

Markov Decision Processes

Chen-Yu Wei

Sequence of Actions

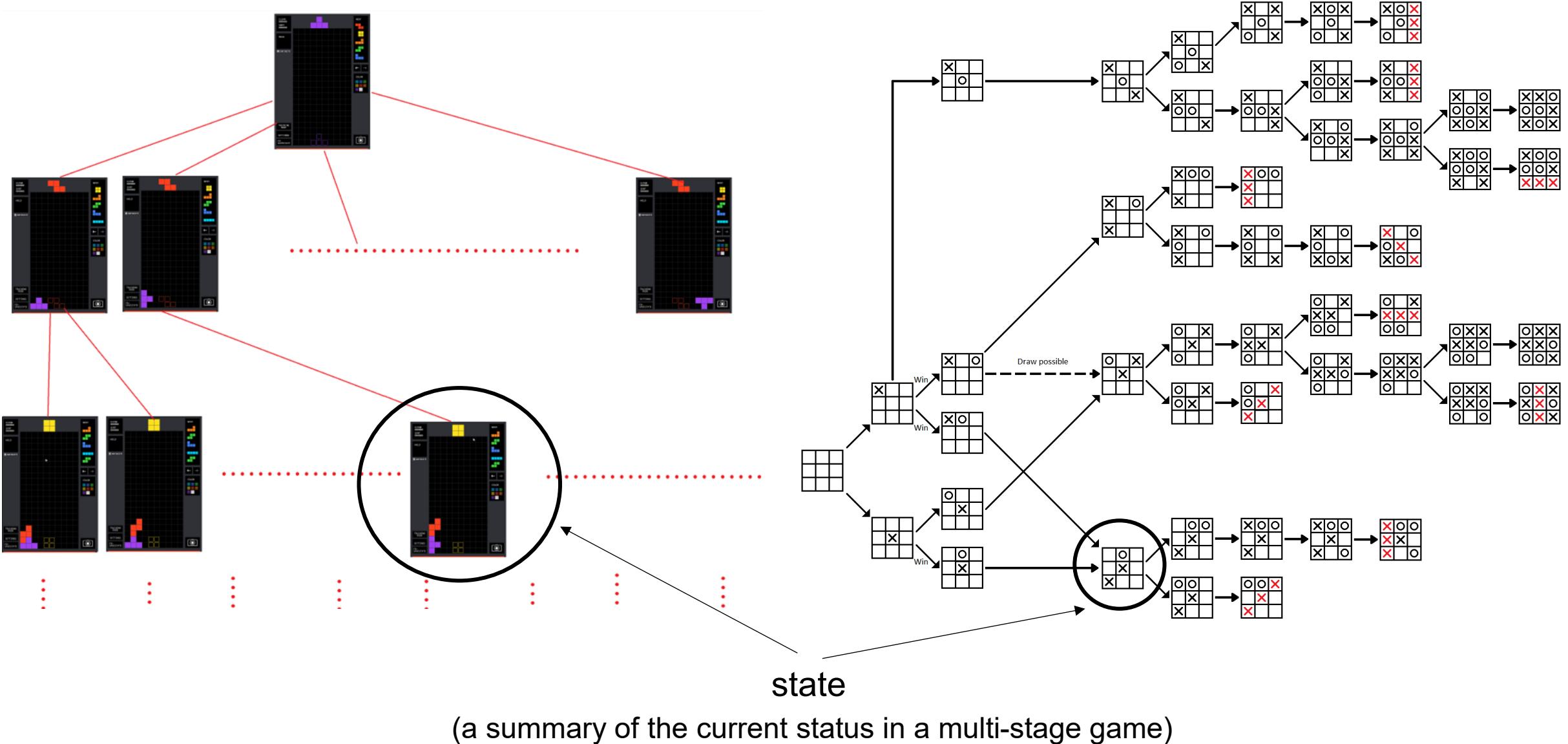


To win the game, the learner has to take a sequence of actions $a_1 \rightarrow a_2 \rightarrow \dots \rightarrow a_H$.

The effect of a particular action may not be revealed instantaneously.

- Some effect may be revealed instantaneously
- Some may be revealed later

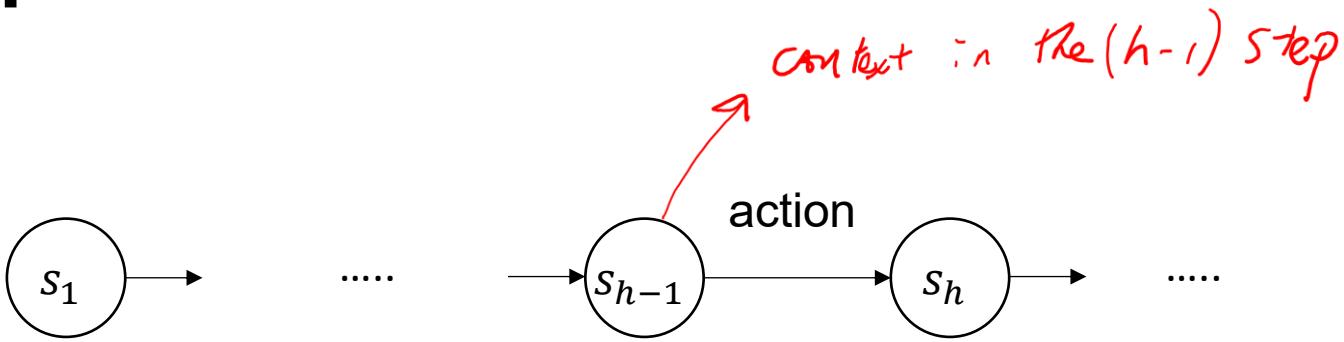
Sequence of Actions



Sequence of Actions

- The number of possible combinations of actions grows **exponentially** with the length of the sequence.
- We would like to decompose the problem so that every single decision in the sequence is easy to make.
- **State:** a **summary** of the **status of the world** and the **progress of the learner**, so that all future decisions can only depend on the state and not on everything else.
 - Games (Go, Chess): To decide future moves, the player only need the current board configuration.
 - Robot navigation to a goal: only need the current position and not the exact path reaching the current position.
 - Inventory management: only need the current inventory level, and not the sequence of past sales.

Sequence of Actions



Like a sequential contextual bandit problem – except that future contexts depends on the learner's past decisions.

Interaction Protocol (Episodic Setting)

For episode $t = 1, 2, \dots, T$:

$$h \leftarrow 1$$

✓ Environment generates initial state $s_{t,1}$

While episode t has not ended:

Learner chooses an action $a_{t,h}$

Learner observes instantaneous reward $r_{t,h}$ with $\mathbb{E}[r_{t,h}] = R(s_{t,h}, a_{t,h})$

Environment generates next state $s_{t,h+1} \sim P(\cdot | s_{t,h}, a_{t,h})$

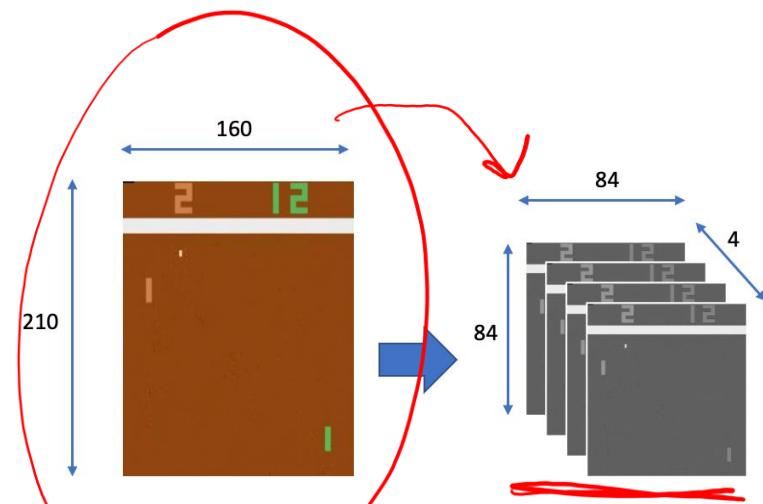
$$h \leftarrow h + 1$$

Markov assumption:

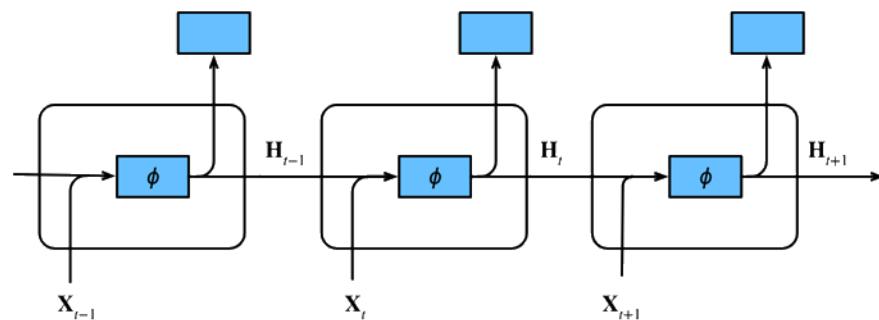
~~$r_{t,h}$ and $s_{t,h+1}$ are conditionally independent of $(s_{t,1}, a_{t,1}, \dots, s_{t,h-1}, a_{t,h-1})$ given $s_{t,h}$~~

Goal: maximize
$$\sum_{t=1}^T \sum_{h=1}^{\tau_t} R(s_{t,h}, a_{t,h})$$
 ← length of episode

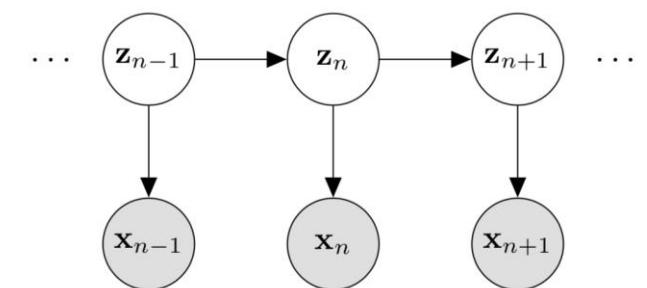
From Observations to States



Stacking recent observations



Recurrent neural network

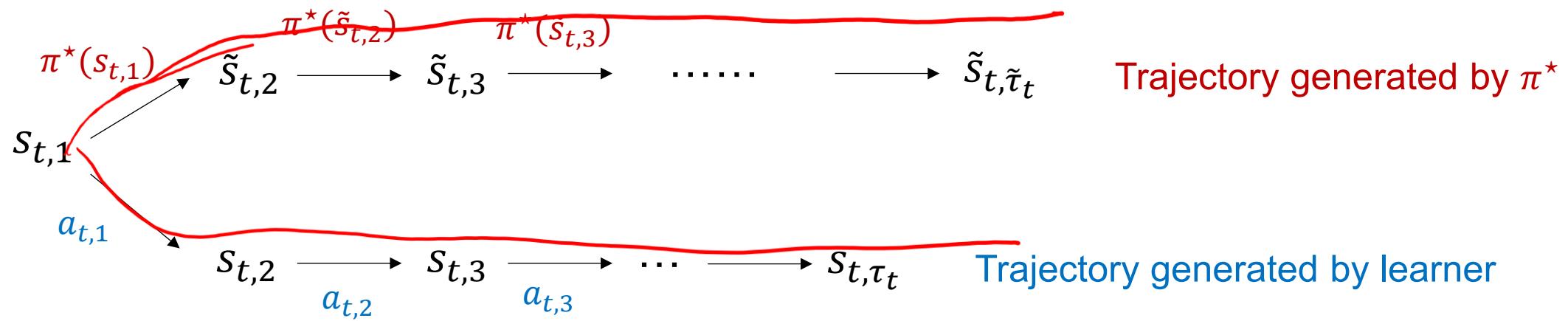


Hidden Markov model

Regret (Episodic Setting)

Policy : mapping from state to action
(action distribution)

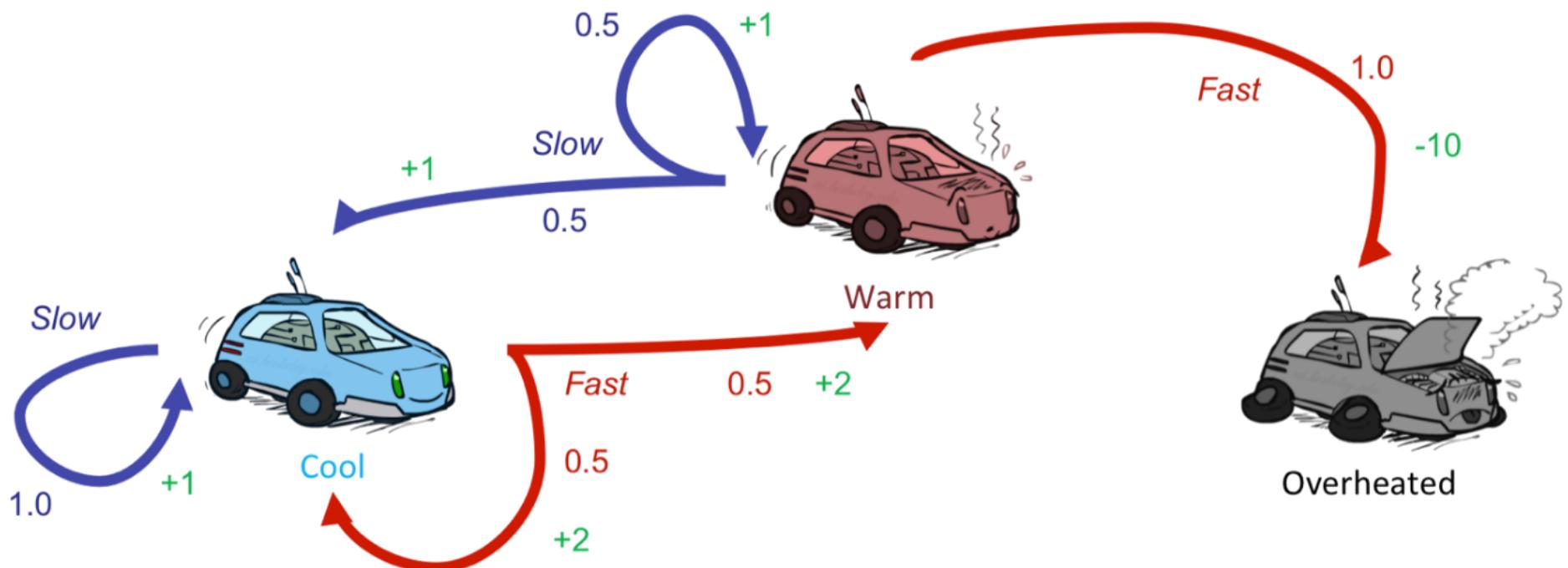
$$\text{Regret} = \underbrace{\max_{\pi^*} \mathbb{E}^{\pi^*} \left[\sum_{t=1}^T \sum_{h=1}^{\tilde{\tau}_t} R(\tilde{s}_{t,h}, \pi^*(\tilde{s}_{t,h})) \right]}_{\text{Benchmark}} - \sum_{t=1}^T \sum_{h=1}^{\tau_t} R(s_{t,h}, a_{t,h})$$



Example: Racing

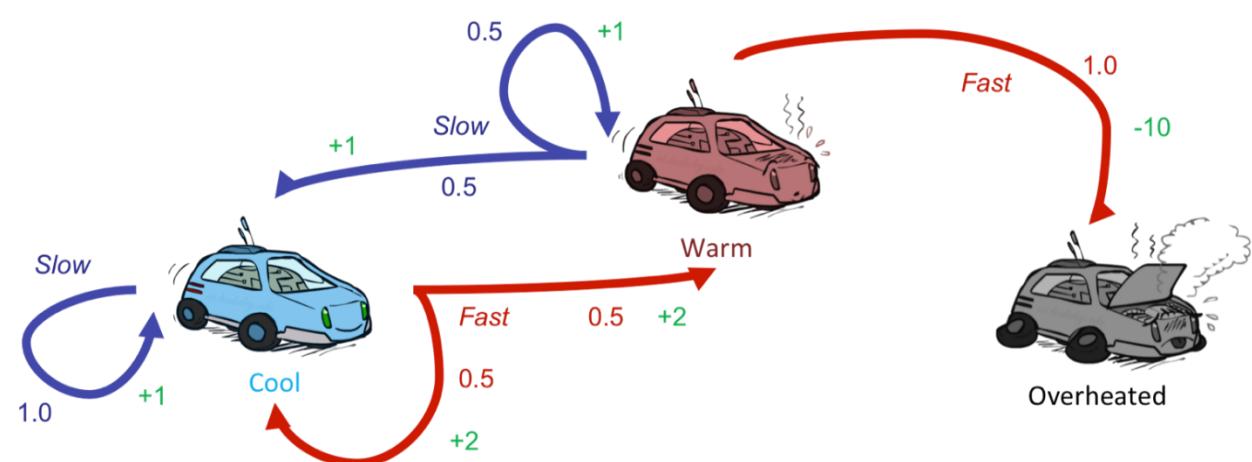
- A robot car wants to travel far, quickly
- Three states: Cool, Warm, Overheated
- Two actions: Slow, Fast
- Going faster gets double reward

$$\left\{ \begin{array}{l} R(s,a) \\ P(s'|s,a) \end{array} \right.$$



Example: Racing

s	a	s'	$P(s' s, a)$	$R(s, a)$
	Slow		1.0	+1
	Fast		0.5	+2
	Fast		0.5	+2
	Slow		0.5	+1
	Slow		0.5	+1
	Fast		1.0	-10
	(end)		1.0	0



Formulations

- Interaction Protocol
 - Fixed-Horizon
 - Variable-Horizon
- Performance Metric
 - Total Reward
 - Discounted Reward
- Policy
 - Markov policy
 - Stationary policy

Horizon = Length of an episode

Interaction Protocols (1/2): Fixed-Horizon

Horizon length is a fixed number H

$h \leftarrow 1$

Observe initial state $s_1 \sim \rho$

While $h \leq H$:

 Choose action a_h

 Observe reward r_h with $\mathbb{E}[r_h] = R(s_h, a_h)$

 Observe next state $s_{h+1} \sim P(\cdot | s_h, a_h)$

Examples: games with a fixed number of time

Interaction Protocols (2/2): Variable-Horizon

The learner interacts with the environment until reaching **terminal states** $\mathcal{T} \subset \mathcal{S}$

$h \leftarrow 1$

Observe initial state $s_1 \sim \rho$

While $s_h \notin \mathcal{T}$:

 Choose action a_h

 Observe reward r_h with $\mathbb{E}[r_h] = R(s_h, a_h)$

 Observe next state $s_{h+1} \sim P(\cdot | s_h, a_h)$

$h \leftarrow h + 1$

Examples: video games, robotics tasks, personalized recommendations, etc.

Formulations

- Interaction Protocol
 - Fixed-Horizon
 - Variable-Horizon
- Performance Metric
 - Total Reward
 - Discounted Reward
- Policy
 - Markov policy
 - Stationary policy

Horizon = Length of an episode

Performance Metric

τ : the step where the episode ends

Total Reward:

$$r_h \in [-1, 1]$$
$$\Rightarrow \left| \text{Discounted total reward} \right| \leq 1 + \gamma + \gamma^2 + \dots + \gamma^{\tau-1}$$
$$\leq \frac{1}{1-\gamma} \quad \checkmark$$

Discounted Total Reward:

$$\sum_{h=1}^{\tau} r_h$$

$$\sum_{h=1}^{\tau} \gamma^{h-1} r_h$$

$$\Downarrow \underline{r_1 + \gamma r_2 + \gamma^2 r_3 + \dots +}$$

$\gamma \approx 0.9$ $\rightarrow \frac{1}{1-\gamma} = 10$

$\approx 0.99 \rightarrow \frac{1}{1-\gamma} = 100$

ergen kart

Due to discounting, the future reward starting from any state is always upper bounded by $\frac{\text{range of } r}{1-\gamma}$, even if the episode length is very very long.

Without discounting, the range of future reward could be unbounded \rightarrow making it hard to optimize

There is a potential mismatch between our ultimate goal and what we really optimized.

Formulations

- Interaction Protocol
 - Fixed-Horizon
 - Variable-Horizon
- Performance Metric
 - Total Reward
 - Discounted Reward
- Policy
 - Markov policy
 - Stationary policy

Policy for MDPs

Markov Policy

$$a_h \sim \pi_h(\cdot | s_h)$$
$$a_h = \pi_h(s_h)$$

$$a_h \sim \pi(\cdot | s_1, a_1, s_2, a_2, \dots, s_h)$$

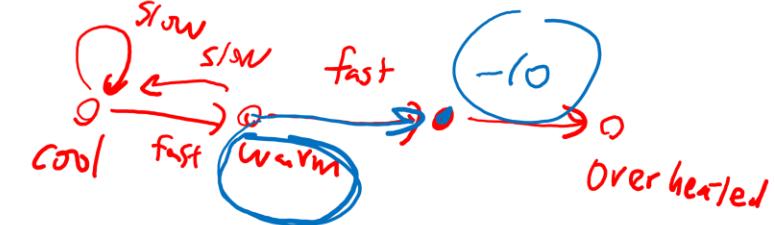
history-dependent

For **fixed-horizon** setting, there exists an optimal policy in this class

Stationary Policy

$$a_h \sim \pi(\cdot | s_h)$$
$$a_h = \pi(s_h)$$

For **variable-horizon** settings, there exists an optimal policy in this class



$$\text{slow : } +1$$
$$\text{fast : } 2$$

Markov Policy = Stationary Policy where the state is augmented with **the timestep**.

A stationary policy specifies

$$\pi(\text{Slow} \mid \text{Cool})$$

$$\pi(\text{Fast} \mid \text{Cool})$$

$$\pi(\text{Slow} \mid \text{Warm})$$

$$\pi(\text{Fast} \mid \text{Warm})$$

A Markov policy specifies

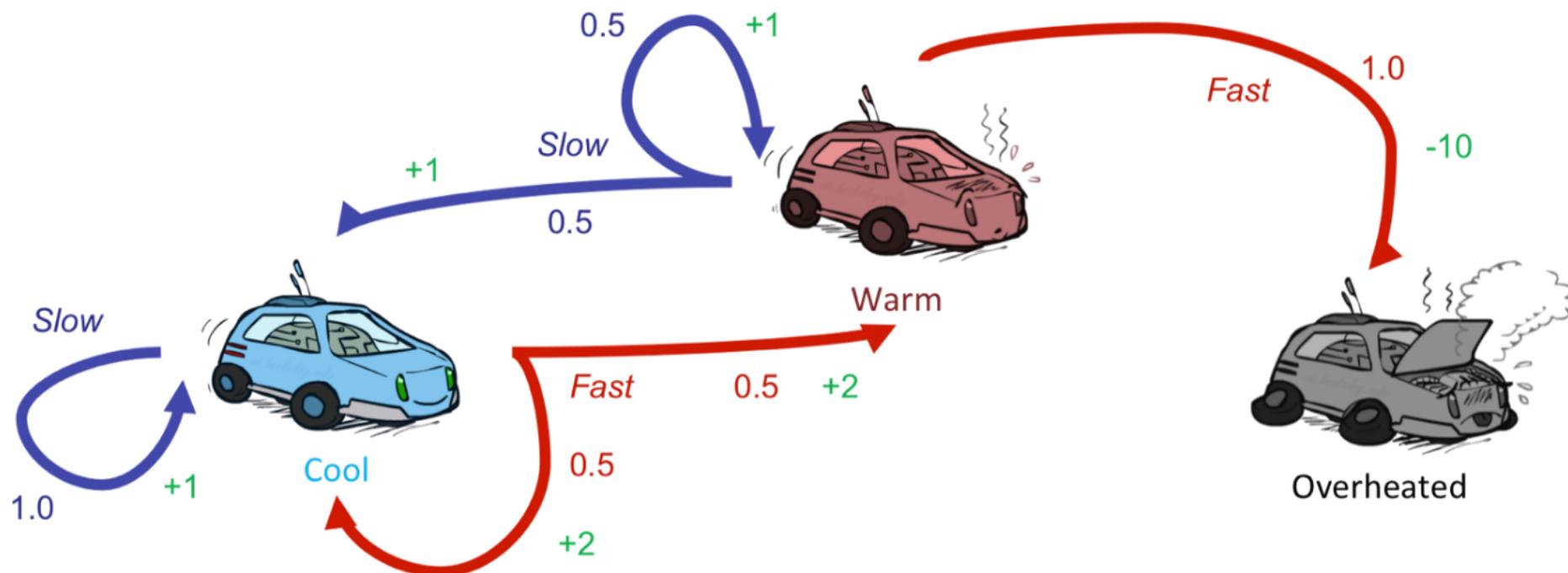
$$\pi_h(\text{Slow} \mid \text{Cool})$$

$$\pi_h(\text{Fast} \mid \text{Cool})$$

$$\pi_h(\text{Slow} \mid \text{Warm})$$

$$\pi_h(\text{Fast} \mid \text{Warm})$$

$$\forall h$$

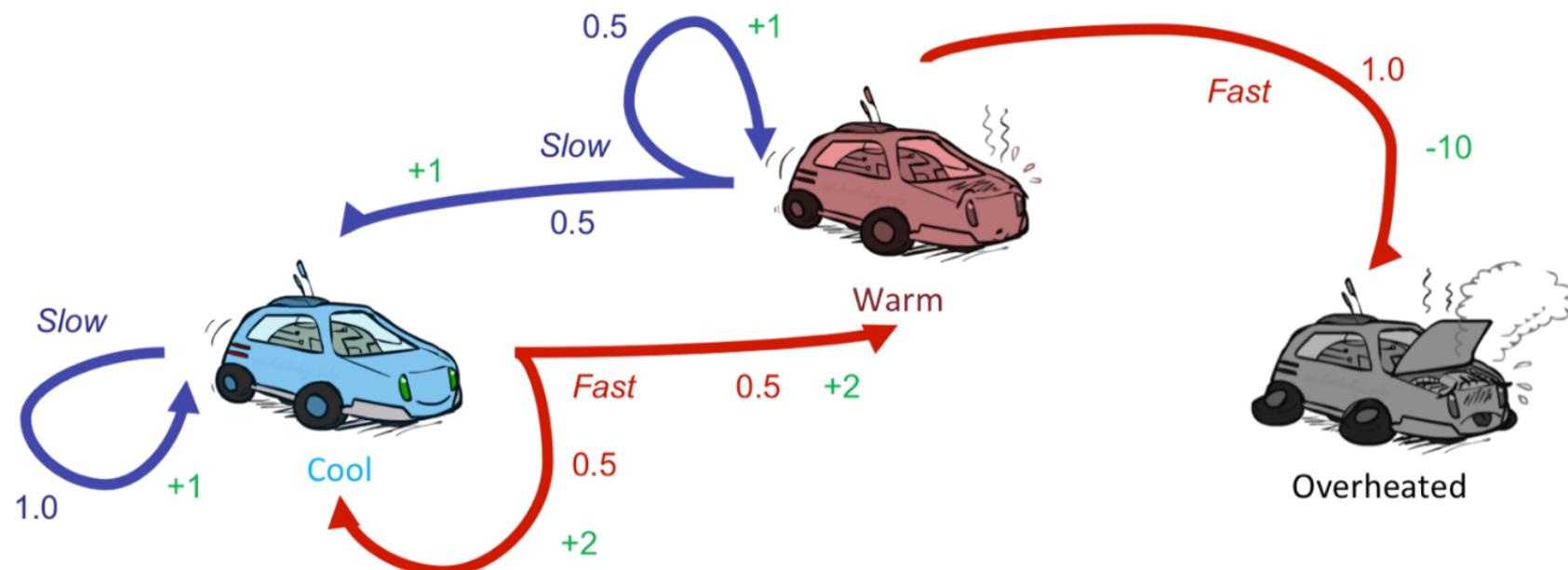


Value Iteration

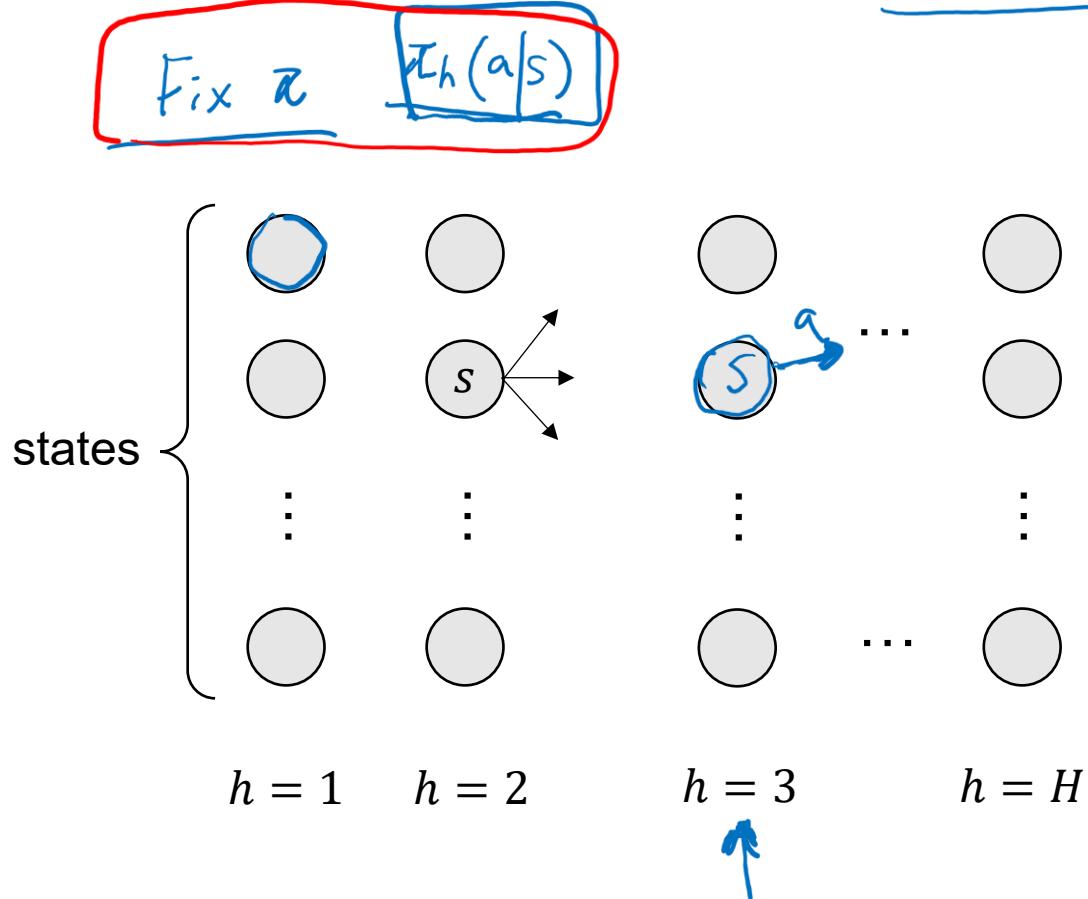
(Fixed-Horizon + Total-Reward)

Two Tasks

- Policy Evaluation:** Calculate the expected total reward of a given policy
What is the expected total reward for the policy $\pi(\text{cool}) = \text{fast}$, $\pi(\text{warm}) = \text{slow}$?
- Policy Optimization:** Find the best policy
What is the policy that achieves the highest expected total reward?



Value Iteration for Policy Evaluation



State transition: $P(s'|s, a)$

Reward: $R(s, a)$

$$\cancel{V_h^\pi(s) =}$$

$$E^\pi \left[\sum_{k=h}^H R(s_k, a_k) \mid s_h = s \right]$$

$$Q_h^\pi(s, a) = E^\pi \left[\sum_{k=h}^H R(s_k, a_k) \mid (s_h, a_h) = (\underline{s}, \underline{a}) \right]$$

$$\checkmark V_h^\pi(s) = E^\pi \left[\sum_{k=h}^H R(s_k, a_k) \mid s_h = s \right]$$

Backward induction:

$$V_{H+1}^\pi(s) = 0 \quad \forall s$$

For $h = H, \dots, 1$: for all s, a

$$\boxed{Q_h^\pi(s, a) = R(s, a) + \sum_{s'} P(s'|s, a) V_{h+1}^\pi(s')}$$

Expected total reward
of π from step $h + 1$

$$\boxed{V_h^\pi(s) = \sum_a \pi_h(a|s) Q_h^\pi(s, a)}$$

Bellman Equation

Q_h^π is called “the state-action value functions of policy π ”
 V_h^π is called “the state value function of policy π ”
Both can be just called “**value functions**”

$$Q_h^\pi(s, a) = R(s, a) + \sum_{s'} P(s'|s, a) V_{h+1}^\pi(s')$$

$$V_h^\pi(s) = \sum_a \pi_h(a|s) Q_h^\pi(s, a)$$

or

$$\underline{Q_h^\pi(s, a)} = R(s, a) + \sum_{s', a'} P(s'|s, a) \pi_{h+1}(a'|s') \underline{Q_{h+1}^\pi(s', a')}$$

✓

or

$$\underline{V_h^\pi(s)} = \sum_a \pi_h(a|s) \left(R(s, a) + \sum_{s'} P(s'|s, a) \underline{V_{h+1}^\pi(s')} \right)$$

✓

The Meaning of Bellman Equations

Definitions

$\mathcal{R}, \mathcal{P}, \pi$

$$Q_h^\pi(s, a) \triangleq \mathbb{E}^\pi \left[\sum_{k=h}^H R(s_k, a_k) \mid (s_h, a_h) = (s, a) \right]$$

$$V_h^\pi(s) \triangleq \mathbb{E}^\pi \left[\sum_{k=h}^H R(s_k, a_k) \mid s_h = s \right]$$

Relations (Bellman Equations)

$$Q_h^\pi(s, a) = R(s, a) + \sum_{s'} P(s'|s, a) V_{h+1}^\pi(s')$$

$$V_h^\pi(s) = \sum_a \pi_h(a|s) Q_h^\pi(s, a)$$

Calculation (VI)

Calculate

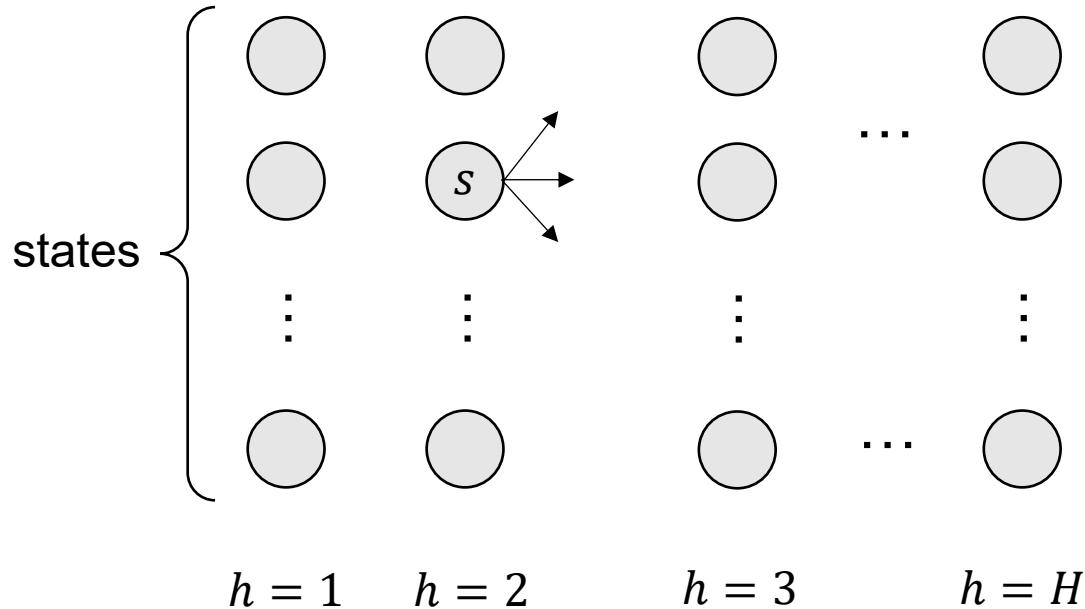
$Q_h^\pi(s, a), V_h^\pi(s) \forall s, a$
from $h = H$ to $h = 1$



Based on Dynamic Programming

$V_1^*(s)$

Value Iteration for Policy Optimization



- ✓ State transition: $P(s'|s, a)$
- ✓ Reward: $R(s, a)$

$$Q_h^*(s, a) = \max_{\pi \in \text{Markov Policy}} \mathbb{E}^\pi \left[\sum_{k=h}^H R(s_k, a_k) \mid (s_h, a_h) = (s, a) \right]$$

$$\rightarrow V_h^*(s) = \max_{\pi \in \text{Markov Policy}} \mathbb{E}^\pi \left[\sum_{k=h}^H R(s_k, a_k) \mid s_h = s \right]$$

Backward induction:

$$V_{H+1}^*(s) = 0 \quad \forall s$$

$\pi(a|s)$: distributions over actions
 $\pi(s)$: action

For $h = H, \dots, 1$: for all s, a

$$Q_h^*(s, a) = R(s, a) + \sum_{s'} P(s'|s, a) V_{h+1}^*(s')$$

Expected optimal total reward from step $h + 1$

$$V_h^*(s) = \max_a Q_h^*(s, a)$$

$$\pi_h^*(s) = \operatorname{argmax}_a Q_h^*(s, a)$$

Exercise

✓

s	a	s'	$P(s' s, a)$	$R(s, a)$
	Slow		1.0	+1
	Fast		0.5	+2
	Fast		0.5	+2
	Slow		0.5	+1
	Slow		0.5	+1
	Fast		1.0	-10
	(end)		1.0	0

$$Q_3^*(s, a)$$

$$Q_3^*(\text{cool, slow}) = 1$$

$$Q_3^*(\text{cool, fast}) = 2$$

$$Q_3^*(\text{warm, slow}) = 1$$

$$Q_3^*(\text{warm, fast}) = -10$$

$$V_3^*(s)$$

$$\underline{V_3^*(\text{cool})} \approx 2 \quad \pi_3^*(\text{cool}) = \text{fast}$$

$$V_3^*(\text{warm}) = 1 \quad \pi_3^*(\text{warm}) = \text{slow}$$

$$Q_2^*(s, a) = R(s, a) + \sum_s P(s'|s, a) V_3^*(s')$$

$$Q_2^*(\text{cool, slow}) = 1 + V_3^*(\text{cool}) = 1+2=3$$

$$Q_2^*(\text{cool, fast}) = 2 + \frac{1}{2} V_3^*(\text{cool}) + \frac{1}{2} V_3^*(\text{warm}) \approx 3,5$$

$$Q_2^*(\text{warm, slow}) = 1 + \frac{1}{2} V_3^*(\text{cool}) + \frac{1}{2} V_3^*(\text{warm}) = 2,5$$

$$Q_2^*(\text{warm, fast}) = -10 + \underline{V_3^*(\text{overheat})} = -10$$

$$V_2^*(s)$$

$$V_2^*(\text{cool}) \approx 3,5 \quad \pi_2^*(\text{cool}) = \text{fast}$$

$$V_2^*(\text{warm}) \approx 2,5 \quad \pi_2^*(\text{warm}) = \text{slow}$$

Assume $H = 3$

Bellman Optimality Equation

Q_h^* : optimal state-action value functions
 V_h^* : optimal state value functions
or “optimal value functions”

$$Q_h^*(s, a) = R(s, a) + \sum_{s'} P(s'|s, a) V_{h+1}^*(s')$$

✓ $V_h^*(s) = \max_a Q_h^*(s, a)$

or

$$Q_h^*(s, a) = R(s, a) + \sum_{s'} P(s'|s, a) \left(\max_{a'} Q_{h+1}^*(s', a') \right)$$

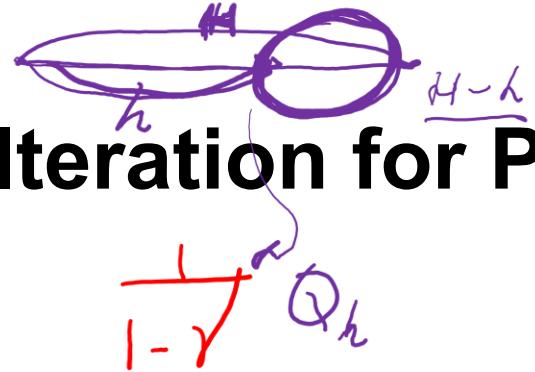
or

$$V_h^*(s) = \max_a \left(R(s, a) + \sum_{s'} P(s'|s, a) V_{h+1}^*(s') \right)$$

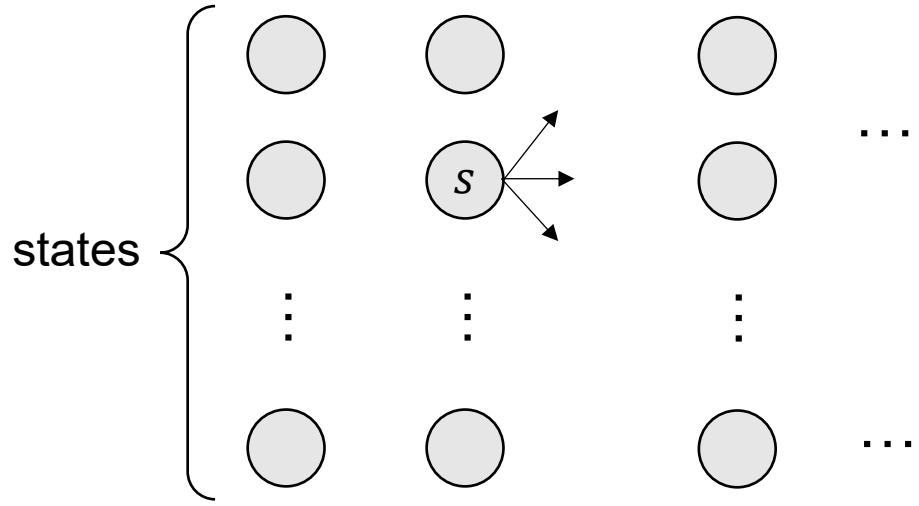
$$\pi_h^*(s) = \operatorname{argmax}_a Q_h^*(s, a)$$

Value Iteration

(Variable-Horizon + Discounted Reward)



Value Iteration for Policy Evaluation

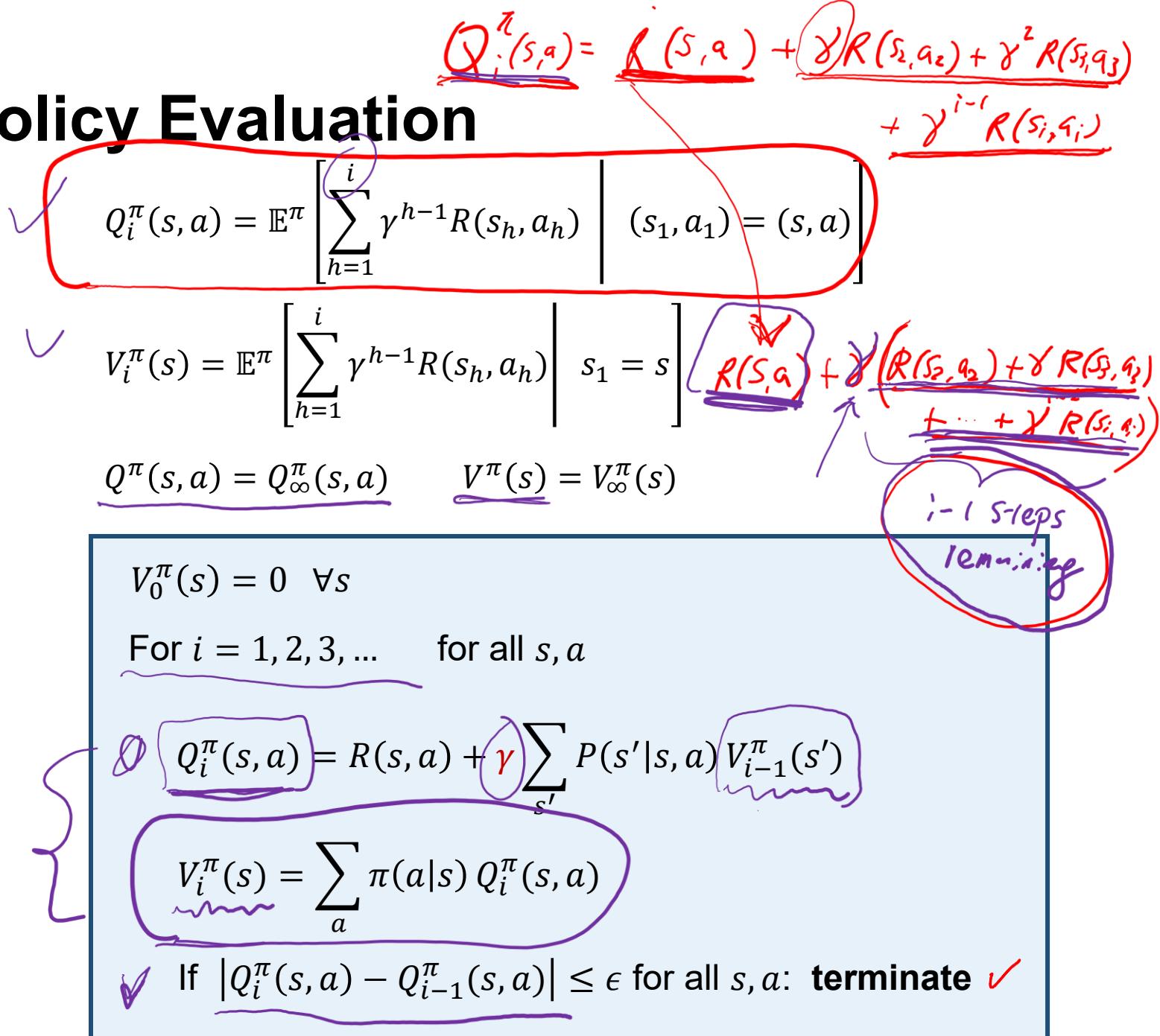


$h = 1 \quad h = 2 \quad h = 3$

weight 1 γ γ^2

State transition: $P(s'|s, a)$

Reward: $R(s, a)$



Bellman Equation

$$Q^\pi(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^\pi(s')$$

$$V^\pi(s) = \sum_a \pi(a|s) Q^\pi(s, a)$$

or

$$Q^\pi(s, a) = R(s, a) + \gamma \sum_{s', a'} P(s'|s, a) \pi(a'|s') Q^\pi(s', a')$$

or

$$V^\pi(s) = \sum_a \pi(a|s) \left(R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^\pi(s') \right)$$

The Meaning of Bellman Equations

Definitions

$$Q^\pi(s, a) = \mathbb{E}^\pi \left[\sum_{h=1}^{\infty} \gamma^{h-1} R(s_h, a_h) \mid (s_1, a_1) = (s, a) \right]$$

$$V^\pi(s) = \mathbb{E}^\pi \left[\sum_{h=1}^{\infty} \gamma^{h-1} R(s_h, a_h) \mid s_1 = s \right]$$

Relations (Bellman Equations)

$$Q^\pi(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^\pi(s')$$

$$V^\pi(s) = \sum_a \pi(a|s) Q^\pi(s, a)$$

Calculation (VI)

Calculate
 $Q_i^\pi(s, a), V_i^\pi(s) \forall s, a$
for $i = 1, 2, \dots$
until terminated

The Quality of $Q_i^\pi(s, a)$ when VI Terminates

Unanswered questions:

1. Will VI (for policy evaluation) always terminate?

2. At termination, we know $\max_{s,a} |Q_i^\pi(s, a) - Q_{i-1}^\pi(s, a)| \leq \epsilon$, $\xrightarrow{10^{-6}}$ Bellman Error (Q_i^π) $\leq \epsilon$
but our goal is to approximate $Q^\pi(s, a)$.

What can we say about $\max_{s,a} |Q_i^\pi(s, a) - Q^\pi(s, a)|$? $\xrightarrow{\text{ValueError } (Q_i^\pi)} \text{ValueError } (Q^\pi) \leq \frac{\epsilon}{1-\gamma} = \frac{10^{-6}}{0.99} = 10^{-4}$

The Quality of $Q_i^\pi(s, a)$ when VI Terminates

$f(s, a)$

Let $f: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ be **any** function. Define

If $f = Q^\pi \Rightarrow \text{BellmanError}(f) = 0$

$$\text{BellmanError}(f) = \max_{s, a} \left| f(s, a) - \left(R(s, a) + \gamma \sum_{s', a'} P(s'|s, a) \pi(a'|s') f(s', a') \right) \right|$$

$$\text{ValueError}(f) = \max_{s, a} |f(s, a) - Q^\pi(s, a)|$$

Theorem

$$\text{ValueError}(f) \leq \frac{\text{BellmanError}(f)}{1 - \gamma}$$

With this theorem, we can argue the quality of $Q_i^\pi(s, a)$ when VI terminates through the following:

1. Prove that when VI terminates, $\text{BellmanError}(Q_i^\pi) \leq \epsilon$
2. Using the theorem, we get $\text{ValueError}(Q_i^\pi) \leq \frac{\epsilon}{1 - \gamma}$

1. When VI terminates. (return Q_i^π), $\text{BellmanError}(Q_i^\pi) \leq \epsilon$

Proof. By the definition of Q_{i+1}^π ,

$$\left\{ \begin{array}{l} Q_{i+1}^\pi(s, a) = \underbrace{R(s, a) + \gamma \sum_{s'} \sum_{a'} P(s'|s, a) \pi(a'|s') Q_i^\pi(s', a')} \\ \text{When terminates, } |Q_i^\pi(s, a) - \underline{Q_{i+1}^\pi(s, a)}| \leq \epsilon \quad \forall s, a \end{array} \right.$$

$$\Rightarrow \left| Q_i^\pi(s, a) - \left(R(s, a) + \gamma \sum_{s'} \sum_{a'} P(s'|s, a) \pi(a'|s') Q_i^\pi(s', a') \right) \right| \leq \epsilon$$

$$\Rightarrow \text{BellmanErr}(Q_i^\pi) \leq \epsilon$$

$$2. \quad \text{Value Err}(f) \leq \frac{\text{Bellman Err}(f)}{1-\gamma}$$

Proof. $\text{Bellman Err}(f) \stackrel{?}{=} \varepsilon$

$$\Rightarrow |f(s,a) - (R(s,a) + \gamma \sum_{s',a'} P(s'|s,a) \pi(a'|s') f(s',a'))| \leq \varepsilon \quad (1)$$

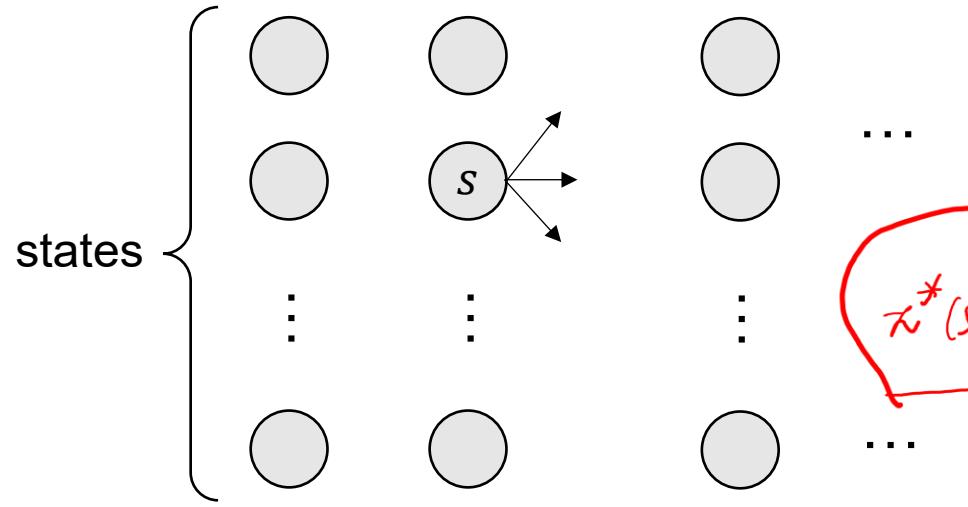
We know by BE.

$$Q^z(s,a) = R(s,a) + \gamma \sum_{s',a'} P(s'|s,a) \pi(a'|s') Q^z(s',a')$$

$$(1) \Rightarrow \underbrace{f(s,a)}_{\geq} \leq R(s,a) + \gamma \sum_{s',a'} P(s'|s,a) \pi(a'|s') f(s',a') + \varepsilon$$

$$\begin{aligned} \cancel{f(s,a)} - Q^z(s,a) &\leq \gamma \sum_{s',a'} P(s'|s,a) \pi(a'|s') (f(s',a') - Q^z(s',a')) + \varepsilon \\ &\leq \gamma \max_{s',a'} (f(s',a') - Q^z(s',a')) + \varepsilon \\ \max_{s,a} f(s,a) - Q^z(s,a) &\leq \gamma \max_{s',a'} (f(s',a') - Q^z(s',a')) + \varepsilon \leq \frac{\varepsilon}{1-\gamma} \\ &\geq -\frac{\varepsilon}{1-\gamma} \end{aligned}$$

Value Iteration for Policy Optimization



$h = 1 \quad h = 2 \quad h = 3$

weight

1

γ

γ^2

State transition: $P(s'|s, a)$

Reward: $R(s, a)$

$$\pi^*(s) = \arg \max_a Q_i^*(s, a)$$

$$Q_i^*(s, a) = \max_{\pi} \mathbb{E}^{\pi} \left[\sum_{h=1}^i \gamma^{h-1} R(s_h, a_h) \mid (s_0, a_0) = (s, a) \right]$$

$$V_i^*(s) = \max_{\pi} \mathbb{E}^{\pi} \left[\sum_{h=1}^i \gamma^{h-1} R(s_h, a_h) \mid s_0 = s \right]$$

$$\pi^*(s) = \arg \max_a Q_i^*(s, a)$$

$$Q^*(s, a) = Q_\infty^*(s, a) \quad V^*(s) = V_\infty^*(s)$$

$$V_0^*(s) = 0 \quad \forall s$$

For $i = 1, 2, 3, \dots$ for all s, a

$$Q_i^*(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) V_{i-1}^*(s')$$

$$V_i^*(s) = \max_a Q_i^*(s, a)$$

If $|Q_i^*(s, a) - Q_{i-1}^*(s, a)| \leq \epsilon$ for all s, a : **terminate**

Bellman Optimality Equation

$$\pi^*(s) = \operatorname{argmax}_a Q^*(s, a)$$

$$Q^*(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^*(s')$$

$$V^*(s) = \max_a Q^*(s, a)$$

or

$$Q^*(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) \max_{a'} Q^*(s', a')$$

or

$$V^*(s) = \max_a \left(R(s, a) + \gamma \sum_{s'} P(s'|s, a) V^*(s') \right)$$

The Solution Quality when VI Terminates

Unanswered questions:

- ✓ 1. Will VI (for policy optimization) always terminate? ✓
- ✓ 2. At termination, we know $\max_{s,a} |Q_i^*(s, a) - Q_{i-1}^*(s, a)| \leq \epsilon$, $\Rightarrow \text{Bellman Error}(Q_i^*) \leq \epsilon$
What can we say about $\max_{s,a} |Q_i^*(s, a) - Q^*(s, a)|$? $\leq \text{Value Err}(Q_i^*) \leq \frac{\epsilon}{1-\gamma}$
- 3. And what can we say about the **performance of the greedy policy $\hat{\pi}$**

defined as $\hat{\pi}(a|s) = \mathbb{I}[a = \operatorname{argmax}_{a'} Q_i^*(s, a')]$? or simply $\hat{\pi}(s) = \operatorname{argmax}_{a'} Q_i^*(s, a')$

$$\underline{V^*(s) - V^{\hat{\pi}}(s)} \leq ?$$

$\hat{\pi}$

The Solution Quality when VI Terminates (1/2)

Let $f: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ be **any** function. Define

$$\text{BellmanError}(f) = \max_{s,a} \left| f(s,a) - \left(R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} f(s',a') \right) \right|$$

$$\text{ValueError}(f) = \max_{s,a} |f(s,a) - Q^*(s,a)|$$

Theorem

$$\text{ValueError}(f) \leq \frac{\text{BellmanError}(f)}{1 - \gamma}$$

The Solution Quality when VI Terminates (2/2)

Let $f: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ be **any** function. Define

$$\pi_f(s) = \operatorname{argmax}_a f(s, a)$$

Theorem

Proven in HW3

$$V^*(\rho) - V^{\pi_f}(\rho) \leq \frac{2}{1-\gamma} \operatorname{ValueError}(f)$$

R : true reward

\hat{R} : estimated reward

$$\hat{a}^* = \operatorname{argmax}_a R(a)$$

$$\hat{a} = \operatorname{argmax}_a \hat{R}(a)$$

$$R(a^*) - R(\hat{a})$$

$$= \frac{\hat{R}(a^*) - \hat{R}(\hat{a})}{\leq 0} + \underbrace{R(a^*) - \hat{R}(a^*)}_{\text{curly brace}} + \underbrace{\hat{R}(\hat{a}) - R(\hat{a})}_{\text{curly brace}}$$

$$\leq 2 \max_a |R(a) - \hat{R}(a)|$$

$$\left| \begin{array}{c} Q^* \\ f \end{array} \right|$$

Combining the two theorems, we know that when VI (for policy optimization) terminates,

$$V^*(\rho) - V^{\hat{\pi}}(\rho) \leq \frac{2}{1-\gamma} \operatorname{ValueError}(Q_i^*) \leq \frac{2}{(1-\gamma)^2} \operatorname{BellmanError}(Q_i^*) \leq \frac{2\epsilon}{(1-\gamma)^2}$$

where $\hat{\pi}(s) = \operatorname{argmax}_a Q_i^*(s, a)$

Remark

For the discounted setting, value iteration converges to the same value regardless of the initial value

$$V_0^\pi(s) = \text{arbitrary} \quad \forall s$$

For $i = 1, 2, 3, \dots$ for all s, a

$$Q_i^\pi(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) V_{i-1}^\pi(s')$$

$$V_i^\pi(s) = \sum_a \pi(a|s) Q_i^\pi(s, a)$$

If $|Q_i^\pi(s, a) - Q_{i-1}^\pi(s, a)| \leq \epsilon$ for all s, a : **terminate**

$$V_0^*(s) = \text{arbitrary} \quad \forall s$$

For $i = 1, 2, 3, \dots$ for all s, a

$$Q_i^*(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a) V_{i-1}^*(s')$$

$$V_i^*(s) = \max_a Q_i^*(s, a)$$

If $|Q_i^*(s, a) - Q_{i-1}^*(s, a)| \leq \epsilon$ for all s, a : **terminate**

$$\pi^*(x) = \arg \max_a R(x, a)$$

$$\hat{\pi}(x) = \arg \max_a f(x, a)$$

$$f(x, a) \approx \underline{R(x, a)}$$

Policy Iteration

MPP

$$f(s, a) \approx Q^*(s, a)$$

$$\pi^*(s) = \arg \max_a Q^*(s, a)$$

$$\hat{\pi}(s) = \arg \max_a f(s, a)$$

Policy Iteration

$$Q^{\pi_i}$$

Policy Iteration

For $i = 1, 2, \dots$

$$\forall s, \quad \pi_i(s) \leftarrow \operatorname{argmax}_a Q^{\pi_{i-1}}(s, a)$$

*inner
for-loop* *Calculate $Q^{\pi_i}(s, a)$ by s, a* *(VI for policy evaluation)*

Theorem (monotonic improvement). Policy Iteration ensures

$$\forall s, a, \quad Q^{\pi_i}(s, a) \geq Q^{\pi_{i-1}}(s, a)$$

When converged (i.e., $\pi_i = \pi_{i-1}$), we have $\pi_i = \pi^*$.

(We will prove this later.)

- Value Iteration
 - For policy evaluation: approximating $Q^\pi(s, a), V^\pi(s)$ for given π
 - For policy optimization: approximating $Q^*(s, a), V^*(s)$
- Policy iteration
 - For policy optimization: approximating π^*

Generalized Policy Iteration

$N = \infty \Rightarrow$ Policy Iteration

$N = 1 \Rightarrow$ Value Iteration for policy optimization

For $i = 1, 2, \dots$

$$\pi_i(s) = \max_a Q_i(s, a) \quad \xleftarrow{\text{Policy update}}$$

$$Q \leftarrow Q_i$$

Repeat for N times:

$$Q(s, a) \leftarrow R(s, a) + \gamma \sum_{s', a'} P(s'|s, a) \pi_i(a'|s') Q(s', a') \quad \xleftarrow{\text{Value update}}$$

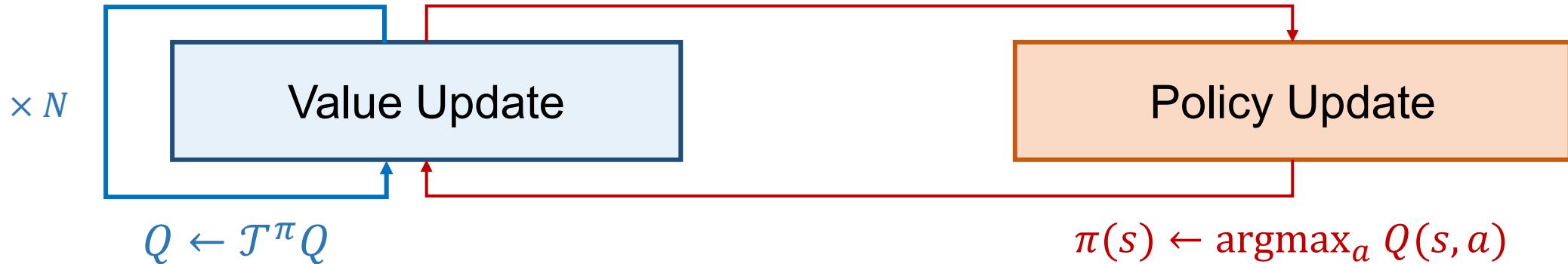
$$Q_{i+1} \leftarrow Q$$

Notice: in value iteration for PO, there may not exist a policy π such that $Q_i = Q^\pi$

In contrast, in policy iteration we have $Q_i = Q^{\pi_{i-1}}$

VI for PO can be viewed as PI with incomplete policy evaluation

Generalized Policy Iteration



$$Q \leftarrow \mathcal{T}^\pi Q \text{ means } Q(s, a) \leftarrow R(s, a) + \gamma \sum_{s', a'} P(s'|s, a) \pi_i(a'|s') Q(s', a') \quad \text{for all } s, a$$

- Provide a unified view for algorithms approximating optimal policy with known P, R
- Provide a unified view for “value-based” and “policy-based” algorithms
- Compared with bandits:
 - The R is replaced by Q (could be Q^π or Q^*) to capture the long-term goal of the learner

Summary

- Value Iteration for Policy Optimization (VI for PO)
 - Is essentially a **dynamic programming** algorithm
 - Finds the value functions of the optimal policy
- Value Iteration for Policy Evaluation (VI for PE)
 - Also a **dynamic programming** algorithm
 - Finds the value functions of the given policy
- Policy Iteration (PI)
 - An iterative policy improvement algorithm
 - Each iteration involves a policy evaluation subtask
- VI for PO and PI can be viewed as special cases of Generalized PI

Performance Difference Lemma

Unanswered Questions

$$V^\pi(\rho) \triangleq \mathbb{E}_{s \sim \rho}[V^\pi(s)]$$

- For an estimation $\hat{Q}(s, a) \approx Q^*(s, a)$ with error, how can we bound

$$V^*(\rho) - V^{\hat{\pi}}(\rho) \quad \text{where } \hat{\pi}(s) = \operatorname{argmax}_a \hat{Q}(s, a)?$$

- How to show that Policy Iteration leads to monotonic policy improvement?
- Also, how are these methods (VI and PI) related to the third challenge of online RL: credit assignment?

Recall: Regret

$$\text{Regret} = \max_{\pi^*} \mathbb{E}^{\pi^*} \left[\sum_{t=1}^T \sum_{h=1}^{\tilde{\tau}_t} R(\tilde{s}_{t,h}, \pi^*(\tilde{s}_{t,h})) \right] - \sum_{t=1}^T \sum_{h=1}^{\tau_t} R(s_{t,h}, a_{t,h})$$

$$\mathbb{E}[\text{Regret}] = \mathbb{E} \left[\sum_{t=1}^T (V_1^*(s_{t,1}) - V_1^{\pi_t}(s_{t,1})) \right]$$

$$= \mathbb{E} \left[\sum_{t=1}^T (V_1^*(\rho) - V_1^{\pi_t}(\rho)) \right] \quad V_1^\pi(\rho) \triangleq \mathbb{E}_{s \sim \rho}[V_1^\pi(s)]$$

Performance Difference Lemma

Proven in HW3

For any two stationary policies π' and π in the discounted setting,

$$\begin{aligned}\mathbb{E}_{s \sim \rho} [V^{\pi'}(s)] - \mathbb{E}_{s \sim \rho} [V^{\pi}(s)] &= \sum_{s,a} d_{\rho}^{\pi'}(s) (\pi'(a|s) - \pi(a|s)) Q^{\pi}(s, a) \\ &= \sum_{s,a} d_{\rho}^{\pi'}(s, a) (Q^{\pi}(s, a) - V^{\pi}(s))\end{aligned}$$

$$d_{\rho}^{\pi}(s) \triangleq \mathbb{E}^{\pi} \left[\sum_{h=1}^{\infty} \gamma^{h-1} \mathbb{I}\{s_h = s\} \mid s_1 \sim \rho \right] \quad \text{Discounted occupancy measure on state } s$$

$$d_{\rho}^{\pi}(s, a) \triangleq \mathbb{E}^{\pi} \left[\sum_{h=1}^{\infty} \gamma^{h-1} \mathbb{I}\{(s_h, a_h) = (s, a)\} \mid s_1 \sim \rho \right]$$

Performance Difference Lemma

We can also swap the roles of π' and π and apply the same lemma

$$\mathbb{E}_{s \sim \rho}[V^\pi(s)] - \mathbb{E}_{s \sim \rho}[V^{\pi'}(s)] = \sum_{s,a} d_\rho^\pi(s) (\pi(a|s) - \pi'(a|s)) Q^{\pi'}(s, a)$$

$$\times (-1) \Rightarrow \mathbb{E}_{s \sim \rho}[V^{\pi'}(s)] - \mathbb{E}_{s \sim \rho}[V^\pi(s)] = \sum_{s,a} d_\rho^{\pi'}(s) (\pi'(a|s) - \pi(a|s)) Q^\pi(s, a)$$

||

Original version:

$$\mathbb{E}_{s \sim \rho}[V^{\pi'}(s)] - \mathbb{E}_{s \sim \rho}[V^\pi(s)] = \sum_{s,a} d_\rho^{\pi'}(s) (\pi'(a|s) - \pi(a|s)) Q^\pi(s, a)$$

Performance Difference Lemma (Fixed-Horizon)

For any two Markov policies π' and π in the fixed-horizon setting,

$$\begin{aligned}\mathbb{E}_{s_1 \sim \rho} [V_1^{\pi'}(s_1)] - \mathbb{E}_{s_1 \sim \rho} [V_1^{\pi}(s_1)] &= \sum_{h=1}^H \sum_{s,a} d_{\rho,h}^{\pi'}(s) (\pi'_h(a|s) - \pi_h(a|s)) Q_h^{\pi}(s, a) \\ &= \sum_{h=1}^H \sum_{s,a} d_{\rho,h}^{\pi'}(s, a) (Q_h^{\pi}(s, a) - V_h^{\pi}(s))\end{aligned}$$

$$d_{\rho,h}^{\pi}(s) \triangleq \mathbb{E}^{\pi}[\mathbb{I}\{s_h = s\} \mid s_1 \sim \rho] = \mathbb{P}^{\pi}(s_h = s \mid s_1 \sim \rho)$$

$$d_{\rho,h}^{\pi}(s, a) \triangleq \mathbb{E}^{\pi}[\mathbb{I}\{(s_h, a_h) = (s, a)\} \mid s_1 \sim \rho] = \mathbb{P}^{\pi}((s_h, a_h) = (s, a) \mid s_1 \sim \rho)$$

The Meaning of Performance Difference Lemma

It tells us how **credit** are assigned to each state/step

The sub-optimality of a policy π :

$$\mathbb{E}_{s \sim \rho}[V^*(s)] - \mathbb{E}_{s \sim \rho}[V^\pi(s)]$$

If π is highly sub-optimal, then we can always find

- 1) An (s, a) -pair on the path of π such that $V^*(s) - Q^*(s, a)$ is positive and large
- 2) An (s, a) -pair on the path of π^* such that $Q^\pi(s, a) - V^\pi(s)$ is positive and large

$$= \sum_{s,a} d_\rho^\pi(s) (\pi^*(a|s) - \pi(a|s)) Q^*(s, a)$$

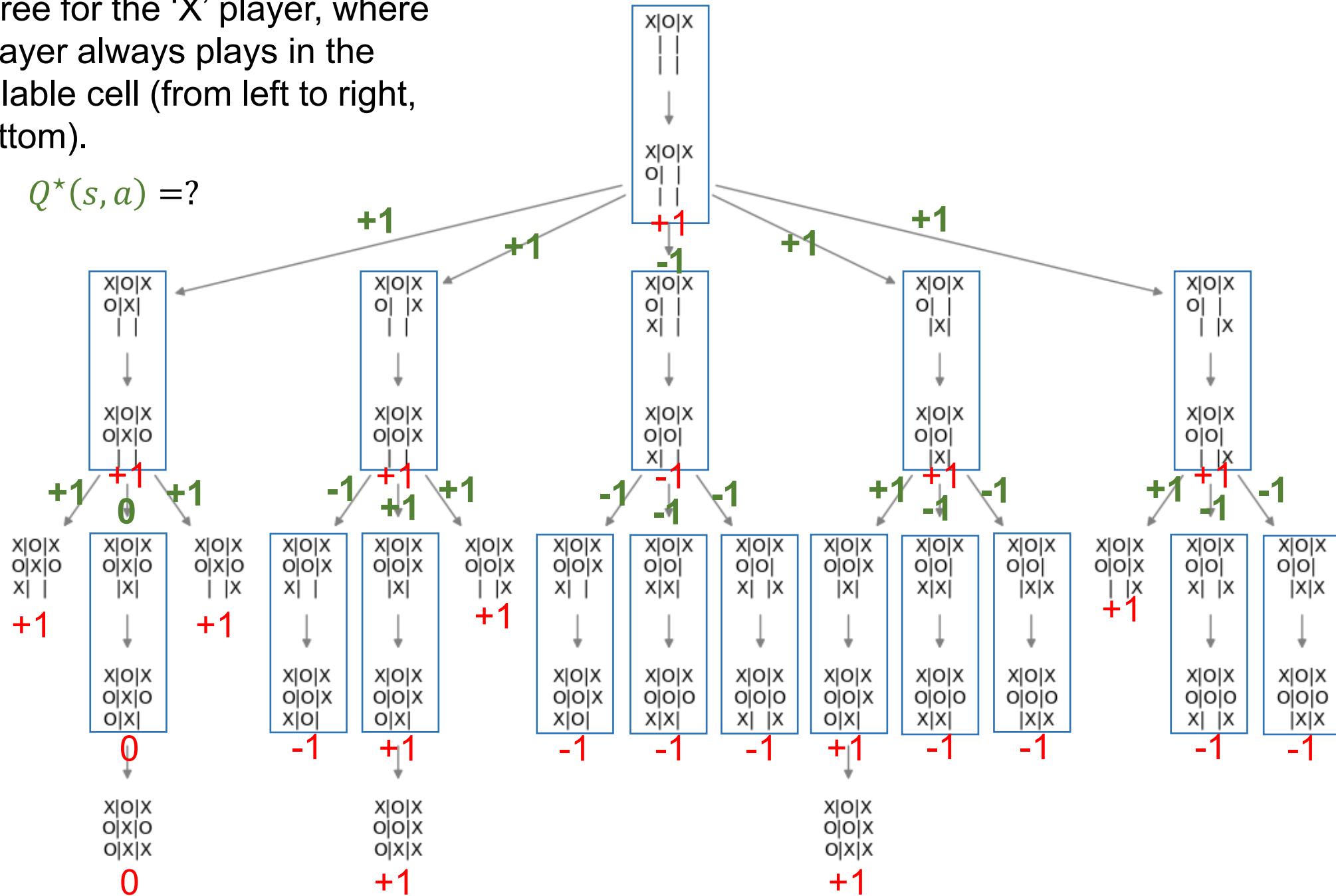
$$= \sum_{s,a} d_\rho^\pi(s, a) (V^*(s) - Q^*(s, a))$$

$$= \sum_{s,a} d_\rho^{\pi^*}(s) (\pi^*(a|s) - \pi(a|s)) Q^\pi(s, a)$$

$$= \sum_{s,a} d_\rho^{\pi^*}(s, a) (Q^\pi(s, a) - V^\pi(s))$$

A game tree for the ‘X’ player, where the ‘O’ player always plays in the **first** available cell (from left to right, top to bottom).

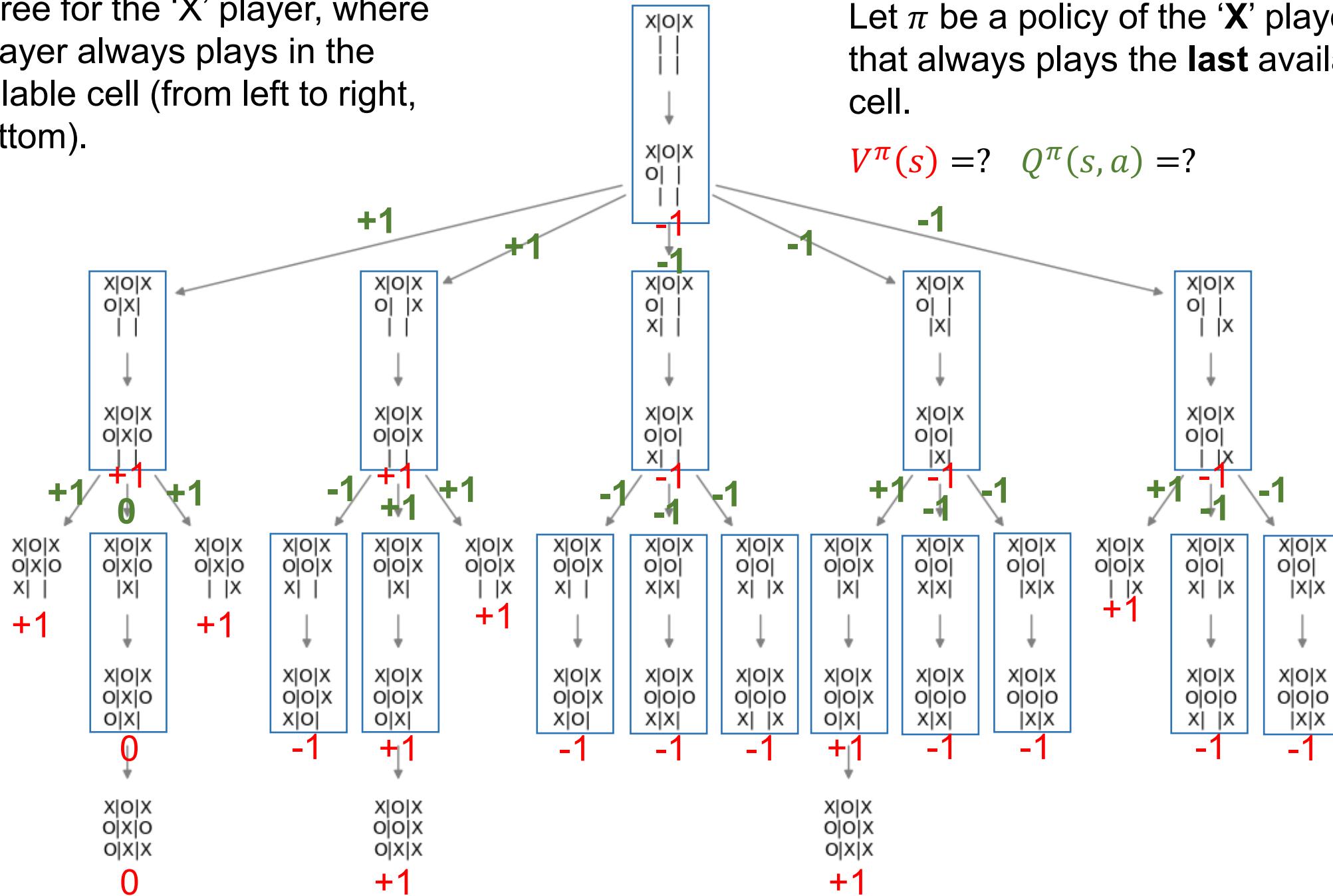
$$V^*(s) =? \quad Q^*(s, a) =?$$



A game tree for the ‘X’ player, where the ‘O’ player always plays in the **first** available cell (from left to right, top to bottom).

Let π be a policy of the 'X' player that always plays the **last** available cell.

$$V^\pi(s) = ? \quad Q^\pi(s, a) = ?$$



Unanswered Question 1

Suboptimality $\leq (1 - \gamma)^{-1}$ Value Error

Let $f: \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ be **any** function

If

$$|f(s, a) - Q^*(s, a)| \leq \epsilon \quad \forall s, a$$

then

$$V^*(s) - V^{\pi_f}(s) \leq \frac{2\epsilon}{1 - \gamma} \quad \forall s$$

where $\pi_f(s) = \operatorname{argmax}_a f(s, a)$

Unanswered Question 2

Policy Iteration ensures

$$\forall s, a, \quad Q^{\pi_i}(s, a) \geq Q^{\pi_{i-1}}(s, a)$$

When converged (i.e., $\pi_i = \pi_{i-1}$), we have $\pi_i = \pi^*$.

$$\pi_i = \pi_{i-1}$$

$$\Rightarrow \pi_i(s) = \operatorname{argmax}_a Q^{\pi_i}(s, a)$$

$$\Rightarrow Q^{\pi_i}(s, a) = R(s, a) + \gamma \sum P(s'|s, a) \pi_i(a'|s') Q^{\pi_i}(s', a') = R(s, a) + \gamma \sum P(s'|s, a) \max_{a'} Q^{\pi_i}(s', a')$$

$\Rightarrow Q^{\pi_i}$ satisfies the Bellman optimality equation

$$\Rightarrow \text{BellmanError}(Q^{\pi_i}) = 0$$

$$\Rightarrow Q^{\pi_i}(s, a) = Q^\star(s, a) \text{ by the "ValueError} \leq \frac{1}{1-\gamma} \text{ BellmanError" lemma}$$

$$\Rightarrow \pi_i(s) = \operatorname{argmax}_a Q^\star(s, a) = \pi^\star(s).$$

Recap: MDP

- Definitions of $Q^\pi(s, a), V^\pi(s), Q^*(s, a), V^*(s)$
- Bellman equations (= the update of dynamic programming)
- Value Iteration to approximate $Q^\pi(s, a)/V^\pi(s)$ or $Q^*(s, a)/V^*(s)$
- Policy Iteration to find π^* --- involving $Q^\pi(s, a)/V^\pi(s)$ approximation
- Unified by Generalized Policy Iteration
- Performance difference lemma to decompose $\mathbb{E}_{s \sim \rho} [V^{\pi'}(s)] - \mathbb{E}_{s \sim \rho} [V^\pi(s)]$
 - Credit assignment
 - $= \sum_{s,a} d_\rho^\pi(s, a) (V^{\pi'}(s) - Q^{\pi'}(s, a))$ (helpful in analyzing VI by letting $\pi' = \pi^*$)
 - $= \sum_{s,a} d_\rho^{\pi'}(s, a) (Q^\pi(s, a) - V^\pi(s))$ (helpful in analyzing PI by letting $\pi' = \pi_{i+1}$)

Next

- Our discussion indicates there are two potential ways to find optimal policy
 - Value-Iteration-based: approximate $\hat{Q}(s, a) \approx Q^*(s, a)$ and let $\hat{\pi}(s) = \operatorname{argmax}_a \hat{Q}(s, a)$
 - Policy-Iteration-based: approximate $\hat{Q}(s, a) \approx Q^\pi(s, a)$ and let $\hat{\pi}(s) = \operatorname{argmax}_a \hat{Q}(s, a)$
- RL algorithms we will discuss:
 - Performing approximate VI or PI using samples