

# Lecture 1 - Markov Decision Process

Guiliang Liu

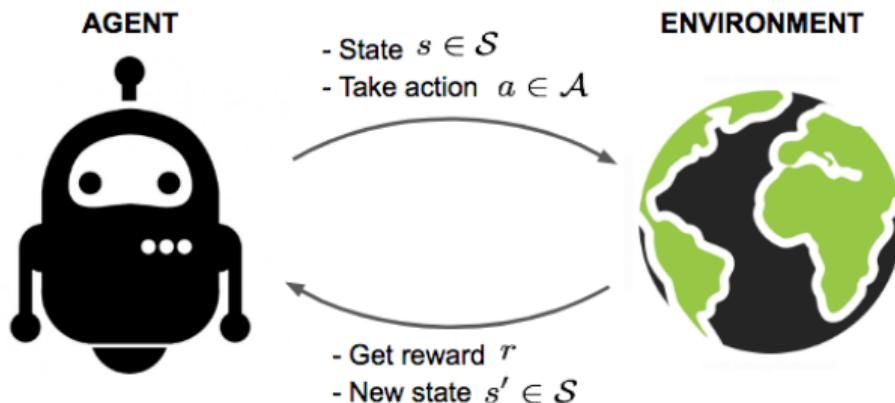
The Chinese University of Hong Kong, Shenzhen

DDA4230: Reinforcement Learning

Course Page: [\[Click\]](#)

# Reinforcement Learning

A reinforcement learning **agent** interacts with its **world** and from that learns how to **maximize cumulative reward** over time.



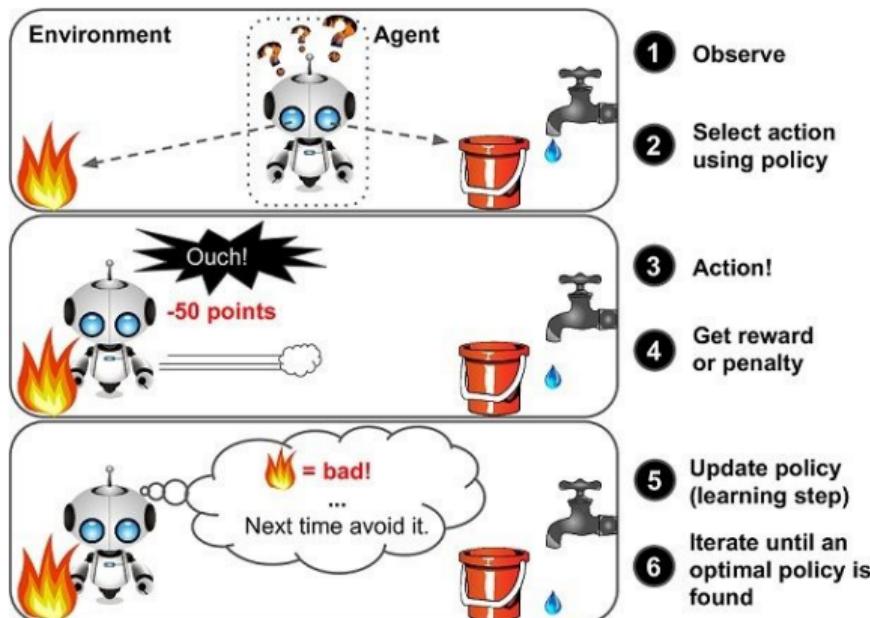
Reference: <https://lilianweng.github.io/posts/2018-02-19-rl-overview/>



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Reinforcement Learning



No **teacher** or **knowledge** of the world model. Learn to act through **trial and error**.

- E1: Agent + Fire  $\rightarrow$  -50.
- E2: Agent + Bucket + Fire  $\rightarrow$  -50.
- E3: Agent + Bucket + Water + fire  $\rightarrow$  +50.
- E4: What will the agent do?

Reference: <https://www.odinschool.com/>



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Connection with other learning problems

## Reinforcement Learning

- No **teacher** or **knowledge** of the world model (e.g., environment).

## Supervised Learning



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Connection with other learning problems

## Reinforcement Learning

- No **teacher** or **knowledge** of the world model (e.g., environment).
- **Interact** with the environment.

## Supervised Learning

- Given a **dataset**, which consists of examples and labels (knowledge).



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Connection with other learning problems

## Reinforcement Learning

- No **teacher** or **knowledge** of the world model (e.g., environment).
- **Interact** with the environment.
- Making **sequential decisions**.

## Supervised Learning

- Given a **dataset**, which consists of examples and labels (knowledge).
- **No interaction**. Only an **offline** dataset.



香港中文大學(深圳)  
The Chinese University of Hong Kong, Shenzhen

# Connection with other learning problems

## Reinforcement Learning

- No **teacher** or **knowledge** of the world model (e.g., environment).
- **Interact** with the environment.
- Making **sequential decisions**.
- Learn a policy to maximize **cumulative** rewards.

## Supervised Learning

- Given a **dataset**, which consists of examples and labels (knowledge).
- **No interaction**. Only an **offline** dataset.
- Making **one-step predictions**.



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Connection with other learning problems

## Reinforcement Learning

- No **teacher** or **knowledge** of the world model (e.g., environment).
- **Interact** with the environment.
- Making **sequential decisions**.
- Learn a policy to maximize **cumulative** rewards.

## Supervised Learning

- Given a **dataset**, which consists of examples and labels (knowledge).
- **No interaction**. Only an **offline** dataset.
- Making **one-step predictions**.
- Learn a predictor to maximizing **point-wise prediction accuracy**.



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Connection with other learning problems

## Reinforcement Learning

- No **teacher** or **knowledge** of the world model (e.g., environment).
- **Interact** with the environment.
- Making **sequential decisions**.
- Learn a policy to maximize **cumulative** rewards.

## Unsupervised (Contrastive) Learning

- Given a **dataset**, which consists of only examples (No labels).
- **No interaction**. Only an **offline** dataset.
- Learning **latent structure** of dataset.
- Learn the **latent features** for classification or identification.



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Challenges in Reinforcement Learning

- How to balance exploration and exploitation?
- How to generalize its experience?
- How to model the delayed consequences of actions (look-ahead).



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Markov Decision Process (MDP)

Discrete-time Markov decision process (MDP), denoted as the tuple  $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \rho_0, \gamma)$ .

- $\mathcal{S}$  the state space;
- $\mathcal{A}$  the action space.  $\mathcal{A}$  can depend on the state  $s$  for  $s \in \mathcal{S}$ ;
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$  the environment transition probability function;
- $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \Delta(\mathbb{R})$  the reward function;
- $\rho_0 \in \Delta(\mathcal{S})$  the initial state distribution;
- $\gamma \in [0, 1]$  the discount factor.

Note that  $\Delta(\mathcal{X})$  denotes the set of all distributions over set  $\mathcal{X}$



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Markov Decision Process (MDP)

Map the Reinforcement Learning (RL) environment to an MDP.

Application 1: Autonomous Driving.



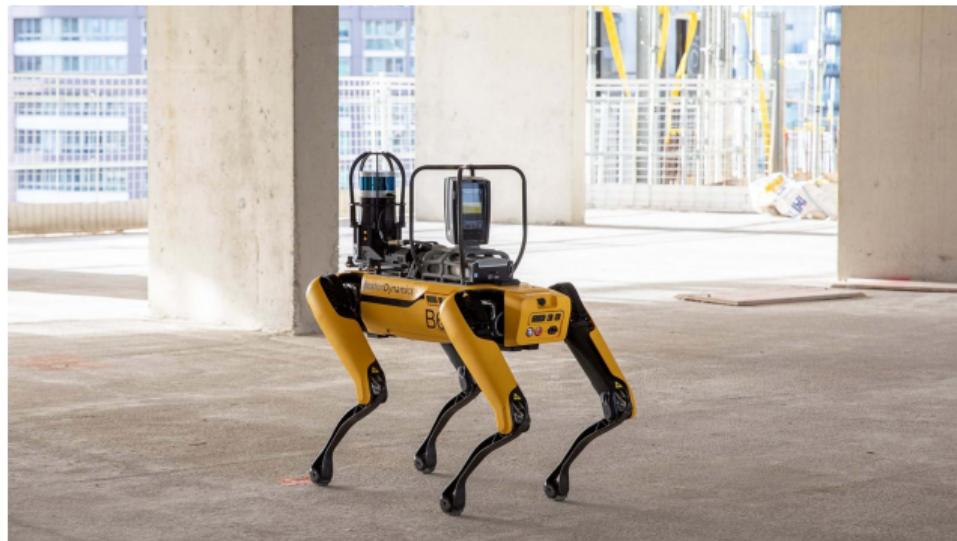
香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Markov Decision Process (MDP)

Map the Reinforcement Learning (RL) environment to an MDP.

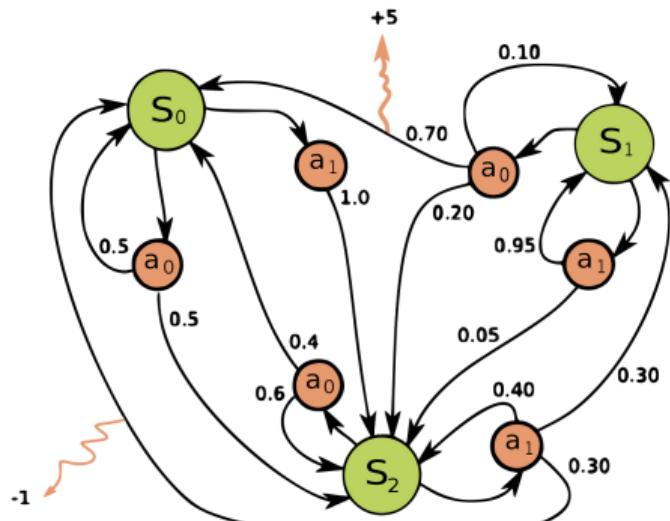
Application 2: Robot Control.



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Markov Decision Process (MDP)



For  $t = 0, 1, \dots$ , start with  $s_0 \sim \rho_0$ .

- The agent observes the state  $s_t$ ;
- The agent chooses an action  $a_t$ ;
- The agent receives the reward  
 $r_t \sim \mathcal{R}(s_t, a_t)$ ;
- The environment transitions to a subsequent state:  $s_{t+1} \sim \mathcal{T}(s_t, a_t)$ .

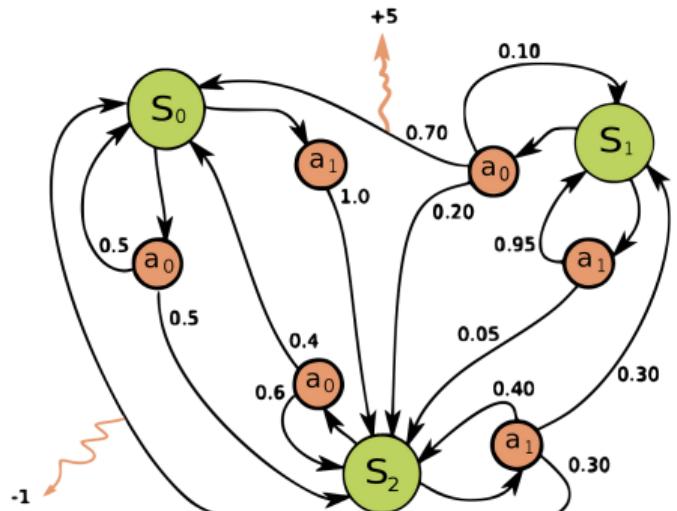
Reference: [https://en.wikipedia.org/wiki/Markov\\_decision\\_process](https://en.wikipedia.org/wiki/Markov_decision_process)



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Markov Decision Process (MDP)



Trajectory generation.

- This process generates the sequence  $s_0, a_0, r_0, s_1, \dots$ , until when  $s_T$  is a terminal state, or indefinitely.
- The sequence up to time  $t$  is defined as the **trajectory** indexed by  $t$ , as  $\tau_t = (s_0, a_0, r_0, s_1, \dots, r_t)$ .

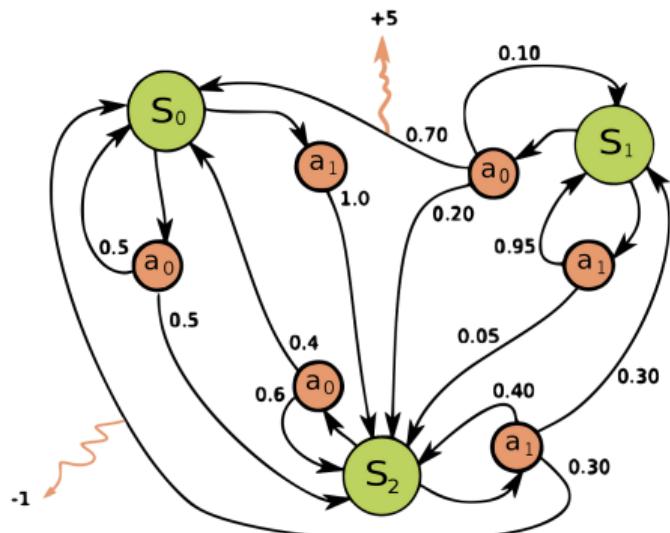
Reference: [https://en.wikipedia.org/wiki/Markov\\_decision\\_process](https://en.wikipedia.org/wiki/Markov_decision_process)



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Markov Decision Process (MDP)



In an MDP, the optimal choice of action  $a_t$  depends only on the state  $s_t$ . A **policy** defines the mapping from  $\mathcal{S}$  to  $\mathcal{A}$ .

- **Stochastic policy:**  $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$
- **Deterministic policy:**  $a_t = \pi(s_t)$

Reference: [https://en.wikipedia.org/wiki/Markov\\_decision\\_process](https://en.wikipedia.org/wiki/Markov_decision_process)



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Important Variables in MDPs

- The **return** is defined as the **discounted cumulative reward** as a **random variable**.

$$R_t = \sum_{t' \geq t}^{\infty} \gamma^{t'} r_{t'}.$$

- The **expectation of the return** is the objective to be maximized by the agent

$$J = \mathbb{E}_{s_t, a_t, r_t, t \geq 0} [R_0] = \mathbb{E}_{s_t, a_t, r_t, t \geq 0} \left[ \sum_{t=0}^{\infty} \gamma^t r_t \right] \text{ and } \pi = \arg \max_{\pi} J$$

The expectation is subject to random variables  $(s_0, a_0, r_0, s_1, \dots, r_\infty)$  with a complicate state space  $(\mathcal{S} \times \mathcal{A} \times \mathcal{R})^T$ . The optimization problem is **not characterisable** (non-linear, non-convex, non-quadratic..) in general.



# Important Variables in MDPs

- The **action value Q-function** of a given policy  $\pi$

$$Q^\pi(s, a) = \mathbb{E}_{s_t, a_t, r_t, t \geq 0} \left[ \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 = s, a_0 = a \right]$$

which is the expected return of policy  $\pi$  at state  $s$  after taking action  $a$ .

- The **state value function** of a given policy  $\pi$

$$V^\pi(s) = \mathbb{E}_{a \sim \pi(s)} [Q^\pi(s, a)]$$

which is the expected return given the initial state only.



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Important Variables in MDPs

Example: Game Go



- Rewards:  $r_0 = 0, r_2 = 0, \dots, r_{T-1} = 0$ . If win  $r_T = 1$  otherwise  $r_T = 0$ .
- Discount:  $\gamma \rightarrow 1$ .
- $Q^\pi(s, a)$ : winning probability of making a move  $a$  under state  $s$  by following policy  $\pi$ .
- $V^\pi(s)$ : winning probability of at the state  $s$  by following policy  $\pi$ .



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Important Variables in MDPs

- The advantage function of a given policy  $\pi$  can be defined as:

$$A^\pi(s, a) = Q^\pi(s, a) - V^\pi(s)$$

- Based on the value functions, define the temporal-difference error

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t).$$

Remark:

- $r_t$ : rewards at  $t$ .
- $V(s_{t+1})$ : expected cumulative rewards at  $t+1, t+2, \dots$
- $V(s_t)$ : expected rewards at  $t, t+1, t+2, \dots$
- if  $\pi$  is optimal,  $\mathbb{E}[\delta_t] = 0$ .



香港中文大學(深圳)  
The Chinese University of Hong Kong, Shenzhen

# The Bellman Equation

- State-value Bellman equation (named after Richard E. Bellman):

$$V(s_t) = \mathbb{E}[r_t + \gamma V(s_{t+1})] \quad \text{and} \quad V(s_t) = \mathbb{E}[r_t].$$

for non-terminal and terminal states, respectively.

- Action-value Bellman equation:

$$Q(s_t, a_t) = \mathbb{E}[r_t + \gamma Q(s_{t+1}, a) \mid a \sim \pi(a \mid s_{t+1})] \quad \text{and} \quad Q(s_t, a_t) = \mathbb{E}[r_t]$$

for non-terminal and terminal states, respectively.



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Stationarity of MDPs and Agents

## Stationarity MDP

- Markovian dynamics of  $s_{t+1}$  depends **only** on  $s_t$  and  $a_t$  as  $s_{t+1} \sim \mathcal{T}(s_t, a_t)$ .
- The reward  $r_t$  depends **only** on  $s_t$  and  $a_t$  as  $r_t \sim \mathcal{R}(s_t, a_t)$ .
- A **policy** is stationary if the action **depends only** on the state:  $a_t \sim \pi(s_t)$ .

## Non-Stationarity MDP

- The transition dynamics depend on the time  $t$  as  $s_{t+1} \sim \mathcal{T}_t(s_t, a_t)$ .
- The reward depend on the **time  $t$**  as  $r_t \sim \mathcal{R}_t(s_t, a_t)$ .
- A **policy** is non-stationary if the action **also depends** on  $t$ :  $a_t \sim \pi_t(s_t)$ .

Remark: If there is an optimal policy, **there is an optimal stationary policy** if the process has a non-fixed horizon.



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# State and Action Spaces

Two common settings of the state space and the action space are:

- $\mathcal{S} \in \mathbb{R}^n$  the *n* dimensional state space,  $\mathcal{A} \in \mathbb{R}^m$  the *m* dimensional action space;
- $\mathcal{S} \in [n]$  the *size-n* discrete state space,  $\mathcal{A} \in [m]$  the *size-m* discrete action space.



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Stochasticity of MDPs and agents

The **stochasticity of a Markov chain** given the MDP and the policy can come from three components:

- Stochastic Markovian dynamics:  $P_{\mathcal{T}}(s_{t+1}|s_t, a_t)$ ;
- Stochastic rewards:  $P_{\mathcal{R}}(r_{t+1}|s_t, a_t)$ ;
- Stochastic policies:  $\pi(a_t|s_t)$ ;



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Discount of Rewards

The discount factor  $\gamma \in [0, 1]$  balances the short-term and long-term rewards. When the objective is discounted ( $\gamma < 1$ )

$$R_0 = r_0 + \gamma r_1 + \gamma^2 r_2 + \dots,$$

Two extreme cases are  $\gamma = 0$  and  $\gamma = 1$ , where the former corresponds to  $R_0 = r_0$  as a one-step MDP and the latter corresponds to  $R_0 = r_0 + r_1 + r_2 + \dots$ .



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Agents of RL

We can classify our agents in a number of ways:

| Agent type   | Policy   | Value Function | Model |
|--------------|----------|----------------|-------|
| Value-based  | Implicit | ✓              | ?     |
| Policy-based | ✓        | ✗              | ?     |
| Actor-critic | ✓        | ✓              | ?     |
| Model-based  | ?        | ?              | ✓     |
| Model-free   | ?        | ?              | ✗     |

- ✓ indicates that the agent has the component.
- ✗ indicates that it must not have the component.
- ? indicates that the agent may have that component.

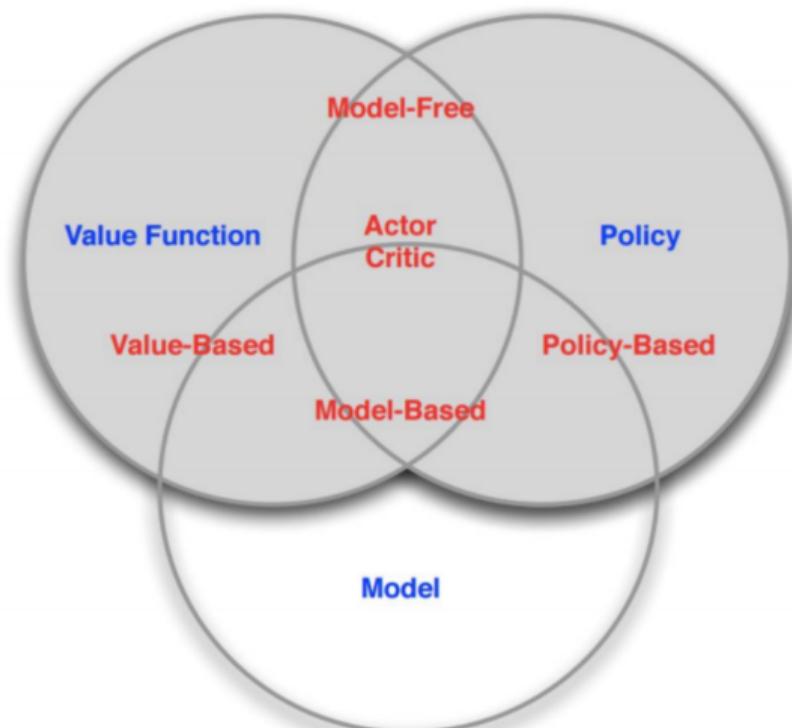


香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Agents of RL

Classification of different reinforcement learning agents.



香港中文大學(深圳)  
The Chinese University of Hong Kong, Shenzhen

# Classification of Markov structures

| Markov structure                       |     | Do actions have influence over<br>the state transitions? |   |
|--|-----|--|---|
|  |     | NO   | YES   |
| Are the states<br>fully<br>observable? | YES | Markov process<br>(Markov chain)                         | Markov decision process                         |
|  | NO  | Hidden Markov model                                      | Partially observable<br>Markov decision process |



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Classification of Markov structures

Fully observable Markov decision process



- All the players are observable.



香港中文大學(深圳)  
The Chinese University of Hong Kong, Shenzhen

# Classification of Markov structures

Partially observable Markov decision process



- Only a partial number of the players are observable.



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

# Question and Answering (Q&A)



香港中文大學(深圳)  
The Chinese University of Hong Kong, Shenzhen