https://introml.mit.edu/

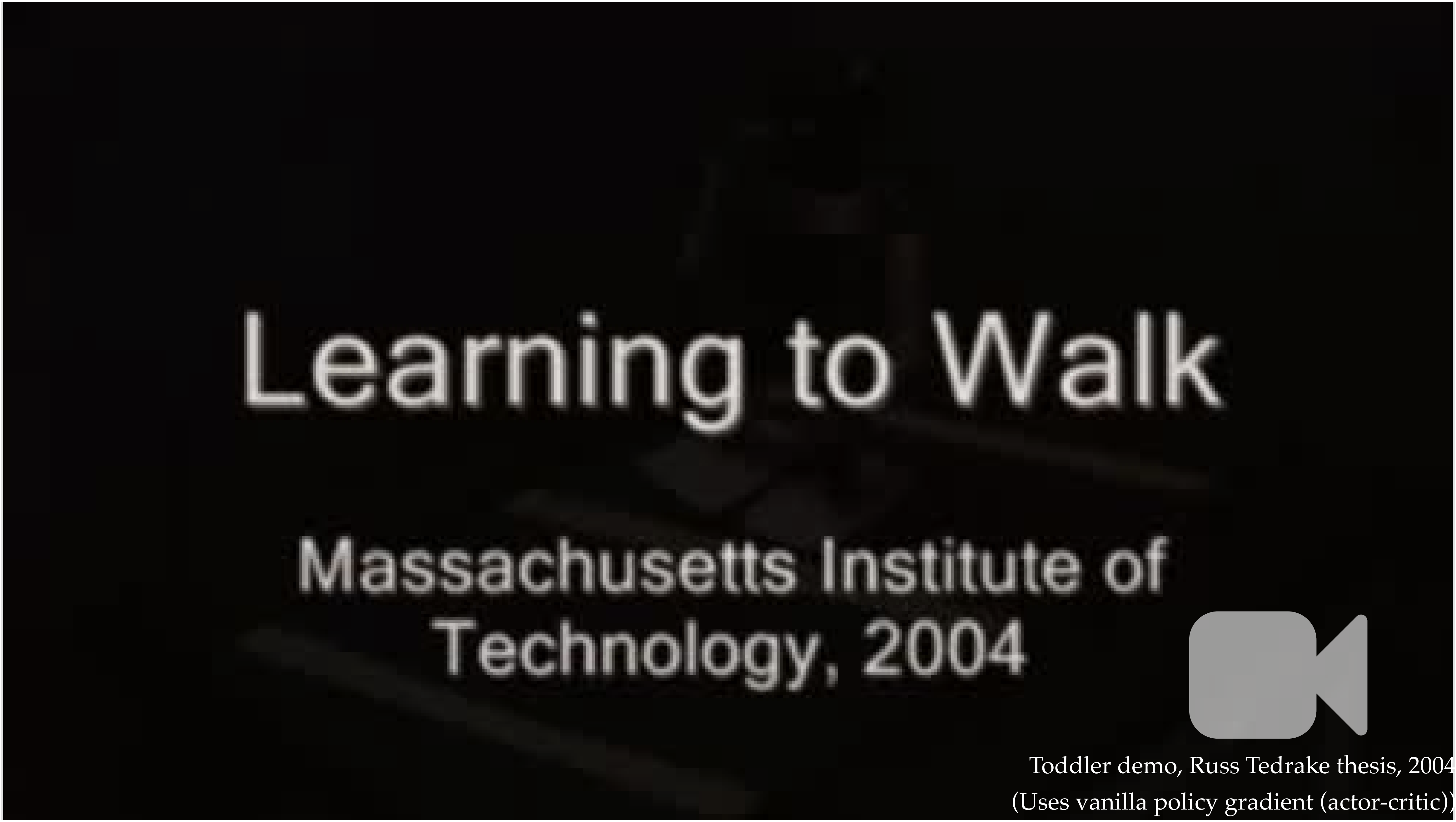
6.390 Intro to Machine Learning

Lecture 10: Markov Decision Processes

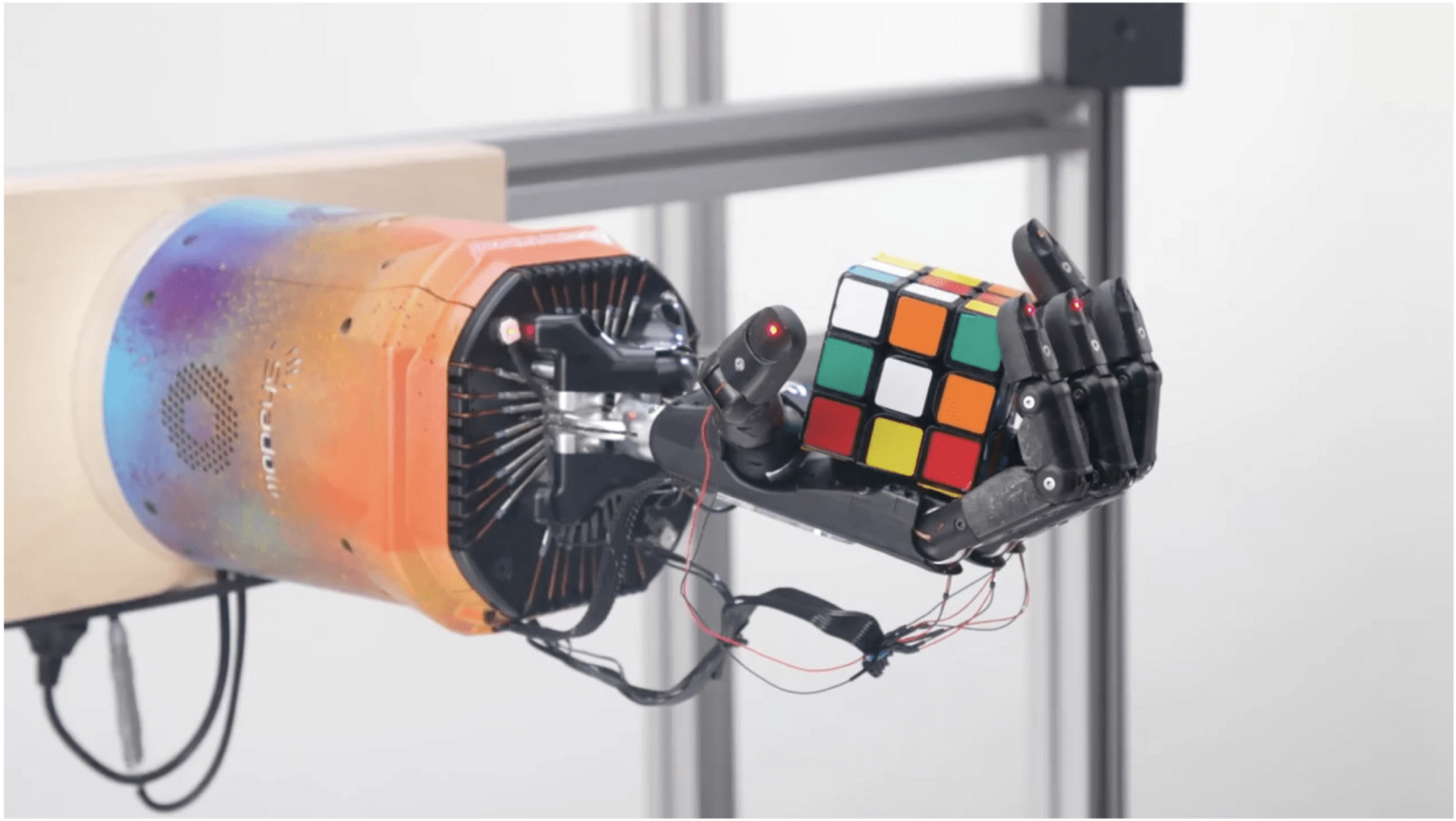
Shen Shen

April 18, 2025

11am, Room 10-250



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



[Hu … 15]

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

Discovering faster matrix multiplication algorithms with reinforcement learning

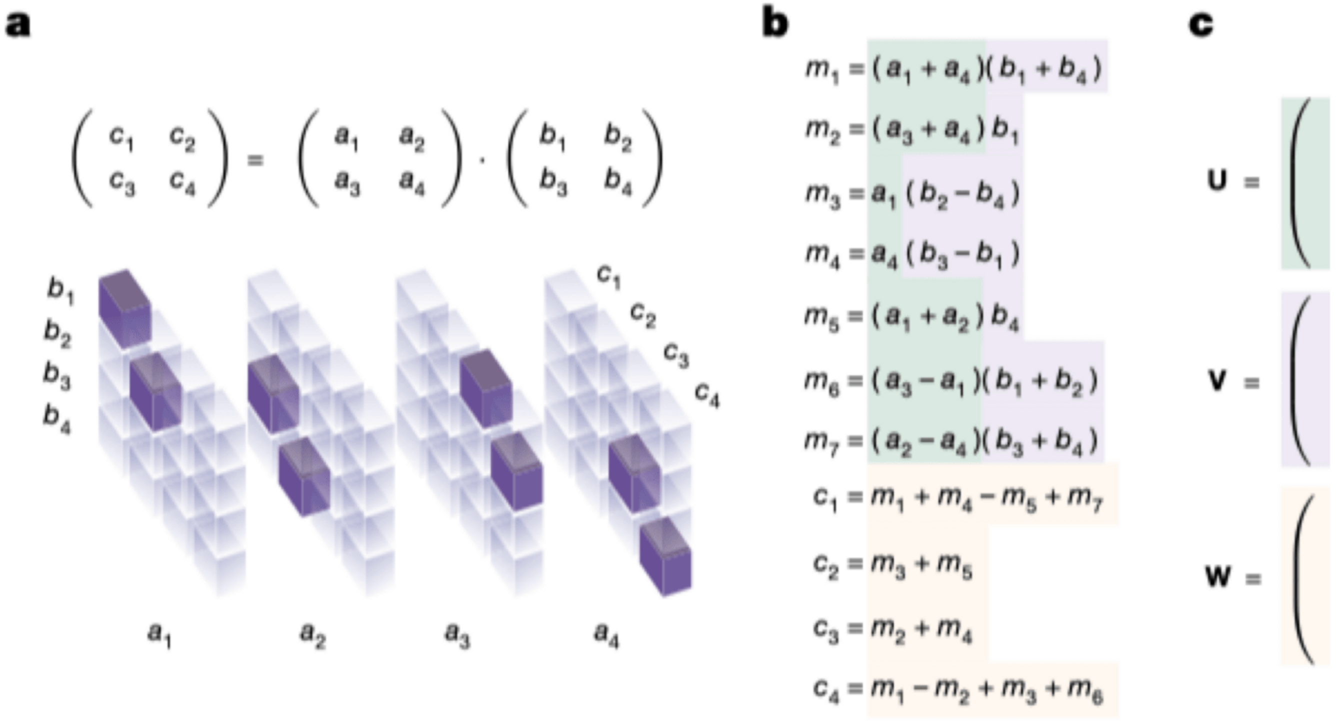
https://doi.org/10.1038/s41586-022-05172-4

Received: 2 October 2021

Accepted: 2 August 2022

Published online: 5 October 2022

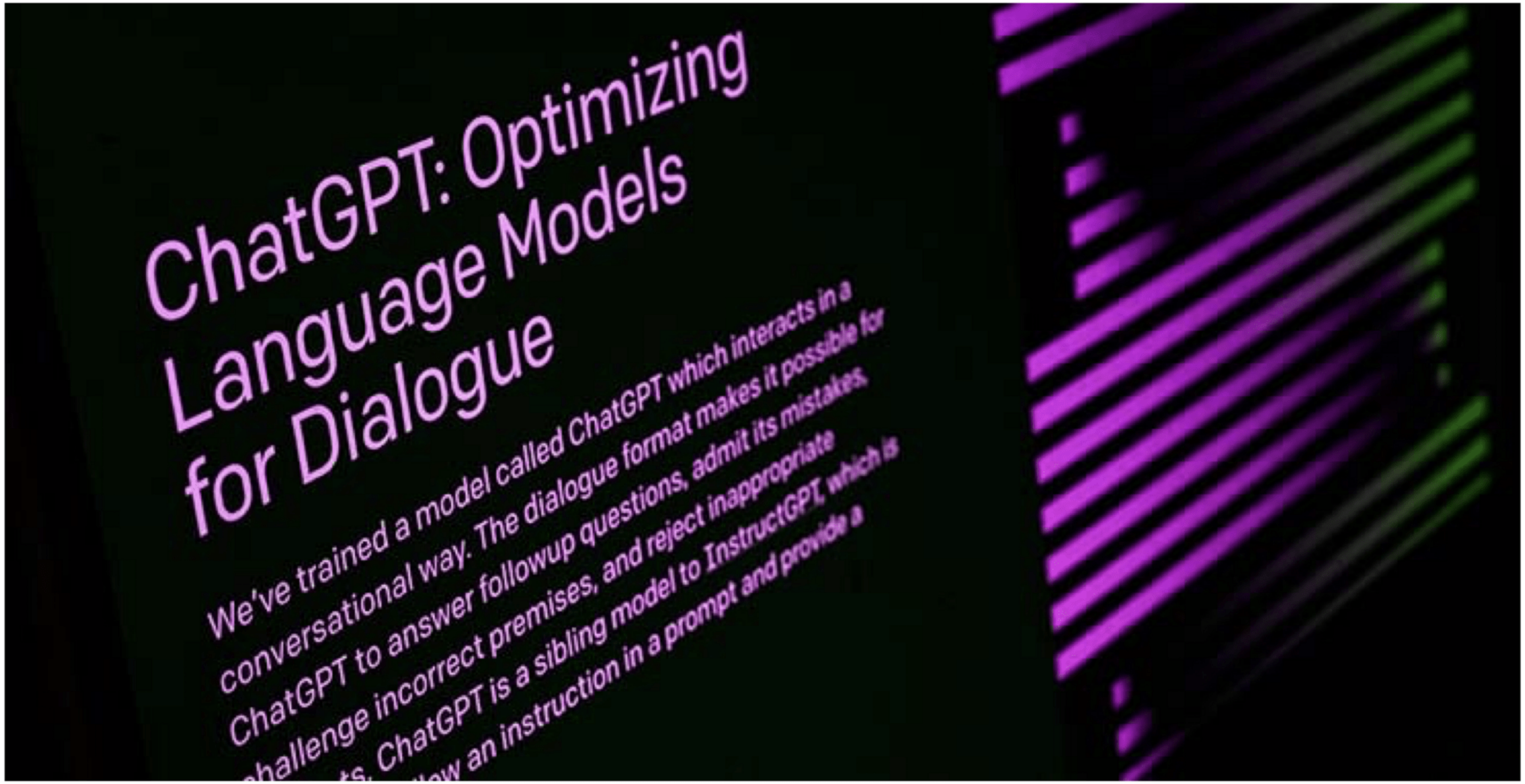
Alhussein Fawzi[1,2✉], Matej Balog[1,2], Aja Huang[1,2], Thomas Hubert[1,2], Bernardino Romera-Paredes[1,2], Mohammadamin Barekatain[1], Alexande
Francisco J. R. Ruiz[1], Julian Schrittwieser[1], Grzegorz Swirszcz[1], David Si
& Pushmeet Kohli[1]



X

Size (n, m, p)	Best method known	Best rank known	AlphaTensor rank Modular	Standard
(2, 2, 2)	(Strassen, 1969)[2]	7	7	7
(3, 3, 3)	(Laderman, 1976)[15]	23	23	23
(4, 4, 4)	(Strassen, 1969)[2] (2, 2, 2) ⊗ (2, 2, 2)	49	47	49
(5, 5, 5)	(3, 5, 5) + (2, 5, 5)	98	96	98
(2, 2, 3)	(2, 2, 2) + (2, 2, 1)	11	11	11
(2, 2, 4)	(2, 2, 2) + (2, 2, 2)	14	14	14
(2, 2, 5)	(2, 2, 2) + (2, 2, 3)	18	18	18
(2, 3, 3)	(Hopcroft and Kerr, 1971)[16]	15	15	15
(2, 3, 4)	(Hopcroft and Kerr, 1971)[16]	20	20	20
(2, 3, 5)	(Hopcroft and Kerr, 1971)[16]	25	25	25
(2, 4, 4)	(Hopcroft and Kerr, 1971)[16]	26	26	26
(2, 4, 5)	(Hopcroft and Kerr, 1971)[16]	33	33	33
(2, 5, 5)	(Hopcroft and Kerr, 1971)[16]	40	40	40
(3, 3, 4)	(Smirnov, 2013)[18]	29	29	29
(3, 3, 5)	(Smirnov, 2013)[18]	36	36	36
(3, 4, 4)	(Smirnov, 2013)[18]	38	38	38
(3, 4, 5)	(Smirnov, 2013)[18]	48	47	47
(3, 5, 5)	(Sedoglavic and Smirnov, 2021)[19]	58	58	58
(4, 4, 5)	(4, 4, 2) + (4, 4, 3)	64	63	63
(4, 5, 5)	(2, 5, 5) ⊗ (2, 1, 1)	80	76	76

Reinforcement Learning with Human Feedback



[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, terminologies, and policy
- Policy Evaluation
  - State Value Functions V^{π}
  - Bellman recursions and Bellman equations
- Policy Optimization
  - Optimal policies π^*
  - Optimal action value functions: Q^*
  - Value iteration

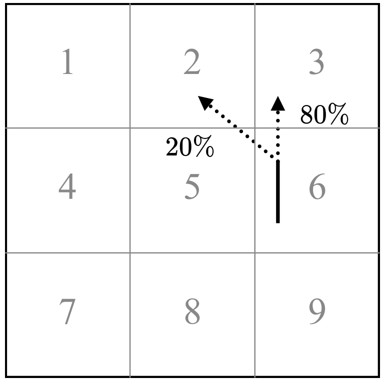
Outline

- Markov Decision Processes Definition, terminologies, and policy
- 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

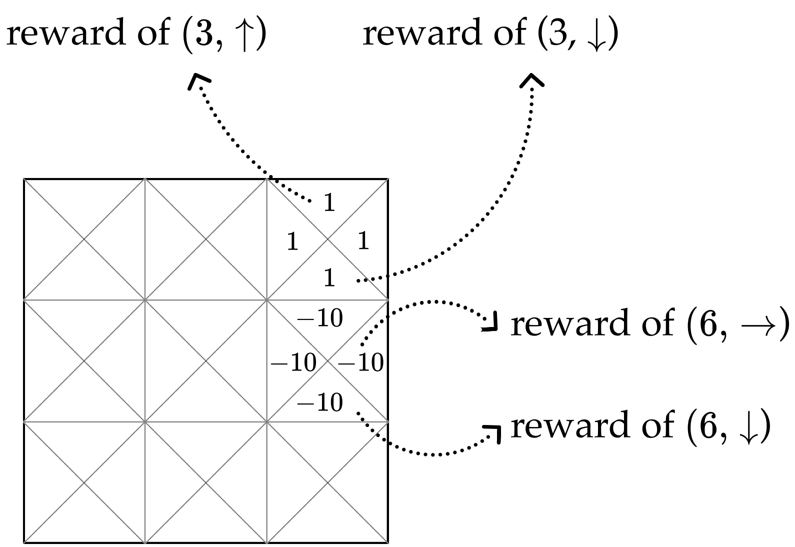
- Research area initiated in the 50s by Bellman, known under various names:
  - Stochastic optimal control (Control theory)
  - Stochastic shortest path (Operations research)
  - Sequential decision making under uncertainty (Economics)
  - Reinforcement learning (Artificial intelligence, Machine learning)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

Running example: Mario in a grid-world



- 9 possible **states *s***
- 4 possible **actions *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 that reduces the "worth" of rewards, depending on the timing Mario gets the rewards.
  - e.g., say this factor is 0.9. then, for $(3, \leftarrow)$ pair, Mario gets a reward of 1 at the start of the game; at the 2nd time step, a discounted reward of 0.9; at the 3rd time step, it is further discounted to $(0.9)^2$, and so on.

Markov Decision Processes - Definition and terminologies

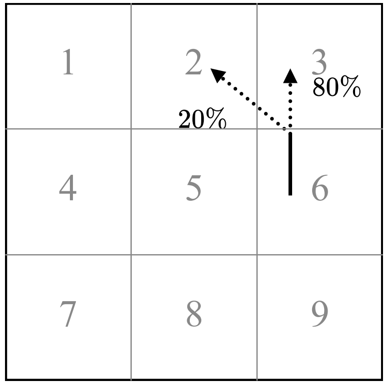
- $\mathcal{S}$: state space, contains all possible states s.
- $\mathcal{A}$: action space, contains all possible actions a.

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.
- $\mathrm{T}\left(s, a, s'\right)$: 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,

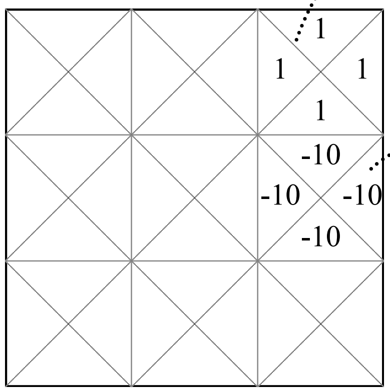
- $\mathcal{S}$ and $\mathcal{A}$ are small discrete sets, unless otherwise specified.
- s' and a' are short-hand for the next-timestep

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

Markov Decision Processes - Definition and terminologies

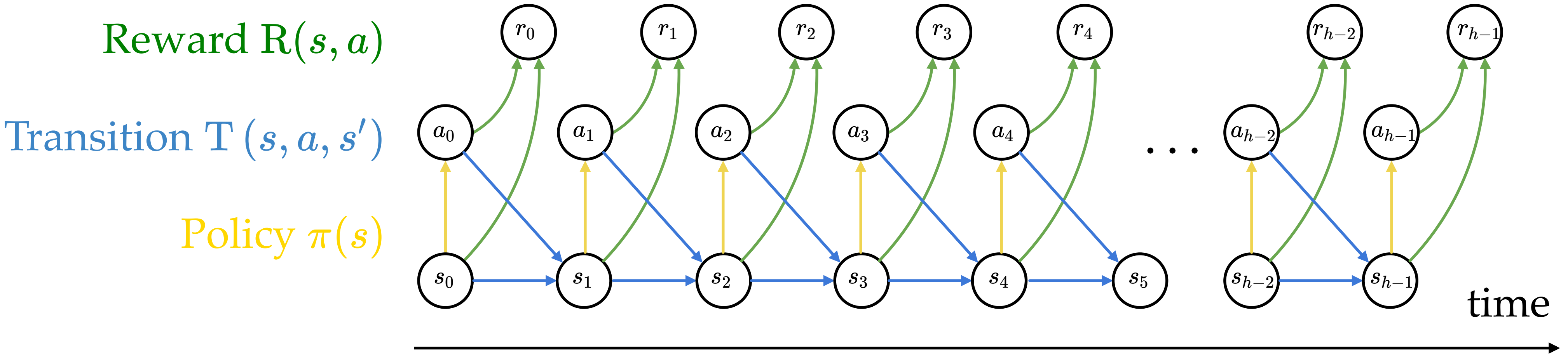
- $\mathcal{S}$: state space, contains all possible states s.
- $\mathcal{A}$: action space, contains all possible actions a.
- $\mathrm{T}\left(s, a, s'\right)$: the probability of transition from state s to s' when action a is taken.
- $\mathrm{R}(s, a)$: reward, takes in a (state, action) pair and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



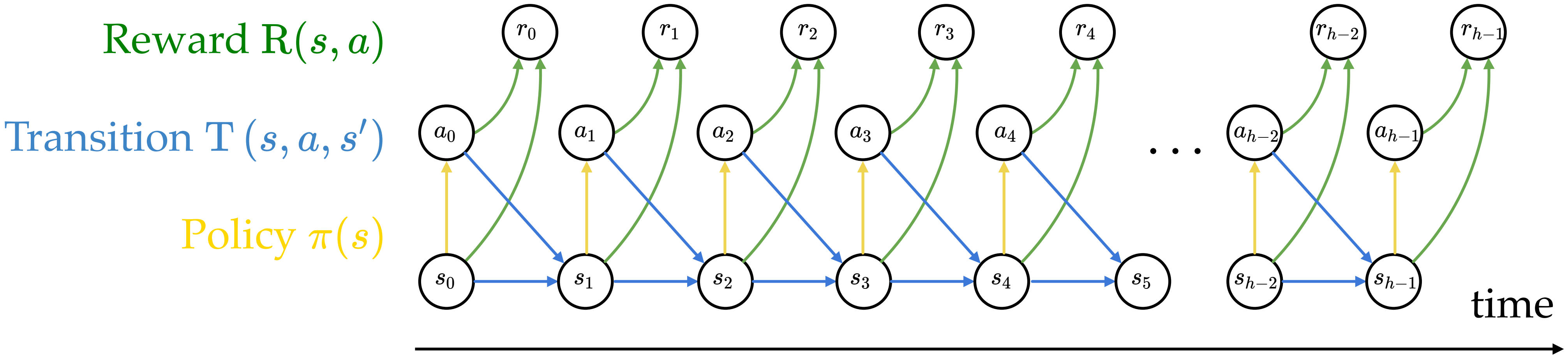
a trajectory (aka, an experience, or a rollout), of horizon h

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

initial state

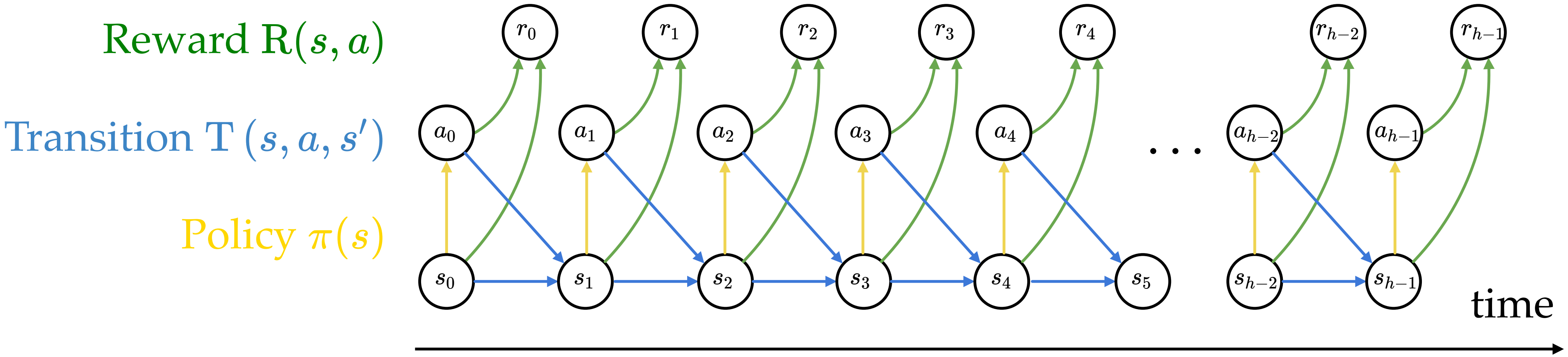
all depends on π

- $a_t = \pi(s_t)$
- $r_t = \mathrm{R}(s_t, a_t)$
- $\Pr(s_t = s' \mid s_{t-1} = s, a_{t-1} = a) = \mathrm{T}(s, a, s')$



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

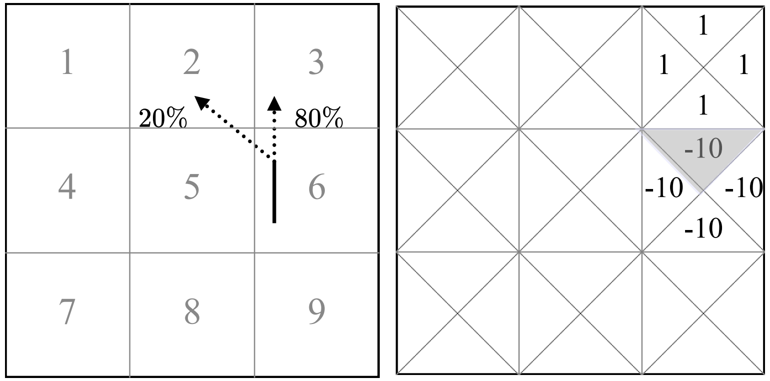
One idea: $\mathrm{R}(s_0, a_0) + \gamma \mathrm{R}(s_1, a_1) + \gamma^2 \mathrm{R}(s_2, a_2) + \gamma^3 \mathrm{R}(s_3, a_3) + \dots + \gamma^{h-1} \mathrm{R}(s_{h-1}, a_{h-1})$



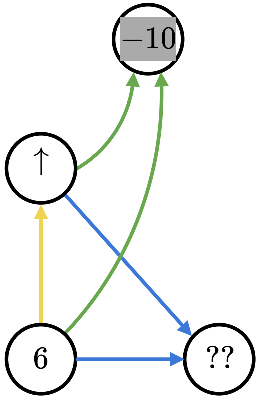
Starting in a given s_0, how "good" is it to follow a policy π for h time steps?

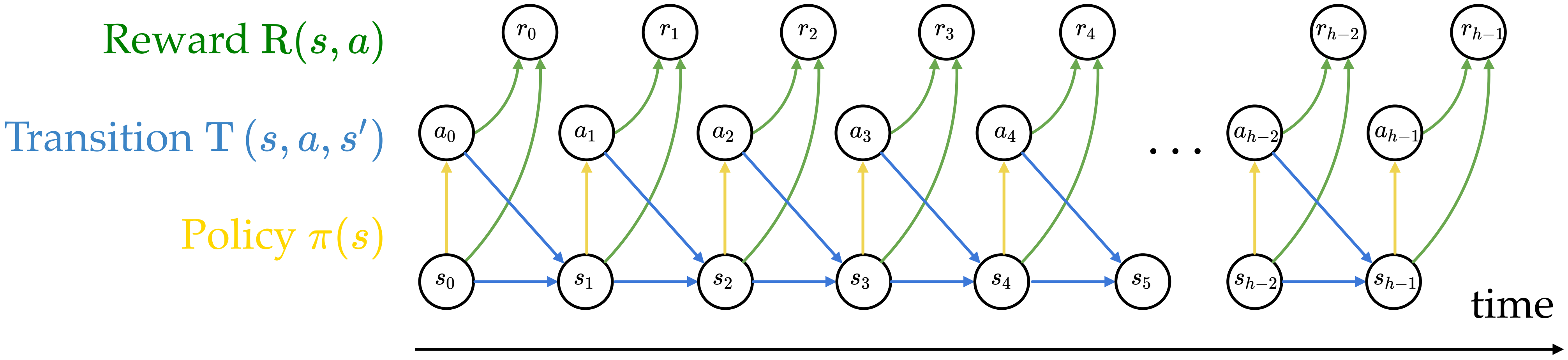
One idea: $\mathrm{R}(s_0, a_0) + \gamma \mathrm{R}(s_1, a_1) + \gamma^2 \mathrm{R}(s_2, a_2) + \gamma^3 \mathrm{R}(s_3, a_3) + \dots + \gamma^{h-1} \mathrm{R}(s_{h-1}, a_{h-1})$

But in Mario game:



if start at $s_0 = 6$ and policy $\pi(s) = \uparrow, \forall s$, i.e., always up





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

$$\mathbb{E}\Big[\underbrace{\mathrm{R}(s_0, a_0) + \gamma\mathrm{R}(s_1, a_1) + \gamma^2\mathrm{R}(s_2, a_2) + \gamma^3\mathrm{R}(s_3, a_3) + \dots + \gamma^{h-1}\mathrm{R}(s_{h-1}, a_{h-1})}_{h \text{ terms}}\Big]$$

in 390, this expectation is only w.r.t. the transition probabilities $\mathrm{T}(s, a, s')$

Outline

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

$$\mathbb{E}\Big[\mathrm{R}(s_0, a_0) \;+\; \gamma \mathrm{R}(s_1, a_1) \;+\; \gamma^2 \mathrm{R}(s_2, a_2) \;+\; \gamma^3 \mathrm{R}(s_3, a_3) \;+\; \ldots \;+\; \gamma^{h-1} \mathrm{R}(s_{h-1}, a_{h-1})\Big]$$

Definition: For a *given* policy $\pi(s)$, the state **value functions**

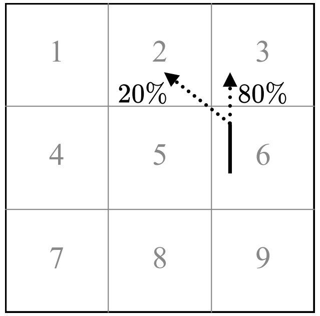
$$\mathrm{V}_h^\pi(s) := \mathbb{E}\left[\sum_{t=0}^{h-1} \gamma^t \mathrm{R}\left(s_t, \pi\left(s_t\right)\right) \mid s_0 = s, \pi\right], \forall s, h$$

- value functions $\mathrm{V}_h^\pi(s)$: the expected sum of discounted rewards, starting in state s, and follow policy π for h steps.
- horizon-0 values defined as 0.
- value is long-term, reward is short-term (one-time).

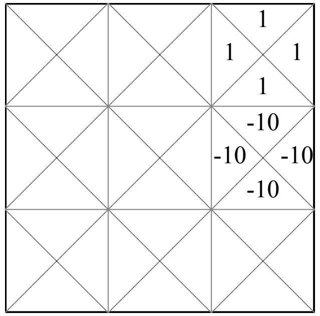
evaluate the "$\pi(s) = \uparrow$, for all s, i.e. the always $\uparrow$" policy

states and

one special transition:



rewards



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

expanded form $\mathbb{E}\left[\underbrace{R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \ldots}_{h \text{ terms}}\right]$

horizon $h = 0$: no step left

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

0	0	0
0	0	0
0	0	0

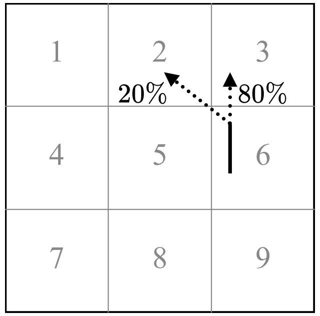
horizon $h = 1$: receive the rewards

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

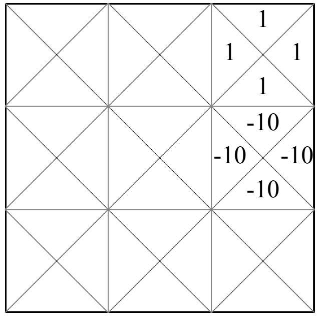
0	0	1
0	0	-10
0	0	0

horizon $h = 2$:

states and one special transition:



rewards



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

$$V_2^{\uparrow}(s) = \mathbb{E}\Big[\underbrace{R(s_0, a_0) + \gamma R(s_1, a_1)}_{\text{2 terms inside}}\Big]$$

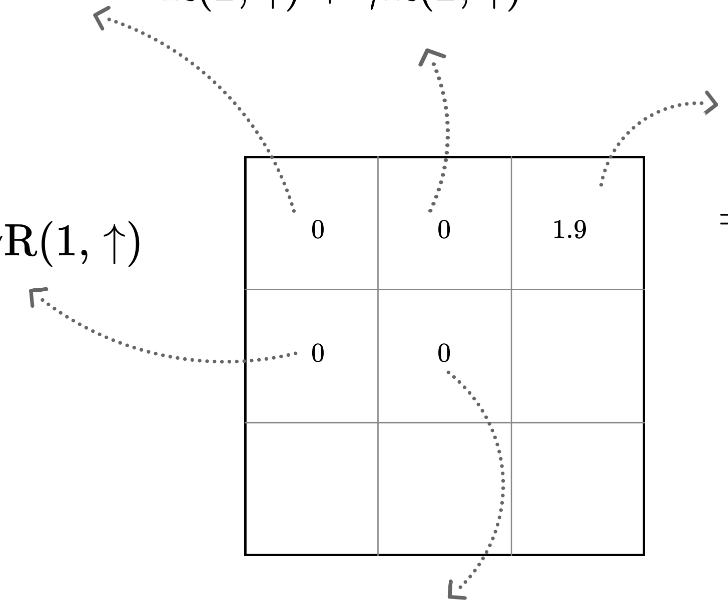
$R(1, \uparrow) + \gamma R(1, \uparrow)$

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

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

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

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

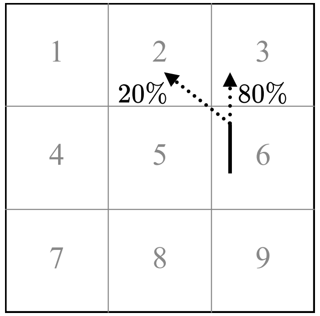


$R(5, \uparrow) + \gamma R(2, \uparrow)$

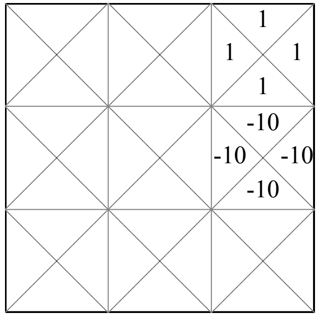
horizon $h = 2$:

states and

one special transition:



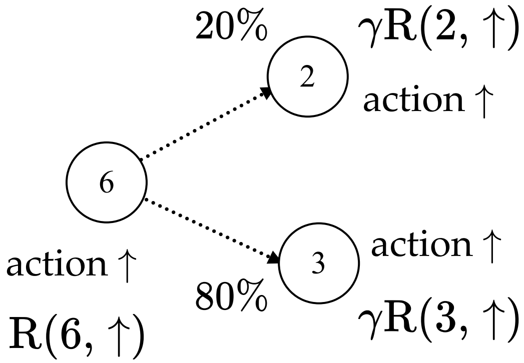
rewards



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

$$V_2^{\uparrow}(s) = \mathbb{E}\left[\, R(s_0, a_0) \;+\; \gamma R(s_1, a_1) \,\right]$$

2 terms inside



0	0	1.9
0	0	−9.28
0	0	−9

$$R(6, \uparrow) + \gamma[.2R(2, \uparrow) + .8R(3, \uparrow)]$$
$$= -10 + 0.9 * (0.2 * 0 + 0.8 * 1)$$
$$= -9.28$$

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

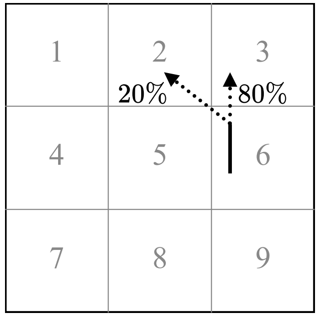
$R(8, \uparrow) + \gamma R(5, \uparrow)$

$R(9, \uparrow) + \gamma R(6, \uparrow) = 0 + 0.9 * (-10)$

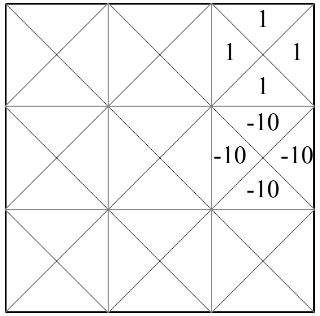
horizon $h = 3$: $\mathbb{E}\Big[\, \mathrm{R}(s_0, a_0) + \gamma \mathrm{R}(s_1, a_1) + \gamma^2 \mathrm{R}(s_2, a_2) \,\Big]$

states and

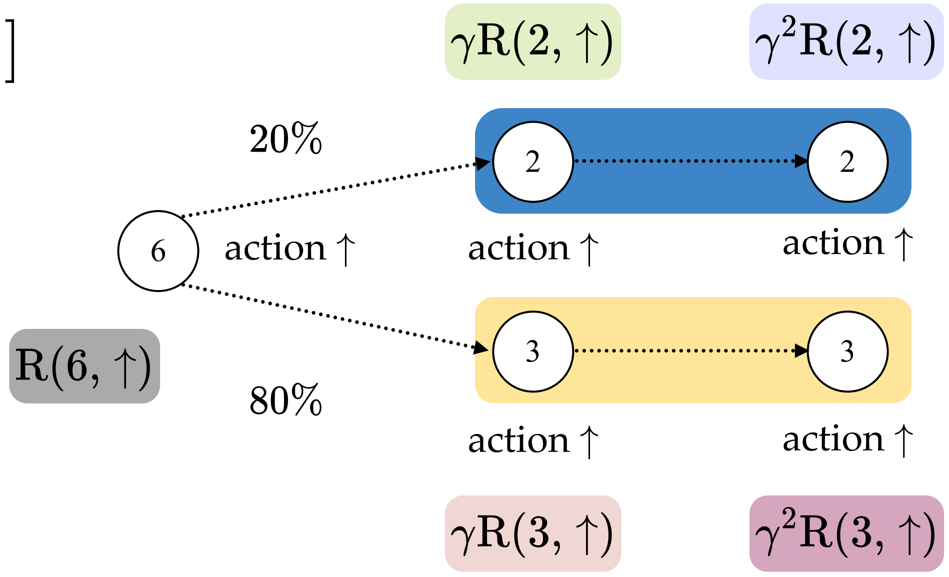
one special transition:



rewards



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



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

$$= \mathrm{R}(6,\uparrow) + 20\%\ [\ \gamma\mathrm{R}(2,\uparrow) + \gamma^2\mathrm{R}(2,\uparrow)\ ] + 80\%\ [\ \gamma\mathrm{R}(3,\uparrow) + \gamma^2\mathrm{R}(3,\uparrow)\ ]$$

$$= \mathrm{R}(6,\uparrow) + 20\%\ \gamma\ [\mathrm{R}(2,\uparrow) + \gamma\mathrm{R}(2,\uparrow)] + 80\%\ \gamma\ [\mathrm{R}(3,\uparrow) + \gamma\mathrm{R}(3,\uparrow)]$$

$$= \mathrm{R}(6,\uparrow) + 20\%\ \gamma\ V_2^{\uparrow}(2) + 80\%\ \gamma\ \mathrm{V}_2^{\uparrow}(3)$$

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

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

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

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

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

sum up future values weighted by the probability of getting to that next state s'

discounted by γ

Bellman Recursion

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

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

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

$V_3^\uparrow(s)$

0.00	0.00	2.71
0.00	0.00	-8.63
0.00	0.00	-8.35

$V_4^\uparrow(s)$

0.00	0.00	3.44
0.00	0.00	-8.05
0.00	0.00	-7.77

$V_5^\uparrow(s)$

0.00	0.00	4.10
0.00	0.00	-7.52
0.00	0.00	-7.24

$V_6^\uparrow(s)$

0.00	0.00	4.69
0.00	0.00	-7.05
0.00	0.00	-6.77

$$V_6^\uparrow(6) = R(6, \uparrow) + \gamma[.2 V_5^\uparrow(2) + .8 \times V_5^\uparrow(3)]$$

$$-7.048 = -10 + .9[.2 * 0 + 0.8 * 4.10]$$

$\dots$

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

0.00	0.00	9.98
0.00	0.00	-2.81
0.00	0.00	-2.53

$V_{62}^\uparrow(s)$

0.00	0.00	9.99
0.00	0.00	-2.81
0.00	0.00	-2.53

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

approaches infinity

If the horizon h goes to infinity

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

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

$|\mathcal{S}|$ many linear equations, one equation for each state

0.00	0.00	10.00
0.00	0.00	-2.80
0.00	0.00	-2.52

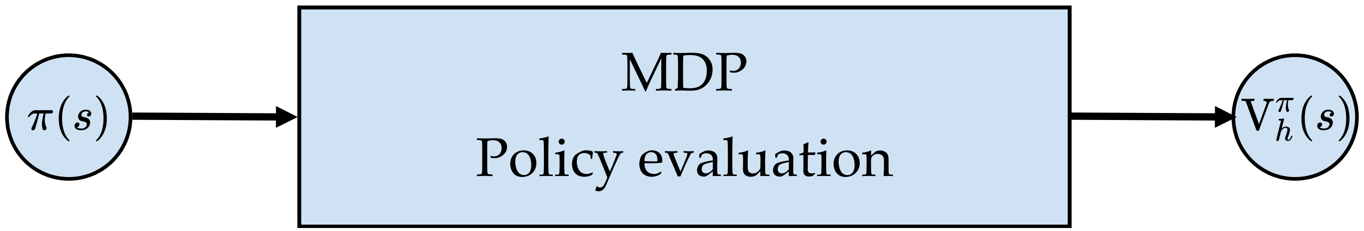
$10 = \mathrm{V}_\infty^{\uparrow}(3) = \mathrm{R}(3, \uparrow) + \gamma[\mathrm{V}_\infty^{\uparrow}(3)] = 1 + .9 \times 10$

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

$-2.52 = \mathrm{V}_\infty^{\uparrow}(9) = \mathrm{R}(9, \uparrow) + \gamma[\mathrm{V}_\infty^{\uparrow}(6)] = 0 + .9 \times (-2.8)$

typically $\gamma < 1$ in MDP definition, motivated to make $\mathrm{V}_\infty^{\pi}(s) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathrm{R}\left(s_t, \pi\left(s_t\right)\right) \mid s_0 = s, \pi\right]$ finite.

`Quick summary`



1. By summing h terms:

Recall: For a *given* policy $\pi(s)$, the (state) **value functions**

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

2. By leveraging structure:

finite-horizon Bellman recursions

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

infinite-horizon Bellman equations

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

Outline

- Markov Decision Processes Definition, terminologies, and policy
- 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^*

Definition: for a given MDP and a fixed horizon h (possibly infinite), π^* is an optimal policy if $\mathrm{V}_h^{\pi^*}(s) = \mathrm{V}_h^*(s) \geqslant \mathrm{V}_h^{\pi}(s)$ **for all** $s \in \mathcal{S}$ and **for all** possible policy π.

- An MDP has a unique optimal value $\mathrm{V}_h^*(s)$.
- Optimal policy π^* might not be unique (think, e.g. symmetric world).
- For finite h, optimal policy π_h^* depends on how many time steps left.
- When $h \to \infty$, time no longer matters, i.e., there exists a stationary π^*.
- Under optimal policy, recursion holds too

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

Definition: for a given MDP and a fixed horizon h (possibly infinite), π^* is an optimal policy if $\mathrm{V}_h^{\pi^*}(s) = \mathrm{V}_h^*(s) \geqslant \mathrm{V}_h^{\pi}(s)$ **for all** $s \in \mathcal{S}$ and **for all** possible policy π.

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

How to search for an optimal policy π^*?

- One idea: enumerate over all π, do policy evaluation, compare V^π, get $\mathrm{V}^*(s)$
- tedious, and even with $\mathrm{V}^*(s)$... not super clear how to act

$\mathrm{V}_{61}^*(s)$

8.08	8.98	9.98
7.27	8.08	-1.20
6.54	7.27	6.54

$\mathrm{V}_{62}^*(s)$

8.09	8.99	9.99
7.28	8.09	-1.19
6.55	7.28	6.55

$\cdots$

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

8.10	9.00	10.00
7.29	8.10	-1.18
6.56	7.29	6.56

Outline

- Markov Decision Processes Definition, terminologies, and policy
- 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 state-action value functions $\mathrm{Q}_h^*(s, a)$

$\mathrm{Q}_h^*(s, a)$: the expected sum of discounted rewards for

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

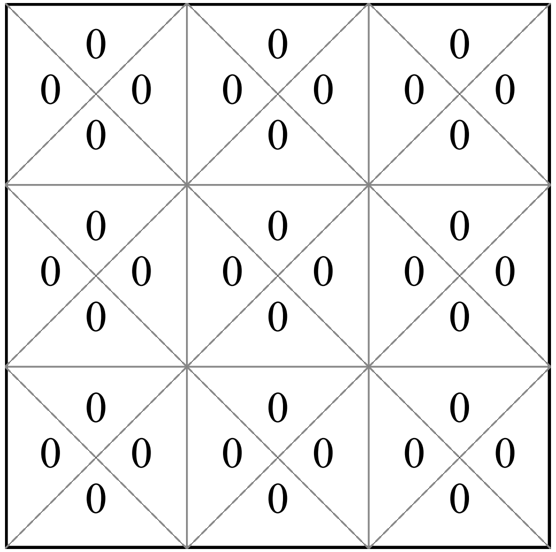
$$\mathrm{Q}_h^*(s, a) = \mathrm{R}(s, a) + \gamma \sum_{s'} \mathrm{T}\left(s, a, s'\right) \max_{a'} \mathrm{Q}_{h-1}^*\left(s', a'\right), \forall s, a, h$$

recursively finding $Q^*_h(s, a)$

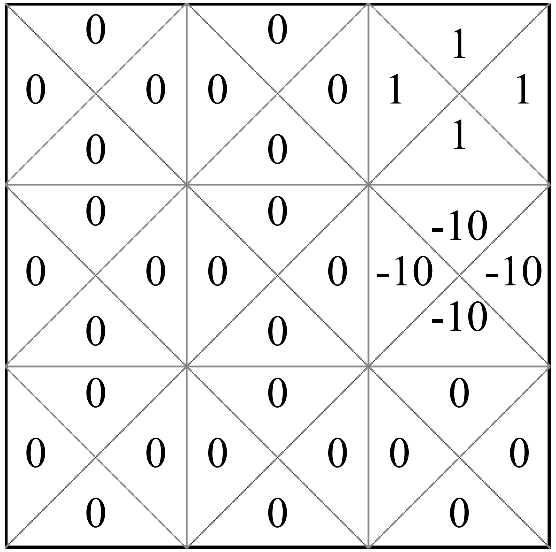
$Q^*_h(s, a)$: the expected sum of discounted rewards for

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

$Q^*_0(s, a)$

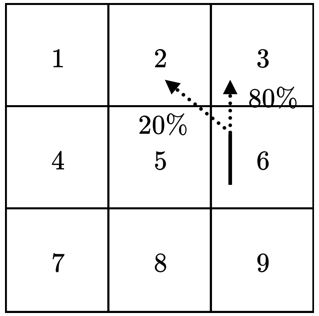


$Q^*_1(s, a) = R(s, a)$

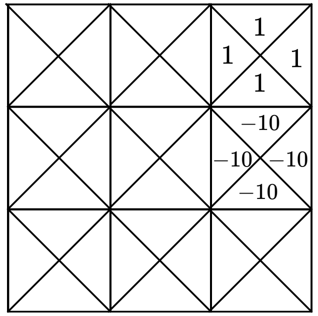


Recall: $\gamma = 0.9$

States and one special transition:



$R(s, a)$

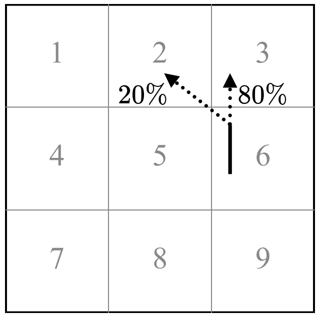


$Q^*_h(s,a)$: the value for

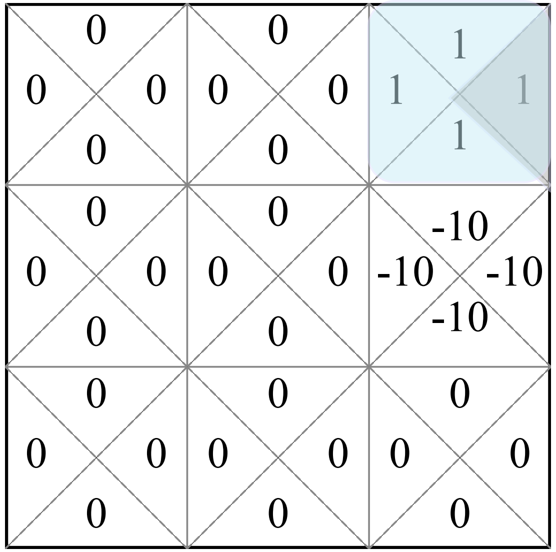
- starting in state s,
- take action a, for one step
- act **optimally** there afterwards for the remaining $(h-1)$ steps

Recall: $\gamma = 0.9$

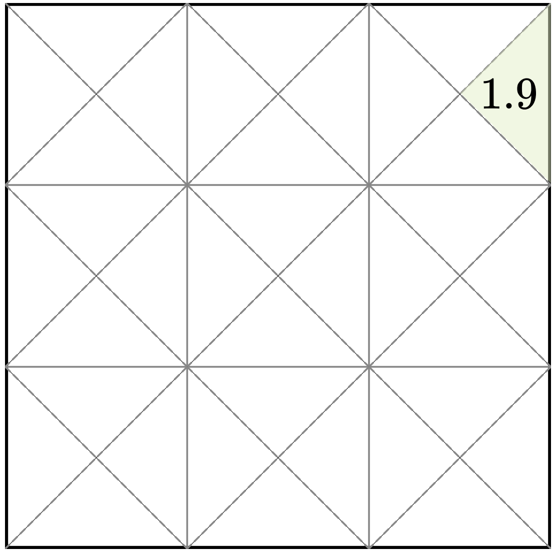
States and one special transition:



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



$$Q^*_2(s,a)$$



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

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

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

$$= 1 + .9 \max_{a'} Q^*_1(3, a')$$

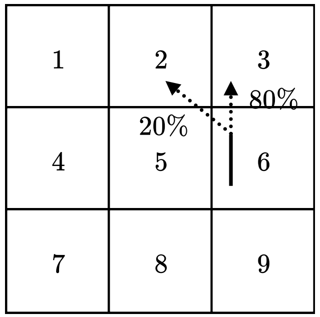
$$= 1.9$$

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

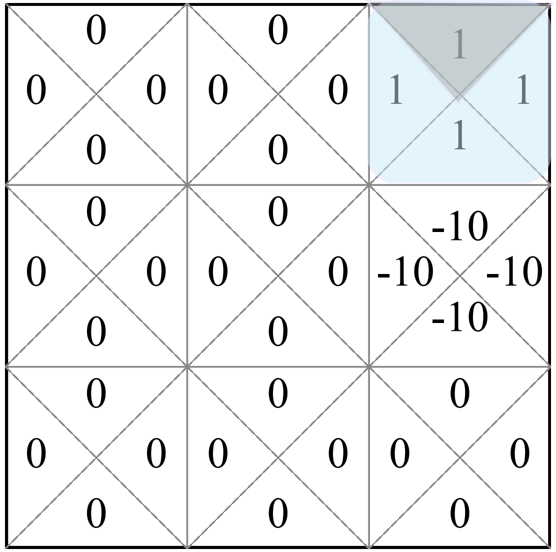
- starting in state s,
- take action a, for one step
- act **optimally** there afterwards for the remaining $(h-1)$ steps

Recall: $\gamma = 0.9$

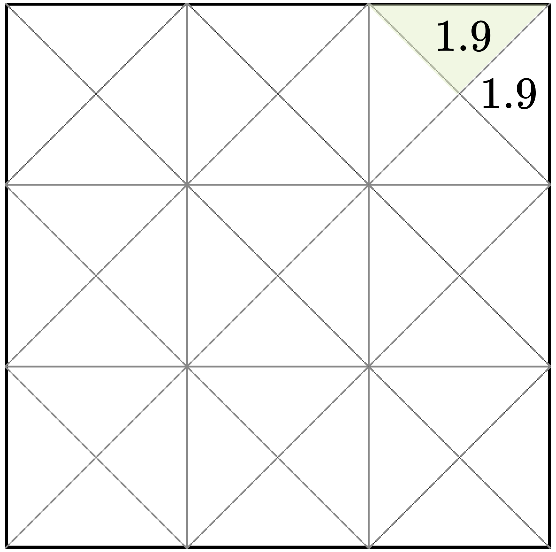
States and one special transition:



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



$Q^*_2(s,a)$



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

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

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

$$= 1 + .9 \max_{a'} Q^*_1(3, a')$$

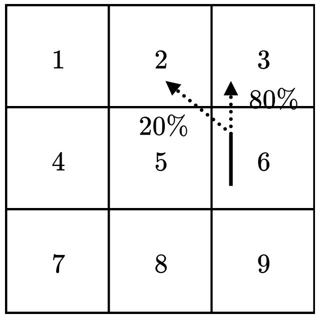
$$= 1.9$$

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

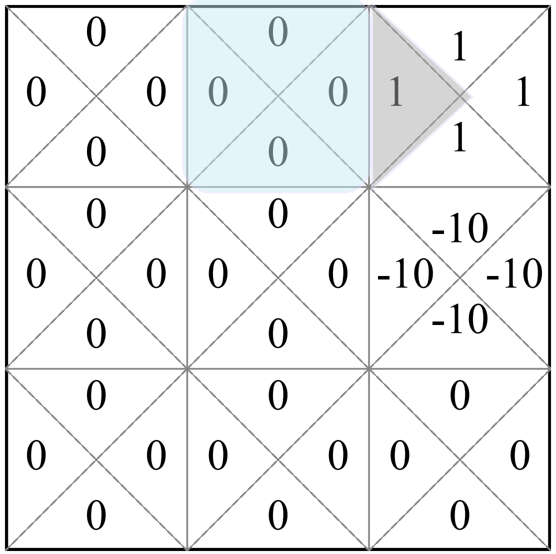
- starting in state s,
- take action a, for one step
- act **optimally** there afterwards for the remaining $(h-1)$ steps

Recall: $\gamma = 0.9$

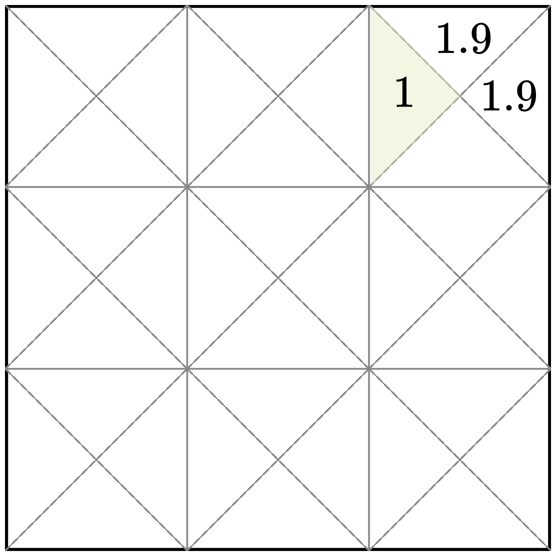
States and one special transition:



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



$$Q^*_2(s, a)$$



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

- receive $R(3, \leftarrow)$
- next state $s' = 2$, act **optimally** for the remaining one timestep
  - receive $\max_{a'} Q^*_1(2, a')$

$$Q^*_2(3, \leftarrow) = R(3, \leftarrow) + \gamma \max_{a'} Q^*_1(2, a')$$

$$= 1 + .9 \max_{a'} Q^*_1(2, a')$$

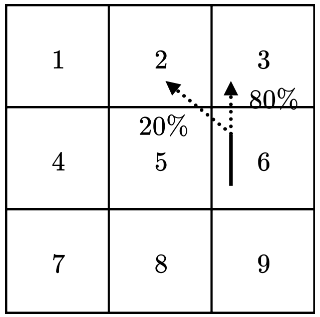
$$= 1$$

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

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

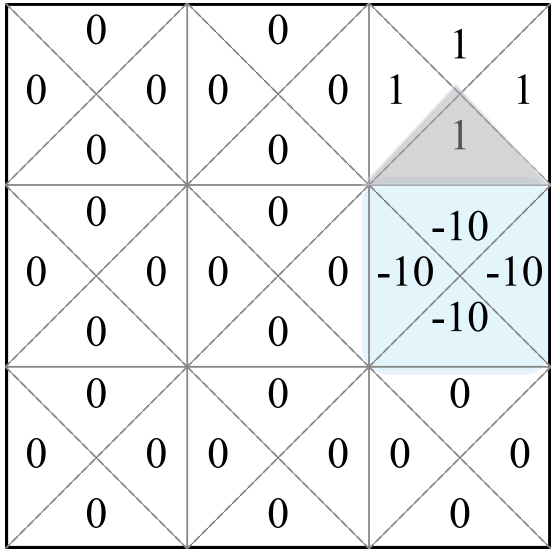
Recall: $\gamma = 0.9$

States and one special transition:

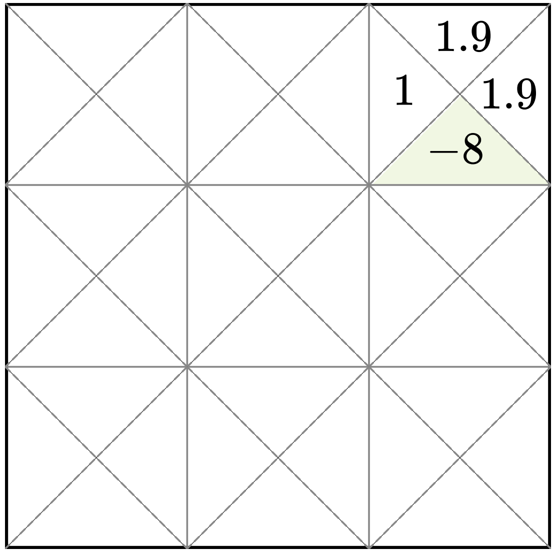


Let's consider $Q^*_2(3, \downarrow)$

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



$$Q^*_2(s,a)$$



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

$$Q^*_2(3,\downarrow) = R(3,\downarrow) \ + \gamma \max_{a'} Q^*_1(2, a')$$

$$= 1 + .9 \max_{a'} Q^*_1(6, a')$$

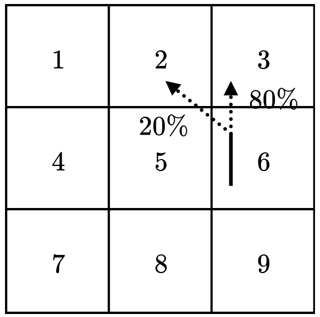
$$= -8$$

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

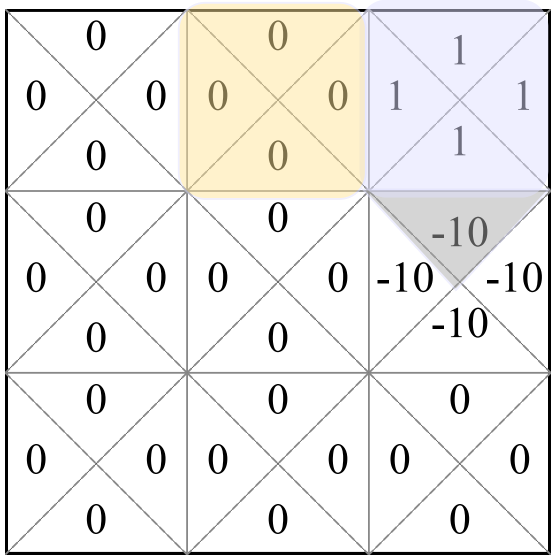
- starting in state s,
- take action a, for one step
- act **optimally** there afterwards for the remaining $(h-1)$ steps

Recall: $\gamma = 0.9$

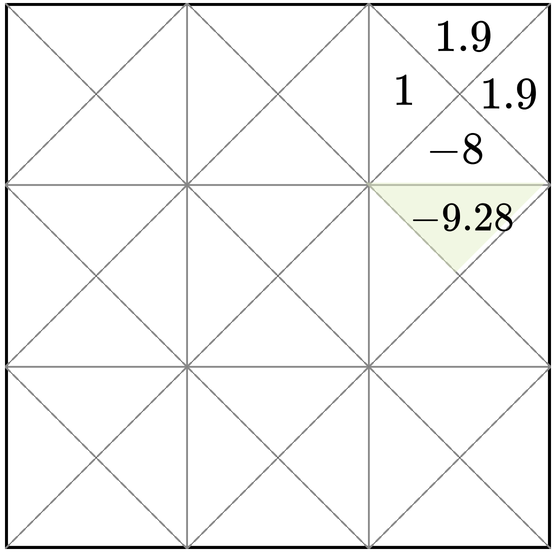
States and one special transition:



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



$Q^*_2(s,a)$



- receive $R(6, \uparrow)$
- act **optimally** for one more timestep, at the next state s'
  - 20% chance, $s' = 2$, act optimally, receive $\max_{a'} Q^*_1(2, a')$
  - 80% chance, $s' = 3$, act optimally, receive $\max_{a'} Q^*_1(3, a')$

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

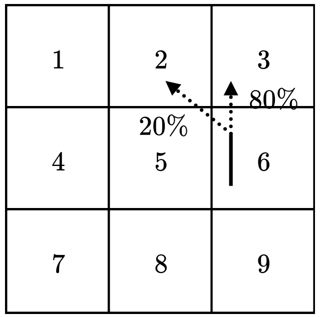
$$= -10 + .9[.2 \times 0 + .8 \times 1] = -9.28$$

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

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

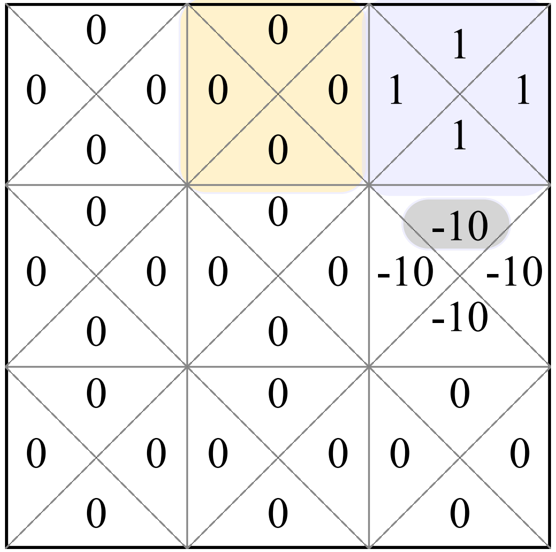
Recall: $\gamma = 0.9$

States and one special transition:

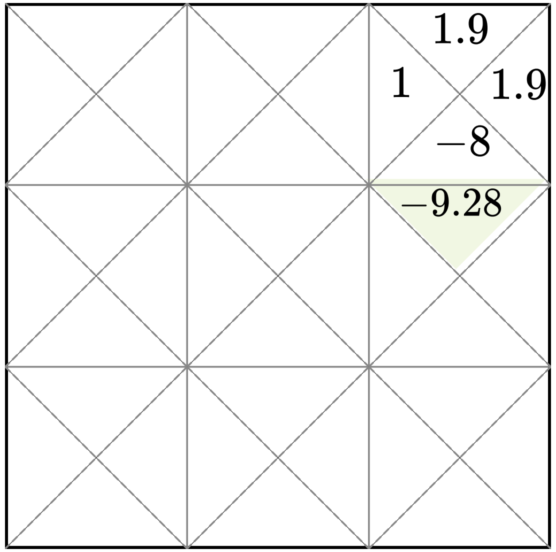


$Q^*_1(s,a)$

$= R(s,a)$



$Q^*_2(s,a)$



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

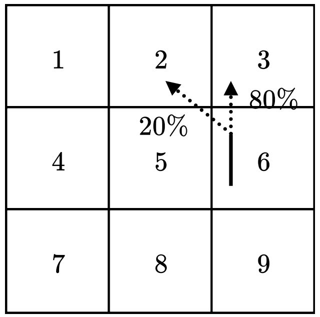
in general $$Q^*_h(s,a) = R(s,a) + \gamma \sum_{s'} T(s,a,s') \max_{a'} Q^*_{h-1}(s',a'), \forall s,a,h$$

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

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

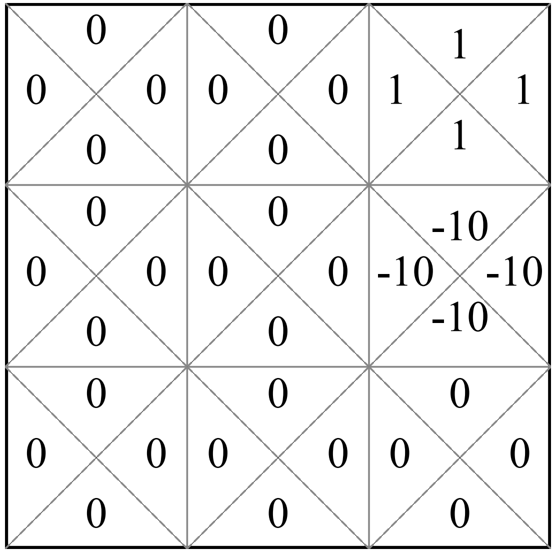
Recall: $\gamma = 0.9$

States and one special transition:

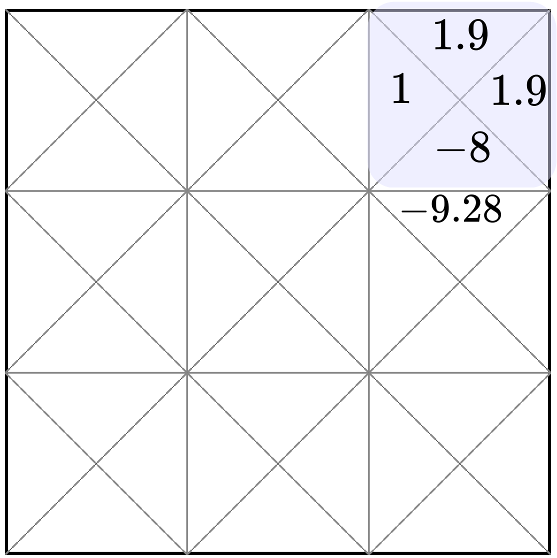


$Q^*_1(s, a)$

$= R(s, a)$



$Q^*_2(s, a)$



what's the optimal action in state 3, with horizon 2, given by $\pi^*_2(3) = ?$

either up or right

in general $\pi^*_h(s) = \arg\max_a Q^*_h(s, a), \forall s, h$

Outline

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

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

we can have an infinite horizon equation

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

```
Value Iteration
```

1. **for** $s \in \mathcal{S}, a \in \mathcal{A}$:
2. $\quad Q_{\text{old}}(\mathrm{s},\mathrm{a}) = 0$
3. **while** True:
4. $\quad$ **for** $s \in \mathcal{S}, a \in \mathcal{A}$:
5. $\quad\quad Q_{\text{new}}(s,a) \leftarrow R(s,a) + \gamma \sum_{s'} T(s,a,s') \max_{a'} Q_{\text{old}}(s',a')$
6. $\quad$ **if** $\max_{s,a} |Q_{\text{old}}(s,a) - Q_{\text{new}}(s,a)| < \epsilon$:
7. $\quad\quad$ **return** Q_{new}
8. $\quad Q_{\text{old}} \leftarrow Q_{\text{new}}$

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

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

V values vs. Q values

- V is defined over state space; Q is defined over (state, action) space.
- $V_h^*(s)$ can be derived from $Q_h^*(s, a)$:, and vise versa.
- Q^* is easier to read "optimal actions" from.
- We care more about V^π and Q^*

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

0.00	0.00	9.98
0.00	0.00	-2.81
0.00	0.00	-2.53

$V^{\uparrow}_{62}(s)$

0.00	0.00	9.99
0.00	0.00	-2.81
0.00	0.00	-2.53

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

0.00	0.00	10.00
0.00	0.00	-2.80
0.00	0.00	-2.52

$V^{*}_{61}(s)$

8.08	8.98	9.98
7.27	8.08	-1.20
6.54	7.27	6.54

$V^{*}_{62}(s)$

8.09	8.99	9.99
7.28	8.09	-1.19
6.55	7.28	6.55

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

8.10	9.00	10.00
7.29	8.10	-1.18
6.56	7.29	6.56

$$V^{*}_{h}(s) = \max_a \left[Q^{*}_{h}(s,a)\right]$$

$Q^{*}_{61}(s,a)$

↑ 7.27, ← 7.27, → 8.08, ↓ 6.54	↑ 8.08, ← 7.27, → 8.98, ↓ 7.27	↑ 9.98, ← 9.08, → 9.98, ↓ -0.08
↑ 7.27, ← 6.54, → 7.27, ↓ 5.89	↑ 8.08, ← 6.54, → -1.08, ↓ 6.54	↑ -1.20, ← -2.73, → -11.08, ↓ -4.11
↑ 6.54, ← 5.89, → 6.54, ↓ 5.89	↑ 7.27, ← 5.89, → 5.89, ↓ 6.54	↑ -1.08, ← 6.54, → 5.89, ↓ 5.89

$Q^{*}_{62}(s,a)$

↑ 7.28, ← 7.28, → 8.09, ↓ 6.55	↑ 8.09, ← 7.28, → 8.99, ↓ 7.28	↑ 9.99, ← 9.09, → 9.99, ↓ -0.08
↑ 7.28, ← 6.55, → 7.28, ↓ 5.89	↑ 8.09, ← 6.55, → -1.08, ↓ 6.55	↑ -1.19, ← -2.72, → -11.08, ↓ -4.11
↑ 6.55, ← 5.89, → 6.55, ↓ 5.89	↑ 7.28, ← 5.89, → 5.89, ↓ 6.55	↑ -1.08, ← 6.55, → 5.89, ↓ 5.89

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

↑ 7.29, ← 7.29, → 8.10, ↓ 6.56	↑ 8.10, ← 7.29, → 9.00, ↓ 7.29	↑ 10.00, ← 9.10, → 10.00, ↓ -0.06
↑ 7.29, ← 6.56, → 7.29, ↓ 5.90	↑ 8.10, ← 6.56, → -1.06, ↓ 6.56	↑ -1.18, ← -2.71, → -11.06, ↓ -4.10
↑ 6.56, ← 5.90, → 6.56, ↓ 5.90	↑ 7.29, ← 5.90, → 5.90, ↓ 6.56	↑ -1.06, ← 6.56, → 5.90, ↓ 5.90

$$\pi^{*}_{h}(s) = \arg\max_a \left[Q^{*}_{h}(s,a)\right]$$

Summary

- Markov decision processes (MDP) is 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, we can calculate $V^\pi(s)$ via
  - the summation over rewards definition
  - Bellman recursion for finite horizon, 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/DefAyvq8KA9kg37X8

We'd love to hear your thoughts.

Thanks!