Mathematics of Reinforcement Learning

沈威宇

February 26, 2025

Contents

1 Discrete-Time Markov Decision Process (DTMDP)

I Terms

- State s: The state of the agent with respect to the environment.
- State space S: The set of all possible states.
- Action: A choice the agent can make to interact with the environment, changing its state.
- Action space of a state A_s: The set of all possible actions of a state s.
- State transition: The transition from a state to next state.
- State transition probability (in a discrete-time Markov process) $p(s'|s)$: A probability mass function that defines the likelihood of an agent transitioning to s' from s.
- State transition matrix (in a discrete-time Markov process) P: Suppose the states could be indexed as $s_i \ (_{i=1}^{n})$. The state transition matrix is defined to be:

$$P \in [0,1]^{n \times n}, \quad P_{ij} := p(s_j|s_i).$$

- State transition probability (of an action) $p(s'|s,a)$: A probability mass function that defines the likelihood of an agent transitioning to s' given that the agent takes an action a in a state s.
- Policy $\pi(a|s)$: A policy is a probability function from the state space to the action spaces an agent follows to select actions based on its current state. It defines the conditional probability of the agent taking action a when in state s.
- Deterministic policy: A policy is deterministic if, for each state s, there exists exactly one action a such that $\pi(a|s) = 1$ and $\pi(a'|s) = 0$ for all other actions $a' \neq a$.
- Stochastic policy: A policy that is not deterministic.
- State transition probability (given a policy) $p_\pi(s'|s)$:

$$p_\pi(s'|s) := \sum_a \pi(a|s) p(s'|s,a).$$

- State transition matrix (given a policy) P_π: Suppose the states could be indexed as $s_i \ (_{i=1}^{n})$. The state transition matrix of a policy π is defined to be:

$$P_\pi \in [0,1]^{n \times n}, \quad (P_\pi)_{ij} := p_\pi(s_j|s_i).$$

- Reward r: A real number the agent gets after taking a action.
- Reward transition probability $p(r|s,a)$: A probability mass function that defines the likelihood of an agent receiving reward r given that the agent takes an action a in a state s.

- Reward (given a policy) $r_\pi(s)$:

$$r_\pi(s) := E[r|S_t = s, A_t = a] = \sum_a \pi(a|s) \sum_r p(r|s,a)r.$$

- Reward vector (given a policy) r_π: Suppose the states could be indexed as $s_i \left(_{i=1}^{n}\right)$. The reward vector of a policy π is defined to be:

$$r_\pi := [r_\pi(s_i) \left(_{i=1}^{n}\right)]^\top \in \mathbb{R}^n.$$

- Trajectory: A finite or infinite state-action-reward chain that an agent can take by taking a chain of actions in the action space of the state it's in, moving along a chain of states in the state space, and receiving rewards along the way. In a trajectory, the ith state (probability) is called S_{i-1}, the ith action (probability) taken is called A_{i-1}, the ith reward received is called R_i, making the trajectory $S_0 \xrightarrow[R_1]{A_0} S_1 \ldots S_{i-1} \xrightarrow[R_i]{A_{i-1}} S_i \ldots$.
- Return: The sum of all rewards the agent receives along a trajectory.
- Discounted return G_t: The discounted reward G_t at step t given that the discount rate is $\gamma \in [0, 1)$, the reward at step i is r_i, and the final step in the trajectory is the step $t + n$ ($n = \infty$ for infinite trajectory), is defined to be:

$$G_t := \sum_{i=0}^{n} \gamma^i R_{t+i}.$$

- Terminal state: The state that the agent is in after its last action in a finite trajectory.
- Episode or Trial: A trajectory with a terminal state.
- Episodic task: A task with a terminal state.
- Continuing task: A task without a terminal state.
- Target state: The terminal state in a finite trajectory, or the state that the agent stays in since a specific action is taken and that the agent takes a same action that doesn't change its state afterwards in an infinite trajectory. Not all infinite trajectories have a target state.
- Absorbing state: A target state in an infinite trajectory that any action of the agent after it yields zero reward. Not all infinite trajectories have an absorbing state.
- System model, Transition model, or Model (of a Markov decision process): the state transition probability of each action in the action space of each state, and the reward transition probability of each action in the action space of each state of a Markov decision process.
- Discrete-time Markov process or Discrete-time Markov chain (DTMC) $(S, p(s'|s))$: A stochastic process describing a sequence of possible events in which the probability of each event depends only on the current state in the previous event. A Markov process is given by a two-tuple of the state space S and state transition probability $p(s'|s)$, that the following property, called Markov property or Memoryless property, holds:

$$p(S_{t+1}|S_t, S_{t-1}, \ldots, S_0) = p(S_{t+1}|S_t).$$

- Discrete-time Markov decision process (DTMDP) $(S, A_s, p(s'|s,a), p(r|s,a))$: A Markov decision process is given by a four-tuple of the state space S, the action spaces A_s of each state s, the state transition probability $p(s'|s,a)$ of each action a in the action space of each state s, and the reward transition probability $p(r|s,a)$ of each action a in the action space of each state s, that the following property, called Markov property or Memoryless property, holds:

$$p(S_{t+1}|A_t, S_t, A_{t-1}, S_{t-1}, \dots, A_0, S_0) = p(S_{t+1}|A_t, S_t),$$
$$p(R_{t+1}|A_t, S_t, A_{t-1}, S_{t-1}, \dots, A_0, S_0) = p(R_{t+1}|A_t, S_t).$$

  If a policy is given, a Markov decision process becomes a Markov process.

- State value $v_\pi(s)$: The state value, a function of state s given the policy π, is the expected value of the discounted return G_t given $S_t = s$, that is,

$$v_\pi(s) := E[G_t|S_t = s].$$

- State value vector v_π: Suppose the states could be indexed as $s_i \left(_{i=1}^{n}\right)$. The state value vector of a policy π is defined to be:

$$v_\pi := [v_\pi(s_i) \left(_{i=1}^{n}\right)]^\top \in \mathbb{R}^n.$$

- Policy evaluation: Given a policy, finding out the corresponding state values of all states is called policy evaluation.

- Action value or Q-value $q_\pi(s,a)$ or $Q^\pi(s,a)$: The action value, a function of state-action pair (s,a) given the policy π, is the expected value of the discounted return G_t given $S_t = s$ and $A_t = a$, that is,

$$q_\pi(s,a) := E[G_t|S_t = s, A_t = a] = E[R_{t+1} + \gamma v_\pi(S_{t+1})|S_t = s, A_t = a].$$

- Optimal Policy π^*: Given two policies π_1 and π_2, if

$$v_{\pi_1}(s) \geq v_{\pi_2}(s), \quad \forall s \in S,$$

  then we say π_1 is "better" than π_2.
  A policy π^* is optimal if for any other policy π

$$v_{\pi^*}(s) \geq v_\pi(s), \quad \forall s \in S.$$

  Given a Markov decision process, there must exist an optimal policy, but it is not necessarily unique.

II Bellman Equation

i Bellman Equation Elementwise Form

The elementwise form of the Bellman equation of a given policy π is

$$\begin{aligned} v_\pi(s) =& E[G_t|S_t = s] \\ =& E[R_{t+1}|S_t = s] + \gamma E[G_{t+1}|S_t = s] \\ =& \sum_a \pi(a|s) \sum_r p(r|s,a)r + \gamma \sum_a \pi(a|s) \sum_{s'} p(s'|s,a)v_\pi(s') \\ =& \sum_a \pi(a|s) \left(\sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v_\pi(s') \right) \\ =& r_\pi(s) + \gamma \sum_{s'} p_\pi(s'|s)v_\pi(s'), \quad \forall s \in S. \end{aligned}$$

ii Bellman Equation Matrix-Vector Form

The matrix-vector form of the Bellman equation of a given policy π is

$$v_\pi = r_\pi + \gamma P_\pi v_\pi.$$

iii Bellman Equation Closed Form Solution

$$v_\pi = (I - \gamma P_\pi)^{-1} r_\pi.$$

iv Bellman Equation Iterative Solution Matrix-Vector Form

Consider a sequence $\{v_k\}$ where v_0 is any arbitrary vector $\in \mathbb{R}^{|S|}$, and

$$v_k = r_\pi + \gamma P_\pi v_{k-1}, \quad k \in \mathbb{N},$$

then

$$v_\pi = \lim_{k\to\infty} v_k.$$

In practice, we usually stop when $\|v_k - v_{k-1}\|$ is sufficiently small or when k is sufficiently large.

v Bellman Equation Iterative Solution Elementwise Form

Consider a sequence $\{v_k(s)\}$ where $v_0(s)$ is any arbitrary value, and

$$v_k = \sum_a \pi(a|s) \left(\sum_r p(r|s,a)r + \gamma \sum_{s'} p(s'|s,a)v_{k-1}(s) \right), \quad k \in \mathbb{N},$$

then

$$v_\pi(s) = \lim_{k\to\infty} v_k.$$

In practice, we usually stop when $|v_k(s) - v_{k-1}(s)|$ is sufficiently small or when k is sufficiently large.

vi Value Improvement Theorem

Consider a sequence $\{v_k\}$ where v_0 is any arbitrary vector $\in \mathbb{R}^{|S|}$, and

$$v_k = r_\pi + \gamma P_\pi v_{k-1}, \quad k \in \mathbb{N},$$

then

$$v_{k+1} \geq v_k.$$

III State-Action Value Function or Value Function (VF)

Compare

$$v_\pi(s) = \sum_a \pi(a|s) q_\pi(s,a)$$

and the Bellman equation, we have the action-value function:

$$q_\pi(s,a) = \sum_r p(r|s,a) r + \gamma \sum_{s'} p(s'|s,a) v_\pi(s').$$

IV Contraction Mapping Theorem

i Fixed Point

Given $f: X \to X$, $x \in X$ is a fixed point if

$$f(x) = x.$$

ii Contraction Mapping or Contractive Function

$f: X \to X$ is a contraction mapping if

$$\exists \gamma \in [0,1): \ \|f(x_1) - f(x_2)\| \leq \gamma \|x_1 - x_2\|, \quad \forall x_1, x_2 \in X,$$

where $\|\cdot\|$ can be any vector norm.

iii Contraction Mapping Theorem

For any contraction mapping,

- Existence: there exists a fixed point x^* satisfying $f(x^*) = x^*$.
- Uniqueness: the fixed point x^* is unique.
- Algorithm: Consider a sequence $\{x_k\}$ where x_0 is any arbitrary value and $x_{k+1} = f(x_k), \quad k \in \mathbb{N}$, then

  $$\lim_{k \to \infty} x_k = x^*.$$

  Moreover, the convergence rate is exponential and determined by γ.

V Bellman Optimality Equation (BOE)

i Bellman Optimality Equation Elementwise Form

$$v(s) = \max_{\pi} \left(\sum_{a} \pi(a|s) q(s,a) \right), \quad s \in S.$$

ii Bellman Optimality Equation Matrix-Vector Form

$$v = \max_{\pi} \left(r_\pi + \gamma P_\pi v_\pi \right).$$

iii Value Iteration (VI) Matrix-Vector Form

Let

$$f(v) = \max_{\pi} \left(r_\pi + \gamma P_\pi v \right),$$

where v is a state value given policy π.
Because $f(v)$ is a contraction mapping, it satisfies the contraction mapping theorem, that is,

- Existence and uniqueness:

  $$\exists! v^* \text{ such that } v^* = f(v^*),$$

- Iterative algorithm: Consider a sequence $\{v_k\}$ where v_0 is any arbitrary value, and $v_k = f(v_{k-1}), \quad k \in \mathbb{N}$. It converges to v^* in an exponential rate determined by γ as k approaching ∞.

One iteration in the value iteration algorithm,

$$v_{k+1} = f(v_k) = \max_{\pi} \left(r_\pi + \gamma P_\pi v_k \right), \quad k+1 \in \mathbb{N},$$

can be decomposed into two steps,

1. **Policy update (PU)**: Solve

   $$\pi_{k+1} = \arg\max_{\pi} (r_\pi + \gamma P_\pi v_k)$$

   for π_{k+1} given v_k.

2. **Value update (VU)**: Solve

   $$v_{k+1} = r_{\pi_{k+1}} + \gamma P_{\pi_{k+1}} v_k),$$

   for v_{k+1} given π_{k+1} and v_k.

In practice, we usually stop when $\|v_k - v_{k-1}\|$ is sufficiently small or when k is sufficiently large.

iv Value Iteration (VI) Elementwise Form

In elementwise form, the two steps of value iteration can be written as,

1. **Policy update (PU)**: Solve

   $$\pi_{k+1}(s) = \arg\max_{\pi} \sum_{a} \pi(a|s) q_{\pi_k}(s,a), \quad s \in S$$

for $\pi_{k+1}(s)$ given $v_k(s')$ for all $s' \in S$.
Let $a_k^*(s) = \arg\max_a q_{\pi_k}(s,a)$. We select

$$\pi_{k+1}(a|s) = \begin{cases} 1, & a = a_k^*(s) \\ 0, & a \neq a_k^*(s) \end{cases},$$

called "greedy policy" because it simply selects the greatest policy value.

2. **Value update (VU)**: Solve

$$v_{k+1}(s) = \sum_a \pi_{k+1}(a|s) q_{\pi_k}(s,a), \quad s \in S$$

for $v_{k+1}(s)$ given $\pi_{k+1}(a|s)$ for all $a \in A_s$ for all $s \in S$, and $v_k(s')$ for all $s' \in S$.
Since $\pi_{k+1}(a|s)$ is greedy,

$$v_{k+1}(s) = \max_a q_{\pi_k}(s,a).$$

In practice, we usually stop when $|v_k(s) - v_{k-1}(s)|$ is sufficiently small or when k is sufficiently large.

v Optimality Theorem

Suppose v^* is the solution to a Bellman optimality equation, that is,

$$v^* = \max_\pi \left(r_\pi + \gamma P_\pi v^* \right).$$

Suppose

$$\pi^* = \arg\max_\pi \left(r_\pi + \gamma P_\pi v^* \right).$$

Then

$$v^* = r_{\pi^*} + \gamma P_{\pi^*} v^*.$$

v^* is the optimal state value, and π^* is the optimal policy.

vi Optimal Policy Invariance Theorem

Consider a Markov decision process with $v^* \in \mathbb{R}^{|S|}$ as the optimal state value satisfying $v^* = \max_\pi(r_\pi + \gamma P_\pi v^*$. If every reward r is changed by an affine transformation to $ar + b$, where $a, b \in \mathbb{R}$ and $a \neq 0$, then the corresponding optimal state value v' is also an affine transformation of v^*:

$$v' = av^* + \frac{b}{1-\gamma}\mathbf{1},$$

where $\gamma \in [0,1)$ is the discount rate and $\mathbf{1} = [1, \ldots, 1]^\top \in \mathbb{R}^{|S|}$.
Consequently, the optimal policies are invariant to any affine transformation of the reward signals.

vii Policy Improvement Theorem

If

$$\pi_{k+1} = \arg\max_\pi(r_\pi + \gamma P_\pi v_{\pi_k}),$$

then $\|v_{\pi_{k+1}}\| \geq \|v_{\pi_k}\|$ for any k.

viii Convergence of Policy Iteration Theorem

The state value sequence $\{v_{\pi_k}\}_{k=0}^{\infty}$ generated by the policy iteration algorithm converges to the optimal state value v^*. Consequently, the policy sequc $\{\pi_k\}_{k=0}^{\infty}$ coverages to an optimal policy.

ix Policy Iteration (PI) Matrix-Vector Form

An arbitrary initial policy π_0 is given. One iteration in the policy iteration algorithm can be decomposed into two steps,

1. **Policy evaluation (PE)**: Solve the Bellman equation

$$v_{\pi_k} = r_{\pi_k} + \gamma P_{\pi_k} v_{\pi_k}$$

   for v_{π_k} given π_k.

2. **Policy improvement (PI)**: Solve

$$\pi_{k+1} = \arg\max_{\pi}(r_{\pi} + \gamma P_{\pi} v_{\pi_k})$$

   for π_{k+1} given v_{π_k}.

In practice, we usually stop when $\|v_{\pi_k} - v_{\pi_{k-1}}\|$ is sufficiently small or when k is sufficiently large.

x Policy Iteration (PI) Elementwise Form

In elementwise form, the two steps of policy iteration can be written as,

1. **Policy evaluation (PE)**: Solve the Bellman equation

$$v_{\pi_k}(s) = \text{☼}_a \pi(a|s) q_{\pi_k}(s,a), \quad s \in S$$

   for v_{π_k} given π_k.

2. **Policy improvement (PI)**:
   Solve

$$\pi_{k+1}(s) = \arg\max_{\pi} \sum_a \pi(a|s) q_{\pi_k}(s,a), \quad s \in S$$

   for $\pi_{k+1}(s)$ given $v_{\pi_k}(s)$.
   Let $a_k^*(s) = \arg\max_a q_{\pi_k}(s,a)$. We select

$$\pi_{k+1}(a|s) = \begin{cases} 1, & a = a_k^*(s) \\ 0, & a \neq a_k^*(s) \end{cases},$$

   called ”greedy policy” because it simply selects the greatest policy value.

In practice, we usually stop when $|v_{\pi_k}(s) - v_{\pi_{k-1}}(s)|$ is sufficiently small or when k is sufficiently large.

xi Truncated Policy Iteration

The truncated policy iteration is the same as policy iteration with the policy evaluation step using the iterative solution but stopped when $\|v_k - v_{k-1}\|$ is sufficiently small or when k is sufficiently large. If stopping when $k = 1$, the truncated policy iteration becomes value iteration except that the first iteration lacks value update and is initialized with an arbitrary policy; if stopping when $k = \infty$, the truncated policy iteration becomes policy iteration.