AIC/RL – Markov Decision Processes (Part I)

Freek Stulp

Université Paris-Saclay

Outline

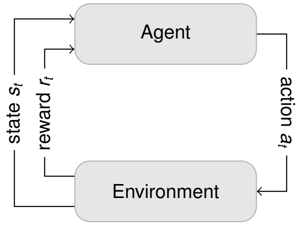


Figure : Agent-environment interface

Interaction: $s_1 \rightarrow a_1 \rightarrow r_2, s_2 \rightarrow a_2 \rightarrow r_3, s_3 \rightarrow \ldots \rightarrow a_{T-1} \rightarrow r_T, s_T$

How does the environment behave?	Markov Decision Process	$\{S, A, \mathcal{P}, \mathcal{R}\}$
How does the agent behave?	policy	$\pi(s, a)$
What should the agent do?	optimize returns!	$argmax_\pi \, \mathrm{E}\{R \mid \pi\}$

Outline

1. Markov Decision Process
2. Policy
3. Returns
4. Dimensions of RL

How does the environment behave?	Markov Decision Process	$\{S, A, \mathcal{P}, \mathcal{R}\}$
How does the agent behave?	policy	$\pi(s, a)$
What should the agent do?	optimize returns!	$argmax_{\pi}\ \mathrm{E}\{R\|\pi\}$

Markov Decision Process (SUBA3.6)

S State space *"all possible states the environment can have"*

Gridworld example

$$S = \{s^0, s^1, s^2, s^3, s^4, s^5, s^6, s^7\}$$

s^0	s^1	s^2	s^3
s^4	s^5	s^6	s^7

Markov Decision Process (SuBa3.6)

S State space *"all possible states the environment can have"*

A Action space *"all possible actions the agent can take"*

Gridworld example

$$S = \{s^0, s^1, s^2, s^3, s^4, s^5, s^6, s^7\}$$

$$A = \{a^0, a^1, a^2, a^3\} = \{a_{\text{UP}}, a_{\text{RIGHT}}, a_{\text{LEFT}}, a_{\text{DOWN}}\}$$

s^0	s^1	s^2	s^3
s^4	s^5	s^6	s^7

Markov Decision Process (SuBa3.6)

- S State space — *"all possible states the environment can have"*
- A Action space — *"all possible actions the agent can take"*
- $\mathcal{P}^a_{ss'}$ Transition function — *"probability of going from s to s′ when doing a"*

$$\mathcal{P}^a_{ss'} = Pr\{s_{t+1} = s' | s_t = s, a_t = a\}$$

Gridworld example

$$S = \{s^0, s^1, s^2, s^3, s^4, s^5, s^6, s^7\}$$

$$A = \{a^0, a^1, a^2, a^3\} = \{a_{\text{UP}}, a_{\text{RIGHT}}, a_{\text{LEFT}}, a_{\text{DOWN}}\}$$

$$P^{a_{LEFT}}_{s^2 s^1} = 0.8, \quad P^{a_{LEFT}}_{s^2 s^2} = 0.2, \quad \text{etc.}$$

s^0	s^1	s^2	s^3
s^4	s^5	s^6	s^7

Markov Decision Process (SuBa3.6)

S State space — *"all possible states the environment can have"*

A Action space — *"all possible actions the agent can take"*

$\mathcal{P}^a_{ss'}$ Transition function — *"probability of going from s to s′ when doing a"*

$$\mathcal{P}^a_{ss'} = Pr\{s_{t+1} = s' | s_t = s, a_t = a\}$$

$\mathcal{R}^a_{ss'}$ Reward function — *"immediate reward in s / when going from s to s′"*

$$\mathcal{R}^a_{ss'} = \mathrm{E}\{r_{t+1} | s_t = s, a_t = a, s_{t+1} = s'\}$$

Reward function

Expected immediate reward in state s

$$\mathcal{R}_s = \mathrm{E}\{r_{t+1} | s_t = s\} \tag{1}$$

Expected immediate reward for going from s to s′

$$\mathcal{R}_{ss'} = \mathrm{E}\{r_{t+1} | s_t = s, s_{t+1} = s'\} \tag{2}$$

Exp. imm. reward for performing a in s which leads to s′

$$\mathcal{R}^a_{ss'} = \mathrm{E}\{r_{t+1} | s_t = s, a_t = a, s_{t+1} = s'\} \tag{3}$$

In our code we use (2)

Markov Decision Process (SuBa3.6)

S State space — *"all possible states the environment can have"*

A Action space — *"all possible actions the agent can take"*

$\mathcal{P}^a_{ss'}$ Transition function — *"probability of going from s to s′ when doing a"*

$$\mathcal{P}^a_{ss'} = Pr\{s_{t+1} = s' | s_t = s, a_t = a\}$$

$\mathcal{R}^a_{ss'}$ Reward function — *"immediate reward in s / when going from s to s′"*

$$\mathcal{R}^a_{ss'} = \mathrm{E}\{r_{t+1} | s_t = s, a_t = a, s_{t+1} = s'\}$$

Gridworld example

$$\forall s \Rightarrow \mathrm{E}\{r_{t+1} \mid s_t = s,\ s_{t+1} = s^0\} = 100 \quad (1)$$

$$\forall s, s',\ \ s' \neq s^0 \Rightarrow \mathrm{E}\{r_{t+1} \mid s_t = s,\ s_{t+1} = s'\} = -1 \quad (2)$$

s^0	100	s^1	-1	s^2	-1	s^3
100				-1		
s^4		s^5		s^6		s^7

Markov Decision Process (SuBa3.6)

S State space — *"all possible states the environment can have"*

$T \subset S$ Terminal states — *"in which states does an episode end"*

$\mathcal{I}_s$ Initial state distribution — *"probabilities of starting in each state"*

$$\mathcal{I}_s = Pr\{s_t = s | t = 1\}$$

Gridworld example

$$S = \{s^0, s^1, s^2, s^3, s^4, s^5, s^6, s^7\}$$

$$T = \{s^0\}$$

$$\mathcal{I}_s = \begin{cases} 0 & \text{if } s = s^0 \\ \frac{1}{7} & \text{otherwise} \end{cases}$$

s^0	s^1	s^2	s^3
s^4	s^5	s^6	s^7

Policy

- Behavior of environment as MDP: $\{S, A, \mathcal{P}, \mathcal{R}\}$ (and $\{T, \mathcal{I}\}$)
- Behavior of agent a a policy: $\pi(s, a) = Pr\{a_t = a | s_t = s\}$
  "probability of doing action a in state s"

Policy

$\pi(s, UP)$

T	0.1	0.1	0.1
0.7	0.1	0.1	0.1

$\pi(s, LEFT)$

T	0.7	0.7	0.7
0.1	0.7	0.7	0.7

$\pi(s, RIGHT)$

T	0.1	0.1	0.1
0.1	0.1	0.1	0.1

$\pi(s, DOWN)$

T	0.1	0.1	0.1
0.1	0.1	0.1	0.1

$\text{argmax}_a\, \pi(s, a)$

T	$<$	$<$	$<$
$\wedge$	$<$	$<$	$<$

Policy

- Behavior of environment as MDP: $\{S, A, \mathcal{P}, \mathcal{R}\}$ (and $\{T, \mathcal{I}\}$)
- Behavior of agent a a policy: $\pi(s, a) = Pr\{a_t = a | s_t = s\}$
  "probability of doing action a in state s"

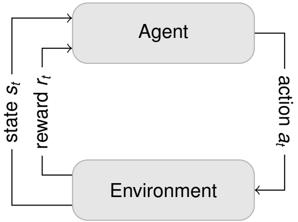


Figure : Agent-environment interface

Interaction: $s_1 \rightarrow a_1 \rightarrow r_2, s_2 \rightarrow a_2 \rightarrow r_3, s_3 \rightarrow \cdots \rightarrow a_{T-1} \rightarrow r_T, s_T$

Policy

- Behavior of environment as MDP: $\{S, A, \mathcal{P}, \mathcal{R}\}$ (and $\{T, \mathcal{I}\}$)
- Behavior of agent a a policy: $\pi(s, a) = Pr\{a_t = a | s_t = s\}$

  "probability of doing action a in state s"

- We've defined the interfaces for the environment and the agent...

  now what is the aim of the agent?

  - Informal: *"optimize rewards by choosing the right actions"*
  - Formal: $\pi^* = argmax_\pi \, \mathrm{E}\{R|\pi\}$ (next two slides)

Returns (SuBa3.3)

- The return R is the (discounted) sum over immediate rewards r_t

$$R_t = r_{t+1} + r_{t+2} + r_{t+3} + \cdots + r_T \qquad \text{Episodic Tasks} \qquad (3)$$

$$= \sum_{k=0}^{T} r_k \qquad (4)$$

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \ldots \qquad \text{Continued Tasks} \qquad (5)$$

$$= \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \qquad \text{with } 0 \leq \gamma \leq 1 \qquad (6)$$

$$R_t = \sum_{k=0}^{T} \gamma^k r_{t+k+1} \qquad \text{Unified (SuBa3.4)} \qquad (7)$$

- Discount factor γ: prefer rewards now over rewards in the future
  - $\gamma = 1 \Rightarrow$ 100EUR next year as good as 100EUR now
  - $\gamma = 0 \Rightarrow$ only the next reward counts: “hedonism”

Optimizing returns

- Aim of RL
  - Find the policy that optimizes the expected return
  - 1 simple formula ⇒ 50 years of research!

$$\pi^* = argmax_\pi \, \mathrm{E}\{R|\pi\} \tag{8}$$

- Why is it difficult?
  - What is the state space?
  - Gigantic states/action spaces
  - What exactly is the reward function?
  - Unpredictable environments (e.g. due to multiple agents)
  - Best discount factor?
- Some optimal policies that would be nice to have (increasing difficulty)
  - Optimal autonomous driving (safe, fast, comfortable)
  - Optimal trading on the stock-market
  - Policy that optimizes your happiness during your life
  - Policy that optimizes long-term hapiness of humanity
    - Clearly, discount factor too low now...
- What makes RL easy/difficult ⇒ dimensions of RL

Dimensions of RL

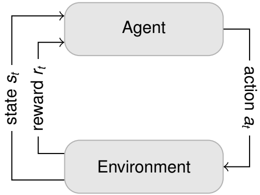


Figure : Agent-environment interface

Finite	vs.	Infinite
Discrete	vs.	Continuous
Model-based	vs.	Model-free
Deterministic	vs.	Stochastic
Episodic	vs.	Continuing
Markovian	vs.	Non-Markovian
Observable	vs.	Partially Observ.

Finite (Discrete) vs. Infinite (Continuous)

- Are the state and action spaces finite or infinite?
- Are the state and action spaces discrete or continuous?
  - Finite/Discrete: chess, flipping a coin, grid world
  - Infinite/Continuous: robot control

Dimensions of RL

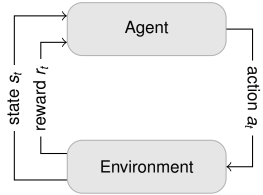


Figure : Agent-environment interface

Finite	vs.	Infinite
Discrete	vs.	Continuous
Model-based	vs.	Model-free
Deterministic	vs.	Stochastic
Episodic	vs.	Continuing
Markovian	vs.	Non-Markovian
Observable	vs.	Partially Observ.

Model-based vs. Model-free

- Do the algorithms that find the optimal policy have access to the MDP?
  - Model-based: optimal control, dynamic programming
  - Model-free: reinforcement learning

Dimensions of RL

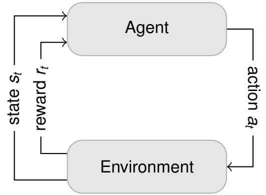


Figure : Agent-environment interface

Finite	vs.	Infinite
Discrete	vs.	Continuous
Model-based	vs.	Model-free
Deterministic	vs.	Stochastic
Episodic	vs.	Continuing
Markovian	vs.	Non-Markovian
Observable	vs.	Partially Observ.

Deterministic vs. Stochastic

- Does executing the same action in the same state always lead to the next same state?
  - Deterministic: chess agains a computer
  - Stochastic: robot control, chess agains a human opponent

Dimensions of RL

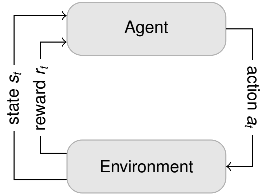


Figure : Agent-environment interface

Finite	vs.	Infinite
Discrete	vs.	Continuous
Model-based	vs.	Model-free
Deterministic	vs.	Stochastic
Episodic	vs.	Continuing
Markovian	vs.	Non-Markovian
Observable	vs.	Partially Observ.

Episodic vs. Continuing

- Does an interaction always end in a terminal state?
  - Episodic: grid world, flip coin
  - Continuing: driving a car

Dimensions of RL

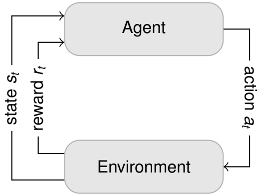


Figure : Agent-environment interface

Finite	vs.	Infinite
Discrete	vs.	Continuous
Model-based	vs.	Model-free
Deterministic	vs.	Stochastic
Episodic	vs.	Continuing
Markovian	vs.	Non-Markovian
Observable	vs.	Partially Observ.

Markov Property (SuBa3.5)

- Does the optimal policy depend only on the current observable state?
  - Markovian: chess, grid world
  - non-Markovian: game of memory, driving a car, life

Dimensions of RL

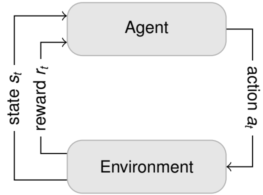


Figure : Agent-environment interface

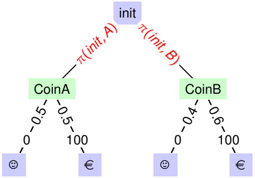
Finite	vs.	Infinite
Discrete	vs.	Continuous
Model-based	vs.	Model-free
Deterministic	vs.	Stochastic
Episodic	vs.	Continuing
Markovian	vs.	Non-Markovian
Observable	vs.	Partially Observ.

Observable vs. Partially Observable

- Can an agent always perfectly observe the state?
  - Observable: chess, grid world
  - Partially Observable: driving a car, life

Summary: Flipping coins example

- Choose one of two coins randomly
  - CoinA is fair, i.e. 50%/50%
  - but CoinB gives tails 60% of the time
- If you get tails you get 100, if heads 0



Corresponding MDP

$$S = \{\ init,\ \ \text{☺}\ (heads),\ \ \text{€}\ (tails)\ \}$$

$$A = \{\ CoinA,\ \ CoinB\ \}\ \ (which\ one\ do\ you\ choose?)$$

$$T = \{\ \text{☺},\ \ \text{€}\ \}$$

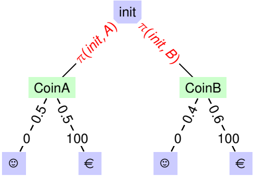
$$\mathcal{I}_s = \begin{cases} 1 & \text{if } s = init \\ 0 & \text{otherwise} \end{cases}$$

$$P^a_{ss'} = [P^{CoinA}_{init\ \text{☺}} = 0.5,\quad P^{CoinA}_{init\ \text{€}} = 0.5,\quad P^{CoinB}_{init\ \text{☺}} = 0.4,\quad P^{CoinB}_{init\ \text{€}} = 0.6]$$

$$\mathcal{R}_{ss'} = [R_{init,\text{☺}} = 0,\quad R_{init,\text{€}} = 100]$$

Summary: Flipping coins example

- Choose one of two coins randomly
  - CoinA is fair, i.e. 50%/50%
  - but CoinB gives tails 60% of the time
- If you get tails you get 100, if heads 0



Policy

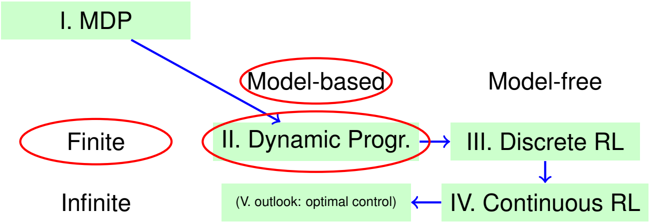
- Random policy

$$\pi(s,a) = [\pi(init, CoinA) = 0.5, \quad \pi(init, CoinB) = 0.5]$$

- Optimal policy (deterministic)

$$\pi^*(s,a) = [\pi(init, CoinA) = 0.0, \quad \pi(init, CoinB) = 1.0]$$

Up next in the lecture



- Algorithms based on “Dynamic Programming” to find optimal policies
  - for finite MDPs $\Rightarrow$ with discrete state space S and action space A
  - with model-based algorithms $\Rightarrow$ they need to know $\mathcal{P}^a_{ss'}$ and $\mathcal{R}^a_{ss'}$