Markov Decision Processes

Chen-Yu Wei

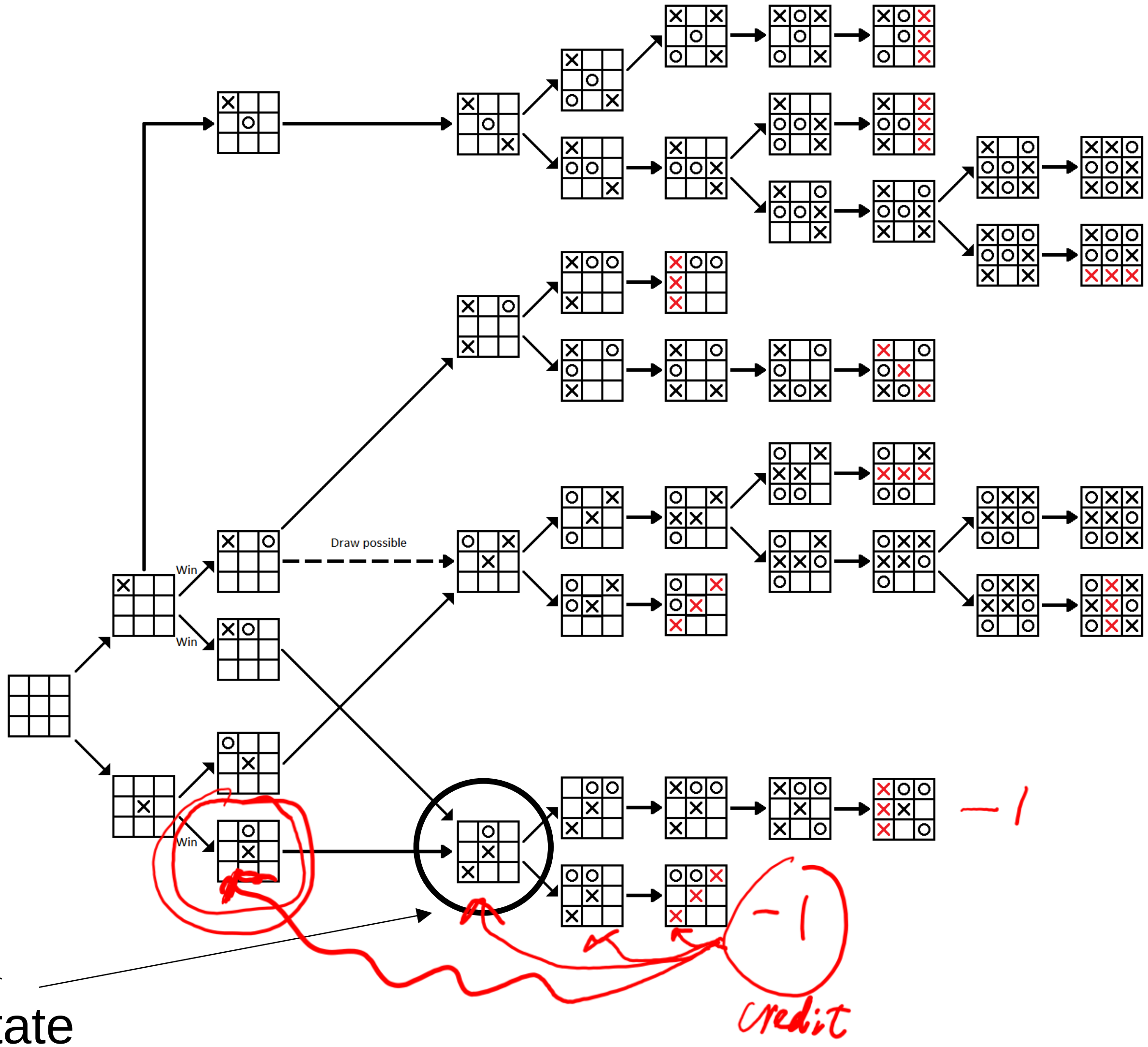
Sequence of Actions

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)

Interaction Protocol (Episodic Setting)

step

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

$h \leftarrow h + 1$

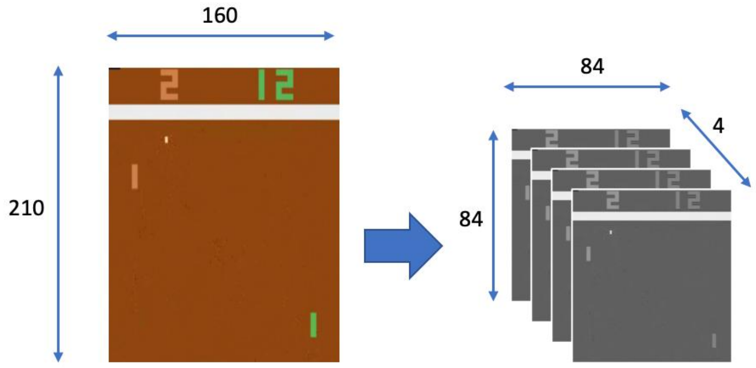
Markov assumption:

$r_{t,h}$ 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 $s_{t,h}$

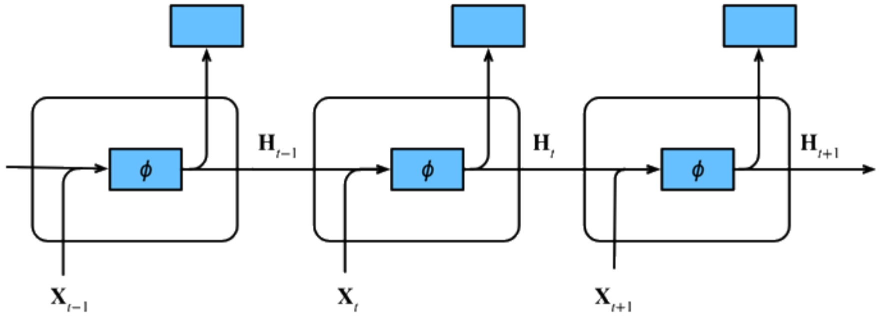
Goal: maximize $\sum_{t=1}^{T} \sum_{h=1}^{\tau_t} R(s_{t,h}, a_{t,h})$

τ_t : length of episode t

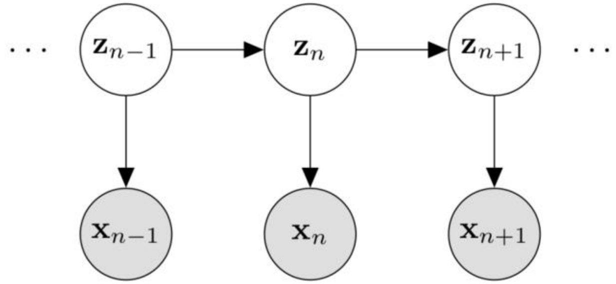
From Observations to States



Stacking recent observations



Recurrent neural network



Hidden Markov model

Regret (Episodic Setting)

$\pi^*: S \to A$

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

CB $\max_{\pi^*} \sum_{t=1}^{T} R(x_t, \pi^*(x_t)) - \sum_{t=1}^{T} R(x_t, a_t)$

$$s_{t,1} \xrightarrow{\pi^\star(s_{t,1})} \tilde{s}_{t,2} \xrightarrow{\pi^\star(\tilde{s}_{t,2})} \tilde{s}_{t,3} \xrightarrow{\pi^\star(\tilde{s}_{t,3})} \cdots\cdots \longrightarrow \tilde{s}_{t,\tilde{\tau}_t}$$

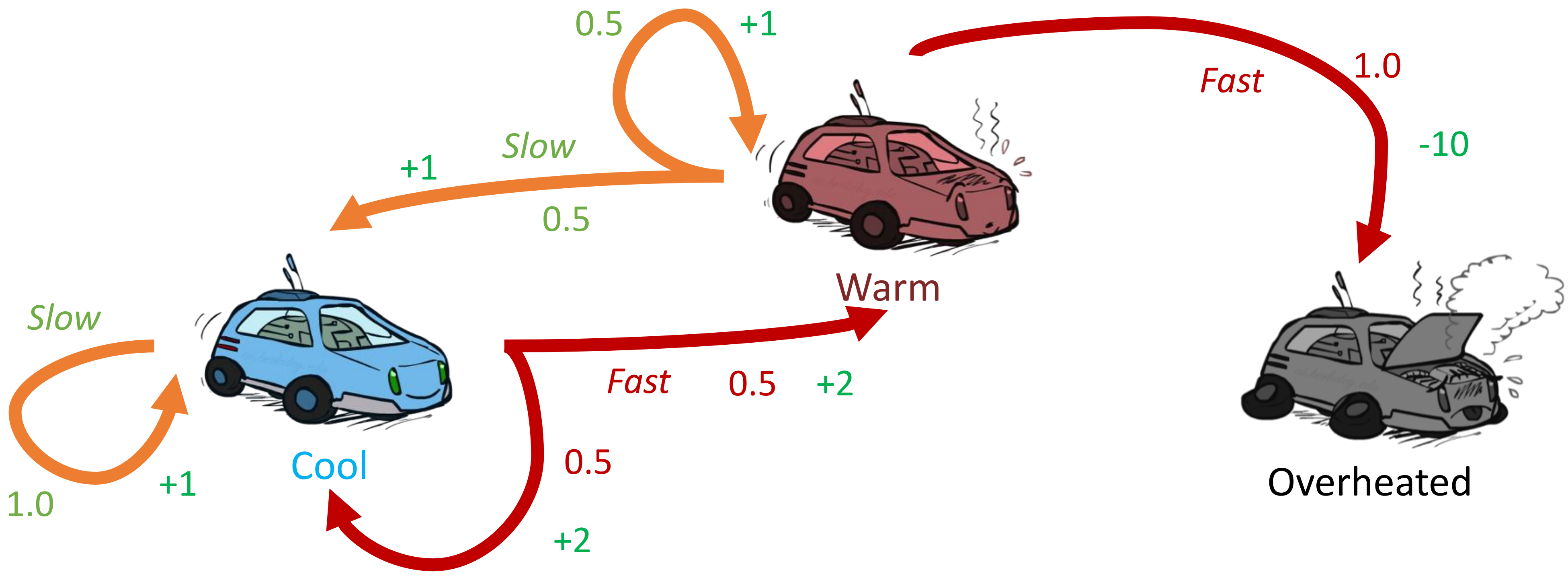
Trajectory generated by $\pi^\star$

$$s_{t,1} \xrightarrow{a_{t,1}} s_{t,2} \xrightarrow{a_{t,2}} s_{t,3} \xrightarrow{a_{t,3}} \cdots \longrightarrow s_{t,\tau_t}$$

Trajectory generated by learner

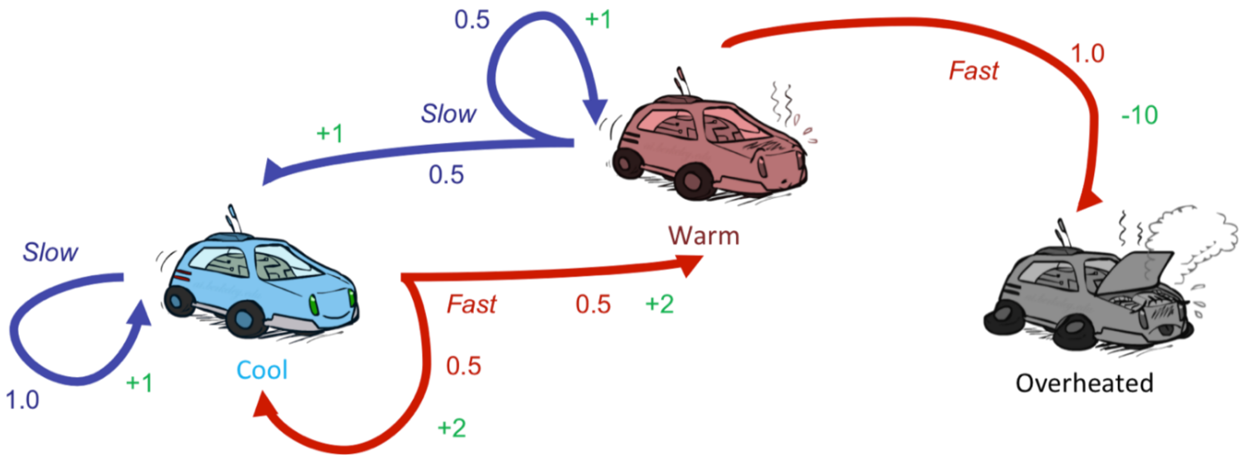
Example: Racing

- 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'	$P(s'\|s,a)$	$R(s,a)$
	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 (Goal-Oriented)
  - Infinite-Horizon
- Performance Metric
  - Total Reward
  - Average Reward
  - Discounted Reward
- Policy
  - Markov policy
  - Stationary policy

Horizon = Length of an episode

Interaction Protocols (1/3): Fixed-Horizon

Horizon length is a fixed number H

$h \leftarrow 1$

Observe initial state $s_1 \sim \rho$

While $\boldsymbol{h \leq H}$:

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 (2/3): Goal-Oriented

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.

Interaction Protocols (3/3): Infinite-Horizon

The learner continuously interacts with the environment

$h \leftarrow 1$

Observe initial state $s_1 \sim \rho$

Loop forever:

  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: network management, inventory management

Formulations

- Interaction Protocol
  - Fixed-Horizon
  - Variable-Horizon (Goal-Oriented)
  - Infinite-Horizon
- Performance Metric
  - Total Reward
  - Average Reward
  - Discounted Reward
- Policy
  - Markov policy
  - Stationary policy

Performance Metric

Total Reward (for episodic setting): $\sum_{h=1}^{\tau} r_h$ (τ: the step where the episode ends)

Average Reward (for infinite-horizon setting): $\lim_{H\to\infty} \frac{1}{H} \sum_{h=1}^{H} r_h$

Discounted Total Reward (for episodic or infinite-horizon): $\sum_{h=1}^{\tau} \gamma^{h-1} r_h$

τ: the step where the episode ends, or ∞ in the infinite-horizon case
$\gamma \in [0,1)$: discount factor

Interaction Protocols vs. Performance Metrics

	“natural” objective		
Fixed-Horizon	$\dashrightarrow$	Total Reward	
Goal-Oriented	$\dashrightarrow$	Total Reward	Could be unbounded
Infinite-horizon	$\dashrightarrow$	Average Reward	Could have constant change for an infinitesimal change in policy

Discounted Total Reward?

Focusing more on the **recent** reward

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

Formulations

- Interaction Protocol
  - Fixed-Horizon
  - Variable-Horizon (Goal-Oriented)
  - Infinite-Horizon
- Performance Metric
  - Total Reward
  - Average 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

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

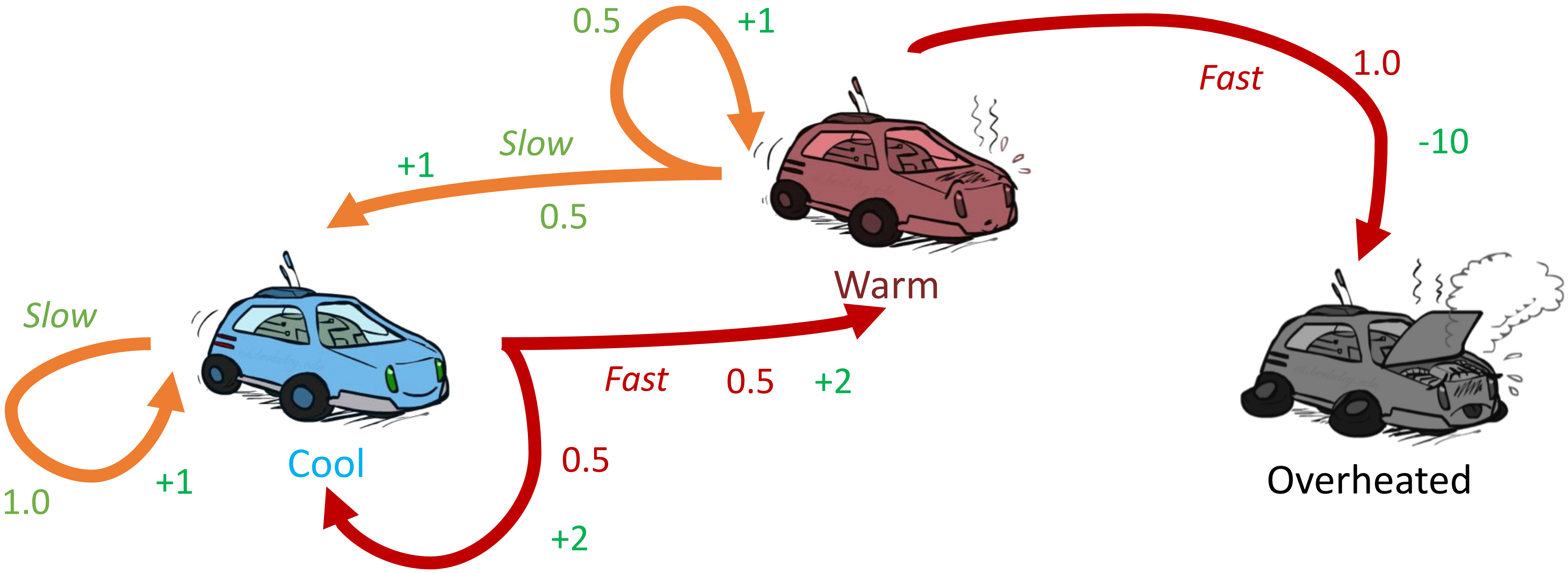
$\pi_h(\text{Slow} \mid \text{Cool})$

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

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

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

$\forall h$



Value Iteration

(Fixed-Horizon)

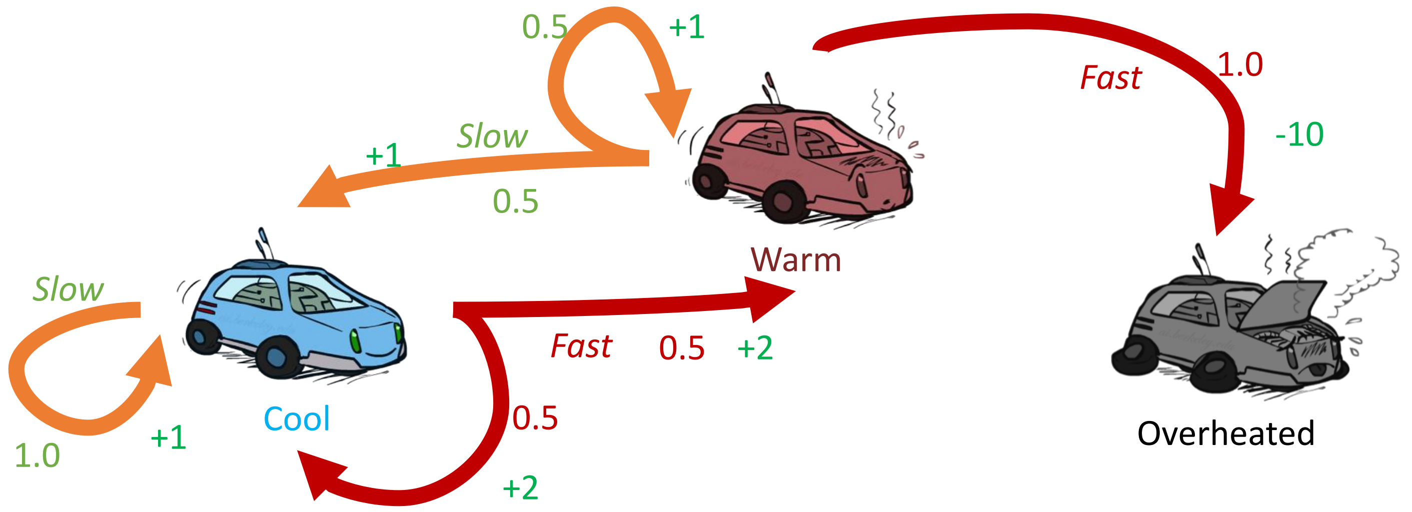
Two Tasks

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

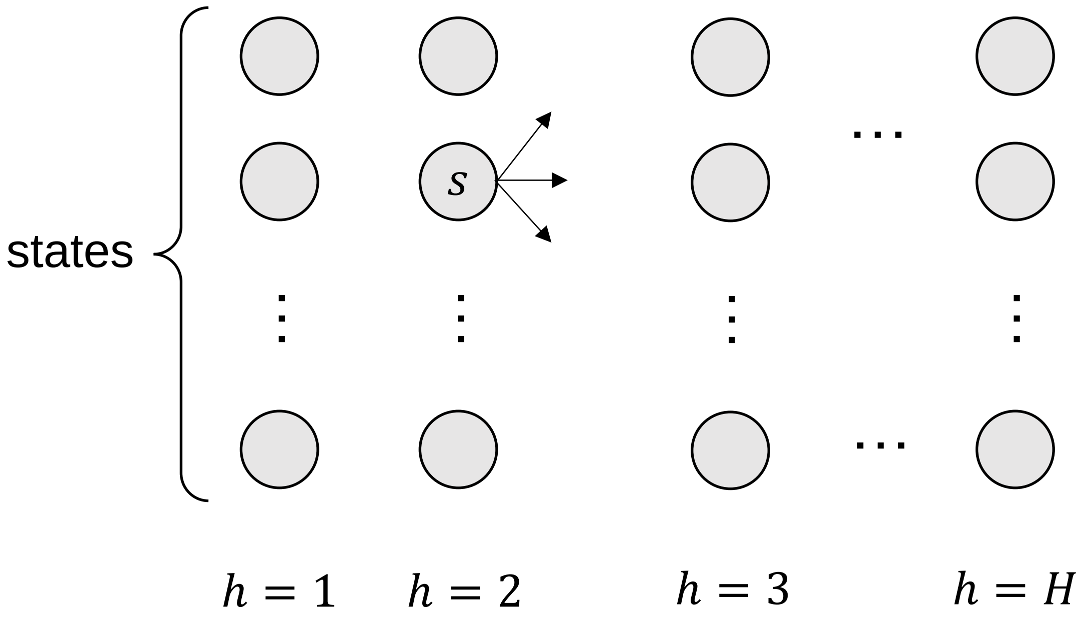
What is the policy that achieves the highest policy expected total reward?



Value Iteration for Policy Evaluation

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

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



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

Reward: $R(s,a)$

Backward induction:

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

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

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

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

Bellman Equation

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

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

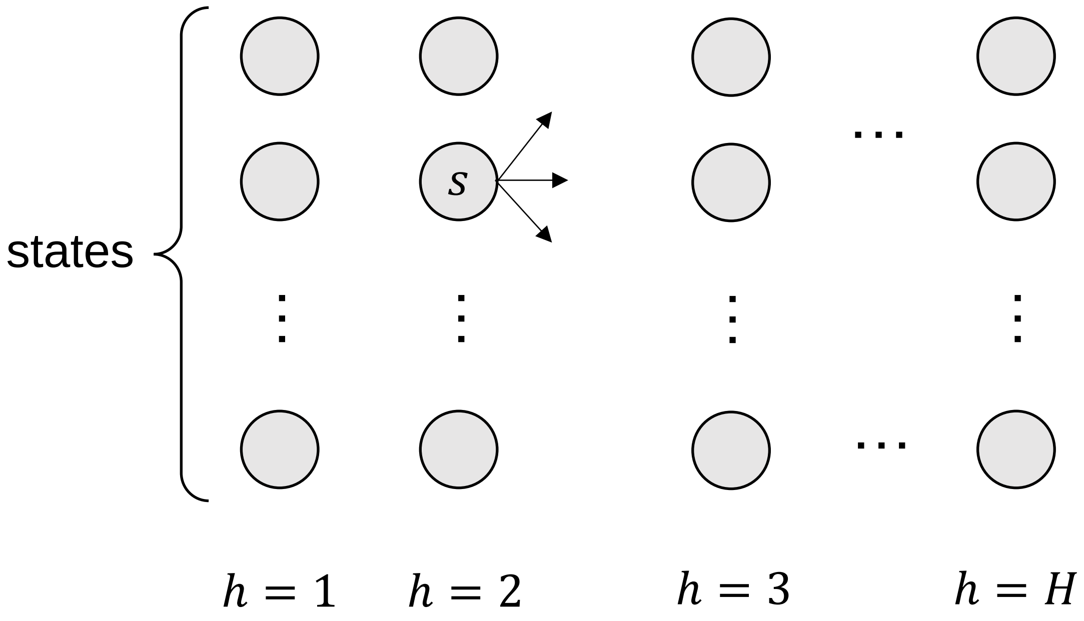
or

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

or

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

Value Iteration for Policy Optimization



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

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

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

Reward: $R(s,a)$

Backward induction:

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

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

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

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

Bellman Optimality Equation

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

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

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

or

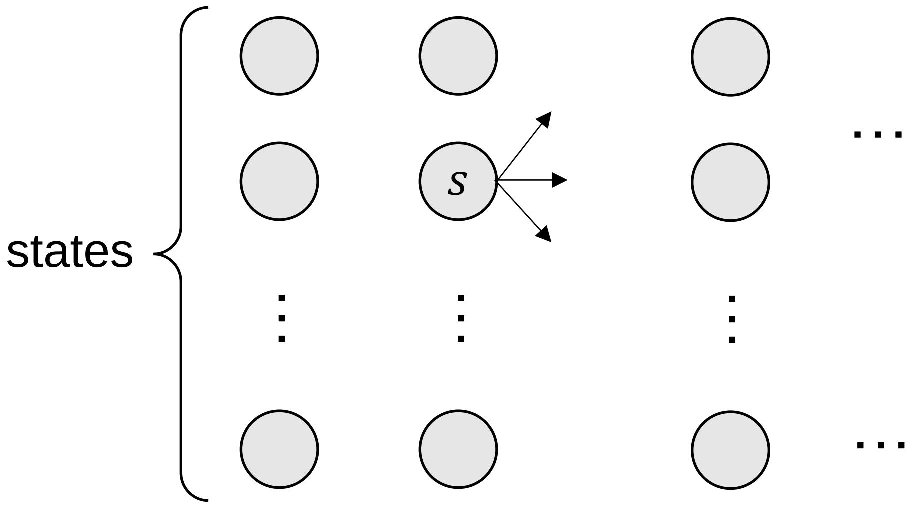
$$Q_h^\star(s,a) = R(s,a) + \sum_{s'} P(s'|s,a) \left(\max_{a'} Q_{h+1}^\star(s',a') \right)$$

or

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

Value Iteration
(Infinite-Horizon)

Value Iteration for Policy Evaluation



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

weight $\quad 1$ $\quad \gamma$ $\quad \gamma^2$

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

Reward: $R(s,a)$

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

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

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

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

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

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

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

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

			
V_2^π	3.35	2.35	0
V_1^π	2	1	0
V_0^π	0	0	0

Assume $\gamma = 0.9$ $\quad \pi(\text{cool}) = \text{fast}, \ \pi(\text{warm}) = \text{slow}$

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

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

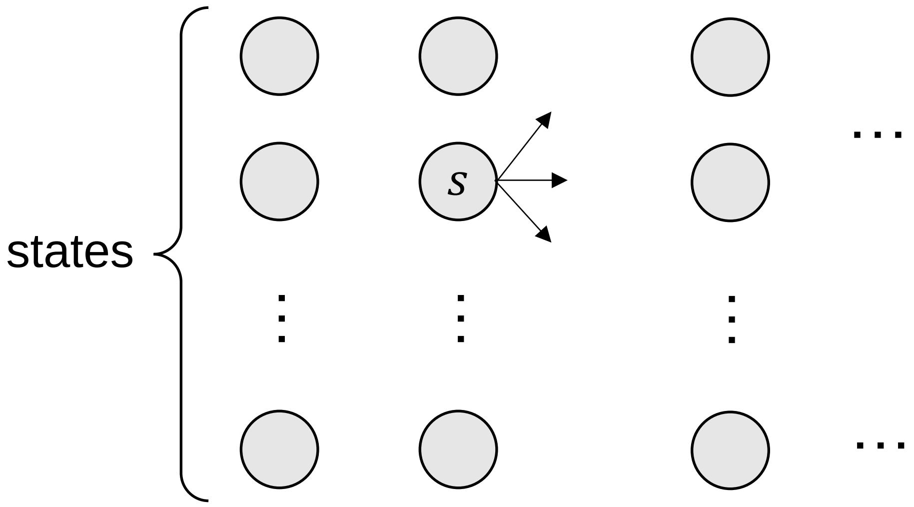
Convergence

Value Iteration ensures

$$\left|Q_i^{\pi}(s,a) - Q^{\pi}(s,a)\right| \leq \gamma^i |Q_0^{\pi}(s,a) - Q^{\pi}(s,a)|$$

$$\left|V_i^{\pi}(s) - V^{\pi}\ (s)\right| \leq \gamma^i |V_0^{\pi}(s) - V^{\pi}(s)|$$

Value Iteration for Policy Optimization



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

weight $\quad$ 1 $\quad$ γ $\quad$ γ^2

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

Reward: $R(s,a)$

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

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

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

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

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

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

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

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

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

Convergence

Value Iteration ensures

$$\left|Q_i^\star(s,a) - Q^\star(s,a)\right| \le \gamma^i |Q_0^\star(s,a) - Q^\star(s,a)|$$

$$\left|V_i^\star(s) - V^\star\ (s)\right| \le \gamma^i |V_0^\star(s) - V^\star(s)|$$

Question

We know $Q^{\star}(s,a) = \lim_{i\to\infty} Q_i^{\star}(s,a)$ recovers the optimal policy by $\pi^{\star}(s) = \underset{a}{\operatorname{argmax}}\, Q^{\star}(s,a)$.

But we only have $Q_i^{\star}(s,a)$ for finite i.

How good is the policy $\hat{\pi}(s) = \underset{a}{\operatorname{argmax}}\, Q_i^{\star}(s,a)$?

Policy Iteration

Policy Iteration

Policy Iteration

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

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

Theorem (monotonic improvement). Policy Iteration ensures

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

Modified Policy Iteration

$N = 1 \Rightarrow$ Value Iteration

$N = \infty \Rightarrow$ Policy Iteration

For $i = 1, 2, \ldots$

$$Q_i(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, V_i(s')$$

$$\pi_{i+1}(s) = \max_a Q_i(s,a)$$

Policy update

$V(s) \leftarrow V_i(s)$

Repeat for N times:

$$V(s) \leftarrow \sum_a \pi_{i+1}(a|s) \left(R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, V(s') \right)$$

$V_{i+1}(s) \leftarrow V(s)$

Value update

Performance Difference Lemma

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

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

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