{R}\left(s_t, \pi\left(s_t\right)\right) \mid s_0 = s, \pi\right] \qquad \text{(eq. 1️⃣)}$$

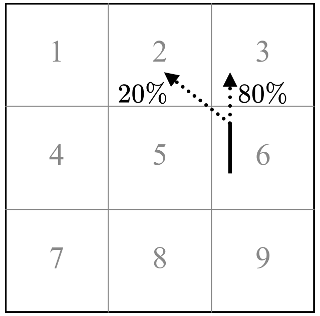
in 6.390, this expectation is only w.r.t. the transition probabilities $\mathrm{T}\left(s, a, s'\right)$

- $\mathrm{V}_h^\pi(s)$: expected sum of discounted rewards starting in state s and follow π for h steps
- Value is long-term; reward is immediate (one-time)
- Horizon-0 values $\mathrm{V}_0^\pi(s)$ defined as 0 for all states

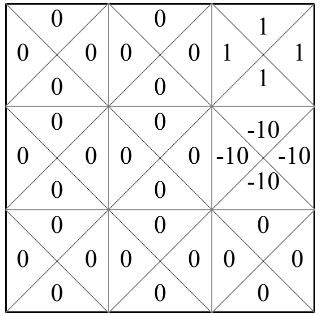
evaluate $V_h^{\pi}(s)$ under the "always-up" policy

states and

one special transition:



rewards



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

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

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

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

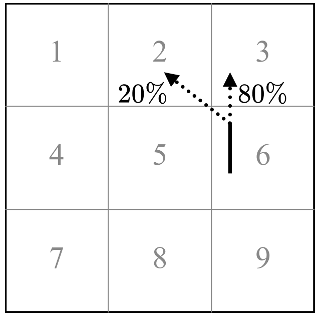
$V_1^{\uparrow}(s) = R(s, \uparrow)$

0	0	1
0	0	-10
0	0	0

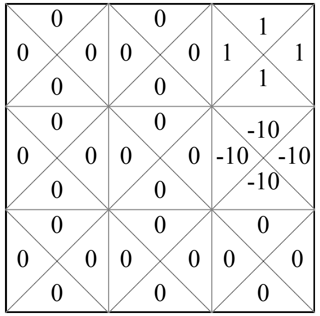
horizon $h = 2$

states and

one special transition:



rewards



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

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

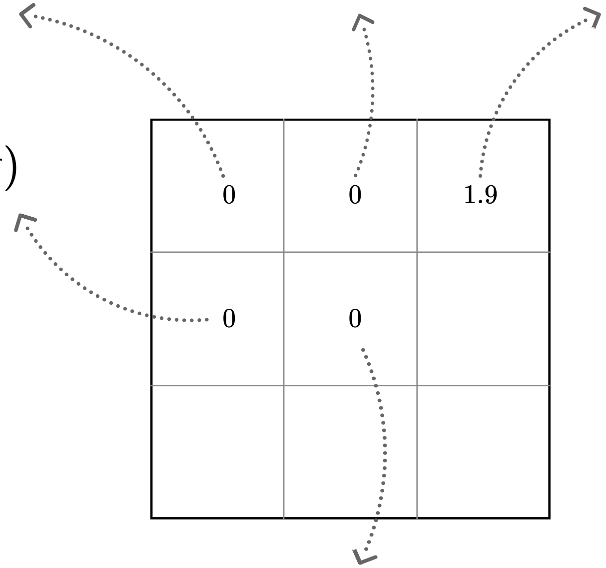
$R(1, \uparrow) + \gamma R(1, \uparrow)$

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

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

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

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

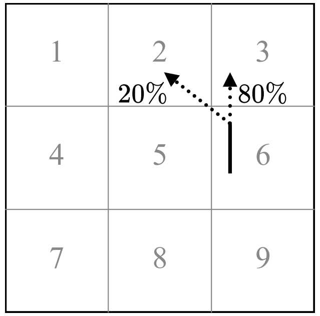


$R(5, \uparrow) + \gamma R(2, \uparrow)$

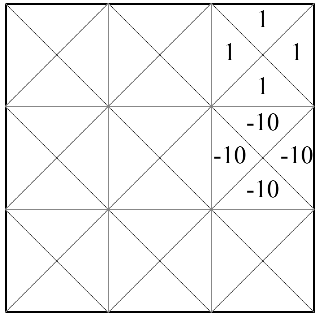
horizon $h = 2$

states and

one special transition:

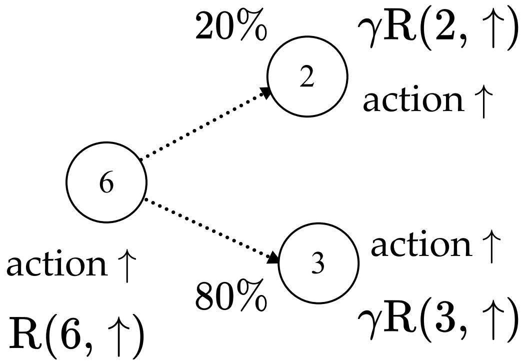


rewards



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

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



0	0	1.9
0	0	-9.28
0	0	-9

$R(6, \uparrow) + \gamma[.2R(2, \uparrow) + .8R(3, \uparrow)]$

$= -10 + 0.9 * (0.2 * 0 + 0.8 * 1)$

$\Rightarrow -9.28$

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

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

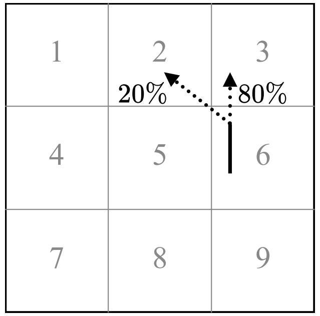
$R(9, \uparrow) + \gamma R(6, \uparrow)$

$= 0 + 0.9 * (-10) \Rightarrow -9$

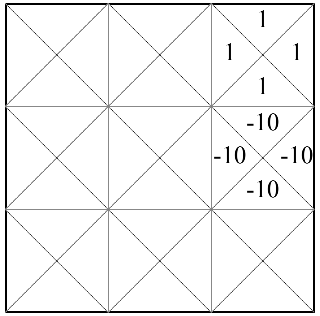
horizon $h = 3$

states and

one special transition:

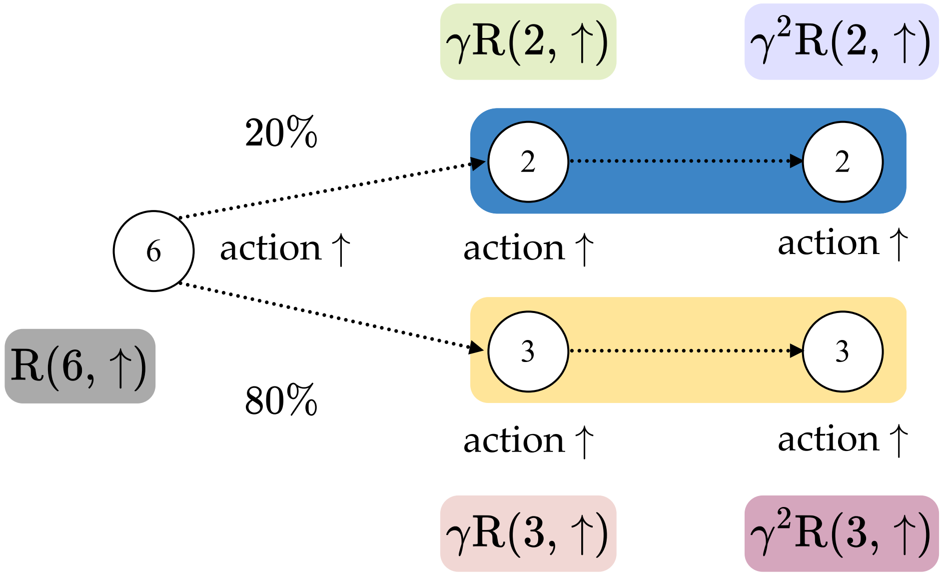


rewards



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

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



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

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

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

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

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'

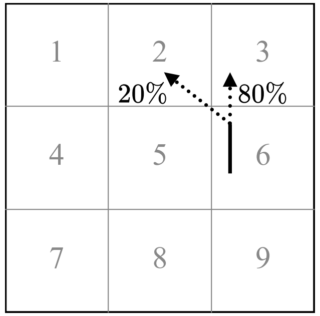
discounted by γ

Bellman Recursion (finite horizon h)

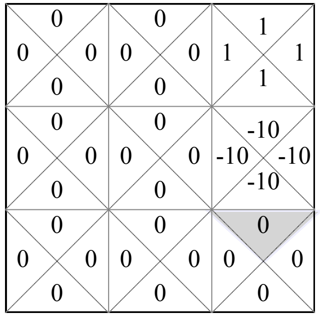
$$V_h^{\pi}(s) = R\left(s, \pi(s)\right) + \gamma \sum_{s'} T\left(s, \pi(s), s'\right) V_{h-1}^{\pi}\left(s'\right)$$

states and

one special transition:



rewards



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

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

0.00	0.00	1.00
0.00	0.00	-10.00
0.00	0.00	0.00

$V_2^{\uparrow}(s)$

0.00	0.00	1.90
0.00	0.00	-9.28
0.00	0.00	-9.00

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

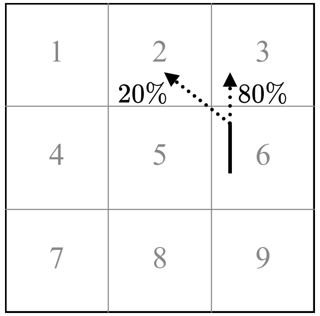
$$= 0 + 0.9 \times [-10]$$

$$= -9$$

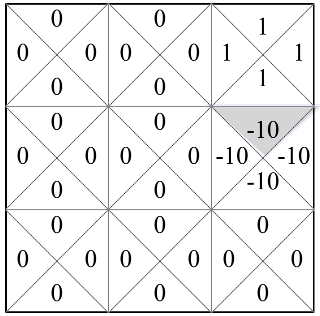
Bellman Recursion (finite horizon h) $\quad \mathrm{V}_h^{\pi}(s) = \mathrm{R}\left(s, \pi(s)\right) + \gamma \sum_{s'} \mathrm{T}\left(s, \pi(s), s'\right) \mathrm{V}_{h-1}^{\pi}\left(s'\right)$

states and

one special transition:



rewards



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

$\mathrm{V}_1^{\uparrow}(s) = \mathrm{R}(s, \uparrow)$

0.00	0.00	1.00
0.00	0.00	-10.00
0.00	0.00	0.00

$\mathrm{V}_2^{\uparrow}(s)$

0.00	0.00	1.90
0.00	0.00	-9.28
0.00	0.00	-9.00

$\mathrm{V}_3^{\uparrow}(s)$

0.00	0.00	2.71
0.00	0.00	-8.63
0.00	0.00	-8.35

$$\mathrm{V}_3^{\uparrow}(6) = \mathrm{R}(6, \uparrow) + \gamma[.2 \times \mathrm{V}_2^{\uparrow}(2) + .8 \times \mathrm{V}_2^{\uparrow}(3)]$$

$$= -10 + .9[.2 \times 0 + 0.8 \times 1.9]$$

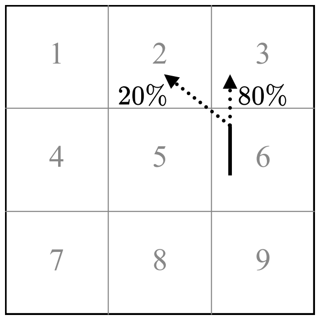
$$= -8.632$$

Bellman Recursion (finite horizon h)

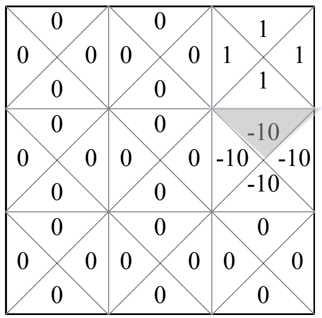
$$\mathrm{V}_h^\pi(s) = \mathrm{R}\left(s, \pi(s)\right) + \gamma \sum_{s'} \mathrm{T}\left(s, \pi(s), s'\right) \mathrm{V}_{h-1}^\pi\left(s'\right)$$

states and

one special transition:



rewards



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

$\mathrm{V}_4^{\uparrow}(s)$

0.00	0.00	3.44
0.00	0.00	-8.05
0.00	0.00	-7.77

$\mathrm{V}_5^{\uparrow}(s)$

0.00	0.00	4.10
0.00	0.00	-7.52
0.00	0.00	-7.24

$\mathrm{V}_6^{\uparrow}(s)$

0.00	0.00	4.69
0.00	0.00	-7.05
0.00	0.00	-6.77

$\cdots$

$$\mathrm{V}_6^{\uparrow}(6) = \mathrm{R}(6, \uparrow) + \gamma[.2 \times \mathrm{V}_5^{\uparrow}(2) + .8 \times \mathrm{V}_5^{\uparrow}(3)]$$

$$= -10 + .9[.2 \times 0 + 0.8 \times 4.10]$$

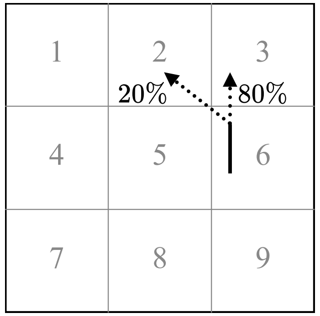
$$= -7.048$$

Bellman Recursion (finite horizon h)

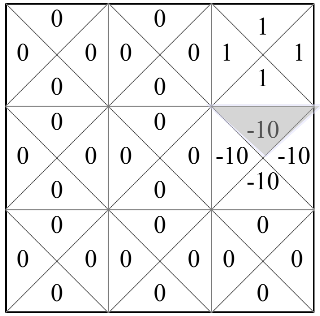
$$V_h^{\pi}(s) = R(s, \pi(s)) + \gamma \sum_{s'} T(s, \pi(s), s') V_{h-1}^{\pi}(s')$$

states and

one special transition:



rewards



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

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

0.00	0.00	9.98
0.00	0.00	-2.82
0.00	0.00	-2.54

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

0.00	0.00	9.98
0.00	0.00	-2.81
0.00	0.00	-2.53

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

0.00	0.00	9.98
0.00	0.00	-2.81
0.00	0.00	-2.53

$\cdots$

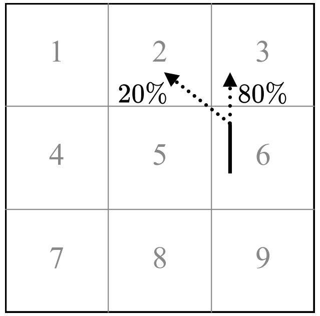
$$V_{61}^{\uparrow}(6) = R(6, \uparrow) + \gamma[.2 \times V_{60}^{\uparrow}(2) + .8 \times V_{60}^{\uparrow}(3)]$$

$$= -10 + .9[.2 \times 0 + 0.8 \times 9.98]$$

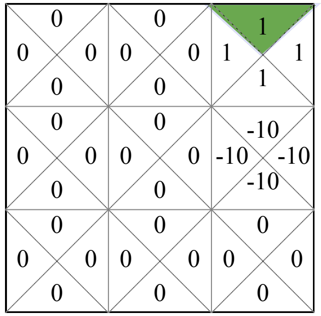
$$= -2.8144$$

Value functions converge as $h \to \infty$

states and
one special transition:



rewards



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

$V^{\uparrow}_{60}(s)$

0.00	0.00	9.98
0.00	0.00	-2.81
0.00	0.00	-2.53

$V^{\uparrow}_{61}(s)$

0.00	0.00	9.98
0.00	0.00	-2.81
0.00	0.00	-2.53

$\cdots$

$V^{\uparrow}_{\infty}(s)$

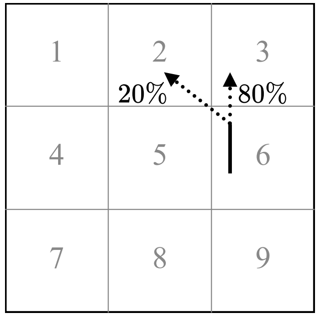
0.00	0.00	10.00
0.00	0.00	-2.80
0.00	0.00	-2.52

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

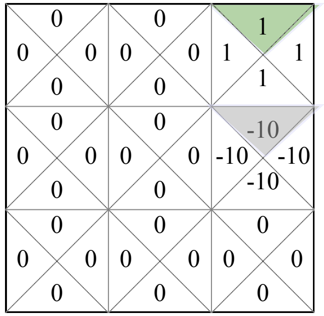
Typically, $\gamma < 1$ to ensure V_∞ is finite.

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

states and
one special transition:



rewards



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

Recursion (finite h) 2️⃣ $\mathrm{V}_h^\pi(s) = \mathrm{R}\left(s, \pi(s)\right) + \gamma \sum_{s'} \mathrm{T}\left(s, \pi(s), s'\right) \mathrm{V}_{h-1}^\pi\left(s'\right)$

Equation ($h \to \infty$) 3️⃣ $\mathrm{V}_\infty^\pi(s) = \mathrm{R}(s, \pi(s)) + \gamma \sum_{s'} \mathrm{T}\left(s, \pi(s), s'\right) \mathrm{V}_\infty^\pi\left(s'\right)$

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

$\mathrm{V}_\infty^{\uparrow}(s)$

0.00	0.00	10.00
0.00	0.00	-2.80
0.00	0.00	-2.52

$$\mathrm{V}_\infty^{\uparrow}(3) = \mathrm{R}(3, \uparrow) + \gamma[\mathrm{V}_\infty^{\uparrow}(3)]$$
$$= 1 + .9 \times 10 \Rightarrow 10$$

$$\mathrm{V}_\infty^{\uparrow}(6) = \mathrm{R}(6, \uparrow) + \gamma[.2 \times \mathrm{V}_\infty^{\uparrow}(2) + .8 \times \mathrm{V}_\infty^{\uparrow}(3)]$$
$$= -10 + .9[.2 \times 0 + 0.8 \times 10] \Rightarrow -2.8$$

```
Quick summary
```

$\pi(s) \longrightarrow$ Policy Evaluation $\longrightarrow \mathrm{V}_h^{\pi}(s)$

Use the definition and sum up expected rewards:

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

Or, leverage the recursive structure:

2️⃣ **finite-**horizon Bellman recursions

$$\mathrm{V}_h^{\pi}(s) = \mathrm{R}(s, \pi(s)) + \gamma \sum_{s'} \mathrm{T}\left(s, \pi(s), s'\right) \mathrm{V}_{h-1}^{\pi}\left(s'\right)$$

3️⃣ **infinite-**horizon Bellman equations

$$\mathrm{V}_{\infty}^{\pi}(s) = \mathrm{R}(s, \pi(s)) + \gamma \sum_{s'} \mathrm{T}\left(s, \pi(s), s'\right) \mathrm{V}_{\infty}^{\pi}\left(s'\right)$$

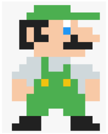
Outline

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

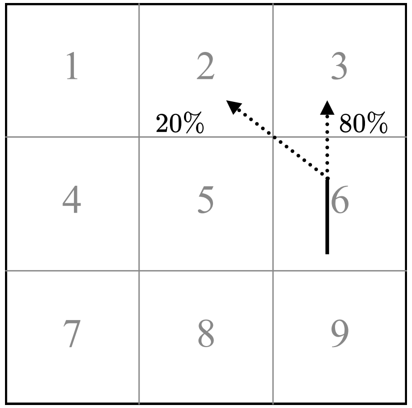
Optimal policy π^*

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

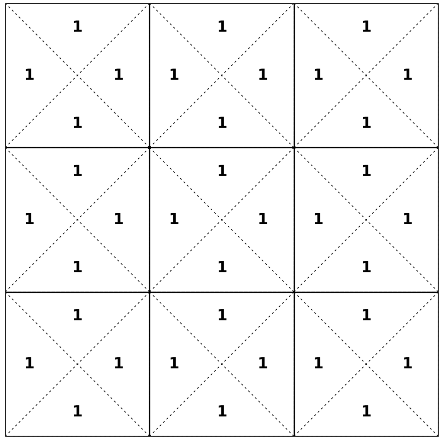
e.g. in the "Luigi game", all rewards are 1,



States and one special transition:



Rewards:



$\gamma = 0.9$

then any policy is an optimal policy

Optimal policy π^*

- Formally: an optimal policy π^* is such that: $\mathrm{V}_h^{\pi^*}(s) = \max_\pi \mathrm{V}_h^\pi(s) = \mathrm{V}_h^*(s), \forall s \in \mathcal{S}$
- How to search for an optimal policy π^*?
- Even if we tediously enumerate over all π, do policy evaluation, compare values to get $\mathrm{V}_h^*(s)$...it's not yet clear how to choose actions.

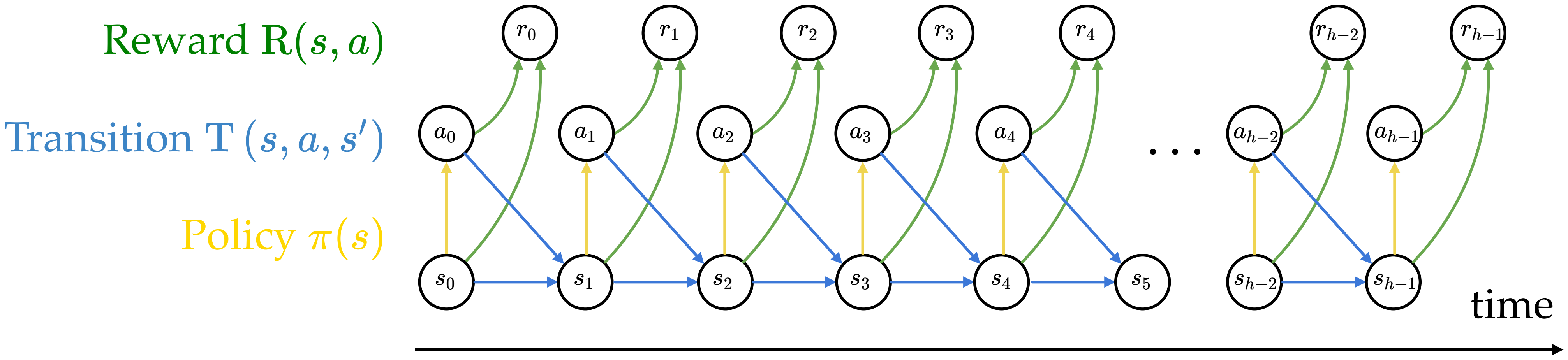
$\mathrm{V}^*(s)$ is defined over states, not actions.

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

Optimality recursion

if we've acted optimally for h steps: $\mathrm{V}^*_h(s)$

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



Bellman recursion under an optimal policy

4️⃣ $$\mathrm{V}^*_h(s) = \max_a \left[\mathrm{R}(s, a) + \gamma \textstyle\sum_{s'} \mathrm{T}(s, a, s')\mathrm{V}^*_{h-1}(s')\right]$$

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

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

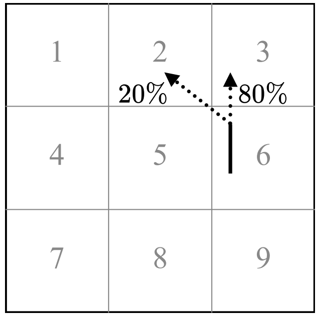
Q^* satisfies the Bellman recursion:

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

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

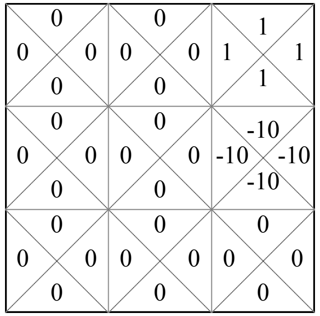
- starting in state s,
- take action a, for one step
- act **optimally** thereafter for the remaining $(h-1)$ steps

States and one special transition:

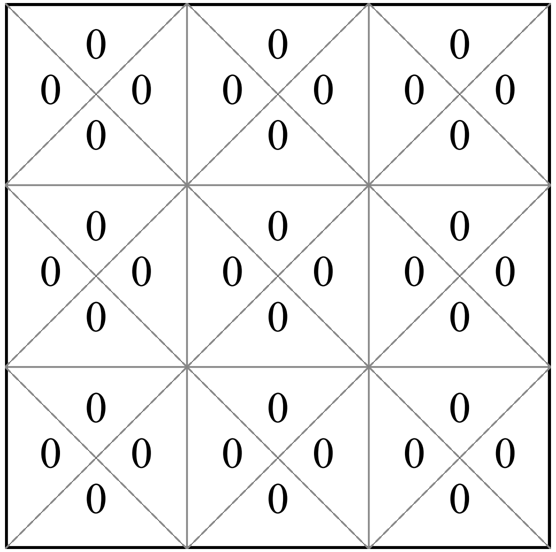


$\gamma = 0.9$

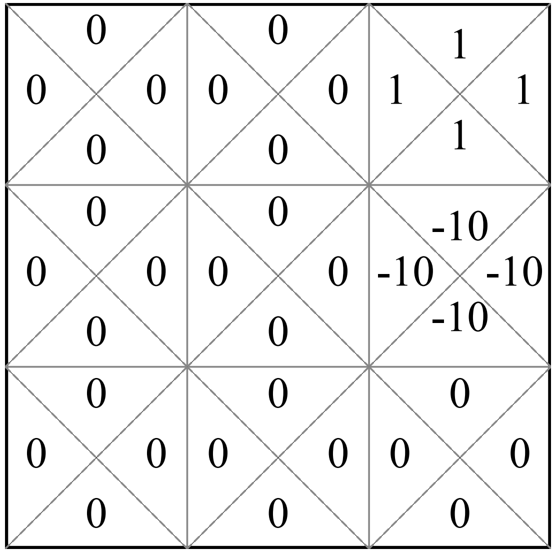
Rewards:



$Q^*_0(s,a)$



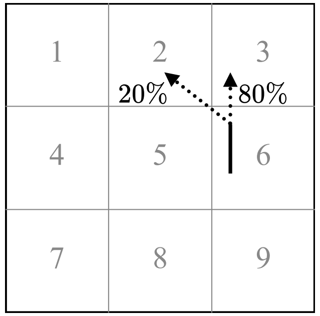
$Q^*_1(s,a) = R(s,a)$



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

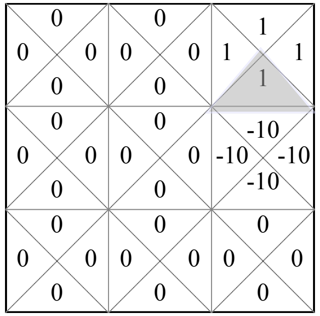
- starting in state s,
- take action a, for one step
- act **optimally** thereafter for the remaining $(h-1)$ steps

States and one special transition:

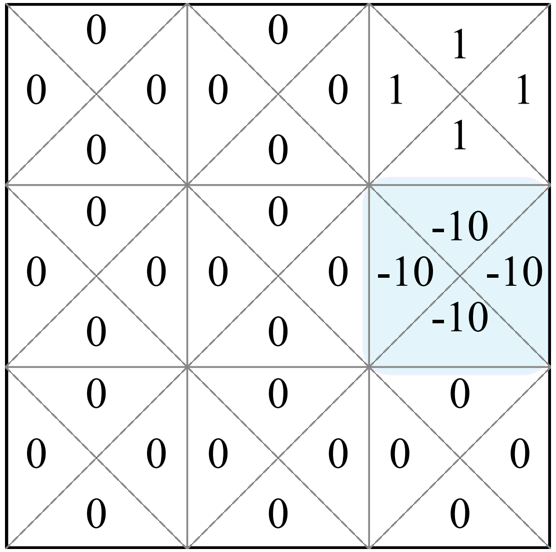


$\gamma = 0.9$

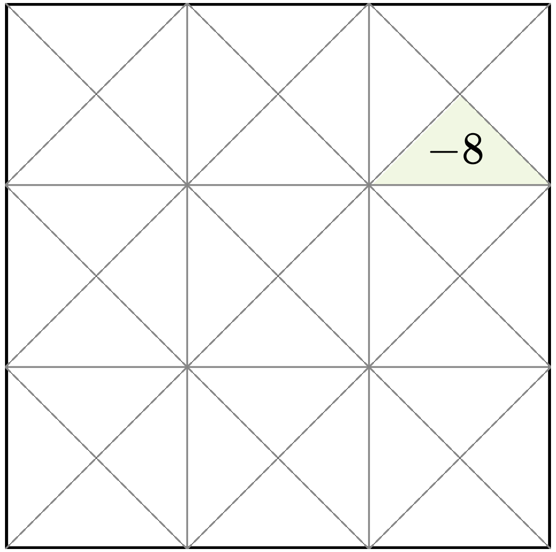
Rewards:



$Q^*_1(s,a)$



$Q^*_2(s,a)$



Consider $Q^*_2(3,\downarrow)$

- receive $R(3,\downarrow)$
- next state $s' = 6$, act **optimally** for the remaining one timestep
  - receive $\max_{a'} Q^*_1(6,a')$

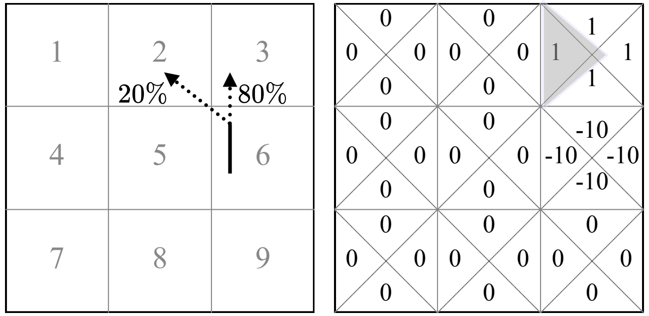
$$Q^*_2(3,\downarrow) = R(3,\downarrow) + \gamma \max_{a'} Q^*_1(6,a')$$
$$= 1 + .9 \times 10$$
$$= -8$$

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

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

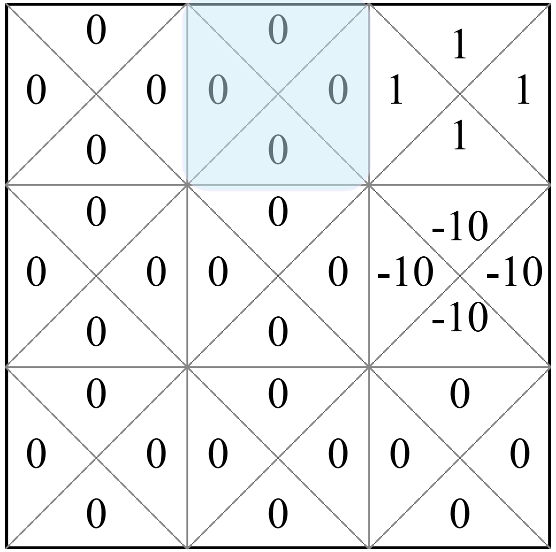
States and one special transition:

Rewards:

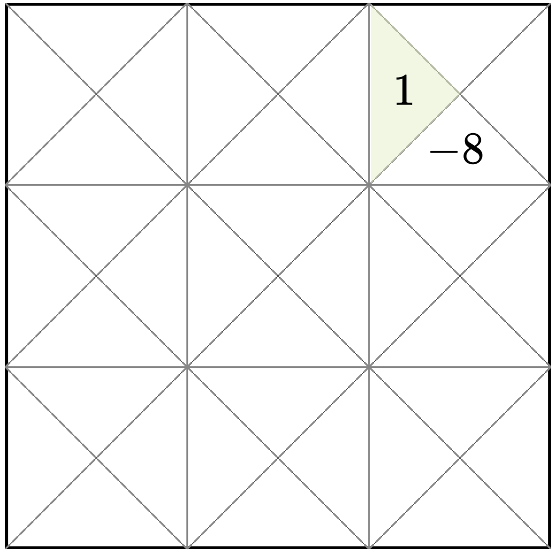


$\gamma = 0.9$

$Q_1^*(s, a)$



$Q_2^*(s, a)$



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

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

$$Q_2^*(3, \leftarrow) = R(3, \leftarrow) + \gamma \max_{a'} Q_1^*(2, a')$$

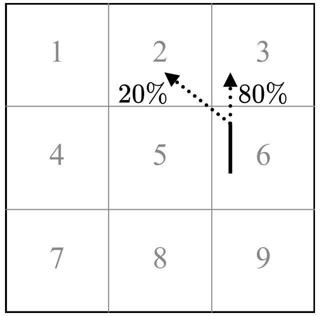
$$= 1 + .9 \times 0$$

$$= 1$$

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

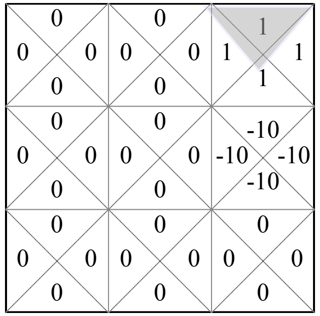
- starting in state s,
- take action a, for one step
- act **optimally** thereafter for the remaining $(h-1)$ steps

States and one special transition:

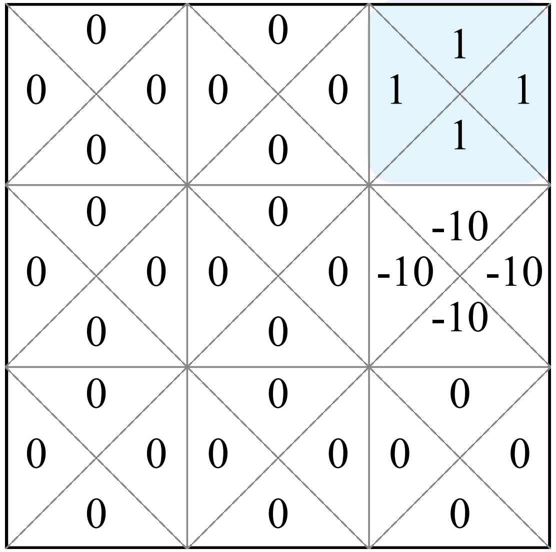


$\gamma = 0.9$

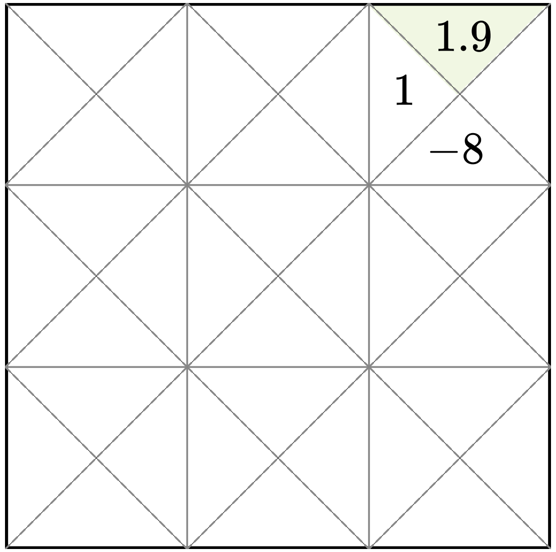
Rewards:



$Q^*_1(s, a)$



$Q^*_2(s, a)$



Let's consider $Q^*_2(3, \uparrow)$

- receive $R(3, \uparrow)$
- next state $s' = 3$, act **optimally** for the remaining one timestep
  - receive $\max_{a'} Q^*_1(3, a')$

$$Q^*_2(3, \uparrow) = R(3, \uparrow) + \gamma \max_{a'} Q^*_1(3, a')$$

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

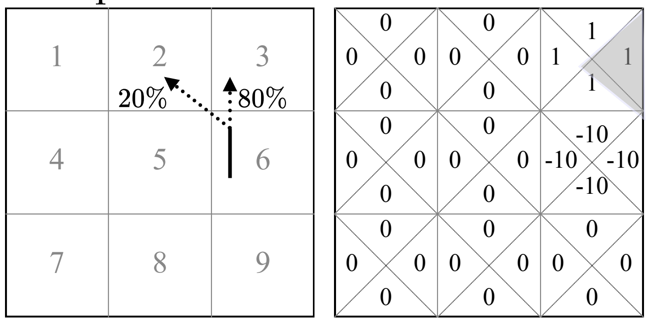
$$= 1.9$$

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

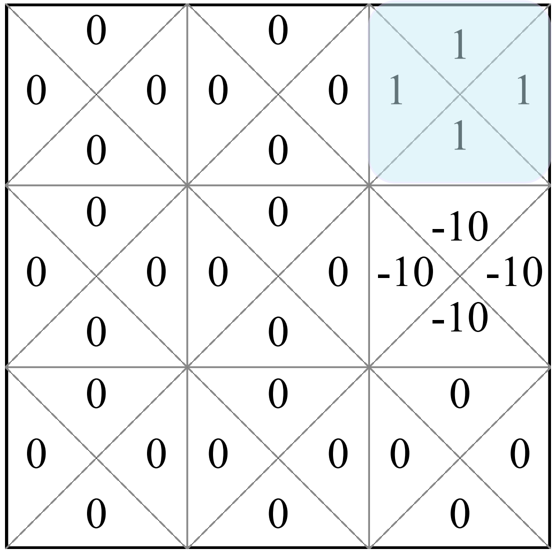
- starting in state s,
- take action a, for one step
- act **optimally** thereafter for the remaining $(h-1)$ steps

States and one special transition:

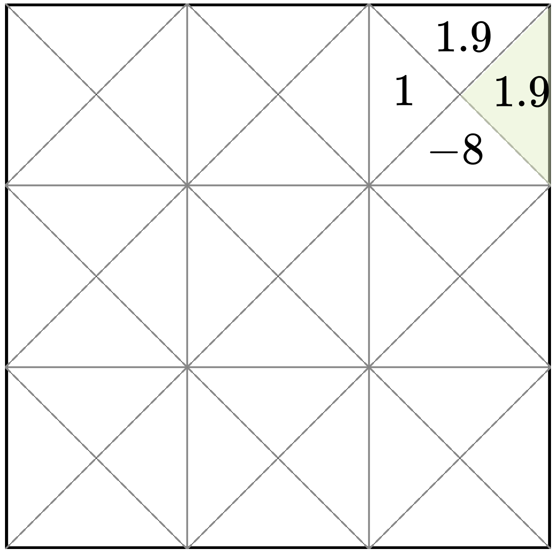
$\gamma = 0.9$



$Q^*_1(s,a)$



$Q^*_2(s,a)$



Let's consider $Q^*_2(3, \rightarrow)$

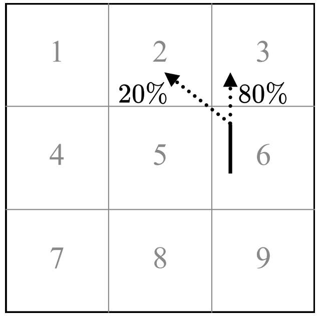
- receive $R(3, \rightarrow)$
- next state $s' = 3$, act **optimally** for the remaining one timestep
  - receive $\max_{a'} Q^*_1(3, a')$

$$Q^*_2(3, \rightarrow) = R(3, \rightarrow) + \gamma \max_{a'} Q^*_1(3, a')$$
$$= 1 + .9 \times 1$$
$$= 1.9$$

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

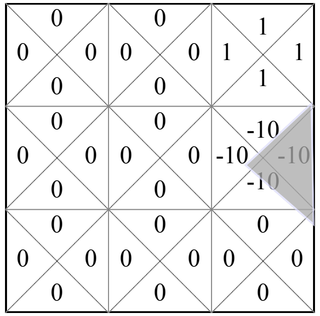
- starting in state s,
- take action a, for one step
- act **optimally** thereafter for the remaining $(h-1)$ steps

States and one special transition:

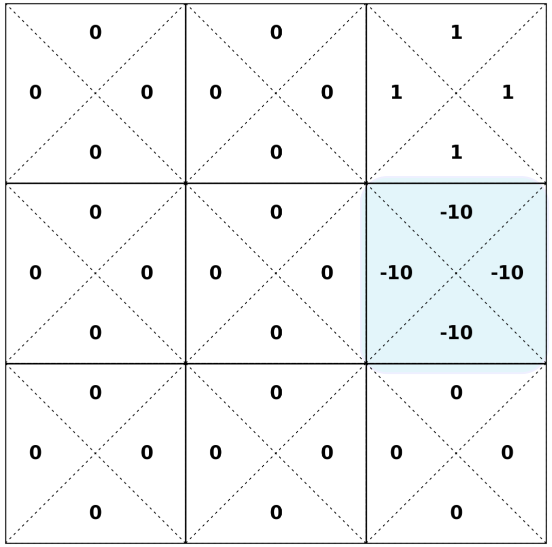


$\gamma = 0.9$

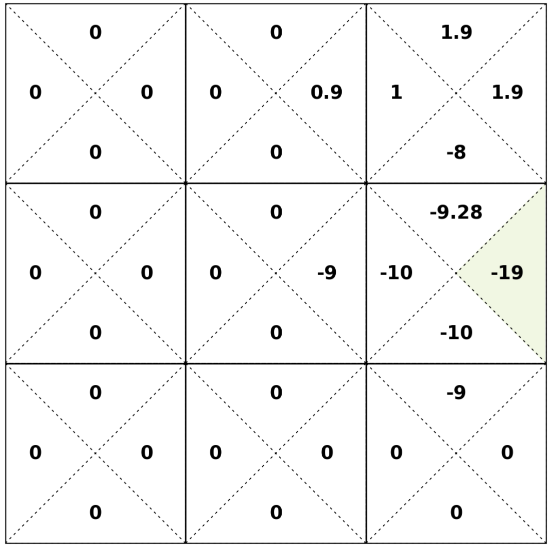
Rewards:



$Q^*_1(s,a)$



$Q^*_2(s,a)$



Let's consider $Q^*_2(6,\rightarrow)$

- receive $R(6,\rightarrow)$
- act optimally at the next state $s' = 6$

  receive $\max_{a'} Q^*_1(6,a')$

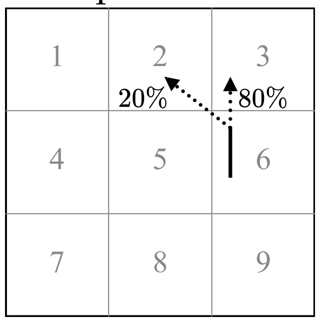
$$Q^*_2(6,\rightarrow) = R(6,\rightarrow) + \gamma[\max_{a'} Q^*_1(6,a')]$$

$$= -10 + .9 \times -10 \Rightarrow -19$$

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

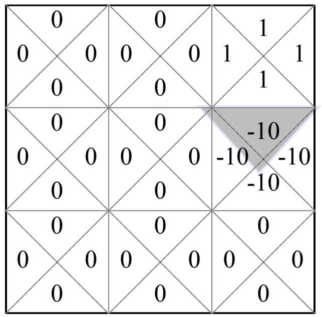
- starting in state s,
- take action a, for one step
- act **optimally** thereafter for the remaining $(h-1)$ steps

States and one special transition:

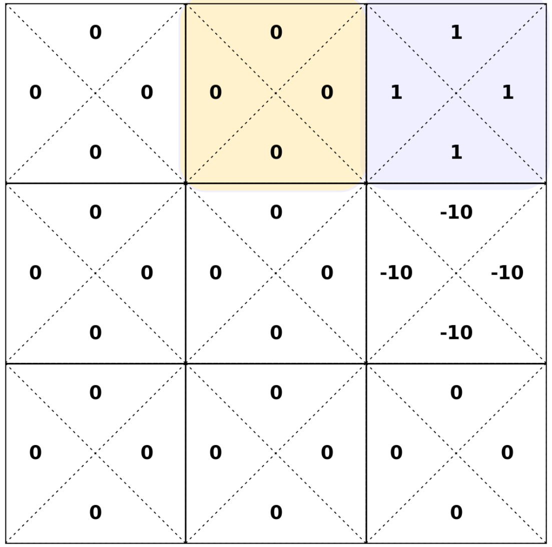


$\gamma = 0.9$

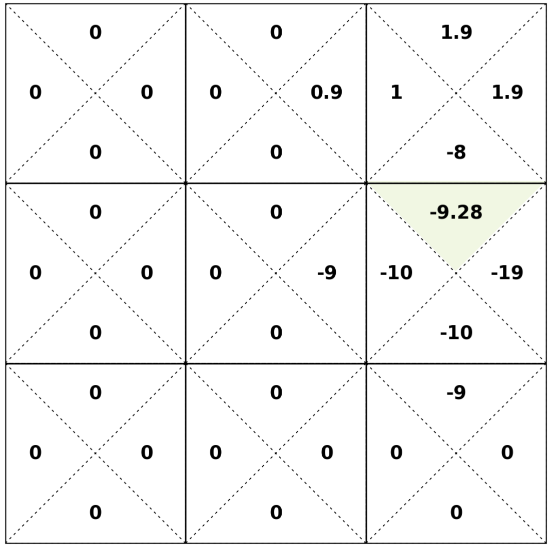
Rewards:



$Q^*_1(s, a)$



$Q^*_2(s, a)$



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 $\max_{a'} Q^*_1(3, a')$

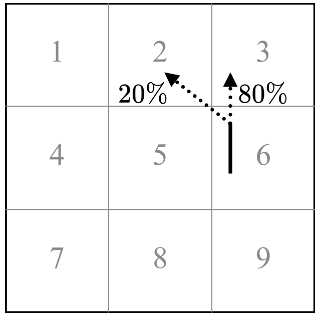
$$Q^*_2(6, \uparrow) = R(6, \uparrow) + \gamma[.2 \max_{a'} Q^*_1(2, a') + .8 \max_{a'} Q^*_1(3, a')]$$

$$= -10 + .9[.2 \times 0 + .8 \times 1] \Rightarrow -9.28$$

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

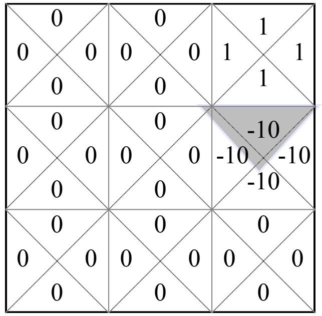
- starting in state s,
- take action a, for one step
- act **optimally** thereafter for the remaining $(h-1)$ steps

States and one special transition:

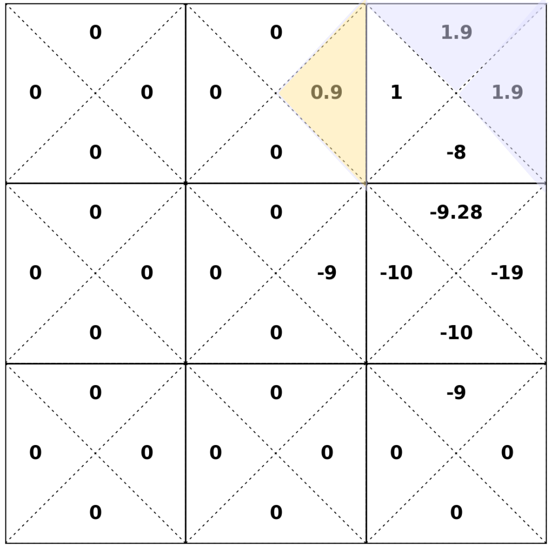


$\gamma = 0.9$

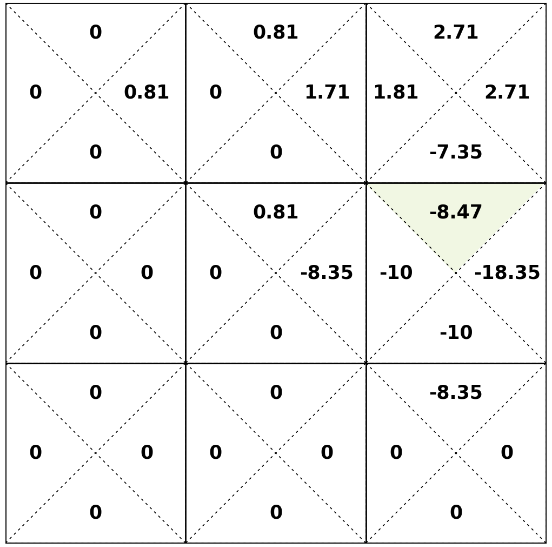
Rewards:



$Q^*_2(s, a)$



$Q^*_3(s, a)$



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 $\max_{a'} Q^*_2(3, a')$

$$Q^*_3(6, \uparrow) = R(6, \uparrow) + \gamma[.2 \max_{a'} Q^*_2(2, a') + .8 \max_{a'} Q^*_2(3, a')]$$

$$= -10 + .9[.2 \times 0.9 + .8 \times 1.9] \Rightarrow -8.47$$

Value iteration: what we just did, iteratively invoke 5️⃣

$$\mathrm{Q}^*_h(s,a) = \mathrm{R}(s,a) + \gamma \sum_{s'} \mathrm{T}(s,a,s') \max_{a'} \mathrm{Q}^*_{h-1}(s',a')$$

`Value Iteration`

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

$\quad \mathrm{Q}_{\text{old}}(\mathrm{s}, \mathrm{a}) = 0$

while True:

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

$\quad\quad \mathrm{Q}_{\text{new}}(s,a) \leftarrow \mathrm{R}(s,a) + \gamma \sum_{s'} \mathrm{T}(s,a,s') \max_{a'} \mathrm{Q}_{\text{old}}(s',a')$

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

$\quad\quad$ **return** Q_{new}

$\quad \mathrm{Q}_{\text{old}} \leftarrow \mathrm{Q}_{\text{new}}$

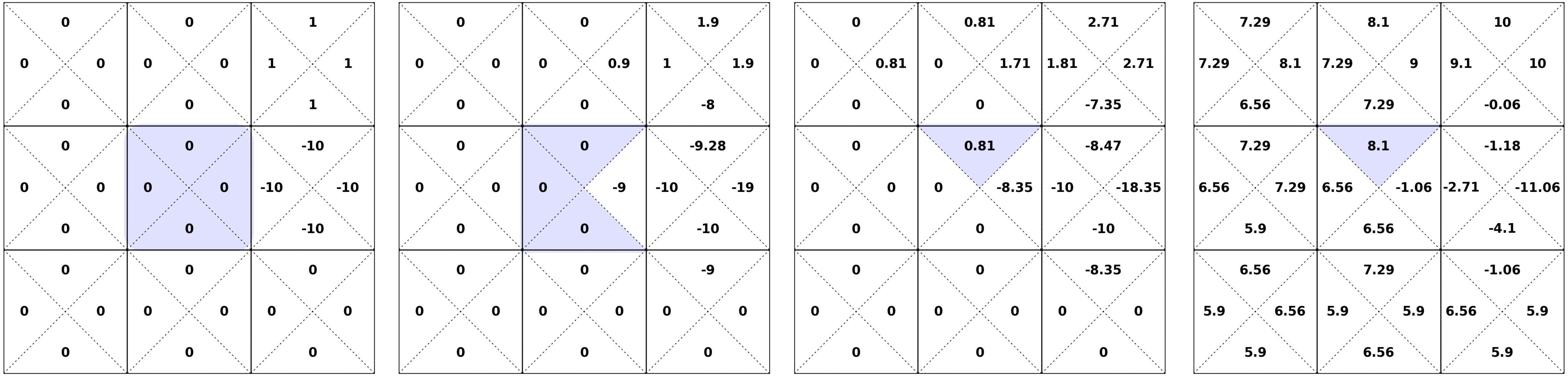
if run this block h times and break, then the returns are exactly Q^*_h

$\mathrm{Q}^*_\infty(s,a)$

Optimal policy easily extracted: 6️⃣ $\pi_h^*(s) = \arg\max_a \mathrm{Q}_h^*(s,a)$

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

$\mathrm{Q}_1^*(s,a)$ $\mathrm{Q}_2^*(s,a)$ $\mathrm{Q}_3^*(s,a)$ $\dots$ $\mathrm{Q}_\infty^*(s,a)$



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

Summary

- Markov decision processes (MDP) are a nice mathematical framework for making sequential decisions. It's the foundation to reinforcement learning.
- An MDP is defined by a five-tuple, and the goal is to find an optimal policy that leads to high expected cumulative discounted rewards.
- To evaluate how good a *given* policy π is, we can calculate $V^{\pi}(s)$ via
  - the summation-over-rewards definition
  - Bellman recursion for finite horizon and Bellman equation for infinite horizon
- To *find* an optimal policy, we can recursively find $Q^{*}(s, a)$ via the value iteration algorithm, and then act greedily w.r.t. the Q^{*} values.

https://forms.gle/6snt5oZgS1N9nZY78

We'd love to hear your thoughts.

Thanks!