# Markov Decision Processes

Chen-Yu Wei

# Sequence of Actions


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

**One option:** view every sequence as a "meta-action": $\bar{a} = (a_1, a_2, \cdots, a_H)$

**Drawback:**

- The number of actions is exponential in horizon
- In stochastic environments**,** this does not leverage intermediate observations

**Solution idea:** dynamic programming

# Interaction Protocol: Fixed-Horizon Case

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

For **step** $h = 1, 2, \dots, H$:

Learner observes an observation $x_{t,h}$

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

Learner receives instantaneous reward $r_{t,h}$

**General case:**

$$\mathbb{E}\left[r_{t,h}\right] = R\left(x_{t,1}, a_{t,1}, \dots, x_{t,h}, a_{t,h}\right), \quad x_{t,h+1} \sim P(\cdot \mid x_{t,1}, a_{t,1}, \dots, x_{t,h}, a_{t,h})$$

$\Rightarrow$ Optimal decisions may depend on the entire history $\mathcal{H}_t = (x_{t,1}, a_{t,1}, \dots, x_{t,h})$

# Interaction Protocol: Fixed-Horizon Case

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

For **step** $h = 1, 2, \dots, H$:

Learner observes an observation $x_{t,h}$

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

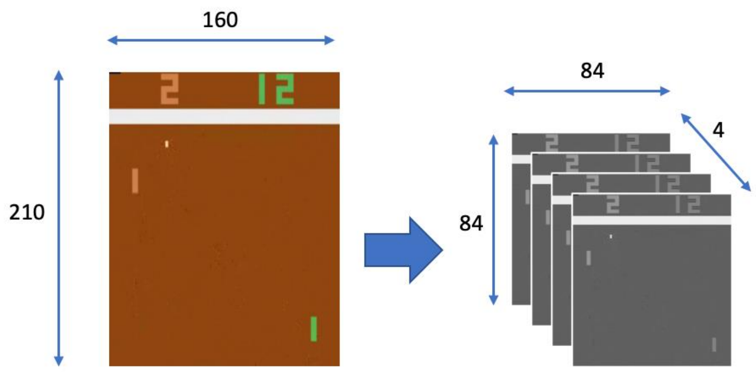
Learner receives instantaneous reward $r_{t,h}$

We assume that the history $\mathcal{H}_t = (x_{t,1}, a_{t,1}, \dots, x_{t,h})$ can be summarized as a **horizon-length-independent** representation $s_{t,h} = \Phi(x_{t,1}, a_{t,1}, \dots, x_{t,h}) \in \mathcal{S}$ so that

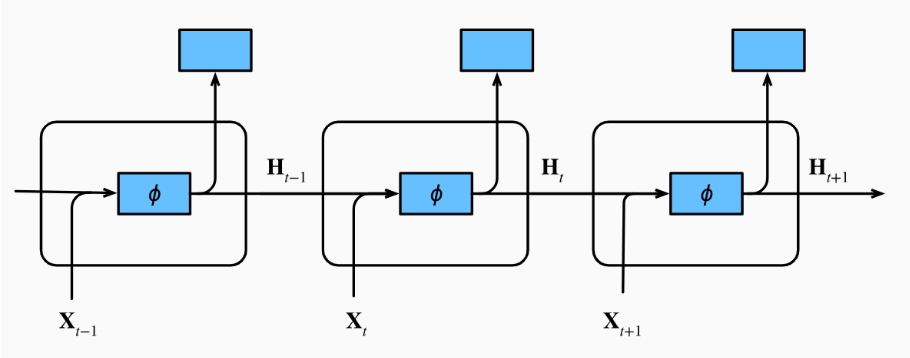
$$\mathbb{E}[r_{t,h}] = R(s_{t,h}, a_{t,h}), \quad x_{t,h+1} \sim P(\cdot \mid s_{t,h}, a_{t,h})$$

$s_{t,h}$ is called the "state" at the step $h$ of episode $t$.

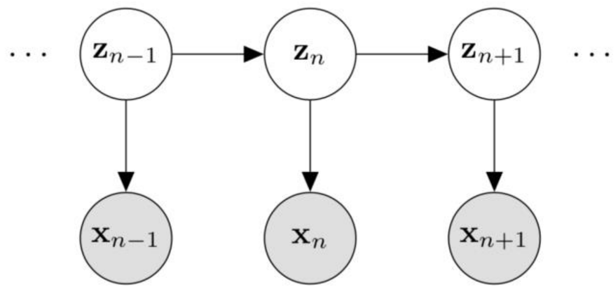
# From Observations to States




Stacking recent observations




Recurrent neural network




Hidden Markov model

# Interaction Protocol: Fixed-Horizon Case

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

For **step** $h = 1, 2, \ldots, H$:

Environment reveals state $s_{t,h}$

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

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

Next state is generated as $s_{t,h+1} \sim P(\cdot \mid s_{t,h}, a_{t,h})$

This is called the Markov decision process.

# MDP as Contextual Bandits?

Viewing states as contexts, and viewing the problem as a contextual bandit problem with $TH$ rounds (what's wrong?)

$(x_{t,1}, a_{t,1}, \ldots, x_{t,h})$

$$\text{Regret (contextual bandit)} = \sum_{t=1}^{T}\sum_{h=1}^{H} \underbrace{\max_a R(s_{t,h}, a)} - \sum_{t=1}^{T}\sum_{h=1}^{T} R(s_{t,h}, a_{t,h})$$

$$\text{Regret (MDP)} = \sum_{t=1}^{T}\left[\sum_{h=1}^{H} R(s^*_{t,h}, a^*_{t,h})\right] - \sum_{t=1}^{T}\sum_{h=1}^{H} R(s_{t,h}, a_{t,h})$$

$s^*_{t,1} = s_{t,1}$

$s^*_{t,h} \neq s_{t,h}$ for $h \geq 2$

# Formulations

- Interaction Protocol
  - Fixed-Horizon
  - Variable-Horizon (Goal-Oriented)
  - Infinite-Horizon
- Performance Metric
  - Total Reward
  - Average Reward
  - Discounted Reward
- Policy
  - History-dependent 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 $s_1$

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

**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 $s_1$

**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.

# Interaction Protocols (3/3): Infinite-Horizon

The learner continuously interacts with the environment

$h \leftarrow 1$

Observe initial state $s_1$

**Loop forever:**

Choose action $a_h$

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

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

$h \leftarrow h + 1$

**Examples:** network management, inventory management

# Formulations for Markov Decision Processes

- Interaction Protocol
  - Fixed-Horizon
  - Variable-Horizon (Goal-Oriented)
  - Infinite-Horizon

Episodic setting

- Performance Metric
  - Total Reward
  - Average Reward
  - Discounted Reward
- Policy
  - History-dependent policy
  - Markov policy
  - Stationary policy

# Performance Metric

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

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

If $|r_h| \le 1$,

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

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

# Interaction Protocols vs. Performance Metrics

"natural" objective

Fixed-Horizon - - - - - - - - - - → Total Reward

Goal-Oriented - - - - - - - - - - → Total Reward — Could be unbonded

Infinite-horizon - - - - - - - - - - → 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.

# Our Focus

In most of the following lectures, we focus on the **goal-oriented / infinite-horizon** setting with **discount total reward** as the performance metric.

# Policy

A mapping from observations/contexts/states to (distribution over) actions

- Contextual bandits

$$a \sim \pi(\cdot \mid x) \qquad \text{(randomized/stochastic)}$$

$$\text{or } a = \pi(x) \qquad \text{(deterministic)}$$

- Multi-armed bandits

$$a \sim \pi$$

$$\text{or } a = a^\star$$

# Policy for MDPs

History-dependent Policy

$$a_h \sim \pi(\cdot \mid s_1, a_1, r_1, s_2, a_2, r_2, \dots, s_h)$$
$$a_h = \pi(s_1, a_1, r_1, s_2, a_2, r_2, \dots, s_h)$$

Markov Policy

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

For **fixed-horizon + total reward** 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 + discounted total reward** setting, there exists an optimal policy in this class

# Fixed-Horizon + Total Reward

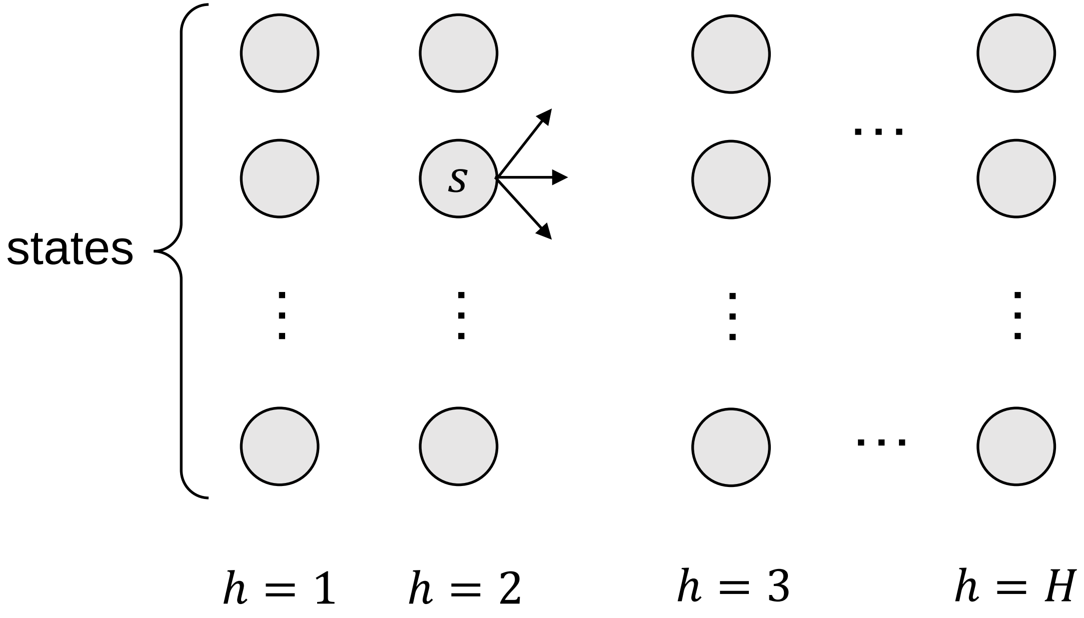
# Dynamic Programming

**Goal:** Calculate the expected total reward of a policy

A (Markov) policy is a mapping from (state, step index) to action distribution, written as

$$\pi_h(\cdot \mid s) \in \Delta(\mathcal{A}) \qquad \text{for } s \in \mathcal{S} \text{ and } h \in \{1, 2, \ldots, H\}$$

# Dynamic Programming




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

Reward: $R(s,a)$

**Key quantity:** $V_h^\pi(s)$ = the expected total reward of policy $\pi$ starting from state $s$ at step $h$.

**Backward calculation:**

$$V_H^\pi(s) = \sum_a \pi_H(a|s)\, R(s,a) \qquad \forall s$$

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

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

# Bellman Equation

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

$$V_h^{\pi}(s) = \sum_a \pi_h(a|s) \underbrace{\left( R(s,a) + \sum_{s'} P(s'|s,a)\, V_{h+1}^{\pi}(s') \right)}_{Q_h^{\pi}(s,a)} \qquad \text{for } h = H, \dots, 1$$

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

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

# Occupancy Measures

$d_\rho^\pi(s)$: the expected number of times state $s$ is visited, under policy $\pi$ and initial state distribution $\rho$

**Key quantity:** $d_{\rho,h}^\pi(s) =$ the probability of state $s$ being visited **at step $h$**, under policy $\pi$ and initial state distribution $\rho$

**Forward calculation:**

$$d_{\rho,1}^\pi(s) = \rho(s) \qquad \forall s$$

For $h = 2, \ldots H$:

$$d_{\rho,h}^\pi(s) = \sum_{s'} d_{\rho,h-1}^\pi(s') \sum_{a'} \pi_{h-1}(a'|s') P(s|s',a') \qquad \forall s$$

# Reverse Bellman Equation

$$d^{\pi}_{\rho,1}(s) = \rho(s)$$

$$d^{\pi}_{\rho,h}(s) = \sum_{s',a'} \underbrace{d^{\pi}_{\rho,h-1}(s')\,\pi_{h-1}(a'|s')}_{d^{\pi}_{\rho,h-1}(s',a')} P(s|s',a') \qquad \text{for } h = 2, \ldots, H$$

$$d^{\pi}_{\rho,h}(s) = \sum_{s',a'} d^{\pi}_{\rho,h-1}(s',a') P(s|s',a')$$

$$d^{\pi}_{\rho,h}(s,a) = d^{\pi}_{\rho,h}(s)\pi_h(a|s)$$

# Dynamic Programming

**Goal:** Find the optimal policy

**Key quantity:** $V_h^\star(s)$ = the optimal expected total reward starting from state $s$ at step $h$.

**Backward calculation:**

$$\left.\begin{array}{l} V_H^\star(s) = \max_a \; R(s,a) \qquad \forall s \\ \text{For } h = H-1, \dots 1: \\ \qquad V_h^\star(s) = \max_a \; R(s,a) + \sum_{s'} P(s'|s,a)\, V_{h+1}^\star(s') \qquad \forall s \end{array}\right\} \text{Value Iteration}$$

$$\pi_h^\star(s) = \operatorname*{argmax}_a \; R(s,a) + \sum_{s'} P(s'|s,a)\, V_{h+1}^\star(s')$$

# Bellman Optimality Equation

$$V^\star_{H+1}(s) = 0$$

$$V^\star_h(s) = \max_a \Big( R(s,a) + \underbrace{\sum_{s'} P(s'|s,a)\, V^\star_{h+1}(s')}_{Q^\star_h(s,a)} \Big) \qquad \text{for } h = H, \ldots, 1$$

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

$$Q^\star_h(s,a) = R(s,a) + \sum_{s'} P(s'|s,a)\, V^\star_{h+1}(s')$$

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

# Recap

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

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

Bellman Equation
(Value Iteration)

$$d_{\rho,h}^\pi(s,a) = d_{\rho,h}^\pi(s) \pi_h(a|s)$$

$$d_{\rho,h}^\pi(s) = \sum_{s',a'} d_{\rho,h-1}^\pi(s',a') P(s|s',a')$$

Reverse Bellman Equation

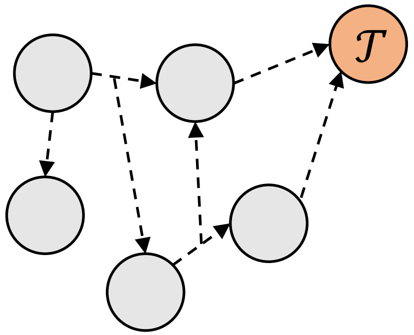
$$V_h^\star(s) = \max_a Q_h^\star(s,a)$$

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

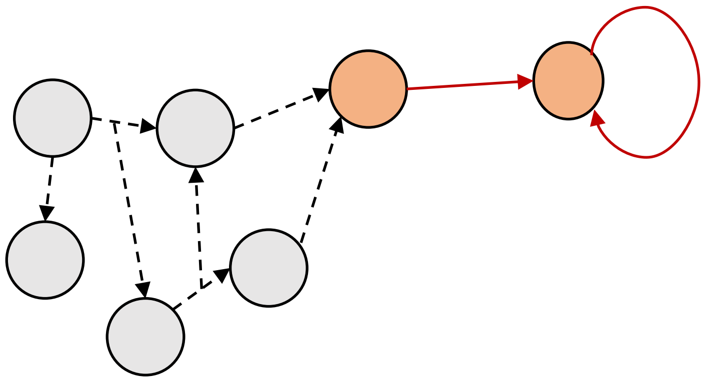
Bellman Optimality Equation
(Value Iteration)

# Infinite-Horizon / Goal-Oriented + Discounted Total Reward

# Equivalent Views



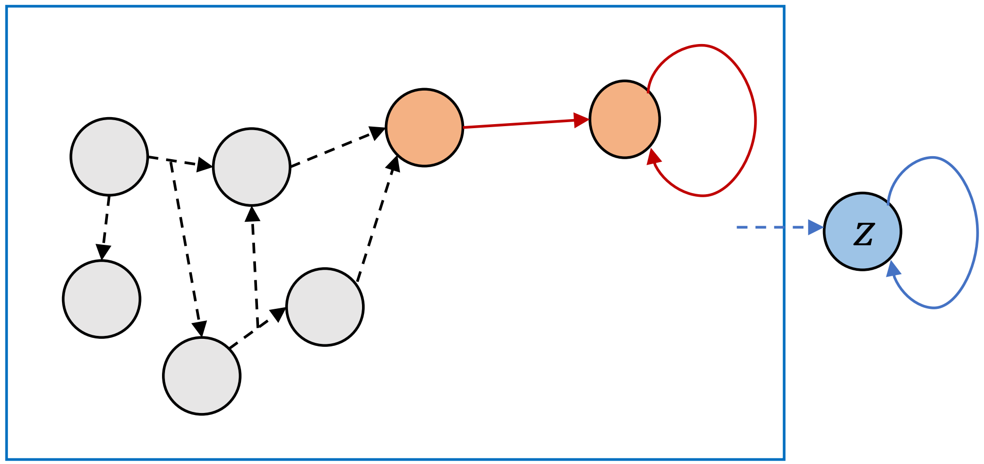

deterministic and zero-reward



Converting goal-oriented to infinite-horizon

$$\mathbb{E}^{\text{new}}\left[\sum_{h=1}^{\infty}\gamma^{h-1}r_h\right] = \mathbb{E}^{\text{old}}\left[\sum_{h=1}^{\tau}\gamma^{h-1}r_h\right]$$

Scale down all transitions by a factor of $\gamma$ and add probability $1-\gamma$ transitioning to $z$




Converting discounted total reward to total reward

$$\mathbb{E}^{\text{new}}\left[\sum_{h=1}^{\infty}r_h\right] = \mathbb{E}^{\text{old}}\left[\sum_{h=1}^{\infty}\gamma^{h-1}r_h\right]$$

# Dynamic Programming

**Goal:** Calculate the expected discounted total reward of a stationary policy $\pi$

$V^{\pi}(s) =$ the expected discounted total reward starting from state $s$

**Key quantity:** $V_i^{\pi}(s) =$ the expected discounted total reward starting from state $s$ supposed that $i$ more steps can be executed

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

For $i = 1, 2, 3$ ...

$$V_i^{\pi}(s) = \sum_a \pi(a|s) \left( R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, V_{i-1}^{\pi}(s') \right) \qquad \forall s$$

$V^{\pi}(s) = \lim_{i \to \infty} V_i^{\pi}(s)$ (need to prove that the limit exists)

# Value Iteration for $V^{\pi}$

Arbitrary $\hat{V}_0(s) \quad \forall s$

For $i = 1, 2, 3$ ...

$$\hat{V}_i(s) = \sum_a \pi(a|s)\left( R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, \hat{V}_{i-1}(s') \right) \qquad \forall s$$

To show that this algorithm converges, we prove the following statement:

For any $\epsilon > 0$, there exists a large enough $N$ such that

$$\left|\hat{V}_i(s) - \hat{V}_j(s)\right| \leq \epsilon$$

for any $i, j \geq N$.

# Proof of Convergence

# Proof of Convergence

For any $\epsilon > 0$, there exists a large enough $N$ such that

$$\left|\hat{V}_i(s) - \hat{V}_j(s)\right| \leq \epsilon$$

for any $i, j \geq N$.

$$\hat{V}(s) = \lim_{i \to \infty} \inf\left\{\hat{V}_j(s) : j \geq i\right\}$$

For any $\epsilon > 0$, there exists a large enough $N$ such that

$$\left|\hat{V}_i(s) - \hat{V}(s)\right| \leq \epsilon$$

for any $i \geq N$.

# Proof of Uniqueness

No matter what the initial values of $\hat{V}_0(s)$ are, the limit $\lim_{i \to \infty} \hat{V}_i(s)$ is the same.

(This value is $V^\pi(s)$)

# Bellman Equation

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

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

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

# Occupancy Measures

$d_\rho^\pi(s)$: the expected number of times state $s$ is visited, under policy $\pi$ and initial state distribution $\rho$

**Key quantity:** $d_{\rho,h}^\pi(s) =$ the probability of state $s$ being visited at step $h$, under policy $\pi$ and initial state distribution $\rho$

**Forward calculation:**

$$d_{\rho,1}^\pi(s) = \rho(s) \qquad \forall s$$

For $h = 2, 3, \ldots$

$$d_{\rho,h}^\pi(s) = \gamma \sum_{s'} d_{\rho,h-1}^\pi(s') \sum_{a'} \pi(a'|s') P(s|s', a') \qquad \forall s$$

# Reverse Bellman Equation

$$d_\rho^\pi(s) = \rho(s) + \gamma \sum_{s',a'} \underbrace{d_\rho^\pi(s')\,\pi(a'|s')}_{d_\rho^\pi(s',a')} P(s|s',a')$$

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

$$d_\rho^\pi(s,a) = d_\rho^\pi(s)\pi(a|s)$$

Another (more common) version makes $d_\rho^\pi(s)$ a distribution over $s$

→ Just change the $\rho(s)$ in the first equation by $(1-\gamma)\rho(s)$

# Dynamic Programming

**Goal:** find optimal policy

**Key quantity:** $V_i^\star(s)$ = the optimal discounted total reward starting from state $s$ supposed that $i$ more steps can be executed

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

For $i = 1, 2, 3$ ...

$$V_i^\star(s) = \max_a \; R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, V_{i-1}^\star(s') \quad \forall s$$

Value Iteration

$$V^\star(s) = \lim_{i \to \infty} V_i^\star(s) \qquad \pi^\star(s) = \operatorname{argmax}_a \; R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, V^\star(s')$$

# Bellman Optimality Equation

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

$$\boxed{\begin{aligned} V^\star(s) &= \max_a Q^\star(s,a) \\ Q^\star(s,a) &= R(s,a) + \sum_{s'} P(s'|s,a)\, V^\star(s') \end{aligned}}$$

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

# Recap

$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|\, s) Q^\pi(s,a)$$

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

Bellman Equation
(Value Iteration)

$$d_\rho^\pi(s,a) = d_\rho^\pi(s)\pi(a|s)$$

$$d_\rho^\pi(s) = (1-\gamma)\rho(s) + \gamma \sum_{s',a'} d_\rho^\pi(s',a') P(s|s',a')$$

Reverse Bellman Equation

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

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

Bellman Optimality Equation
(Value Iteration)

# If Bellman Equations Only Hold Approximately

If $\left|\hat{V}(s) - \max_a \left(R(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}\left[\hat{V}(s')\right]\right)\right| \leq \epsilon \quad \forall s$

then $\left|\hat{V}(s) - V^\star(s)\right| \leq \dfrac{\epsilon}{1-\gamma} \quad \forall s$

# Policy Iteration

# Policy Iteration

**Policy Iteration**

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

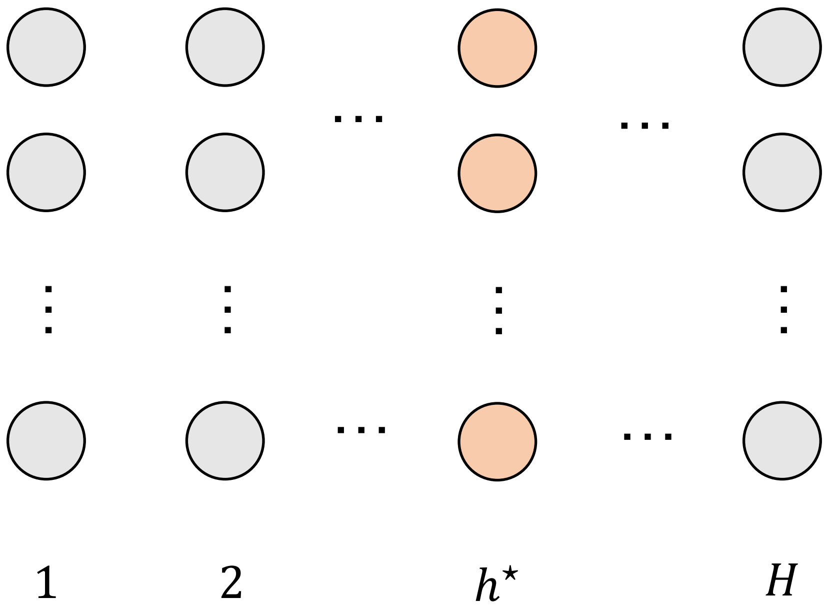
$$\forall s, \qquad \pi^{(k+1)}(s) \ \leftarrow \operatorname*{argmax}_a Q^{\pi^{(k)}}(s,a)$$

**Theorem (monotonic improvement).** Policy Iteration ensures

$$\forall s, \qquad V^{\pi^{(k+1)}}(s) \geq V^{\pi^{(k)}}(s)$$

Below we will establish a more general lemma (not only show monotonic improvement, but also show *how much* the improvement is).

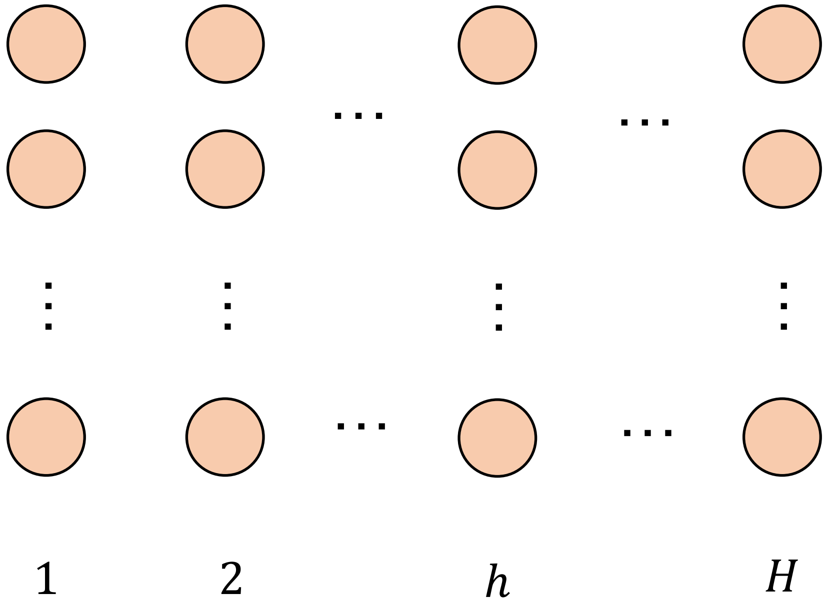
# Single-Step Policy Modification under Fixed Horizon




Assume $\pi'_h(\cdot\,|s) = \pi_h(\cdot|s)$ for all $h \neq h^\star$

$$\mathbb{E}_{s\sim\rho}\left[V_1^{\pi'}(s)\right] - \mathbb{E}_{s\sim\rho}[V_1^{\pi}(s)] = ?$$

# All-Step Policy Modification under Fixed Horizon

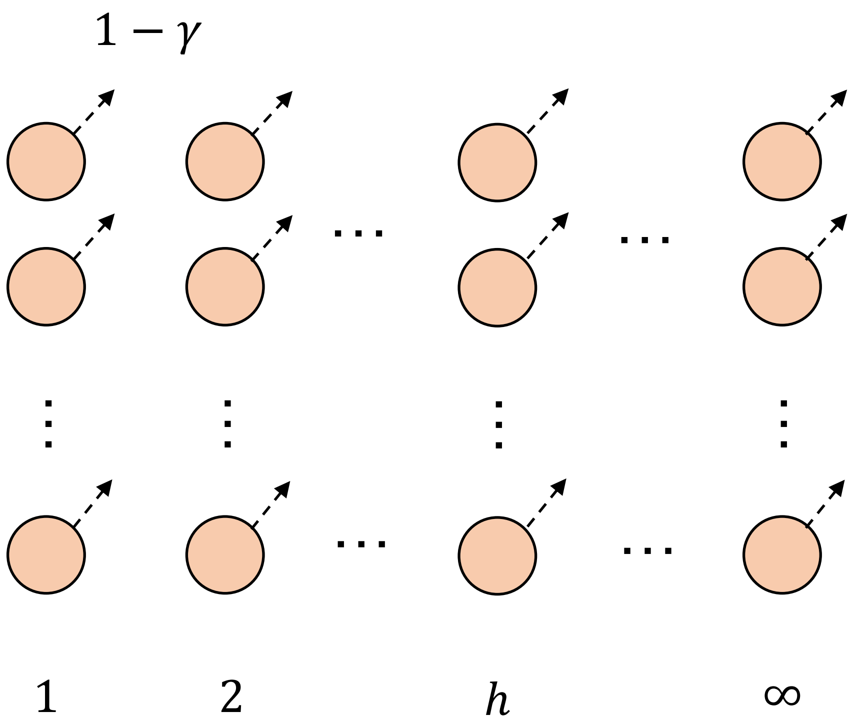



Let $\pi^{(h)}$ be a Markov policy such that it is

$$\begin{cases} \text{same as } \pi' \text{ in steps } 1 \text{ to } h-1 \\ \text{same as } \pi \text{ in steps } h \text{ to } H \end{cases}$$

$\pi' = \pi^{(H+1)}$ and $\pi = \pi^{(1)}$

# Discounted Total Reward Setting



# Performance / Value Difference Lemma

For any two stationary policies $\pi'$ and $\pi$ in the discounted total reward setting,

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

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

# Modified Policy Iteration

**Bellman Operator** $\mathcal{T}^{\pi}$

$$(\mathcal{T}^{\pi}V)(s) = \sum_{a} \pi(a|s)\left(R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, V(s')\right)$$

**Greedy Operator** $\mathcal{G}$

$$(\mathcal{G}V)(s) = \operatorname*{argmax}_{a} \quad R(s,a) + \gamma \sum_{s'} P(s'|s,a)\, V(s')$$

These algorithms just differ in the relative speed between policy and value updates

Value Iteration:

$$\pi_{k+1} = \mathcal{G}V_k$$

$$V_{k+1} = \mathcal{T}^{\pi_{k+1}}V_k$$

Policy Iteration:

$$\pi_{k+1} = \mathcal{G}V_k$$

$$V_{k+1} = (\mathcal{T}^{\pi_{k+1}})^{\infty}V_k$$

Policy update

Value update

MPI:

$$\pi_{k+1} = \mathcal{G}V_k$$

$$V_{k+1} = (\mathcal{T}^{\pi_{k+1}})^{m}V_k$$

# Summary for the Basics of MDPs

- MDPs model decision-making problems where the return depends on sequences of actions.
- “State” summarizes all the information needed to make decisions (in the fixed-horizon setting, the step index is also important).
- While the number of **action sequence is exponential** in the horizon length, the optimal policy can be computed in **poly(#state, #actions, horizon length)** time using dynamic programming techniques (Value Iteration, Bellman optimality equation).
- Interaction Protocols: fixed-horizon, goal-oriented, infinite-horizon
- Performance Metrics: total reward, average reward, discounted total reward
- Policies: history-dependent, Markov, stationary
- Bellman equation, Reverse Bellman equation, Bellman optimality equation
- Policy Iteration and Performance Difference Lemma
- Unifying Value Iteration and Policy Iteration