Markov Decision Processes

Chen-Yu Wei

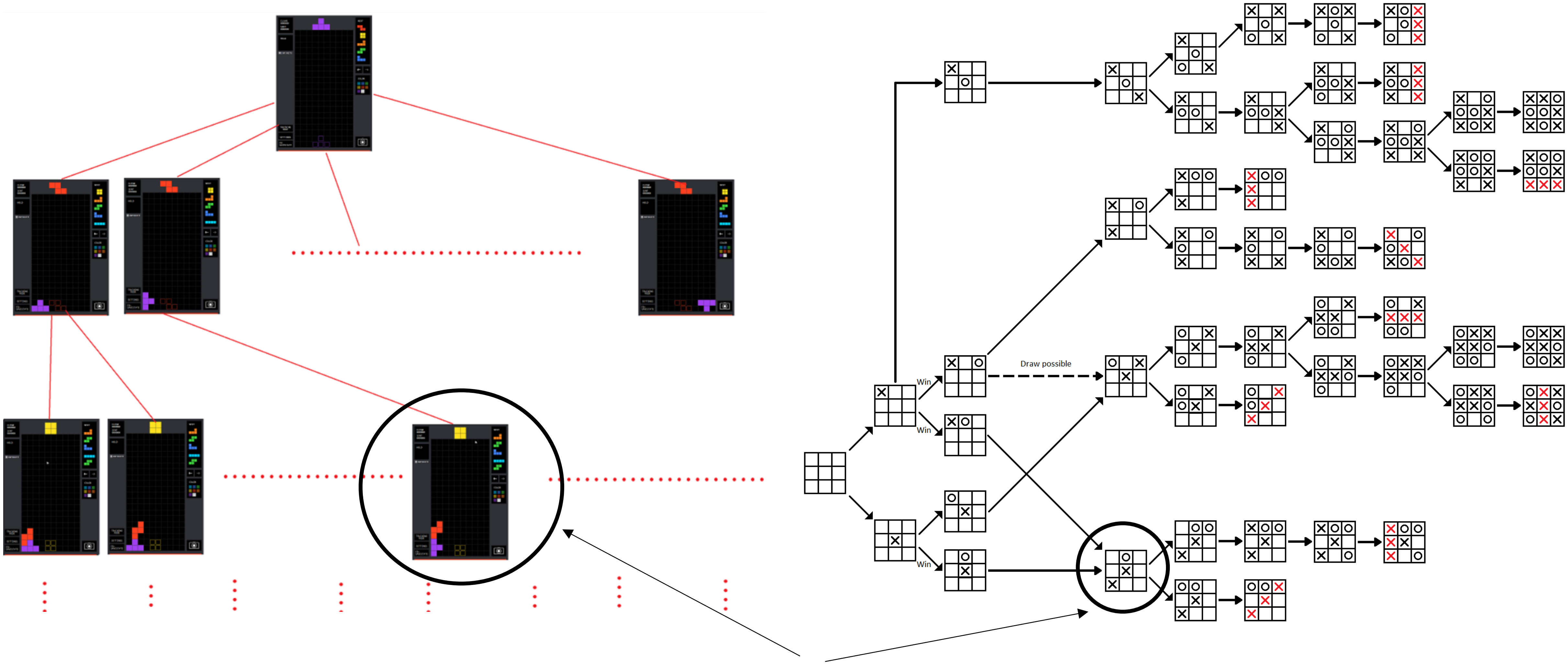
Sequence of Actions

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

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



state

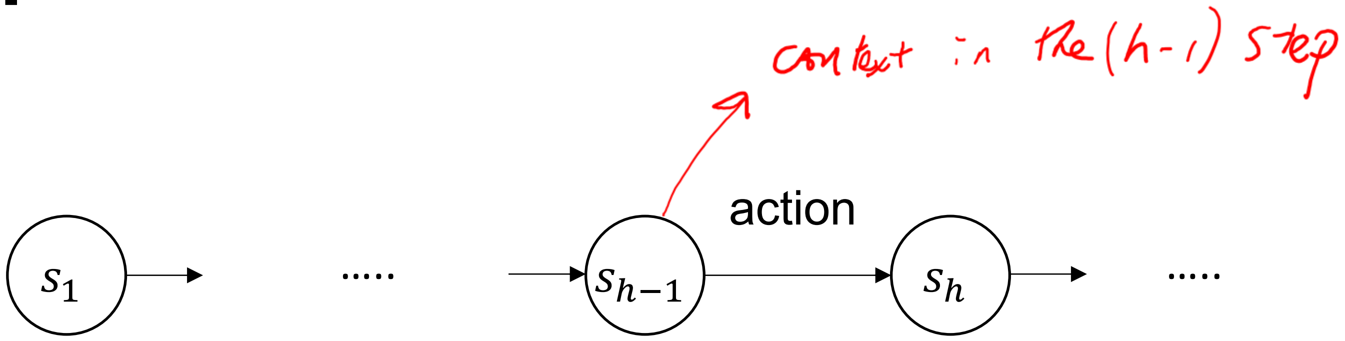
(a summary of the current status in a multi-stage game)

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** $t = 1, 2, \ldots, T$:

$h \leftarrow 1$

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 $s_{t,h+1} \sim P(\cdot \mid s_{t,h}, a_{t,h})$

$h \leftarrow h + 1$

Markov assumption:

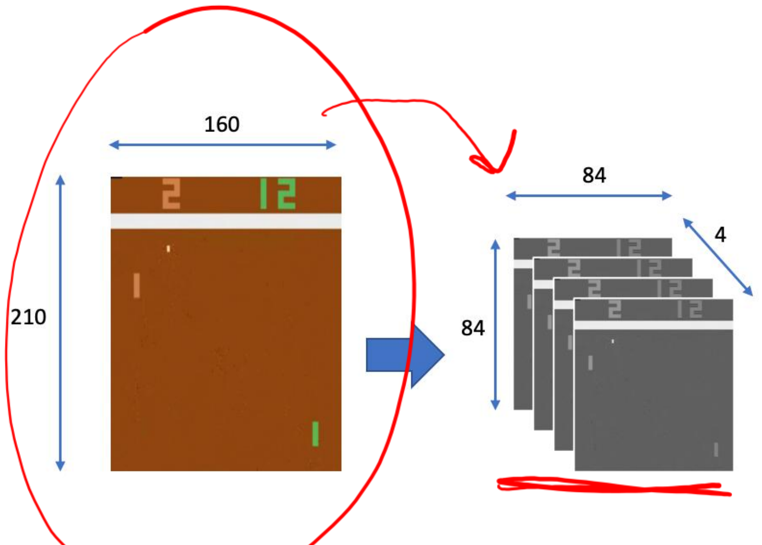
$r_{t,h}$ and $s_{t,h+1}$ are conditionally independent of $(s_{t,1}, a_{t,1}, \ldots, s_{t,h-1}, a_{t,h-1})$ given $s_{t,h}$

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

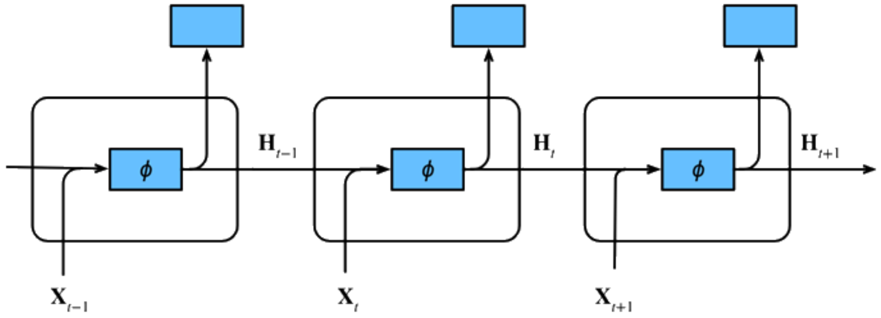
x_t

length of episode

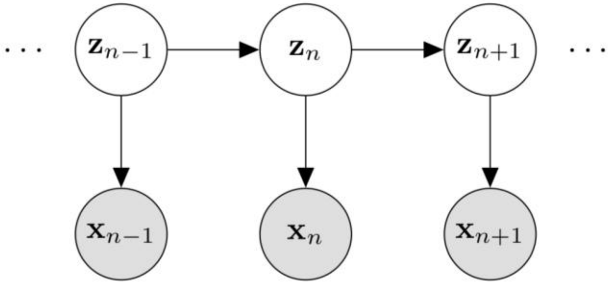
From Observations to States



Stacking recent observations



Recurrent neural network

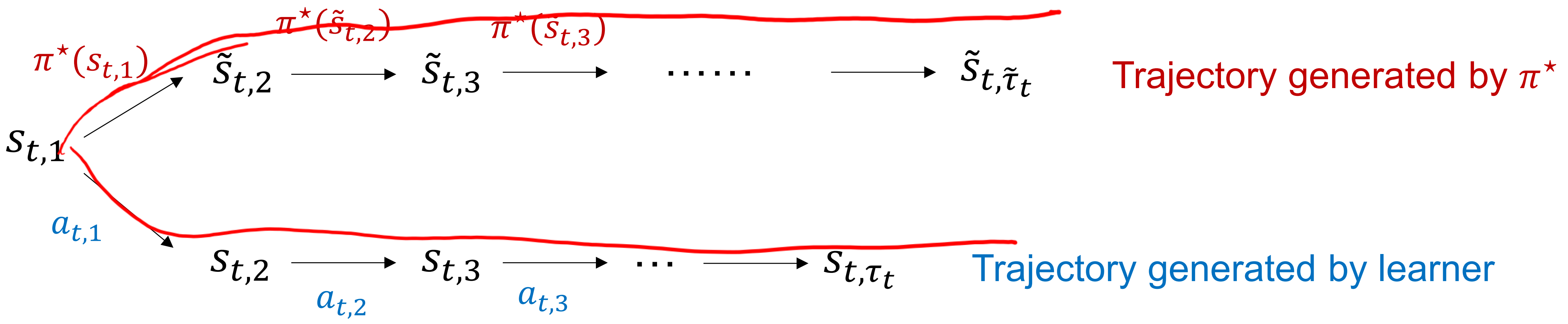


Hidden Markov model

Regret (Episodic Setting)

Policy : mapping from state to action (action distribution)

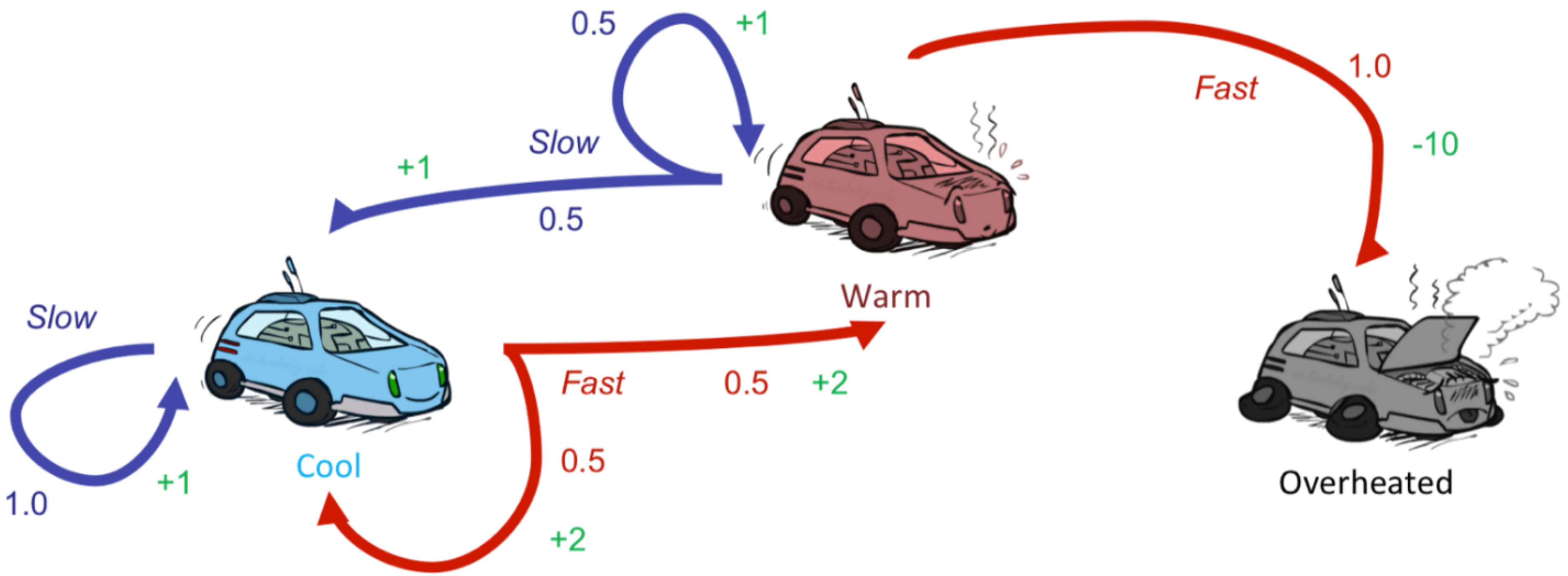
$$\text{Regret} = \underbrace{\max_{\pi^\star} \mathbb{E}^{\pi^\star}\left[\sum_{t=1}^{T}\sum_{h=1}^{\tilde{\tau}_t} R(\tilde{s}_{t,h}, \pi^\star(\tilde{s}_{t,h}))\right]}_{\text{Benchmark}} - \sum_{t=1}^{T}\sum_{h=1}^{\tau_t} R(s_{t,h}, a_{t,h})$$



Example: Racing

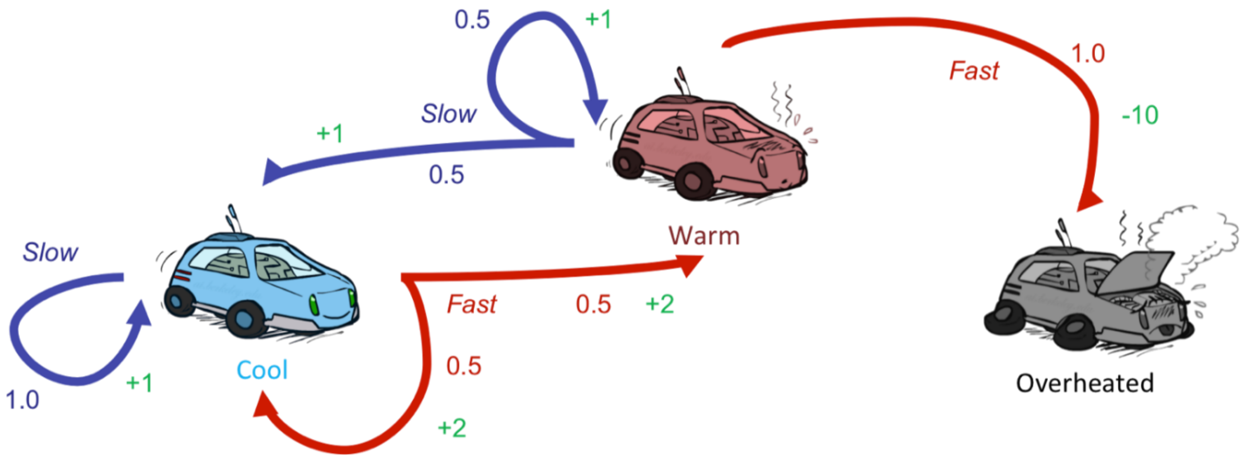
$R(s,a)$
$P(s'|s,a)$

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



Example: Racing

s	a	s'	$P(s' \mid s, a)$	$R(s, a)$
	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 $s_{h+1} \sim P(\cdot \mid s_h, a_h)$

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 $s_{h+1} \sim P(\cdot \mid s_h, a_h)$

$h \leftarrow h + 1$

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

Discounted Total Reward: $$\sum_{h=1}^{\tau} \gamma^{h-1} r_h$$ $\gamma \in [0,1)$: discount factor

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 → 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

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

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

Stationary Policy

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

← For **infinite-horizon/goal-oriented** settings, there exists an optimal policy in this class

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

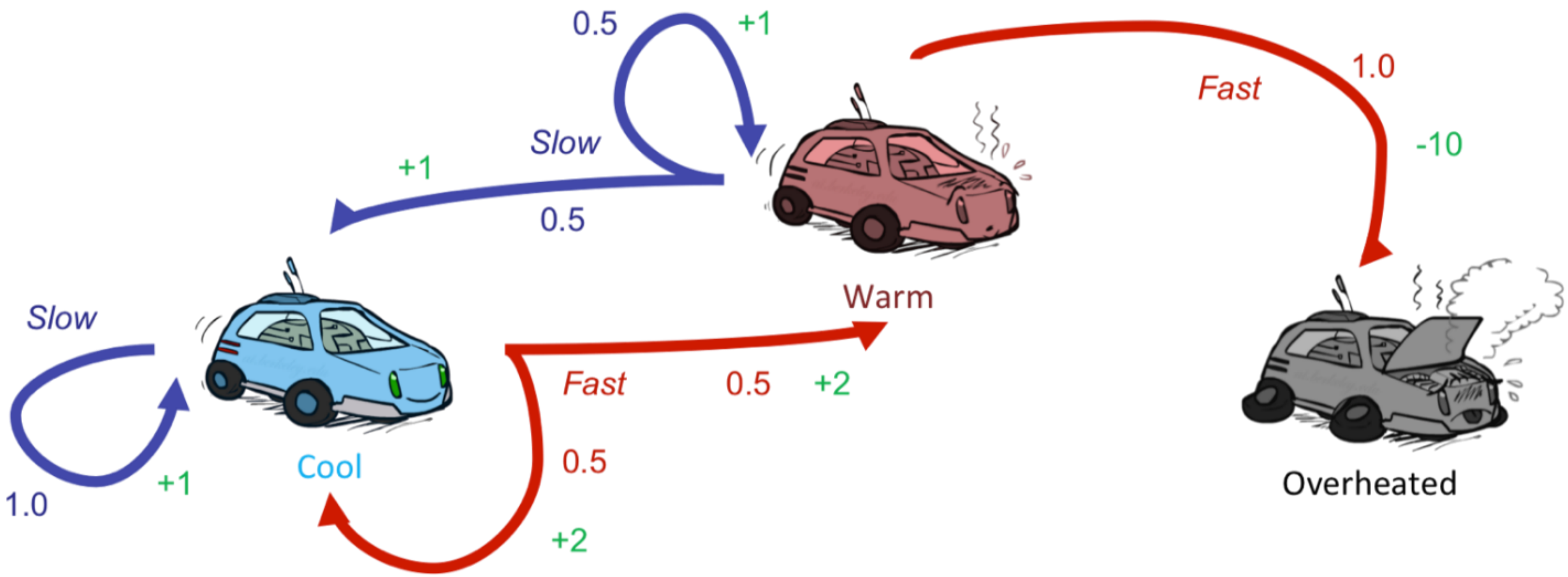
$\pi_h(\text{Slow} \mid \text{Cool})$

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

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

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

$\forall h$



Value Iteration

(Fixed-Horizon + Total-Reward)

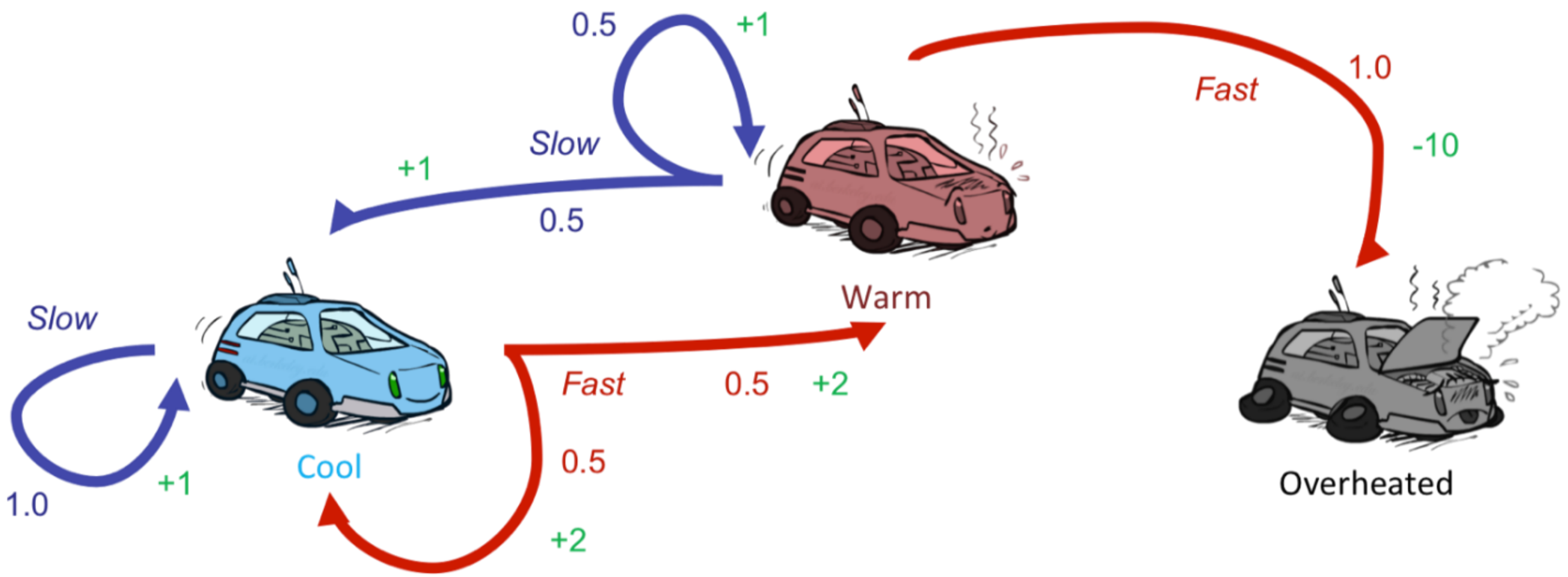
Two Tasks

- **Policy Evaluation:** Calculate the expected total reward of a given policy

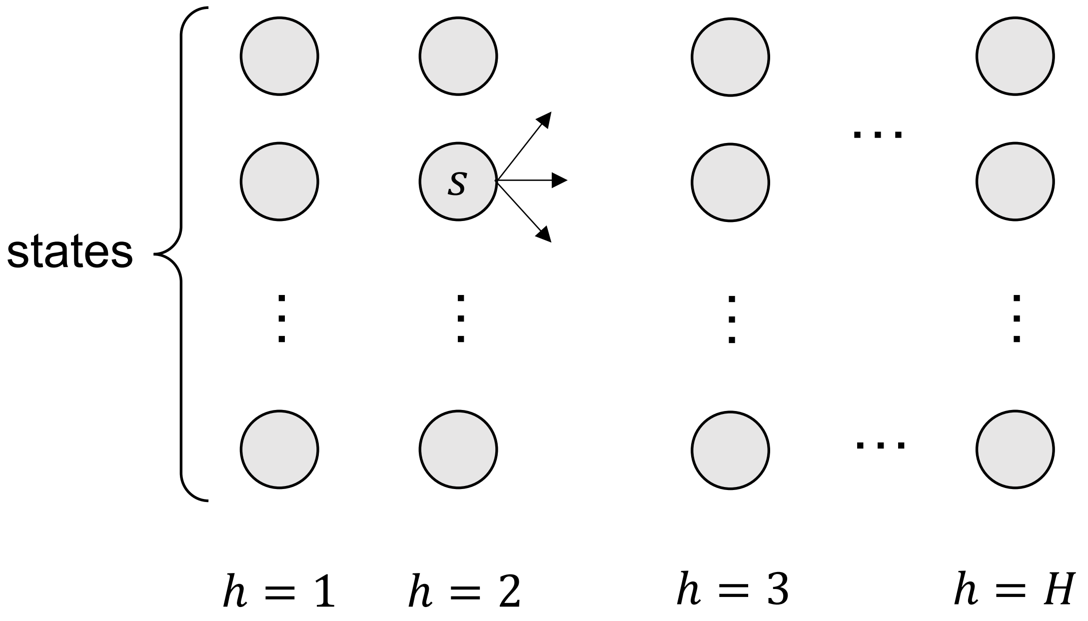
  What is the expected total reward for the policy $\pi(\text{cool}) = \text{fast}, \pi(\text{warm}) = \text{slow}$?

  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: $R(s,a)$

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

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

Backward induction:

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

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

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

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

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

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

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

or

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

The Meaning of Bellman Equations

Definitions

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

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

Relations (Bellman Equations)

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

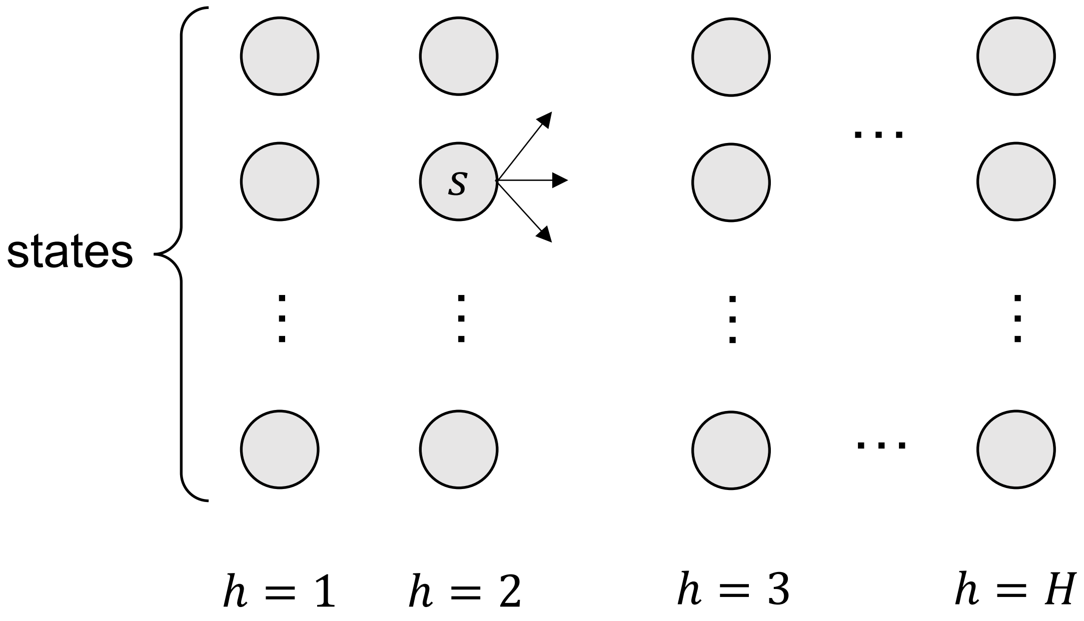
$$V_h^\pi(s) = \sum_{a} \pi_h(a|\, s) Q_h^\pi(s,a)$$

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

Value Iteration for Policy Optimization



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

Reward: $R(s,a)$

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

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

Backward induction:

$V_{H+1}^\star(s) = 0 \qquad \forall s$

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

$$Q_h^\star(s,a) = R(s,a) + \underbrace{\sum_{s'} P(s'|s,a)\, V_{h+1}^\star(s')}_{\text{Expected optimal total reward from step } h+1}$$

Expected optimal total reward from step $h+1$

$$V_h^\star(s) = \max_a Q_h^\star(s,a) \qquad \pi_h^\star(s) = \operatorname{argmax}_a Q_h^\star(s,a)$$

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

Assume $H = 3$

$\boldsymbol{Q_3^\star(s,a)}$

$Q_3^\star(\text{cool}, \text{slow})$

$Q_3^\star(\text{cool}, \text{fast})$

$Q_3^\star(\text{warm}, \text{slow})$

$Q_3^\star(\text{warm}, \text{fast})$

$\boldsymbol{V_3^\star(s)}$

$V_3^\star(\text{cool})$

$V_3^\star(\text{warm})$

$\boldsymbol{Q_2^\star(s,a)}$

$Q_2^\star(\text{cool}, \text{slow})$

$Q_2^\star(\text{cool}, \text{fast})$

$Q_2^\star(\text{warm}, \text{slow})$

$Q_2^\star(\text{warm}, \text{fast})$

$\boldsymbol{V_2^\star(s)}$

$V_2^\star(\text{cool})$

$V_2^\star(\text{warm})$

Bellman Optimality Equation

$Q_h^\star$: optimal state-action value functions
$V_h^\star$: optimal state value functions
or "**optimal value functions**"

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

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

or

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

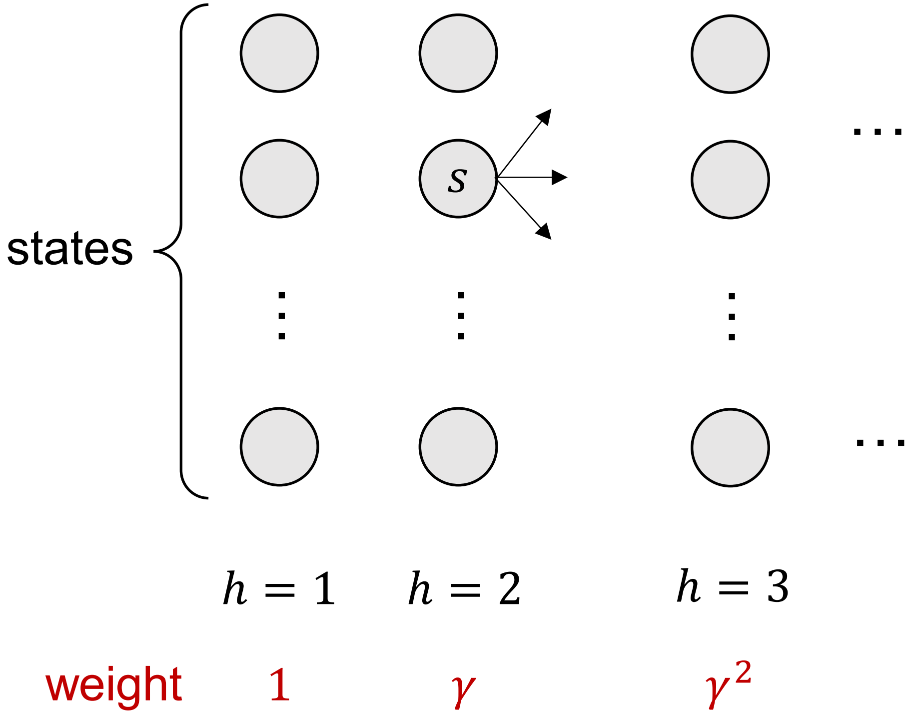
or

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

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

Value Iteration
(Variable-Horizon + Discounted Reward)

Value Iteration for Policy Evaluation



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

Reward: $R(s,a)$

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

$$V_i^\pi(s) = \mathbb{E}^\pi\left[\sum_{h=1}^{i} \gamma^{h-1} R(s_h,a_h) \,\middle|\, s_1 = s\right]$$

$$Q^\pi(s,a) = Q_\infty^\pi(s,a) \qquad V^\pi(s) = V_\infty^\pi(s)$$

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

For $i = 1,2,3,\ldots$ 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 $\left|Q_i^\pi(s,a) - Q_{i-1}^\pi(s,a)\right| \le \epsilon$ for all s,a: **terminate**

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

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

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

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 $i = 1, 2, \ldots$
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 $\max_{s,a} \left| Q_i^\pi(s,a) - Q_{i-1}^\pi(s,a) \right| \leq \epsilon$,

   but our goal is to approximate $Q^\pi(s,a)$.

   What can we say about $\max_{s,a} \left| Q_i^\pi(s,a) - Q^\pi(s,a) \right|$?

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

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

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

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

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:

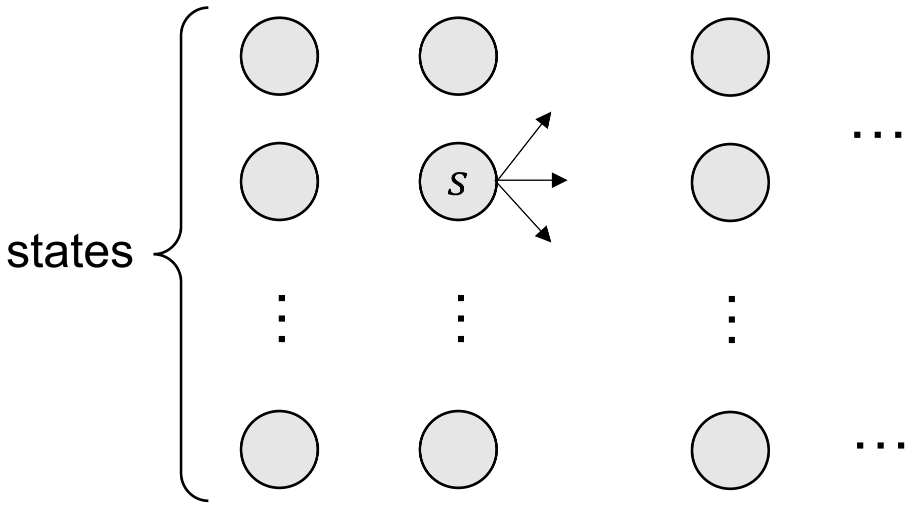
1. Prove that when VI terminates, $\text{BellmanError}(Q_i^\pi)\leq\epsilon$
2. Using the theorem, we get $\text{ValueError}(Q_i^\pi)\leq\frac{\epsilon}{1-\gamma}$

Value Iteration for Policy Optimization

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

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

$$Q^\star(s,a) = Q_\infty^\star(s,a) \qquad V^\star(s) = V_\infty^\star(s)$$



$h = 1$ $h = 2$ $h = 3$

weight 1 γ γ^2

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

Reward: $R(s,a)$

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

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

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

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

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

Bellman Optimality Equation

$$\pi^\star(s) = \underset{a}{\operatorname{argmax}}\ Q^\star(s,a)$$

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

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

or

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

or

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

The Solution Quality when VI Terminates

Unanswered questions:

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

2. At termination, we know $\max_{s,a} \left| Q_i^\star(s,a) - Q_{i-1}^\star(s,a) \right| \leq \epsilon$,

   What can we say about $\max_{s,a} \left| Q_i^\star(s,a) - Q^\star(s,a) \right|$?

   And what can we say about the **performance of the greedy policy $\hat{\pi}$** defined as $\hat{\pi}(a|s) = \mathbb{I}\left[a = \operatorname{argmax}_{a'} Q_i^\star(s,a')\right]$? or simply $\hat{\pi}(s) = \operatorname{argmax}_{a'} Q_i^\star(s,a')$

The Solution Quality when VI Terminates (1/2)

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

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

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

Theorem

$$\mathrm{ValueError}(f) \le \frac{\mathrm{BellmanError}(f)}{1-\gamma}$$

The Solution Quality when VI Terminates (2/2)

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

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

Theorem

$$V^\star(\rho) - V^{\pi_f}(\rho) \le \frac{2}{1-\gamma}\,\mathrm{ValueError}(f)$$

Policy Iteration

Policy Iteration

Policy Iteration

For $i = 1,\ 2, \ldots$

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

Theorem (monotonic improvement). Policy Iteration ensures

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

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

(We will prove this later.)

Generalized Policy Iteration

$N = \infty \Rightarrow$ Policy Iteration

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

For $i = 1, 2, \ldots$

$$\pi_i(s) = \max_a Q_i(s, a)$$ ← **Policy update**

$Q \leftarrow Q_i$

Repeat for N times:

$$Q_{i+1}(s, a) \leftarrow R(s, a) + \gamma \sum_{s', a'} P(s'|s, a)\, \pi_i(a'|s') Q_i(s', a')$$ ← **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**

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

Recall: Regret

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

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

$$= \mathbb{E}\left[\sum_{t=1}^{T}\left(V_1^\star(\rho) - V_1^{\pi_t}(\rho)\right)\right]$$

$$V_1^\pi(\rho) \triangleq \mathbb{E}_{s\sim\rho}[V_1^\pi(s)]$$

Unanswered Questions

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

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

- How to show that Policy Iteration leads to monotonic policy improvement?

- Also, how are these methods related to the third challenge of online RL: credit assignment?

Performance Difference Lemma

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

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

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

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

Discounted occupancy measure on state s

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

Performance Difference Lemma

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

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

$\times(-1)$

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

$$\parallel$$

Original version:

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

Performance Difference Lemma (Fixed-Horizon)

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

$$\mathbb{E}_{s_1\sim\rho}\left[V_1^{\pi'}(s_1)\right] - \mathbb{E}_{s_1\sim\rho}[V_1^{\pi}(s_1)] = \sum_{h=1}^{H}\sum_{s,a} d_{\rho,h}^{\pi'}(s)\left(\pi_h'(a|s) - \pi_h(a|s)\right)Q_h^{\pi}(s,a)$$

$$= \sum_{h=1}^{H}\sum_{s,a} d_{\rho,h}^{\pi'}(s,a)\left(Q_h^{\pi}(s,a) - V_h^{\pi}(s)\right)$$

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

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

The Meaning of Performance Difference Lemma

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

The sub-optimality of a policy π:

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

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

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

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

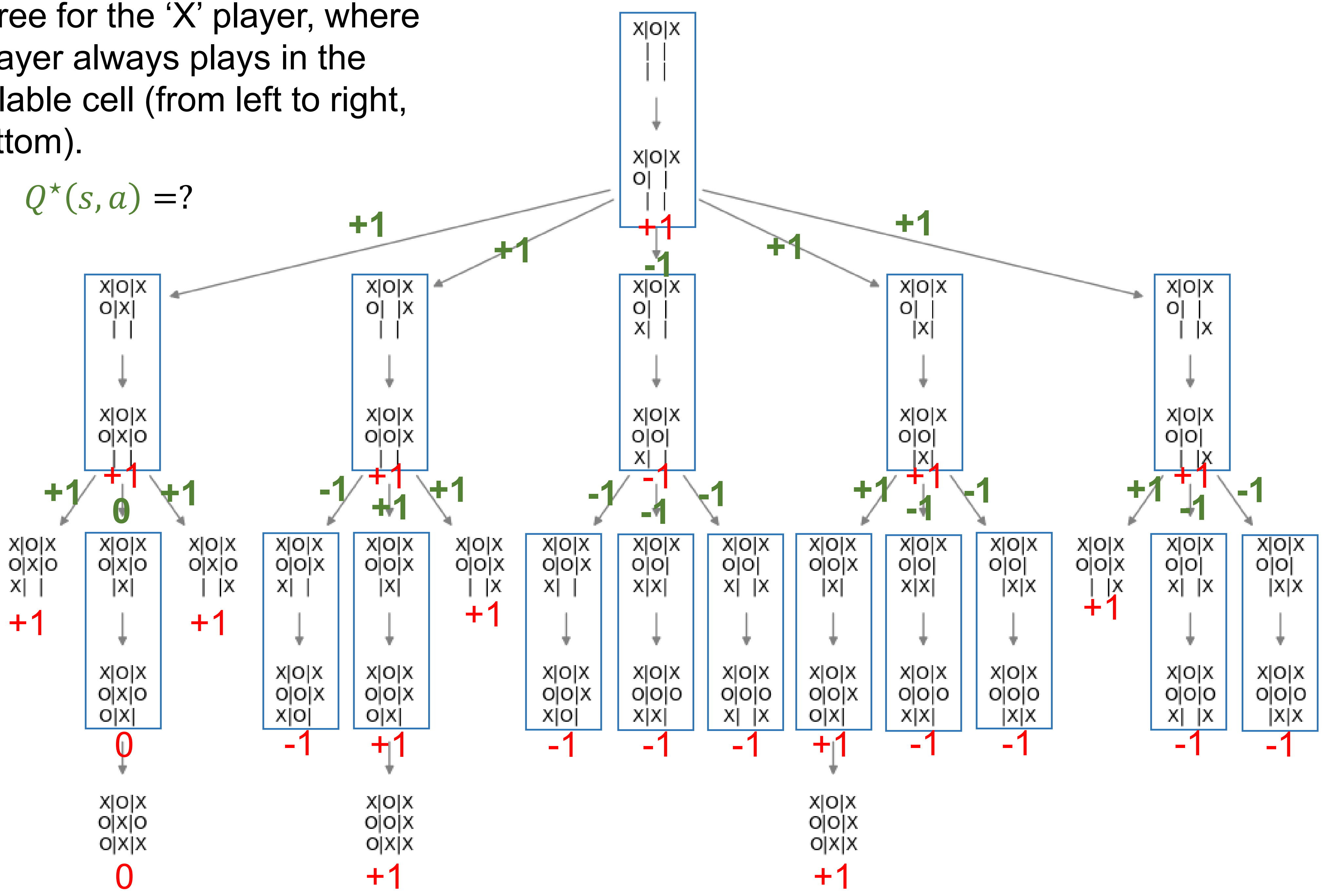
$$= \sum_{s,a} d_\rho^{\pi^\star}(s,a)\,(Q^\pi(s,a) - V^\pi(s))$$ ✓

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

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

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).

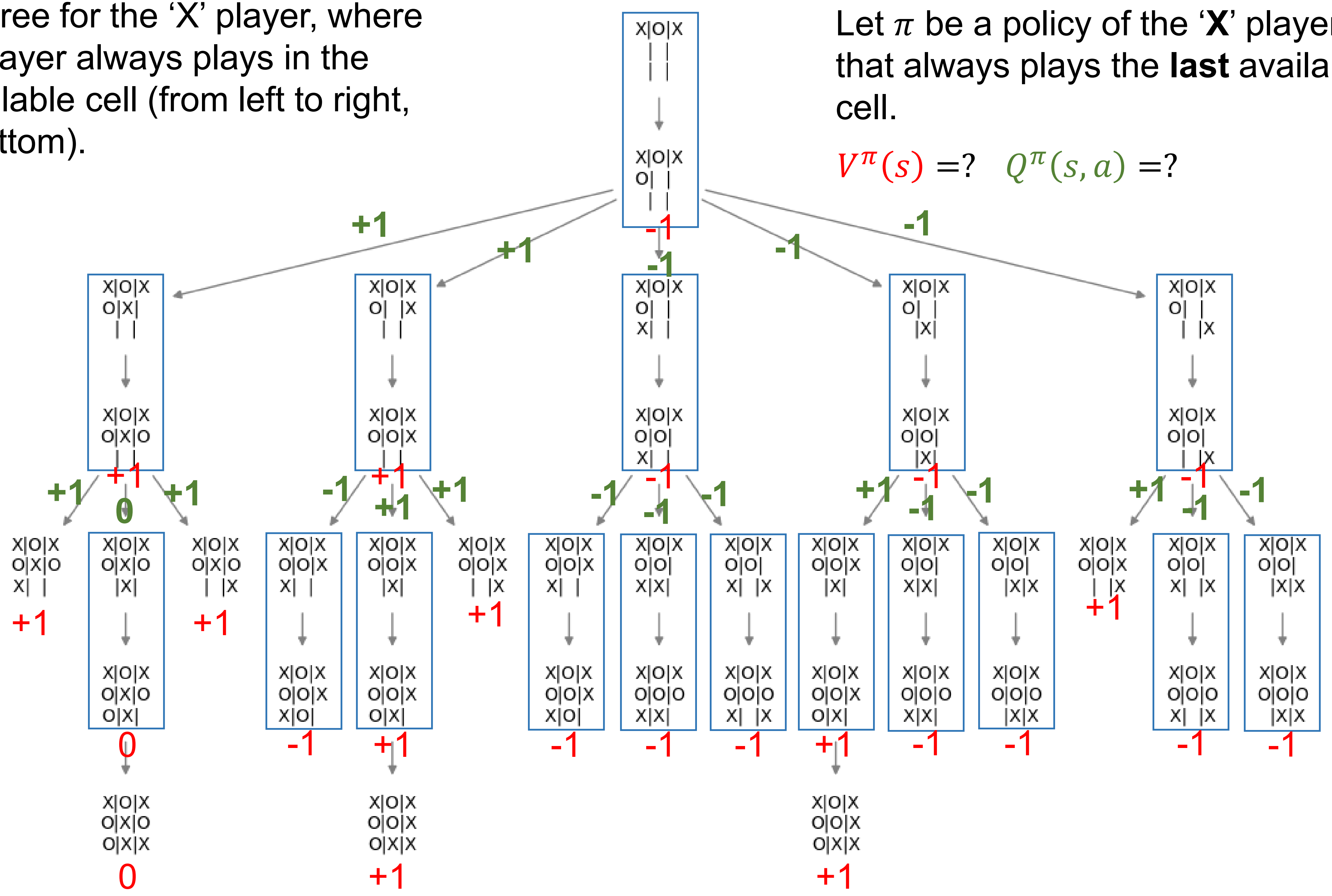
$V^{\star}(s) = ?$ $Q^{\star}(s,a) = ?$



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.

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



Proof (Sketch) of Performance Difference Lemma

h

Unanswered Question 1

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

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

If

$$|f(s,a) - Q^\star(s,a)| \leq \epsilon \qquad \forall s, a$$

then

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

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

Unanswered Question 2

Policy Iteration ensures

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

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

$$
\begin{aligned}
&\pi_i = \pi_{i-1} \\
&\Rightarrow \pi_i(s) = \operatorname*{argmax}_a Q^{\pi_i}(s,a) \\
&\Rightarrow Q^{\pi_i}(s,a) = R(s,a) + \gamma \sum_{s',a'} P(s'|s,a)\pi_i(a'|s')Q^{\pi_i}(s',a') = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q^{\pi_i}(s',a') \\
&\Rightarrow Q^{\pi_i} \text{ satisfies the Bellman optimality equation} \\
&\Rightarrow \text{BellmanError}(Q^{\pi_i}) = 0 \\
&\Rightarrow Q^{\pi_i}(s,a) = Q^\star(s,a) \text{ by the ``ValueError} \le \frac{1}{1-\gamma} \text{ BellmanError'' lemma on Page 38} \\
&\Rightarrow \pi_i(s) = \operatorname*{argmax}_a Q^\star(s,a) = \pi^\star(s).
\end{aligned}
$$

Recap: MDP

- Definitions of $Q^\pi(s,a), V^\pi(s), Q^\star(s,a), V^\star(s)$
- Bellman equations (related to dynamic programming)
- Value Iteration to approximate $Q^\pi(s,a)/V^\pi(s)$ or $Q^\star(s,a)/V^\star(s)$
- Policy Iteration to find $\pi^\star$ --- involving $Q^\pi(s,a)/V^\pi(s)$ approximation
- Unified by Generalized Policy Iteration
- Performance difference lemma to decompose $\mathbb{E}_{s\sim\rho}\left[V^{\pi'}(s)\right] - \mathbb{E}_{s\sim\rho}[V^\pi(s)]$
  - Credit assignment
  - $= \sum_{s,a} d_\rho^\pi(s,a)\left(V^{\pi'}(s) - Q^{\pi'}(s,a)\right)$ (helpful in analyzing VI by letting $\pi' = \pi^\star$)
  - $= \sum_{s,a} d_\rho^{\pi'}(s,a)\left(Q^{\pi}(s,a) - V^{\pi}(s)\right)$ (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 $\hat{Q}(s,a) \approx Q^{\star}(s,a)$ and let $\hat{\pi}(s) = \underset{a}{\operatorname{argmax}}\, \hat{Q}(s,a)$
  - Policy-Iteration-based: approximate $\hat{Q}(s,a) \approx Q^{\pi}(s,a)$ and let $\hat{\pi}(s) = \underset{a}{\operatorname{argmax}}\, \hat{Q}(s,a)$
  - … or something in between (based on generalized policy iteration)
- RL algorithms we will discuss:
  - Performing approximate VI or PI using samples