Introduction to Markov Decision Processes and Reinforcement Learning

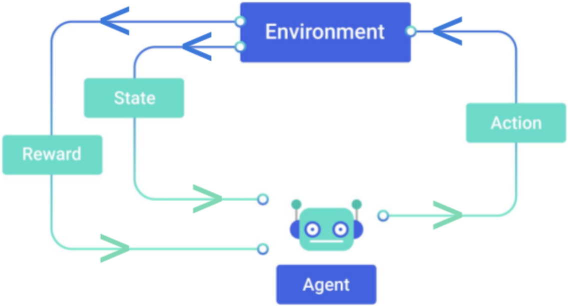
Pedro A. Santos

M4AI May 9th 2025

Contents

Contents

Introduction



$x_0, a_0, r_0, x_1, a_1, r_1, x_2, a_2, r_2, x_3 \ \ldots$

Contents

Markov Decision Processes

Definition (MDP)

A Markov decision process is a tuple $\{\mathcal{X}, \mathcal{A}, \mathcal{P}, r, \gamma\}$, where

- $\mathcal{X}$ denotes a finite set of n states;
- $\mathcal{A}$ a finite set of m actions;
- $\mathcal{P}$ is a set of $n \times n$ stochastic matrices P_a associated with each action $a \in \mathcal{A}$ with entries $[P_a]_{x,y} \in [0, 1]$ representing the probability that the state transitions from x to y given that the action a was performed;
- $R : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ is a reward stochastic mapping;
- $\gamma \in [0, 1]$ is a discount factor.

Many applications

- Inventory Management
- Equipment Repair
- Road maintenance
- Managing Assets
- Recycle bin collection
- Vehicle control
- Pandemic control
- Bird nest abandonment

Any situation of repeated decision under uncertainty where the actions and effects depend only on the current state - Markov Property - can fit into this framework...

The Markov Zoo

States	Actions	State Knowledge	
No	Yes		Multi-armed Bandit
Yes	No	Yes	Markov Process
Yes	No	No	Hidden MM
Yes	Yes	Yes	MDP
Yes	Yes	No	POMDP

An Example

A simple example

We have two states, $1, 2$. Independently of the state, there are two actions available, a^1 and a^2. In state 1, by choosing a^1, the agent gains an immediate reward of 6, and the system in the next decision point is in state 1 with probability $1/2$ and state 2 with probability $1/2$. If the agent chooses action a^2, the agent gains an immediate reward of 10, and the system changes to state 2 with probability 1. In state 2, choosing a^1, gives a negative reward of -10, and the system changes to state 1 with probability $1/10$. Action a^2 gives a negative reward of -1 and the system stays in state 2 with certainty.

An Example

- Decision epochs: $\{0, 1, 2, \dots N\}$, $N \leq \infty$
- State space: $\{1, 2\}$;
- Action set: $\{a^1, a^2\}$;
- Probabilities:

$$P_{a^1} = \begin{bmatrix} 0.5 & 0.1 \\ 0.5 & 0.9 \end{bmatrix}, \; P_{a^2} = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}$$

- Rewards:

$$\begin{aligned} r(1, a^1) &= 6, & r(2, a^1) &= -10 \\ r(1, a^2) &= 10, & r(2, a^2) &= -1; \end{aligned}$$

Rewards, Horizon and discount factor

Objective

The objective of the agent is to maximize the long term reward.

- In a finite horizon situation that can be the sum of rewards received at each step plus the scrap value of the terminal state, or the reward average.
- in a infinite horizon situation, the sums can be infinite, so one can use the concept of discount factor < 1.

Policy

Definition (Policy)

A policy $\pi : \mathcal{X} \to \Delta(\mathcal{A})$ is a (possibly) stochastic map from states to actions.

Policy possibilities:

- Stationary or time-dependent
- Markovian or History-dependent
- Deterministic or Stochastic

We will consider here the simplest kinds, stationary Markovian (stochastic or deterministic) policies

Policy

In our example

Four possible stationary deterministic policies:

- π_{11}: $\pi_{11}(1) = a_1$, $\pi_{11}(2) = a_1$;
- π_{12}: $\pi_{12}(1) = a_1$, $\pi_{12}(2) = a_2$;
- π_{21}: $\pi_{21}(1) = a_2$, $\pi_{21}(2) = a_1$;
- π_{22}: $\pi_{22}(1) = a_2$, $\pi_{22}(2) = a_2$;

Policy

In our example

An example of a static stochastic policy:

$$\pi : \qquad \pi(a_1|1) = 1/3, \quad \pi(a_2|1) = 2/3,$$
$$\pi(a_1|2) = 1/5, \quad \pi(a_2|2) = 4/5;$$

The general case of a stationary stochastic policy for our example is

$$\pi_{\alpha\beta} : \qquad \pi(a_1|1) = \alpha, \quad \pi(a_2|1) = 1 - \alpha$$
$$\pi(a_1|2) = \beta, \quad \pi(a_2|2) = 1 - \beta;$$

State-value function

The state-value function of a given policy π:

Definition (State-value function)

The state-value function $V_\pi : \mathcal{X} \to \mathbb{R}$ is

$$V_\pi^{(N)}(x) := \mathbb{E}[\sum_{t=0}^{N} \gamma^t R_t | x_0 = x],$$

where N is the horizon, and the expectation is taken with respect to the states $x_{t+1} \sim [P_{a_t}]_{x_t,\cdot}$ and the actions $a_t \sim \pi(x_t)$.

The Q function

Definition (State action value function)

The state action value function $Q_\pi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ is

$$Q_\pi(x, a) := \mathbb{E}[\sum_{t=0}^{N} \gamma^t R_t | x_0 = x, a_0 = a],$$

where the expectation is taken with respect to the states $x_{t+1} \sim [P_{a_t}]_{x_t,\cdot}$ and the actions $a_t \sim \pi(x_t)$.

The Q function

,	**Action 1**	**Action 2**
State 1	0	5
State 2	0	5
State 3	0	5
State 4	20	0

The Prediction Problem

Evaluate a policy π by approximating its state value function V_π

Proposition (Fixed point equation for the state value function)

The following relation holds:

$$V_\pi(x) = \mathbb{E}[R(x, a) + \gamma V_\pi(y)], \tag{1}$$

where the expectation is taken with respect to the next state $y \sim [P_a]_{x,\cdot}$ and the action $a \sim \pi(x)$.

Using Matrix Calculus

Given a policy $\pi : \mathcal{X} \to \Delta(\mathcal{A})$ (resp $\pi : \mathcal{X} \to \mathcal{A}$), define the reward vector $r_\pi := [r_\pi(1), r_\pi(2), \ldots]$ with

$$r_\pi(s) := \mathbb{E}\left(\sum_{a\in\mathcal{A}} \pi(a|s)R(s,a)\right) \quad (\text{ resp } r_\pi(s) := \mathbb{E}(R(s,\pi(s))) \text{ }).$$

and define the Markov matrix P_π,

$$[P_\pi]_{ij} := \sum_{a\in\mathcal{A}} \pi(a|s)p(i|j,a) \quad (\text{ resp } [P_\pi]_{ij} := p(i|j,\pi(j)) \text{ }).$$

Using Matrix Calculus

So...

For a fixed policy π, the MDP behaves exactly as a normal Markov process!

Calculating the state-value

$$V_\pi^{(N)}(x) = \mathbb{E}[\sum_{t=0}^{N} \gamma^t R_t | x_0 = x]$$

Using the vector notation

$$\begin{aligned} V_\pi^{(N)} &= r_\pi + r_\pi \gamma P_\pi + r_\pi \gamma^2 P_\pi^2 + \ldots + r_\pi \gamma^N P_\pi^N \\ &= \sum_{t=0}^{N} r_\pi \left(\gamma P_\pi\right)^t \qquad (0 \leq \gamma \leq 1) \end{aligned}$$

Calculating the state-value

Proposition

If π is a stationary markovian policy for an MDP, then

$$V_\pi^\infty = r_\pi (I - \gamma P_\pi)^{-1}, \qquad (0 \leq \gamma < 1)$$

Moreover, V_π^∞ obeys the fixed point equation

$$V_\pi^\infty = r_\pi + V_\pi^\infty \gamma P_\pi$$

This type of fixed point equations are called Bellman equations

The Control Problem

Find a policy π^* that maximizes the cumulative reward $V_{(\cdot)}$

A policy π is better than another policy π' if $V_\pi(x) \geq V_{\pi'}(x)$ for all $x \in \mathcal{X}$.

Definition

An optimal policy π^* is any such that, for any policy π,

$$V_{\pi^*}(x) \geq V_\pi(x),$$

for all $x \in \mathcal{X}$.

Optimality Equations

For a deterministic π, with $r(x, a) = \mathbb{E}[R(x, a)]$

$$V_\pi^{(N)}(x) = r(x, \pi(x)) + \sum_{x' \in \mathcal{X}} \gamma p(x'|x, \pi(x)) V_\pi^{(N-1)}(x')$$

$$V_\pi^{\infty}(x) = r(x, \pi(x)) + \sum_{x' \in \mathcal{X}} \gamma p(x'|x, \pi(x)) V_\pi^{\infty}(x')$$

Bellman Equations for the optimal policy

The state-value function of the optimal policy π^*, which we shall denote by V_*, verifies

$$V_*^{(N)}(x) = \max_{a \in \mathcal{A}} \left(r(x, a) + \sum_{x' \in \mathcal{X}} \gamma p(x'|x, a) V_*^{(N-1)}(x') \right)$$

$$V_*^{\infty}(x) = \max_{a \in \mathcal{A}} \left(r(x, a) + \sum_{x' \in \mathcal{X}} \gamma p(x'|x, a) V_*^{\infty}(x') \right)$$

Optimality Equations

Question...

Does an optimal State Value Function always exist?

Optimal Policy

Theorem (Existence of solution)

There exists an optimal policy π^*.

Corollary (Optimal Value Equation)

The state value function of the optimal policy π^**, which we shall denote by* V^**, verifies*

$$V^*(x) = \max_{a \in \mathcal{A}} \mathbb{E}[R(x, a) + \gamma V^*(y)] \tag{2}$$

for all $x \in \mathcal{X}$*, where the expectation is taken with respect to the next state* $y \sim P_{a_x, \cdot}$.

Proof Sketch

Consider an infinite horizon model, let Π be the set of all deterministic policies and U be the Banach space of the real functions defined on $\mathcal{X}$, with $\|u\| := \sup_{x\in\mathcal{X}} |u(x)|$.

Lemma

Suppose there exists $M > 0$ such that $|r(x,a)| < M$ for all $(x,a) \in \mathcal{X} \times \mathcal{A}$, and $0 \leq \gamma < 1$. Then, for any $u \in U$ and $\pi \in \Pi$, we have that

$$r_\pi + u\gamma P_\pi \in U$$

Proof Sketch

For $u \in U$ define the (non-linear) operator H on U by

$$Hu := \max_{\pi \in \Pi} \{ r_\pi + u\gamma P_\pi \}$$

The solutions of the optimality equation are fixed points of H !

Proposition

The Bellman operator H is a contraction mapping.

Optimal state value function

Letting Q^* denote Q_{π^*}, we have that

Proposition (Optimal Q equation)

The following relation holds:

$$Q^*(x, a) = \mathbb{E}[R(x, a) + \gamma \max_{a' \in \mathcal{A}} Q^*(y, a')], \tag{3}$$

where the expectation is taken with respect to the next state $y \sim [P_a]_{x,\cdot}$ *and the action* $a' \sim \pi(x)$

Finding V_π

- As a system of $n \times n$ linear equations:

$$\begin{aligned} V_\pi(x) &= \mathbb{E}[R(x,a) + \gamma V_\pi(y)] \\ &= \sum_a \pi(a|x) \sum_{y,r} p(y,r|x,a)(r + \gamma V_\pi(y)) \end{aligned}$$

$$V_\pi = TV_\pi$$

- Iterative policy evaluation:

$$V_{k+1} = TV_k$$

Approximating π^*

Given V_π,

$$\pi(x) \leftarrow \underset{a}{\operatorname{argmax}} \sum_{y,r} p(y, r|x, a)(r + \gamma V_\pi(y))$$

Policy Iteration

$$\pi_0 \rightarrow V_{\pi_0} \rightarrow \pi_1 \rightarrow V_{\pi_1} \rightarrow \pi_2 \rightarrow \ldots \approx \pi^*$$

Problems with this simplistic approach

- We need to work with the whole state space

- We need to know the model of the world

Contents

Who needs a model?

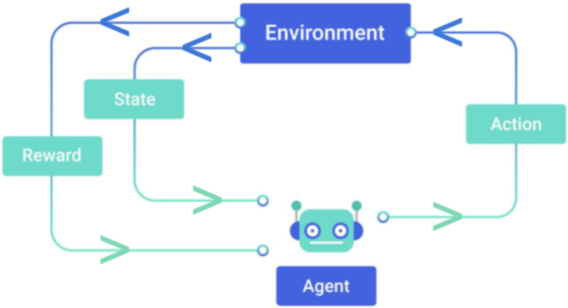
Model-free Methods

- Monte Carlo Methods
- Temporal-Difference Learning

Reinforcement learning

Reinforcement learning

An agent learns how to act optimally in an environment by trial-and-error.



$x_0, a_1, r_1, x_1, a_2, r_2, x_2, a_3, r_3, x_3 \; \ldots$

Temporal-Difference Learning

$$V_\pi(x) = \mathbb{E}\left[R_1 + \gamma V_\pi(x_1) | x_0 = x\right]$$

The $TD(0)$ algorithm

$$V(x_t) \leftarrow (1-\alpha)V(x_t) + \alpha \overbrace{\left(r(x,a) + \gamma V(x_{t+1})\right)}^{\text{TD target}}$$

Learning the Q function

SARSA

$$Q_\pi(x,a) \leftarrow (1-\alpha)Q_\pi(x,a) + \alpha\left(r(x,a) + \gamma Q_\pi(x',a')\right)$$

Q-Learning

$$Q(x,a) \leftarrow (1-\alpha)Q(x,a) + \alpha\left(r(x,a) + \gamma \max_{a'\in\mathcal{A}} Q(x',a')\right)$$

$$Q(x,a) \leftarrow Q(x,a) + \alpha\left(r(x,a) + \gamma \max_{a'\in\mathcal{A}} Q(x',a') - Q(x,a)\right)$$

Q-learning

Q-learning algorithm for estimating π^*

Initialize $Q(x, a)$ for all $x \in \mathcal{X}$ and $a \in \mathcal{A}$ (e.g. $Q(x, a) = 0$);
repeat for each Episode
    Choose an initial state x;
    repeat for each step of Episode
        Choose action a using policy derived from Q (e.g. ϵ-greedy);
        Execute a, observe reward r and new state x';
        $Q(x, a) \leftarrow (1 - \alpha)Q(x, a) + \alpha\big(r + \gamma \max_{a'} Q(x', a')\big)$;
        $x \leftarrow x'$
    until x *is terminal*;
until *satisfied*;

Q-learning with function approximation

Suppose we wish to approximate Q_π, through a parameterized set of functions

$$\mathcal{Q} = \{q_w : w \in \mathbb{R}^k\}$$

Q-learning with function approximation

We wish to approximate Q^* using $\mathcal{Q} = \{q_w : w \in \mathbb{R}^k\}$

Fixed point equation for the optimal state action value function

$$q^*(x,a) = \mathbb{E}[R(x,a) + \gamma \max_{a' \in \mathcal{A}} q^*(y,a')].$$

$$q^* = Hq^*$$

Loss function:

$$L(w) = \frac{1}{2}\mathbb{E}_\mu[(Q^*(x,a) - Q_w(x,a))^2]$$

$$w \leftarrow w + \alpha \mathbb{E}_\mu[(\; q^*(x,a) \; - q_w(x,a))\nabla_w q_w(x,a)]$$

$$w \leftarrow w + \alpha \mathbb{E}_\mu[(\; \boxed{q^*(x,a)} \; - q_w(x,a))\nabla_w q_w(x,a)]$$

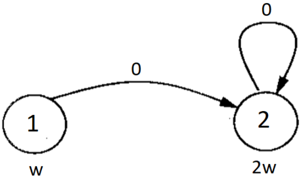
$$w \leftarrow w + \alpha \mathbb{E}_\mu[(\; \boxed{R(x,a) + \gamma \max_{a' \in \mathcal{A}} q_w(y,a')} \; - q_w(x,a))\nabla_w q_w(x,a)]$$

Q-learning with function approximation

The $w \rightarrow 2w$ example (Tsitsiklis and Van Roy 1996)

Consider the state space $\mathcal{X} = \{x_1, x_2\}$, one action, all rewards 0, and the transition matrix

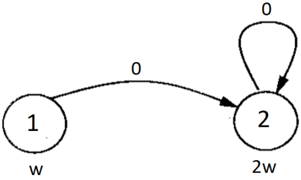
$$P = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$$



$$Q^* = V = 0$$

Q-learning with function approximation

The $w \to 2w$ example (Tsitsiklis and Van Roy 1996)



$$Q^* = V = 0$$

$$\mathcal{Q} = \{w\phi,\ w \in \mathbb{R}\}$$

with $\phi : \mathcal{X} \to \mathbb{R}$ such that $\phi(x_1) = 1, \phi(x_2) = 2$

The spiral example (Tsitsiklis and Van Roy 1997)

Markov chain

$\mathcal{X} = \{s_1, s_2, s_3\}$, $\mathcal{A} = \{a\}$ and

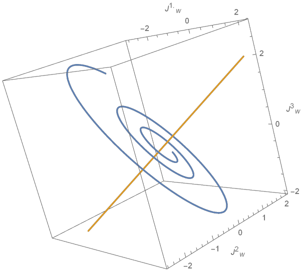
$$\mathbf{P} = \begin{bmatrix} \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 0 \\ 0 & \frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

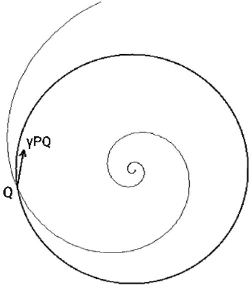
Approximation architecture

$$\frac{dQ_w}{dw} = (S + \epsilon\mathbf{I})Q_w,$$

where ϵ is very small and

$$S = \begin{bmatrix} 1 & \frac{1}{2} & \frac{3}{2} \\ \frac{3}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{3}{2} & 1 \end{bmatrix}.$$





The Deadly Triad

- Function Approximation
- Bootstraping
- Off-policy training

In Part II...

RESEARCH CITADEL
BOOISTRAPPING
STABILITY SWORD
REINFORCEMENT LEARNING DEADLY
DIVERGEIGEICN PIT
GENERALLIZATIDE
PAC LEARNIKG
DIVERGENCE PIT
MATHEMATICS VS. THE RL DEADLY TRIAD