Mathematics of Reinforcement Learning

Technische Universität München

January 3, 2025

Abstract

this is a lecture notes for the graduate level course: Mathematics of Reinforcement Learning at TUM. The references are mainly from the professor's written notes and this open course taught by Prof. Shiyu Zhao. There is also a LaTeX-typed note written by one of my fellow classmate.

Contents

1 Introduction

2 Markov Decision Processes

Goals:

- Develop a mathematical framework for dynamic decision-making problems under uncertainty
- Learn how to solve these problems if the model is known.

2.1 Definition of Markov Decision Processes

We look for stochastic processes (family of random variables indexed by time)

$$\{S_n\}_{n\geq 0} : \text{state} \quad \{A_n\}_{n\geq 0} : \text{action} \quad \{R_n\}_{n\geq 0} : \text{reward}$$

modeling the dynamic evolution of states, actions, and rewards.

Definition 2.1. *A Markov Decision Model (MDM) is a tuple (S, A, D, p, r, γ) consisting of the following components:*

- *A finite set S called state space*

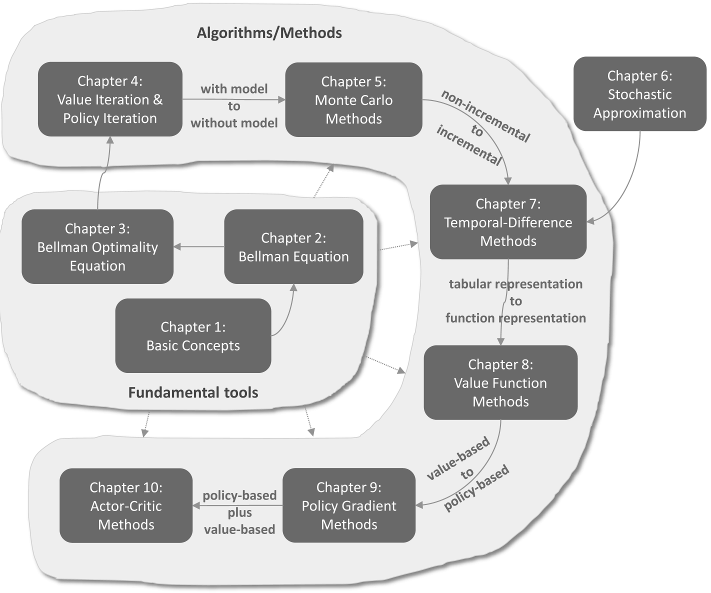

Algorithms/Methods
Chapter 4: Value Iteration & Policy Iteration
with model to without model
Chapter 5: Monte Carlo Methods
non-incremental to incremental
Chapter 6: Stochastic Approximation
Chapter 7: Temporal-Difference Methods
Chapter 3: Bellman Optimality Equation
Chapter 2: Bellman Equation
Chapter 1: Basic Concepts
Fundamental tools
tabular representation to function representation
Chapter 8: Value Function Methods
value-based to policy-based
Chapter 10: Actor-Critic Methods
policy-based plus value-based
Chapter 9: Policy Gradient Methods

Figure 1: Roadmap: Mathematics of Reinforcement Learning(credit to Prof. Shiyu Zhao)

- *A finite set A called action space*
- *A set $D \subseteq S \times A$ whose elements are admissible state-action pairs*
- *A transition probability function $p : S \times S \times A \to [0, 1]$, $(s', a) \mapsto p(s'|s, a)$ satisfying*

$$\sum_{s' \in S} p(s'|s, a) = 1, \quad \forall (s, a) \in S \times A$$

- *A reward function $r : D \to \mathbb{R}$, $(s, a) \mapsto r(s, a)$*
- *A discount factor $\gamma \in (0, 1]$*

Remark.

- *S is the collection of all possible states. A is the collection of all actions. If $(s, a) \in D$, this is interpreted as action a being allowed in state s.*
- *If an agent performs action a in state s, the probability of ending up in state s' is $P(s'|s, a)$, so $s' \mapsto p(s'|s, a)$ is a probability mass function.*
- *The reward the agent receives for action a in state s is $r(s, a)$.*
- *The discount factor γ allows us to encode the time value of rewards. We interpret $\gamma^n r(s, a)$ as the value at time zero of receiving $r(s, a)$ n-steps into the future.*

Definition 2.2. *A policy is a mapping $\pi : S \times A \to [0, 1]$ $(s, a) \mapsto \pi(a|s)$ such that $\sum_{a \in A} \pi(a|s) = 1$ for all $s \in S$ and $\pi(a|s) = 0$ if $(s, a) \notin S \times A|D$.*

We say that π is a deterministic policy if for each $s \in S$ there is a $a \in A$ such that $\pi(a|s) = 1$. We write Π and Π_d for the set of all policies and the subset of deterministic policies respectively.

To wit, $\pi(a|s)$ is interpreted as the probability of choosing action a in state s.

One could also introduce *non-Markovian* policies which depend on the entire history of states and actions, or *non-stationary* policies which depend on the current state and current time.

However, we spare ourselves the trouble since we will eventually see these always as *stationary Markovian* policies as in Definition 2.2.

Intuitively, what happens for a given MDM and a policy π is the following:

1. Start in an initial state $S_0 = s_0$.
2. Use the policy π and randomly draw an action A_0 from $a \sim \pi(a|S_0)$.
3. Collect the reward $R_0 = r(S_0, A_0)$ and draw the next state S_1 from $s' \sim p(s'|S_0, A_0)$.

4. Repeat the procedure to construct $S_0, A_0, R_0, S_1, A_1, R_1, \dots$

Informally, we can introduce the objective

$$\mathbb{E}\left[\sum_{n=0}^{\infty} \gamma^n R_n\right] = \mathbb{E}\left[\sum_{n=0}^{\infty} \gamma^n r(S_n, A_n)\right].$$

the first ingredient is a probability space. we choose the sample space as $\Omega := (S \times A)^{\infty} = \times_{n=1}^{\infty}(S \times A)$ and let the σ-algebra $\mathcal{A}$ on Ω be the power set that is the set of all subsets of Ω. we can safely do so since Ω is countable. In fact, any element $\omega \in \Omega$ takes the form $\omega = (s_0, a_0, s_1, a_1, ...)$

for sequences $\{s_n\}_{n \in \mathbb{N}_0} \subset S$ and $\{a_n\}_{n \in \mathbb{N}_0} \subset A$, we define the following functions:

$$S_n(\omega) = S_n((s_0, a_0, s_1, a_1, ...)) := s_n \quad n \in \mathbb{N}_0$$

$$A_n(\omega) = A_n((s_0, a_0, s_1, a_1, ...)) := a_n \quad n \in \mathbb{N}_0, \omega \in \Omega$$

then S_n, A_n are measurable functions (random variables) taking values in S, A respectively.

what is left is showing that there exists a probability measure $\mathbb{P}^{\pi}_{\mu}$ on $(\Omega, \mathcal{A})$ s.t. $(s_0, a_0, s_1, a_1, ...)$ has the desired distribution. This can be constructed from p (probability function) and π (policy), and an initial distribution μ on S s.t. $S_0 \sim \mu$.

what we would like to do is to define $\mathbb{P}^{\pi}_{\mu}[\{\omega\}] = \mathbb{P}^{\pi}_{\mu}[\{(s_0, a_0, s_1, a_1, ...)\}] := \mu[\{s_0\}]\pi(a_0|s_0)p(s_1|(s_0, a_0))]pi(a_1|s_1)p(s_2|(s_1, a_1))\pi(a_2$

$$\mu[\{s_0\}]\pi(a_0|s_0)\prod_{n=1}^{\infty} p(s_n|(s_{n-1}, a_{n-1}))\pi(a_n|s_n)$$

for all $\omega = (s_0, a_0, s_1, a_1, ...) \in \Omega$. The problem is however, that the resulting probability is usually zero as this is an infinite product with factors valued in $[0, 1]$

Theorem 2.3 (Ionescu-Tulcea for MDMs)**.** *Let $(S, \mathcal{A}, D, \mathbb{P}, r, \mu)$ be an MDM, π a policy and μ a probability measure $\mathbf{P}^{\pi}_{\mu}$ on $(\Omega, \mathcal{A})$ with the property*

$$\mathbf{P}^{\pi}_{\mu}\Big[B \cap \bigtimes_{k=n+1}^{\infty}(S \times A)\Big] = \sum_{(s_0, a_0, s_1, \dots) \in B} \mu[\{s_0\}]\pi(a_0|s_0)\prod_{k=1}^{n} P(s_k|s_{k-1}, a_{k-1})\pi(a_k|s_k)$$

$$\textit{for all } B \subseteq (S \times A)^{n+1}, n \in \mathbb{N}_0.$$

In particular, it holds that

$$\mathbf{P}^{\pi}_{\mu}\big[S_0 = s_0, A_0 = a_0, \dots, S_n = s_n, A_n = a_n\big] = \mu[\{s_0\}]\pi(a_0|s_0)\prod_{k=1}^{n} P(s_k|S_{k-1}, a_{k-1})\pi(a_k|s_k)$$

for all $(s_0, a_0, s_1, a_1, \dots, s_n, a_n) \in (S \times A)^{n+1}$ and $n \in \mathbb{N}_0$.

Proof. The idea is to take the following equations $(*)$ as a definition.

- $\mathbf{P}^{\pi}_{\mu}[S_0 = s_0] = \mu[\{s_0\}]$
- $\mathbf{P}^{\pi}_{\mu}[S_{n+1} = s'|S_n = s, A_n = a] = P(s'|s, a)$
- $\mathbf{P}^{\pi}_{\mu}[A_n = a] = \pi(a|s)$

Note that the sets of the form

$$B \times \bigtimes_{k=n+1}^{\infty}(S \times A)$$

are an intersection stable ring of sets generating $\mathcal{A}$. Moreover, it is not hard to show that $\mathbf{P}^{\pi}_{\mu}$ defined by $(*)$ is a well-defined additive set function. With a bit of work, one can show that $\mathbf{P}^{\pi}_{\mu}$ is even σ-additive, hence a pre-measure. The existence and uniqueness of $\mathbf{P}^{\pi}_{\mu}$ on all of $\mathcal{A}$ is thus a direct consequence of Carathéodory's extension theorem.

Finally, note that

$$\{S_0 = s_0, A_0 = a_0, \dots, S_n = s_n, A_n = a_n\} = \{(s_0, a_0, s_1, a_1, \dots, s_n, a_n)\} \times \bigtimes_{k=n+1}^{\infty}(S \times A),$$

so

$$\mathbf{P}^{\pi}_{\mu}\big[S_0 = s_0, A_0 = a_0, \dots, S_n = s_n, A_n = a_n\big] = \mathbf{P}^{\pi}_{\mu}\Big[\{(s_0, a_0, \dots, s_n, a_n)\} \times \bigtimes_{k=n+1}^{\infty}(S \times A)\Big].$$

$$= \mu[\{s_0\}]\pi(a_0|s_0)\prod_{k=1}^{n} P(s_k|s_{k-1}, a_{k-1})\pi(a_k|s_k).$$

□

Corollary 2.4 (properties of $\mathbb{P}^\pi_\mu$)**.** *in the setting of theorem 2.3, it holds that:*

1. $\mathbb{P}^\pi_\mu[S_0 = s] = \mu[\{s\}]$, $s \in S$
2. $\mathbb{P}^\pi_\mu[S_{N+1} = s'|S_n = s, A_n = a] = p(s'|s,a), n \in \mathbb{N}_0, s, s' \in S, a \in A$
3. $\mathbb{P}^\pi_\mu[A_n = a|S_n = s] = \pi(a|s)$, $n \in \mathbb{N}_0, s \in S, a \in A$
4. $\mathbb{P}^\pi_\mu[S_{n+1} = s_{n+1}, A_{n+1} = a_{n+1}|S_0 = s_0, A_0 = a_0, S_n = s_n, A_n = a_n] = \mathbb{P}^\pi_\mu[S_{n+1} = s_{n+1}, A_{n+1} = a_{n+1}|S_n = s_n, A_n = a_n]$ *for all* $(s_0, a_0, ..., s_{n+1}, a_{n+1}) \in (S \times A)^{n+2}, n \in \mathbb{N}_0$
5. ? $\mathbb{P}^\pi_\mu[S_{n+1} = s_{n+1}|S_0 = s_0, ..., S_n = s_n] = \mathbb{P}^\pi_\mu[S_{n+1} = s_{n+1}|S_n = s_n]$ *for all* $(s_0, s_1, ..., s_{n+1}) \in S^{n+2}$, $n \in \mathbb{N}_0$ *if* π *is deterministic (that is for each* $s \in S$, $\exists a(s) \in A$ *s.t.* $\pi(a(s)|s) = 1$*)*

Proof. see exercises □

2.2 Value Functions and Bellman Equations

Definition 2.5 (state value function, optimal state value function)**.** *For any* $\pi \in \Pi$, *the function* $V^\pi : S \to \mathbb{R}$, *defined as*

$$V^\pi(s) := \mathbb{E}^\pi_s\Big[\sum_{n=0}^{\infty} \gamma^n r(S_n, A_n)\Big],$$

is called the state value function. we have

$$\mathbb{E}^\pi_s[X] = \mathbb{E}^\pi[X|s_0 = s] = \sum_{a \in A} \pi(a|s)[X|s_0 = s]$$

Moreover, if it exists, the function $V : S \to \mathbb{R}$, *defined as*

$$V(s) := \sup_{\pi \in \Pi} V^\pi(s) = \sup_{\pi \in \Pi} \mathbb{E}^\pi_s\Big[\sum_{n=0}^{\infty} \gamma^n r(S_n, A_n)\Big],$$

is called the optimal value function.

Finally, a policy $\pi^* \in \Pi$ *such that*

$$V^{\pi^*}(s) = V(s)$$

is called optimal for the initial state $s \in S$.

the definition leads to many questions:

- does the optimal policy exist?
- is the optimal policy unique?
- is the optimal policy stochastic or deterministic?
- how to obtain the optimal policy?

these questions will be answered after introducing the Bellman optimality equation, (Theorem 2.9)

V^π and V exist under very mild conditions. Indeed, since S and A are finite, we conclude that r is bounded. A sufficient condition for the existence of V^π and V is $\gamma < 1$ almost surely, (-a.s.)

$$\sum_{n=0}^{\infty} \gamma^n r(S_n, A_n) \leq \max_{(s,a) \in D} r(s,a), \quad \sum_{n=0}^{\infty} \gamma^n = \frac{1}{1-\gamma} \max_{(s,a) \in D} r(s,a).$$

For all $\mathbf{P}^\pi_S$, if the MDM has a finite time horizon (meaning that there is $N \in \mathbb{N}_0$ such that $r(S_n, A_n) = 0$ for all $n \geq N$), $\mathbf{P}^\pi_s$-a.s. for all $s \in S$, $\pi \in \Pi$, we don't have to assume $\gamma < 1$.
Standing Assumption: There exists $C > 0$ such that

$$\sum_{n=0}^{\infty} \gamma^n r(S_n, A_n) \leq C, \quad \mathbf{P}^\pi_s\text{-a.s.} \quad \text{for all } \pi \in \Pi,\ s \in S.$$

While it may seem odd that we are now working with an entire family of optimization problems $V(s)$ for any $s \in S$, it is not hard to imagine that these problems are intimately related. We can learn something from the connection.

Lemma 2.6 (time shift)**.** *Let $s, s' \in S$, $a \in \mathcal{A}$, and $\pi \in \Pi$. Define*

$$\mathbf{P}^{\pi}_{s,a,s'} := \mathbf{P}^{\pi}_{s}[\,\cdot\,| S_0 = s, A_0 = a, S_1 = s'].$$

and

$$\widehat{S}_n := S_{n+1}, \quad \widehat{A}_n := A_{n+1}, \quad n \in \mathbb{N}_0.$$

Then the distribution of $\{(\widehat{S}_n, \widehat{A}_n)\}_{n\in\mathbb{N}_0}$ under $\mathbf{P}^{\pi}_{s,a,s'}$ is the same as the distribution of $\{(S_n, A_n)\}_{n\in\mathbb{N}_0}$ under $\mathbf{P}^{\pi}_{s'}$.

Proof. For $n \in \mathbb{N}_0$, $(s_0, a_0, s_1, a_1, \dots, s_n, a_n) \in (S \times A)^{n+1}$, we have to show the following identity:

$$\mathbf{P}^{\pi}_{s,a,s'}[\widehat{S}_0 = s_1, \widehat{A}_0 = a_1, \dots, \widehat{S}_n = s_n, \widehat{A}_n = a_n] = \mathbf{P}^{\pi}_{s}[S_0 = s', A_0 = a_1, \dots, S_n = s_n, A_n = a_n].$$

If $s_0 \neq s'$, then both sides are 0 and the identity is immediate.
Assume $s_0 = s'$. We have:

$$\begin{aligned}
&\mathbf{P}^{\pi}_{s,a,s'}[\widehat{S}_0 = s', \widehat{A}_0 = a_0, \dots, \widehat{S}_n = s_n, \widehat{A}_n = a_n]\\
&= \mathbf{P}^{\pi}_{s}[\widehat{S}_0 = s', \widehat{A}_0 = a_0, \dots, \widehat{S}_n = s_n, \widehat{A}_n = a_n | S_0 = s, A_0 = a, S_1 = s'].\\
&= \frac{\mathbf{P}^{\pi}_{s}[S_0 = s, A_0 = a, S_1 = s', A_1 = a_0, \dots, S_{n+1} = s_n, A_{n+1} = a_n]}{\mathbf{P}^{\pi}_{s}[S_0 = s, A_0 = a, S_1 = s']}\\
&= \frac{\delta_{\{s\}}[\{s\}]\pi(a|s)P(s'|s,a)\pi(a_0|s')\delta_{s'}[\{s'\}]\left[\prod_{k=1}^{n} P(s_k|s_{k-1},a_{k-1})\pi(a_k|s_k)\right]}{\delta_{\{s\}}[\{s\}]\pi(a|s)P(s'|s,a)}.
\end{aligned}$$

Simplifying:

$$= \delta_{s'}[\{s'\}]\pi(a_0|s')\left[\prod_{k=1}^{n} P(s_k|s_{k-1},a_{k-1})\pi(a_k|s_k)\right] = \mathbf{P}^{\pi}_{s}[S_0 = s', A_0 = a, \dots, S_n = s_n, A_n = a_n].$$

□

the following theorem says that the state value function of a state s under a policy π is the expected immediate reward plus the expected discounted state value function of the next state, assuming actions are chosen according to π.

Theorem 2.7 (Bellman Equation)**.** *Let $s \in S$, $\pi \in \Pi$. Then:*

$$V^{\pi}(s) = \sum_{s'\in S, a\in A} \left[r(s,a) + \gamma V^{\pi}(s')\right]\pi(a|s)P(s'|s,a) = \mathbb{E}^{\pi}_{s}\left[r(S_0,A_0) + \gamma V^{\pi}(S_1)\right].$$

here $\mathbb{E}^{\pi}_{s}[X] = \mathbb{E}^{\pi}[X|s_0 = s] = \sum_{a\in A}\pi(a|s)[X|s_0 = s]$ means that the expectation satisfies: the first state s_0 is s and the state value function is under policy π

Proof.

$$V^{\pi}(s) = \mathbb{E}^{\pi}_{s}\Big[\sum_{n=0}^{\infty}\gamma^n r(S_n,A_n)\Big]$$

$$= \sum_{s'\in S, a\in A} \mathbb{E}^{\pi}_{s}\Big[r(S_0,A_0) + \sum_{n=1}^{\infty}\gamma^n r(S_n,A_n)\,\Big|\, S_0 = s, A_0 = a, S_1 = s'\Big]\mathbf{P}^{\pi}_{s}[S_0 = s, A_0 = a, S_1 = s'].$$

By Lemma 2.6, we have that under $\mathbf{P}^{\pi}_{s'}$:

$$\sum_{n=1}^{\infty}\gamma^{n-1} r(S_n,A_n) = \sum_{n=0}^{\infty}\gamma^{n} r(S_{n+1},A_{n+1}) = \sum_{n=0}^{\infty}\gamma^{n}\cdot r(\widehat{S_n},\widehat{A_n}) \sim \sum_{n=0}^{\infty}\gamma^{n} r(S_n,A_n)$$

Substituting back, we get:

$$V^{\pi}(s) = \sum_{s'\in S, a\in A}\Big[r(s,a) + \gamma\mathbb{E}^{\pi}_{s'}\Big[\sum_{n=0}^{\infty}\gamma^n r(S_n,A_n)\Big]\Big]\delta_s[\{s\}]\pi(a|s)P(s'|s,a).$$

Rewriting:

$$V^{\pi}(s) = \sum_{s'\in S, a\in A}\Big[r(s,a) + \gamma V^{\pi}(s')\Big]\delta_s[\{s\}]\pi(a|s)P(s'|s,a) = \mathbb{E}^{\pi}_{s}\Big[r(S_0,A_0) + \gamma V^{\pi}(S_1)\Big].$$

□

Proof. we here give another proof. we first let $R_k = \sum_{n=k}^{\infty} \gamma^n r(s_n, a_n)$ be the total future return at step k

$$V^\pi(s) = \mathbb{E}_s^\pi[R_0] = \mathbb{E}^\pi[R_0|S_0 = s] = \mathbb{E}^\pi[r(S_0, A_0) + \sum_{n=1}^{\infty} \gamma^n r(S_n, A_n)|S_0 = s]$$

$$= \mathbb{E}^\pi[r(S_0, A_0)|S_0 = s] + \mathbb{E}^\pi[R_1|S_0 = s]$$

we calculate the first term:

$$\mathbb{E}^\pi[r(S_0, A_0)|S_0 = s] = \sum_{a\in A} \pi(a|s)\mathbb{E}[r(S_0, A_0)|S_0 = s, A_0 = a] = \sum_{a\in A} \pi(a|s) \sum_r p(r|s, a)r$$

here $\sum_r p(r|s, a) = \sum_r p(S_0 = s, A_0 = a) \triangleq r(s, a)$. the second term:

$$\mathbb{E}^\pi[R_1|S_0 = s_0] = \sum_{s'\in S} \mathbb{E}[R_1|S_1 = s', S_0 = s_0]p(s'|s) = \sum_{s'\in S} \mathbb{E}[R_1|S_1 = s']p(s'|s) = \sum_{s'\in S} V^\pi(s')p(s'|s)$$

$$= \sum_{s'\in S} V^\pi(s') \sum_{a\in A} p(s'|s, a)\pi(a|s)$$

here $\mathbb{E}[R_1|S_1 = s'] \triangleq V^\pi(s')$ and $\sum_{s'\in S} \mathbb{E}[R_1|S_1 = s', S_0 = s_0]p(s'|s) = \sum_{s'\in S} \mathbb{E}[R_1|S_1 = s']p(s'|s)$ is due to the memoryless Markov property. Therefore, we have

$$V^\pi(s) = \sum_{a\in A} \pi(a|s) \left[\sum_r p(r|s, a)r + \gamma \sum_{s'\in S} p(s'|s, a)V^\pi(s')\right]$$

which is actually equivalent to the above bellman equation:

$$V^\pi(s) = \sum_{a\in A} \pi(a|s) \left[\sum_r p(r|s, a)r + \gamma \sum_{s'\in S} p(s'|s, a)V^\pi(s')\right] = \sum_{a\in A} \pi(a|s) \left[r(s, a) + \gamma \sum_{s'\in S} p(s'|s, a)V^\pi(s')\right]$$

$$= \sum_{a\in A} \sum_{s'\in S} \pi(a|s) [p(s'|s, a)r(s, a) + \gamma p(s'|s, a)V^\pi(s')] = \sum_{a\in A} \sum_{s'\in S} \pi(a|s)p(s'|s, a) [r(s, a) + \gamma V^\pi(s')]$$

$$= \sum_{a\in A} \sum_{s'\in S} \delta_s[\{s\}]\pi(a|s)p(s'|s, a) [r(s, a) + \gamma V^\pi(s')] = \mathbb{E}^\pi[r(s, a) + \gamma V^\pi(s')|s_0 = s] = \mathbb{E}_s^\pi [r(s, a) + \gamma V^\pi(s')]$$

the proof is finished. we point out that $p(r|s, a)$ and $p(s'|s, a)$ represent the dynamic model, which are given according to the environment. □

The Bellman equation shows that:

$$V^\pi(s) = \sum_{s'\in S, a\in A} [r(s, a) + \gamma V^\pi(s')] \pi(a|s)P(s'|s, a)$$

Expanding and simplifying:

$$V^\pi(s) = \sum_{a\in A} \left[r(s, a) \sum_{s'\in S} P(s'|s, a) + \sum_{s'\in S} \gamma P(s'|s, a)V^\pi(s')\right]\pi(a|s)$$

This can be written as:

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

This shows that the state value function is the weighted sum of the immediate reward under any action a, plus the discounted state value function in the next state.

Definition 2.8 ((optimal) action value function). *Let $s \in S$, $a \in A$, and $\pi \in \Pi$. We refer to $Q^\pi : S \times A \to \mathbb{R}$, defined as:*

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

*as the **action value function** for the policy π.*
Similarly, we call $Q : S \times A \to \mathbb{R}$, defined as:

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

*the **optimal action value function**.*

We see that the state and action value functions are related via:

$$V^\pi(s) = \sum_{a\in A}\Big[r(s,a) + \gamma \sum_{s'\in S} P(s'|s,a)V^\pi(s')\Big]\pi(a|s)$$

or equivalently:

$$V^\pi(s) = \sum_{a\in A} Q^\pi(s|a)\pi(a|s), \quad \forall s \in S.$$

recall that $V^\pi(s) \triangleq \mathbb{E}_s^\pi[\sum_{n=0}^\infty \gamma^n r(S_n, A_n)] = \mathbb{E}_s^\pi[R_0] = \mathbb{E}^\pi[R_0|S_0 = s] = \sum_{a\in A}\mathbb{E}[R_0|S_0 = s, A_0 = a]\pi(a|s)\delta_s[\{s\}] = \sum_{a\in A}\mathbb{E}[R_0|S_0 = s, A_0 = a]\pi(a|s)$. comparing with the above equation, we have

$$\mathbb{E}[R_0|S_0 = s, A_0 = a] = Q^\pi(s|a)$$

We can thus interpret $Q^\pi(s|a)$ as the expected total return, conditional on the initial state being s and the initial action being a.

Another observation we can draw from the Bellman equation is that:

$$V^\pi(s) = \sum_{s'\in S, a\in A} \left[r(s,a) + \gamma V^\pi(s')\right]\pi(a|s)P(s'|s,a),$$

which just means that V^π can be computed by solving a linear system of equations. If the number of states is sufficiently small, this is indeed feasible.

Similarly, the action value function can be computed by solving the linear system:

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

for $(s,a) \in S \times A$.

We subsequently let:

$$A(s) \triangleq \{a \in A \mid (s,a) \in D\}$$

be the set of admissible actions in any given state $s \in S$. next we show the relation between $V(s)$ and $Q(s|a)$:

$$V(s) = \max_{a\in A(s)} Q(s|a)$$

Proof.

$$V^\pi(s) = \sum_{a\in A}\left[r(s,a) + \gamma \sum_{s'\in S} P(s'|s,a)V^\pi(s')\right]\pi(a|s)$$

$$\leq \max_{a\in A(s)}\left[r(s,a) + \gamma \sum_{s'\in S} P(s'|s,a)V(s')\right]\sum_{a'\in A}\pi(a|s)$$

$$\implies V(s) \leq \max_{a\in A(s)}\left[r(s,a) + \gamma \sum_{s'\in S} P(s'|s,a)V(s')\right]$$

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

The reverse inequality is only derived *heuristically*.

Suppose that we allow non-stationary policies π_a^ϵ, which first choose a fixed action $a \in A(s)$ and, for any $s' \in S$ and some fixed $\epsilon > 0$, follow an ϵ-optimal policy $\pi^{s'}$. (Here non-stationary indicates that the policy π changes for different s') This means:

$$V^{\pi^{s'}}(s') \geq V(s') - \epsilon.$$

The Bellman equation suggests that:

$$V(s) \geq V^{\pi_a^\epsilon}(s),$$

which can be expanded as:

$$V^{\pi^\epsilon_a}(s) = \sum_{a'\in A}\left[r(s,a') + \gamma \sum_{s'\in S} P(s'|s,a')\, \overbrace{V^{\pi^\epsilon_a}(s')}^{V^{s'}(s')}\right]\pi(a'|s),$$

where $\pi(a'|s) = 1$ if $a' = a$, and 0 otherwise. This simplifies to:

$$V^{\pi^\epsilon_a}(s) = r(s,a) + \gamma \sum_{s'\in S} P(s'|s,a) V^{\pi^{s'}}(s').$$

Using the assumption of ϵ-optimality:

$$V^{\pi^{s'}}(s') \geq V(s') - \epsilon,$$

we get:

$$V^{\pi^\epsilon_a}(s) \geq r(s,a) + \gamma \sum_{s'\in S} P(s'|s,a)[V(s') - \epsilon].$$

For any arbitrary $\epsilon > 0$, this implies:

$$V(s) \geq \max_{a\in A(s)} \left[r(s,a) + \gamma \sum_{s'\in S} P(s'|s,a)V(s')\right].$$

□

Theorem 2.9 (Bellman Optimality Equation)**.** *Let $W : S \to \mathbb{R}$ satisfy the Bellman optimality equation:*

$$W(s) = \max_{a\in A(s)} \left[r(s,a) + \gamma \sum_{s'\in S} P(s'|s,a)W(s')\right].$$

Let, moreover, $\pi^* \in \Pi$ be a policy with the property:

$$\pi^*(a|s) > 0 \implies a \in \arg\max_{a\in A(s)} \left[r(s,a) + \gamma \sum_{s'\in S} P(s'|s,a)W(s')\right].$$

If, in addition,

$$\lim_{n\to\infty} \gamma^n W(s_n) = 0 \quad \forall s \in S,$$

and

$$\limsup_{n\to\infty} \gamma^n W(s_n) = 0 \quad \forall s \in S,\, T \in \Pi,$$

then π^* is optimal for all initial states $s \in S$ and $W = V^*$.

Proof. For any $s \in S$ and $\pi \in \Pi$, we have:

$$W(s) = \max_{a\in A(s)} \left[r(s,a) + \gamma \sum_{s'\in S} p(s' \mid s,a) \cdot W(s')\right].$$

$$\geq \sum_{a_0\in A} \left[r(s,a_0) + \gamma \sum_{s_1\in S} p(s_1 \mid s, a_0) \cdot W(s_1)\right] \cdot \pi(a_0 \mid s). \quad \text{(Here we have equality if } \pi = \pi^*.\text{)}$$

$$= \sum_{(s_0,a_0)\in S\times A} [r(s_0,a_0) + \gamma W(s_1)] \cdot \delta_s[\{s_0\}] \cdot \pi(a_0 \mid s_0) \cdot p(s_1 \mid s_0, a_0). \quad \text{(Now we can use the equation for } s = s_1.\text{)}$$

$$\geq \sum_{(s_0,a_0)\in S\times A} \left[r(s_0,a_0) + \gamma \left(\sum_{(a_1,s_2)\in A\times S} (r(s_1,a_1) + \gamma W(s_2)) \cdot \pi(a_1 \mid s_1) \cdot p(s_2 \mid s_1, a_1)\right)\right] \cdot \delta_s[\{s_0\}] \cdot \pi(a_0 \mid s_0) \cdot p(s_1 \mid s_0, a_0).$$

$$= \sum_{(s_0,a_0),(s_1,a_1)\in S\times A, s_2\in S} \left[r(s_0,a_0)+\gamma r(s_1,a_1)+\gamma^2 W(s_2)\right]\cdot \delta_s[\{s_0\}]\cdot \pi(a_0 \mid s_0)\cdot p(s_1 \mid s_0,a_0)\cdot \pi(a_1 \mid s_1)\cdot p(s_2 \mid s_1,a_1).$$

(Apply the equation again for $W(s_2)$, etc.)

$$\geq \sum_{(s_0,a_0),\dots,(s_n,a_n)\in S\times A, s_{n+1}\in S} \left[\sum_{k=0}^{n}\gamma^k r(s_k,a_k)+\gamma^{n+1}W(s_{n+1})\right]\cdot\delta_s[\{s_0\}]\cdot\pi(a_0 \mid s_0)\cdot\prod_{k=1}^{n} p(s_k \mid s_{k-1},a_{k-1})\cdot\pi(a_k \mid s_k)\cdot p(s_{n+1} \mid s_n,a_n).$$

$$= \mathbb{E}_s^\pi\left[\sum_{k=0}^{n}\gamma^k r(S_k,A_k)+\gamma^{n+1}W(S_{n+1})\right].$$

This converges against something which, by dominated convergence, is larger than:

$$\mathbb{E}_s^\pi\left[\sum_{n=0}^{\infty}\gamma^n r(S_n,A_n)\right] = V^\pi(s).$$

Similarly, for $\pi=\pi^*$, we get equalities everywhere, so $W(s)=V^{\pi^*}(s)$. Hence:

$$V^{\pi^*}=W\geq \sup_{\pi\in\Pi} V^\pi = V \geq V^{\pi^*},$$

so:

$$W=V=V^{\pi^*}.$$

Thus, π^* is optimal for all states $s\in S$.

□

we can write the Bellman optimality equation in the following form:

$$W(s)=\max_{\pi}\sum_{a\in A}\pi(a|s)\left(\sum_r p(r|s,a)r+\gamma\sum_{s'\in S}p(s'|s,a)W(s')\right)$$

the red part are known variables from the environment. the optimal policy π^* is the maximal point for the RHS optimization problem. the optimal state value function is the maximum value of the RHS optimization.

Questions about the Bellman optimality equation:

- existence: does this equation have solutions? yes, by the contraction mapping theorem
- uniqueness: is the solution to this equation unique? the answer is that the optimal state value function $V(s)$ is unique but the optimal policy may be not unique.
- algorithm: how to solve this equation? iterative algorithm suggested by the contraction mapping theorem
- optimality: why we study this equation? because its solution corresponds to the optimal state vlaue and optimal policy

Theorem 2.10 (Existence of a fixed point). *Assume that $\gamma<1$ and define an operator T acting on functions $w:S\to\mathbb{R}$ by*

$$T[w](s):=\max_{a\in A(s)}\left[r(s,a)+\gamma\sum_{s'\in S}p(s' \mid s,a)w(s')\right],$$

then T has a unique fixed point $w^*:S\to\mathbb{R}$.

Proof. Denote by $\mathcal{W}$ the space of all functions $w:S\to\mathbb{R}$. This is a Banach space if we equip it with the norm $\|\cdot\|_\infty:\mathcal{W}\to\mathbb{R}$ given by

$$\|w\|_\infty:=\max_{s\in S}|w(s)| \quad \text{for } w\in\mathcal{W}.$$

In order to prove the theorem, it suffices to show that T is a contraction on $\mathcal{W}$ and invoke Banach's Fixed Point Theorem.

To see this, we simply compute for $w, \hat{w} \in \mathcal{W}$:

$$\|T[w] - T[\hat{w}]\|_\infty$$

$$= \max_{s\in S}\left|\max_{a\in A(s)}\left[r(s,a) + \gamma\sum_{s'\in S} p(s' \mid s,a)w(s')\right] - \max_{a\in A(s)}\left[r(s,a) + \gamma\sum_{s'\in S} p(s' \mid s,a)\hat{w}(s')\right]\right|$$

$$\leq \max_{s\in S}\max_{a\in A(s)}\left|r(s,a) + \gamma\sum_{s'\in S} p(s' \mid s,a)w(s') - \left(r(s,a) + \gamma\sum_{s'\in S} p(s' \mid s,a)\hat{w}(s')\right)\right|$$

$$= \max_{s\in S}\max_{a\in A(s)}\gamma\sum_{s'\in S} p(s' \mid s,a)|w(s') - \hat{w}(s')|$$

$$\leq \max_{s\in S}\max_{a\in A(s)}\gamma\sum_{s'\in S} p(s' \mid s,a)\|w - \hat{w}\|_\infty$$

$$= \gamma\|w - \hat{w}\|_\infty, \quad \text{since } \sum_{s'\in S} p(s' \mid s,a) = 1 \text{ and } \gamma < 1.$$

Thus, T is a contraction mapping, completing the proof.

Definition(contraction mapping) f is a contraction mapping if

$$\|f(x_1) - f(x_2)\| \leq \gamma\|x_1 - x_2\|$$

where $\gamma \in (0,1)$.

Theorem (contraction mapping theorem (banach fixed point theorem)) for any equation that has the form of $x = f(x)$, if f is a contraction mapping, then

- 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 $x_{k+1} = f(x_k)$, then $x_k \to x^*$ as $k \to \infty$. moreover, the convergence rate is exponentially fast.

□

Remark. *With the previous theorem, we now have essentially our first algorithm to solve MDPs. Starting from an arbitrary initial estimate $w_0 : S \to \mathbb{R}$ for the value function, we iteratively define:*

$$w_n(s) := T[w_{n-1}](s) = \max_{a\in A(s)}\left[r(s,a) + \gamma\sum_{s'\in S} p(s' \mid s,a)w_{n-1}(s')\right], \quad n \in \mathbb{N}_0.$$

this iterative algorithm is called value iteration.

By Banach's Fixed Point Theorem, for $\gamma < 1$, this sequence converges to the unique fixed point V of T.

An optimal policy is obtained by choosing, for each state $s \in S$, any action $a^*(s)$ maximizing $T[V](s)$. That is:

$$a^*(s) \in \arg\max_{a\in A(s)}\left\{r(s,a) + \gamma\sum_{s'\in S} p(s' \mid s,a)V(s')\right\}.$$

In particular, there always exists a deterministic optimal policy.

2.3 Policy Improvement and Policy Iteration

The fixed point argument developed in the previous section iteratively computes estimates of the value function to solve the MDP. In this section, we instead develop an iterative method to solve MDPs on the level of policies. This works by successively improving a given policy.

The key to constructing such improvements is the following theorem.

Standing assumption: For any $w : S \to \mathbb{R}$,

$$\gamma^n V^{\hat{\pi}}(S_n) \xrightarrow{n\to\infty} 0, \quad \text{where } \mathbf{P}_s^\pi \text{ is a.s.}, \forall \pi, \hat{\pi} \in \Pi, \forall s \in S.$$

Theorem 2.11 (Policy Improvement)**.** *Let $\pi, \pi' \in \Pi$ be two policies such that*

$$\sum_{a \in A} \pi(a \mid s) Q^\pi(s|a) \leq \sum_{a \in A} \pi'(a \mid s) Q^\pi(s|a), \quad \forall s \in S. \tag{$*$}$$

Then π' outperforms π, that is,

$$V^\pi \leq V^{\pi'}.$$

Moreover, if there exists $s_0 \in S$ such that the inequality in $(*)$ is strict, then

$$V^\pi(s_0) < V^{\pi'}(s_0).$$

Note: *The right-hand side of $(*)$ can be interpreted as: following the policy π' in the first step and then switching to π. If this performs better than choosing π to begin with, then π' must be a better strategy overall. This is due to the stationarity of the policies.*

Proof. Let $s \in S$. Then:

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

$$\leq \sum_{a \in A(s)} \pi'(a \mid s) Q^\pi(s|a) \tag{$+$}$$

(by definition of Q^π)

$$= \sum_{a \in A(s)} \pi'(a \mid s) \left[r(s,a) + \gamma \sum_{s' \in S} p(s' \mid s,a) V^{\pi'}(s') \right].$$

$$= \sum_{(s_0,a_0) \in S \times A} \left[r(s_0,a_0) + \gamma V^\pi(s_1) \right] \delta_s[\{s_0\}] p(s_1 \mid s_0, a_0) \pi'(a_0 \mid s_0),$$

$$= \mathbb{E}_s^{\pi'} \left[r(S_0, A_0) + \gamma V^\pi(S_1) \right].$$

Iterating this argument (using Bellman equation towards $V^\pi(S_n)$) yields:

$$V^\pi(s) \leq \mathbb{E}_s^{\pi'} \left[\sum_{k=0}^{n} \gamma^k r(S_k, A_k) + \gamma^{n+1} V^\pi(S_{n+1}) \right].$$

As $n \to \infty$:

$$\mathbb{E}_s^{\pi'} \left[\sum_{n=0}^{\infty} \gamma^n r(S_n, A_n) \right] = V^{\pi'}(s).$$

Finally, if the inequality in $(+)$ is strict, so is the inequality in $(*)$. □

The previous theorem gives an easy way of improving a given policy $\pi \in \Pi$ by choosing $\pi' \in \Pi$ s.t. $\pi'(a^*|s) = 1$ for some $a^* \in \arg\max_{a \in A(s)} Q^\pi(s|a)$. This implies

$$\sum_{a \in A} \pi(a|s) Q^\pi(s|a) \leq \sum_{a \in A} \pi'(a|s) Q^\pi(s|a)$$

i.e., π' is better than π

We say that π' is the *greedy* policy with respect to Q^π (because π' only allows to choose the best action a^*). More generally, let us introduce the following terminology.

Definition 2.12 (ϵ-greedy; ϵ-soft policy)**.** *Let $q : S \times A \to \mathbb{R}$, $\epsilon \in [0,1]$. A policy $\pi \in \Pi$ is called:*

(i) ϵ-soft, *if*

$$\pi(a \mid s) \geq \frac{\epsilon}{|A(s)|}, \quad \forall a \in A(s), s \in S.$$

(ii) ϵ-greedy with respect to q, *if, for all* $s \in S$, $a \in A(s)$*:*

$$\pi(a \mid s) = \begin{cases} (1-\epsilon) + \frac{\epsilon}{|A(s)|}, & \text{if } a = a^*(s), \\ \frac{\epsilon}{|A(s)|}, & \text{if } a \in A(s) \setminus \{a^*(s)\}, \end{cases}$$

where $a^*(s) \in \arg\max_{a \in A(s)} q(s,a)$ *is fixed for each* $s \in S$*. In the case* $\epsilon = 0$*, we simply refer to* π *as* greedy *with respect to* q*.*

To wit, an ϵ-greedy policy chooses an optimal action with respect to q with probability $(1-\epsilon)$ and a random action with probability ϵ chosen uniformly from $A(s)$.

Corollary 2.13 (Greedy Policy Improvement). *Let* $\pi \in \Pi$ *and choose* $\pi' \in \Pi$ *greedy with respect to* Q^π*. Then:*

$$V^\pi(s) \leq V^{\pi'}(s), \quad \forall s \in S.$$

Proof. This follows directly from the Policy Improvement Theorem since:

$$\pi'(a \mid s) \in \arg\max_{a \in A(s)} Q^\pi(s|a) \quad \text{with probability 1.}$$

This implies:

$$\sum_{a \in A(s)} \pi(a \mid s) Q^\pi(a|s) \leq \sum_{a \in A(s)} \pi'(a \mid s) Q^\pi(a|s).$$

□

We have therefore found a way of successively improving (the state value function) from a fixed initial policy $\pi_0 \in \Pi$. We simply choose $\pi_n \in \Pi$ to be greedy with respect to $Q^{\pi_{n-1}}, n \in \mathbb{N}$.

$$a^*(s) \in \arg\max_{a \in A(s)} Q^{\pi_{n-1}}, \qquad \pi_n = \begin{cases} (1-\epsilon) + \frac{\epsilon}{|A(s)|}, & \text{if } a = a^*(s), \\ \frac{\epsilon}{|A(s)|}, & \text{if } a \in A(s) \setminus \{a^*(s)\}, \end{cases}$$

The main question to ask now is, of course, if the policies $\{\pi_n\}_{n \in \mathbb{N}}$ converge and, if so, whether the limit constitutes an optimal policy.

Theorem 2.14 (Optimality via Policy Improvement). *Let* $\pi \in \Pi$ *and denote by* π' *a greedy policy with respect to* Q^π*. If*

$$V^\pi = V^{\pi'} \quad \text{or} \quad Q^\pi = Q^{\pi'},$$

then both π *and* π' *are optimal.*

Proof. Note that, by definition of Q^π and $Q^{\pi'}$, it follows from If $V^\pi = V^{\pi'}$, then also $Q^\pi = Q^{\pi'}$. Hence, we focus on the latter case.

We start by writing:

$$Q^{\pi'}(a|s) \overset{\text{def}}{=} r(s,a) + \gamma \sum_{s' \in S} p(s' \mid s,a) V^{\pi'}(s'),$$

$$= r(s,a) + \gamma \sum_{s' \in S} p(s' \mid s,a) \sum_{a' \in A} \pi'(a' \mid s') Q^{\pi'}(s'|a'),$$

$$= r(s,a) + \gamma \sum_{s' \in S} p(s' \mid s,a) \sum_{a' \in A} \pi'(a' \mid s') \max_{a \in A(s')} Q^\pi(s'|a'),$$

(since π' is greedy w.r.t. Q^π),

$$= r(s,a) + \gamma \sum_{s' \in S} p(s' \mid s,a) \max_{a' \in A(s')} Q^{\pi'}(s'|a').$$

Now, we use:

$$V^{\pi'}(s') = \sum_{a' \in A(s')} \pi'(a' \mid s') Q^{\pi'}(s'|a') \leq \max_{a' \in A(s')} Q^{\pi'}(s'|a')$$

to obtain

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

$$= \sum_{a\in A, s'\in S} \left[r(s,a) + \gamma p(s' \mid s,a) \max_{a'\in A(s')} Q^{\pi'}(s'|a') \right] \pi'(a \mid s),$$

$$\overset{(+)}{\geq} \sum_{a\in A} \left[r(s,a) + \gamma \sum_{s'\in S} p(s' \mid s,a) V^{\pi'}(s') \right] \pi'(a \mid s),$$

$$\text{(by Bellman)} = V^{\pi'}(s).$$

Thus, we have equality in (+), but since r, p, and $\pi' \geq 0$ and

$$\max_{a'\in A(s')} Q^{\pi'}(s'|a') \geq V^{\pi'}(s'),$$

this is only possible if:

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

$$= \max_{a'\in A(s')} \left\{ r(s,a') + \gamma \sum_{s'\in S} p(s' \mid s,a') V^{\pi'}(s') \right\}.$$

Hence, $V^{\pi'} = V^{\pi}$ solves the Bellman optimality equation, implying that π and π' are both optimal.

□

Corollary 2.15 (Suboptimality via Policy Improvement)**.**

Let $\pi \in \Pi$ and π' be greedy with respect to Q^{π}. If π is strictly suboptimal, there exists $s \in S$ such that:

$$V^{\pi}(s) < V^{\pi'}(s).$$

Now suppose that we start with an arbitrary policy $\pi_0 \in \Pi$ and construct an entire sequence of policies $\{\pi_n\}_{n\in\mathbb{N}} \subset \Pi$ by choosing π_n to be greedy with respect to $Q^{\pi_{n-1}}$. Observe that, by construction, each π_n is a deterministic policy. Finally, there are at most $|D|$ Deterministic policies. Since

$$V^{\pi_n} \geq V^{\pi_{n-1}}$$

and there are at most $|D|$ distinct state-value functions corresponding to deterministic policies, we must have

$$V^{\pi_n} = V^{\pi_{n-1}} \quad \text{for all } n \geq |D|,$$

implying that any such π_n is optimal! Hence, policy improvement can indeed be used to solve MDPs.

The policy improvement procedure considered here assumes that we are capable of computing state-value functions V^{π} or Action-value functions Q^{π} associated with arbitrary policies. This could, e.g., be done by solving the linear systems of equations arising from the Bellman equations.

Note, however, that this is only feasible if we actually *know* all transition probabilities p. The case of unknown p, which is at the core of RL, is considered in the next chapter. This is also when ϵ-greedy policies become increasingly important, as they are capable of *exploring* all state-action combinations with small probabilities, which greedy policies, which only *exploit* optimal Actions are not capable of.

Before digging deeper into this matter, we take a closer look at how ϵ-greedy policies behave in terms of policy improvement.

Proposition 2.16 (ϵ-greedy policy improvement)**.** *For $\epsilon \in (0,1]$, suppose that $\pi \in \Pi$ is ϵ-soft and π' is ϵ-greedy with respect to Q^{π}. Then:*

$$V^{\pi}(s) \leq V^{\pi'}(s), \quad \forall s \in S.$$

Proof. We write:

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

$$= \frac{\epsilon}{|A(s)|} \sum_{a\in A(s)} Q^{\pi}(s|a) + (1-\epsilon) \sum_{a\in A(s)} \frac{\pi(a \mid s) - \frac{\epsilon}{|A(s)|}}{1-\epsilon} Q^{\pi}(s|a),$$

$$\leq \frac{\epsilon}{|A(s)|} \sum_{a\in A(s)} Q^{\pi}(s|a) + (1-\epsilon) \max_{a\in A(s)} Q^{\pi}(s|a) \sum_{a\in A(s)} \frac{\pi(a \mid s) - \frac{\epsilon}{|A(s)|}}{1-\epsilon}.$$

Now observe that:

$$\sum_{a \in A(s)} \frac{\pi(a \mid s) - \frac{\epsilon}{|A(s)|}}{1-\epsilon} = 1,$$

since:

$$\sum_{a \in A(s)} \pi(a \mid s) = 1.$$

Thus:

$$\begin{aligned} V^{\pi}(s) &= \frac{\epsilon}{|A(s)|} \sum_{a \in A(s)} Q^{\pi}(s|a) + (1-\epsilon) \max_{a \in A(s)} Q^{\pi}(s|a), \\ &= \sum_{a \in A(s)} \pi'(a \mid s) Q^{\pi}(s|a). \end{aligned}$$

By the policy improvement theorem:

$$V^{\pi}(s) \leq V^{\pi'}(s).$$

□

2.4 Policy iteration algorithms

from above we have reached a way of improving the state value function:

from a fixed initial policy $\pi_0 \in \Pi$. And we call the above process value iteration. We simply choose $\pi_n \in \Pi$ to be greedy with respect to $Q^{\pi_{n-1}}, n \in \mathbb{N}$.

$$a^*(s) \in \arg\max_{a \in A(s)} Q^{\pi_{n-1}}, \quad \pi_n = \begin{cases} (1-\epsilon) + \frac{\epsilon}{|A(s)|}, & \text{if } a = a^*(s), \\ \frac{\epsilon}{|A(s)|}, & \text{if } a \in A(s) \setminus \{a^*(s)\}, \end{cases}$$

but we do not provide a way to find V^{π_n}. Actually, this is the difference among the following three algorithms: Policy iteration algorithm, Value iteration algorithm, and Truncated policy iteration algorithm.

from a fixed initial policy $\pi_0 \in \Pi$ (and an initial state value function V^{π_0}), at step n, we have Q^{π_n}, $a^*(s)$, and π_n. if we use $Q^{\pi_{n-1}}$ to approximate Q^{π_n}, we have

$$V^{\pi_n} = \sum_{a \in A(s)} \pi_n(a|s) Q^{\pi_{n-1}}(s|a) = \sum_{a \in A(s)} \pi_n(a|s) \left[r(s,a) + \gamma \sum_{s' \in S} p(s'|s,a) V^{\pi_{n-1}}(s') \right]$$

if we adopt this strategy, this is called value iteration algorithm.
Clearly V^{π_n} does not satisfy Bellman equation. But after this step we get a new state value function, if we iterate this step, the state value function will finally converge according to the above theorem. Now we get a state value function V^{π_n} which satisfy Bellman equation, i.e., the optimal state value function. if we adopt this strategy, this is called policy iteration algorithm. (every step of the iteration is embedded another iteration to solve the Bellman equation of the current policy)

In real implementation, we set a maximum number of iteration. If the maximum iteration is reached, we truncated the iteration and select the last state value function as the optimal state value function. This strategy is called truncated policy iteration algorithm.

since Bellman equation is satisfied every step in policy iteration, this algorithm outperforms compared with the other algorithms, but it also requires more computational resources.

3 Temporal Difference Learning

Now: Towards Reinforcement Learning

Setting: We do not know the transition probabilities $s' \to p(s' \mid s,a)$, but have an "oracle" which allows us to sample from p and any $\pi \in \Pi$.

Reminder:

$$V(s) = \max_{a \in A(s)} \left[r(s,a) + \gamma \sum_{s'} p(s' \mid s,a) V(s') \right]$$

$$Q^{\pi}(s|a) = r(s,a) + \gamma \sum_{s' \in S, a' \in A} p(s' \mid s,a) \underbrace{\pi(a' \mid s') Q^{\pi}(s'|a')}_{V^{\pi}(s')}$$

$$V^\pi(s) = \mathbb{E}_s^\pi\left[\sum_{n=0}^\infty \gamma^n r(S_n, A_n)\right] = \sum_{a\in A}\left[r(s,a) + \gamma\sum_{s'\in S} p(s' \mid s,a)V^\pi(s')\right]\pi(a \mid s)$$

Observe that, in this setting:

- We cannot solve the Bellman optimality equation as it explicitly involves p.
- The policy improvement algorithm breaks down since we have no means of computing Q^π or V^π without knowing p.

3.1 Temporal Difference Estimation

We begin by investigating how we can estimate V^π and/or Q^π for a fixed policy $\pi \in \Pi$ without knowing P.

The first, ad hoc, idea is to estimate:

$$V^\pi(s) = \mathbb{E}^\pi\left[\sum_{n=0}^\infty \gamma^n r(S_n, A_n)\right]$$

using the Law of Large Numbers.

To simplify notation, we write:

$$R_n = r(S_n, A_n)$$

for the immediate reward at time $n \in \mathbb{N}_0$.

Our assumption of having access to an **oracle** means we can produce An arbitrary number $M \in \mathbb{N}$ of realizations

$$r_n^1, r_n^2, \ldots, r_n^M \in R_n$$

are sampled independently under $\mathbb{P}^\pi$.

More precisely, we assume there are independent stochastic processes

$$\{(S_n^m, A_n^m)\}_{n\in\mathbb{N}_0, m\in\mathbb{N}}$$

which have the same distribution as $\{(S_n, A_n)\}_{n\in\mathbb{N}_0}$ under $\mathbb{P}_s^\pi$.

Writing $R_n^m = r(S_n^m, A_n^m)$, the oracle allows us to obtain one fixed realization of the first $M \in \mathbb{N}$ processes

$$\{(S_n^m, A_n^m, R_n^m)\}_{n\in\mathbb{N}_0}, \quad m = 1, 2, \ldots, M$$

up to a fixed time step $N \in \mathbb{N}_0$. That is, for some $\omega \in \Omega$ (fixed but unknown), we are given the values:

$$S_n^m = S_n^m(\hat{\omega}), \quad A_n^m = A_n^m(\hat{\omega}), \quad R_n^m = R_n^m(\hat{\omega}), \quad m = 1, \ldots, M, \quad n = 0, \ldots, N.$$

Note that:

$$\sum_{n=0}^N \gamma^n R_n \approx \sum_{n=0}^\infty \gamma^n R_n$$

if N is large.

And by the Strong Law of Large Numbers:

$$\frac{1}{M}\sum_{m=1}^M\sum_{n=0}^N \gamma^n R_n^M \to \mathbb{E}^\pi\left[\sum_{n=0}^\infty \gamma^n R_n\right], \quad M\to\infty, \quad \mathbb{P}_s^\pi \text{ a.s.}$$

In combination, it follows that

$$\hat{V}_m^\pi := \frac{1}{M}\sum_{m=1}^M\sum_{n=0}^N \gamma^n r_n^m = \frac{1}{M}\sum_{m=1}^M\sum_{n=0}^N \gamma^n \underbrace{r(S_n^m(\hat{\omega}), A_n^m(\hat{\omega}))}_{R_n^m(\hat{\omega})}$$

$$\xrightarrow{M\to\infty} \mathbb{E}_s^\pi\left[\sum_{n=0}^N \gamma^n r(S_n, A_n)\right] \xrightarrow{N\to\infty} \mathbb{E}_s^\pi\left[\sum_{n=0}^\infty \gamma^n r(S_n, A_n)\right] = V^\pi(s)$$

The quality of approximation clearly depends on the problem at hand.

For example, the approximation in N depends on the runtime of a single episode. In general, there are three different situations which may arise:

Definition 3.1. *A state $s \in S$ is called* ***absorbing*** *if*

$$P(s' = s \mid s, a) = 1, \quad \textit{for all } a \in A(s).$$

Moreover, s is called ***terminal*** *if it is absorbing and:*

$$r(s, a) = 0, \quad \textit{for all } a \in A(s).$$

If the state process $\{S_n\}_{n \in \mathbb{N}_0}$ enters into an absorbing state, it will never leave that state.

If the reward in that state is zero, the Markov Decision Process (MDP) is technically terminated, meaning the episode is effectively over.

Let us write $S_\dagger \subseteq S$ for the set of terminal states.

Definition 3.2. *The function $\zeta : \Omega \to \mathbb{N}_0 \cup \{+\infty\}$, defined as:*

$$\zeta(\omega) = \inf\{n \in \mathbb{N}_0 : S_n(\omega) \in S_t\},$$

is called the ***terminal time*** *of the MDP. There are exactly three different cases:*

1. ***Finite horizon problems:*** *There exists $N \in \mathbb{N}_0$ such that $\zeta \leq N$, $\mathbb{P}^\pi_s$-almost surely, $\forall s \in S$, $\pi \in \Pi$.*
2. ***Infinite horizon problems:*** *$\zeta = +\infty$, $\mathbb{P}^\pi_s$-almost surely, $\forall s \in S$, $\pi \in \Pi$.*
3. ***Random time horizon problems:*** *Any problem which is not a finite or infinite horizon problem.*

In what follows, we refer for each fixed $m \in \{1, \ldots, M\}$ to each sequence:

$$s_0^m, a_0^m, r_0^m, s_1^m, a_1^m, r_1^m, \ldots, s_N^m, a_N^m, r_N^m$$

produced by the oracle as a **roll-out**.

Again, these are interpreted as realizations of the random variables (R.V.):

$$S_0^m, A_0^m, r(S^m, A^m), \ldots, S_N^m, A_N^m, r(S_N^m, A_N^m).$$

In finite horizon problems, we can always guarantee that the length N of a roll-out corresponds to the number of time steps in an episode. Thus, there is no error in the approximation: $\sum_{n=0}^N \gamma^n R_n^m$ of $\sum_{n=0}^\infty \gamma^n R_n$. Conversely, in infinite horizon problems, there will always be an error. In random time horizon problems, one can choose N depending on M with the objective of observing an entire episode. However, there is no guarantee that this is possible.

Update rules

The (random) approximation of $V^\pi(s)$ after roll-out $m \in \{1, \ldots, M\}$ takes the form:

$$\hat{V}^\pi_m(s) = \frac{1}{m} \sum_{l=1}^{m} \sum_{n=0}^{N} \gamma^n \underbrace{r(S_n^l, A_n^l)}_{\triangleq R_n^l}.$$

Note that $\hat{V}^\pi$ can be computed iteratively with respect to m by:

$$\begin{aligned}\hat{V}^\pi_{m+1}(s) &= \frac{1}{m+1} \sum_{l=1}^{m} \sum_{n=0}^{N} \gamma^n R_n^l = \frac{1}{m+1} \sum_{n=0}^{N} \gamma^n R_n^{m+1} + \frac{m}{m+1} \sum_{l=1}^{M} \sum_{n=0}^{N} \gamma^n r_n^l \\ &= \frac{1}{m+1} \sum_{n=0}^{N} \gamma^n R_n^{m+1} + \frac{m}{m+1} \hat{V}^\pi_m(s).\end{aligned}$$

The update rule is much more memory-efficient as we don't have to store the rewards of the previous roll-outs $\ell = 1, 2, \ldots, m$.

Rewriting the update rule above by setting:

$$\alpha_{m+1}(s) = \frac{1}{m+1},$$

and using $\frac{m}{m+1} = 1 - \frac{1}{m+1} = 1 - \alpha_{m+1}(s)$, yields:

$$\underset{\text{new estimate}}{\hat{V}^\pi_{m+1}(s)} \;=\; \underset{\text{old estimate}}{\hat{V}^\pi_m(s)} \;+\; \underset{\text{learning rate}}{\alpha_{m+1}(s)} \underset{\text{error in the estimate}}{\left[\underset{\text{target}}{\sum_{n=0}^{\infty} \gamma^n R_n^{m+1}} - \hat{V}^\pi_m(s) \right]}.$$

note that

$$\hat{V}^\pi_{m+1}(s) - \sum_{n=0}^{\infty} \gamma^n R_n^{m+1} = \hat{V}^\pi_m(s) - \sum_{n=0}^{\infty} \gamma^n R_n^{m+1} - \alpha_{m+1}(s)\left[\hat{V}^\pi_m(s) - \sum_{n=0}^{\infty} \gamma^n R_n^{n+1}\right] = (1-\alpha_{m+1}(s))\left[\hat{V}^\pi_m(s) - \sum_{n=0}^{\infty} \gamma^n R_n^{m+1}\right]$$

clearly we have

$$\left|\hat{V}^\pi_m(s) - \sum_{n=0}^{\infty} \gamma^n R_n^{m+1}\right| \geq \left|\hat{V}^\pi_{m+1}(s) - \sum_{n=0}^{\infty} \gamma^n R_n^{m+1}\right|$$

Some tricks to make the estimation more efficient in practice:

1. Instead of sampling from $\mathbb{P}^\pi_s$, one can sample from $\mathbb{P}^\pi_\mu$, where μ is a distribution on S which puts positive probability on each state $s \in S$. The m-th roll-out $\{(s_n^m, a_n^m, r_n^m)\}, n = 0, \dots, N$ is used to update the estimate of $V^\pi(s_0^m)$ corresponding to the randomly chosen initial state.

2. Each roll-out can be used to update several different estimates of V^π. Indeed, if for $n \in \{1, \dots, N\}$, $s_n^m \notin \{s_0^m, s_1^m, \dots, s_N^m\}$, we can think of (s_j^m, a_j^m, r_j^m), $j = n, \dots, N$ as a roll-out with initial state s_n^m. This approach is called **first-visit Monte Carlo estimation**. note the condition '$s_n^m \notin \{s_0^m, s_1^m, ..., s_N^m\}$', which corresponds to the so-called 'first-visit'

The advantage is that it requires fewer samples compared to the original approach in which each roll-out is only used once. The disadvantage is that the estimate for $V^\pi(s_0^m)$ and $V^\pi(s_n^m)$ are not independent (because they are stochastically dependent on the realizations from the same random variables, namely S_j^m, A_j^m, $j = n, \dots, N$).

3. In practice, one often even goes one step further and uses every subsequence of a roll-out

$$(s_0^m, a_0^m, r_0^m), \dots, (s_N^m, a_N^m, r_N^m)$$

note we may find (s_0^m, a_0^m, r_0^m) several times inside the sequence

to update the estimation of $V^\pi(s_0^m)$

$$(s_1^m, a_1^m, r_1^m), \dots, (s_N^m, a_N^m, r_N^m)$$

similarly we may find (s_1^m, a_1^m, r_1^m) several times inside the sequence

to update the estimation of $V^\pi(s_1^m)$, and so on,

$$(s_j^m, a_j^m, r_j^m), \dots, (s_N^m, a_N^m, r_N^m)$$

to update the estimation of $V^\pi(s_j^m)$.

This is done irrespective of whether or not a state has been visited before. This causes dependence between reward samples within estimates of $V^\pi(s_n^m)$ as soon as s_n^m is visited more than once in one roll-out.

It is hard to justify theoretically that these estimates still converge to the state value function. This approach is called **every visit Monte Carlo estimation**.

3.2 Temporal difference estimation: SARSA

In Monte Carlo, we approximate:

$$V^\pi(s) = \mathbb{E}^\pi_s\left[\sum_{n=0}^{\infty} \gamma^n r(s_n, a_n)\right].$$

Instead, we could also try to approximate:

$$V^\pi(s) = \mathbb{E}^\pi_s\left[r(S_0, A_0) + \gamma V^\pi(s_1)\right].$$

This is achieved as follows: instead of using the MC update rule,

$$\hat{V}^\pi_{m+1}(s) = \hat{V}^\pi_m(s) + \alpha_{m+1}(s)\left[\sum_{n=0}^{N} \gamma^n r_n^m - \hat{V}^\pi_m(s)\right],$$

we change the target as follows. Let:

$$\ell = \{1, \dots, M\}, \quad n \in \{0, \dots, N-1\},$$

and write:

$$(s, a, r, s', a') = \left(s_n^\ell, a_n^\ell, r(s_n^\ell, a_n^\ell), s_{n+1}^\ell, a_{n+1}^\ell\right).$$

Suppose we have an estimate $\hat{V}^\pi_m(s)$, which has been updated m times before. Then the update rule is:

$$\hat{V}^\pi_{m+1}(s) = \hat{V}^\pi_m(s) + \alpha_{m+1}(s)\left[r + \gamma \hat{V}^\pi_m(s') - \hat{V}^\pi_m(s)\right], \tag{+}$$

where $\{\alpha_{m+1}(s)\}_{m\in\mathbb{N}} \subset [0,\infty)$ is a sequence such that:

$$\sum_{m=1}^{\infty} \alpha_m(s) = \infty, \quad \text{but} \quad \sum_{m=1}^{\infty} \alpha_m^2(s) < \infty. \quad \text{Robbins-Monro condition}$$

if we select $\alpha_m(s) \equiv \frac{1}{m}$, this is implicitly taking the average value towards the m estimates of $\hat{V}_m^\pi(s)$. The update rule corresponds to an every visit Monte Carlo estimation in which the learning rate has been generalized (ever so slightly) and, more importantly, the target

$$\sum_{n=0}^{\infty} \gamma^n R_n$$

has been replaced with

$$R_0 + \gamma V^\pi(S_1)$$

inspired by the Bellman equation.
The estimation procedure based on (+) is referred to as *(zero step) temporal difference estimation*, or *TD(0)* for short. We note here that there is also an *n-step variant TD(n)* in which the target is

$$\sum_{k=0}^{n} \gamma^k R_k + \gamma^{n+1} V^\pi(S_{n+1}),$$

which we will not discuss any further. We merely note that every visit Monte Carlo estimation can be thought of as $TD(\infty)$.

Finally, let us note that there is also a version of temporal difference learning for the action value function. Since we may interpret

$$Q^\pi(s|a) = \sum_{s'\in S, a'\in A} [r(s,a) + \gamma Q^\pi(s'|a')] \cdot p(s'|s,a) \cdot \pi(a'|s')$$

as

$$Q^\pi(s|a) = \mathbb{E}_s^\pi \left[r(S_0, A_0) + \gamma \cdot Q^\pi(S_1|A_1) \mid A_0 = a \right],$$

this actually another form of Bellman equation. we obtain an update rule for the estimate $\hat{Q}_m^\pi(s|a)$ of $Q^\pi(s|a)$ of the form

$$\hat{Q}_{m+1}^\pi(s|a) = \hat{Q}_m^\pi(s|a) + \alpha_{m+1}(s) \left[r + \gamma \hat{Q}_m^\pi(s'|a') - \hat{Q}_m^\pi(s|a) \right].$$

the form is similar to the estimation of $\hat{V}_{m+1}^\pi(s)$ in the above. This estimation procedure is referred to as the *SARSA(0)* estimation rule SARSA is the abbreviation of state-action-reward-state-action, as updates are produced from a sequence (s, a, r, s', a') of states, actions and a reward. Of course, there are also n-step variants *SARSA(n)*.

Remark. *what SARSA does is actually solving the Bellman equation.*

3.3 Temporal Difference Control

Now that we have found a way to estimate V^π and Q^π without explicitly knowing the transition probability function p, let us devise a way of constructing good or even optimal policies without knowing p.

The central idea is to combine the temporal difference estimation, or more precisely SARSA(0), and policy improvement. The policy improvement step is performed after every update of our estimates for the action value function. This results in the following algorithm:

1. Initialize a function $q : S \times A \to \mathbb{R}$,

$$(s,a) \mapsto q(s,a)$$

   arbitrarily (e.g. $q \equiv 0$) except that $q(s|a) = 0$ if $s \in S_\dagger$ (terminal state), $a \in A$.

2. Initialize a learning rate function $\alpha : S \times A \to [0,\infty]$ (e.g. $\alpha \equiv 1$).

3. Loop $m = 1, \ldots, M$ (looping over episodes)

   (a) Initialize S_0^m (e.g. randomly)

   (b) Choose an action A_0^m for S_0^m using a policy derived from q (typically ϵ-greedy).

   (c) Loop over $n = 0, 1, \ldots, N$ (looping over time steps)

   - take action A_n^m and observe both the reward $r(S_n^m, A_n^m)$ and the next state S_{n+1}^m.
   - choose an action A_{n+1}^m for S_{n+1}^m using a policy derived from q (typically ϵ-greedy). recall ϵ-greedy means that $\pi_{n+1}^m(a_{n+1}^m|s_{n+1}^m) = 1 - \frac{\epsilon}{|\mathcal{A}|}(|\mathcal{A}| - 1)$ if $a_{n+1}^m = \arg\max_a q(s_{n+1}^m, a)$, otherwise $\pi_{n+1}^m = \frac{\epsilon}{|\mathcal{A}|}$

- update the function q by setting (note that here q is the action at $(S_n^m|A_n^m)$).

$$q(S_n^m|A_n^m) \leftarrow q(S_n^m|A_n^m) + \alpha(S_n^m, A_n^m)\left[r(S_n^m, A_n^m) + \gamma q(S_{n+1}^m|A_{n+1}^m) - q(S_n^m|A_n^m)\right]$$

- update the learning rate function α in the state S_n^m, A_n^m (e.g. $\alpha(S_n^m, A_n^m) = \frac{1}{1+\#\text{visits of }(S_n^m, A_n^m)}$)
- stop this inner loop early if S_n^m is a terminal state.

(Here we mention the relation between SARSA and MC: in the step of updating the function q, recall that the definition of $SARSA(0)$ value is $Q^\pi(s|a) = \mathbb{E}_s^\pi[r(S_0, A_0) + \gamma \cdot Q^\pi(S_1|A_1)|A_0 = a]$. if we change the update rule: $Q^\pi(s|a) = \mathbb{E}_s^\pi[r(S_0, A_0) + \gamma \cdot r(S_1, A_1) + \gamma^2 \cdot Q^\pi(S_2|A_2)|A_0 = a] = ... = \mathbb{E}_s^\pi[r(S_0, A_0) + \gamma \cdot r(S_1, A_1) + ... + \gamma^{n-1} \cdot r(S_{n-1}, A_{n-1}) + \gamma^n \cdot Q^\pi(S_n|A_)|A_0 = a]$, this is the $SARSA(n)$. If we have infinitely many r: $Q^\pi(s|a) = \mathbb{E}_s^\pi[r(S_0, A_0) + \gamma \cdot r(S_1, A_1) + ... + \gamma^n r(S_n, A_n) + ...|A_0 = a]$, this is Monte Carlo (also we need to choose $\alpha \equiv 1$ to be MC)).

In plain words, the algorithm works by alternating between estimation of action value functions using SARSA(0) and an update of the policy. For updating the policy, one might be inclined to use *greedy actions*, instead of ϵ-greedy actions, as this has given us the best chances of convergence to an optimal policy in the past. But this is not a good idea. Be aware that we have to balance *optimization* and *estimation*. The policy determines which states and actions are being updated in the estimation step. By choosing greedy actions, some state-action pairs might never be visited (although these might optimal ones). This problem is referred to as the *exploration-exploitation trade-off*.

In practice, one deals with this problem by choosing ϵ-greedy actions instead. This ensures that all state-action pairs are eventually visited. The downside of this approach is that, if ϵ is kept fixed, we can at most attain an ϵ-soft-optimal policy.

There is another way of dealing with this issue. Roughly, the idea is to use two different policies.

- a *behavioral policy* π_b which is used to generate the MDP. It should be chosen s.t. the state and action spaces are being explored (ϵ-greedy, say), which is used to generate experience samples.

- a *control policy* π^* which is used to update the estimator q toward an optimal policy. Choosing this policy to be greedy w.r.t. q essentially means that we have to replace the SARSA(0) target

  $$r(S_n^m, A_n^m) + \gamma \cdot q(S_{n+1}^m|A_{n+1}^m)$$

  by

  $$r(S_n^m, A_n^m) + \gamma \cdot \max_{a \in A(S_{n+1}^m)} q(S_{n+1}^m|a).$$

This method is referred to as *off-policy temporal difference control*, or more compactly and easier to remember **Q-learning**.

Remark. *when the behavioral policy is different with the control policy, the learning is called off-policy. If they are the same, the learning is called on-policy.*

Remark. *what Q-learning does mathematically is to solve the equation:*

$$q(S_n^m|A_n^m) = \mathbb{E}_s^\pi[r(S_n^m, A_n^m) + \gamma \cdot \max_{A \in A(S_{n+1}^m)} q(S_{n+1}^m|a)]$$

which is indeed solving the Bellman optimality equation (expressed in terms of action values).

The algorithm before (with the actual SARSA(0) update) is called accordingly *on-policy temporal difference control* or **SARSA** for short.

Finally, there is another version of the algorithm called *expected SARSA* where the target is given by

$$r(S_n^m, A_n^m) + \gamma \cdot \sum_{a' \in A(S_{n+1}^m)} q(S_{n+1}^m|a')\pi_b(a'|S_{n+1}^m).$$

where π_b is the behavioral policy. Finally, there are of course n-step variants of these algorithms.

SARSA is an on-policy learning: the policy π_{n+1}^m derived from q generates a_{n+1}^m and $s_n^m, a_n^m, r(s_n^m, a_n^m), s_{n+1}^m$ are not dependent to policy. So π_{n+1}^m is the behavioral policy. And review the algorithm we note that q is updated by A_{n+1}^m and $S_{n+1}^m, ..., \pi_{n+1}^m$ clearly is used to update q. Similarly, MC learning is on-policy.

Q-learning is off-policy: since we introduce the optimization inside the form of target, the behavioral policy to generate a_{n+1}^m from s_{n+1}^m maybe not optimal policy, but the control policy is the optimal policy. (but if the 2 policies happen to be the same, Q-learning is on-policy)

and since the behavioral policy π_b is used to generate the MDP, π^* can be greedy rather than ϵ-greedy as the exploration is accomplished by π_b

4 Exercises

4.1 Exercise 2

Exercise 4.1. *Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a finite probability space, i.e., $\Omega = \{\omega_1, \dots, \omega_n\}$. Show that there exists no sequence $\{Z_n\}_{n\in\mathbb{N}}$ of independent, almost surely non-constant random variables.*

Proof. W.l.o.g. assume $\mathbb{P}(\{\omega_i\}) > 0$, $\forall i = 1, \dots, n$.

Let $\{Z_k\}_{k\in\mathbb{N}}$ be a sequence of independent random variables. Towards a contradiction, assume each Z_k, $k \in \mathbb{N}$, is not constant.

As $|\Omega| < \infty$, we get $|\mathcal{A}| < \infty$ (as $A \in \mathcal{P}(\Omega)$ and $|\mathcal{P}(\Omega)| = 2^{|\Omega|} < \infty$).

With that, there exists $k_1, k_2 \in \mathbb{N}$, $k_1 \neq k_2$, with

$$\sigma(Z_{k_1}) = \sigma(Z_{k_2}) \subseteq \mathcal{A}.$$

For any $A \in \sigma(Z_{k_1}) = \sigma(Z_{k_2})$, we get

$$\mathbb{P}(A) = \mathbb{P}(A \cap A) = \mathbb{P}(A)^2 \quad \Rightarrow \quad \mathbb{P}(A) \in \{0, 1\}.$$

$$\Rightarrow \sigma(Z_{k_1}) = \sigma(Z_{k_2}) \subseteq \{\Omega, \emptyset\}$$

$$\Rightarrow Z_{k_1} \text{ a.s. constant}$$

which is a contradiction. □

Exercise 4.2. *Let $(\Omega, \mathcal{F})$ be a measurable space with $\Omega = \{\omega_1, \dots, \omega_n\}$ finite. Show that there exists exactly one partition $\mathcal{P}$ of Ω which generates $\mathcal{F}$.*

Exercise 4.3. *Let $(\Omega, \mathcal{F})$ be a measurable space with $\Omega = \{\omega_1, \dots, \omega_n\}$ finite. Let moreover $X : \Omega \to \mathbb{R}$ and denote by $\mathcal{P} = \{P_1, \dots, P_j\}$ the unique partition of Ω which generates $\mathcal{F}$. Show that X is $\mathcal{F}$-measurable if and only if X is constant on each $P \in \mathcal{P}$.*

Proof. Note: As Ω is finite, $X(\Omega) = \{x_1, \dots, x_m\}$ is finite.

$\Leftarrow$ Assume X is constant on every $P \in \mathcal{P}$ (i.e., $X = x_j$ on P_k).

$$\{\omega \in \Omega \mid X(\omega) = x_j\} = \bigcup_{\substack{k=1,\dots,j \\ X=x_i \text{ on } P_k}}^{m} P_k \in \mathcal{F}, \quad \forall i = 1, \dots, m.$$

$$\Rightarrow X \text{ is } \mathcal{F}\text{-measurable.}$$

$\Rightarrow$ Assume X is $\mathcal{F}$-measurable. By last exercise, $\mathcal{P} = \{P_\omega \mid \omega \in \Omega\}$, with $P_\omega = \bigcap\{A \in \mathcal{F} \mid \omega \in A\}$, $\omega \in \Omega$.
Assume there exists $p \in \mathcal{P}$, $\omega_1, \omega_2 \in p$, s.t. $x_1 := X(\omega_1) \neq X(\omega_2) =: x_2$.

$$\mathcal{F} := \{\omega \in \Omega \mid X(\omega) = x_j\} \in \mathcal{F} = \bigcup_{\omega \in \mathcal{F}} P_\omega$$

As $\omega_1 \in F$ and $\omega_2 \in F \Rightarrow$ contradiction, as $X(\omega_2) = x_2 \neq x_1$.

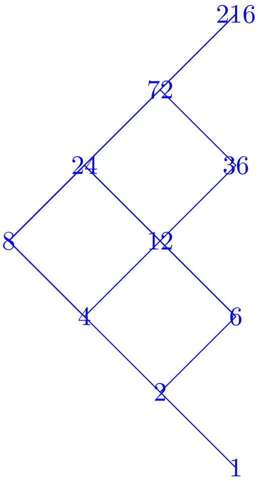
$$\Rightarrow X \text{ constant on every } P \in \mathcal{P}.$$

□

Exercise 4.4. *Let P be the price of a financial asset in the CRR model with time horizon $T = 3$, $P_0 = 8$, $d = -0.5$, $u = 2$, and $q = 0.5$. Define a Markov decision model that allows trading of the asset P.*

For that, assume that the agent starts out with an initial wealth of $w_0 := 100$ and they are only allowed to hold integer amounts of P. Negative positions (short-selling) and positions exceeding the total wealth (borrowing) are not allowed. The objective of the agent is to maximize expected log-utility of terminal wealth, that is, the expectation of $\log(W_T)$ with W_T denoting the wealth at time T.

Hint: States should be three-dimensional and contain the current time, the financial asset price, and the agent's wealth. Actions can be modeled as the amount of wealth invested into financial asset.



solution.

We start with the following definitions:

$$P_0 := \{8\}, \quad W_0 := \{100\} \quad \text{(possible asset prices and wealth at } t = 0\text{)}$$
$$A_0 := \{a \in \mathbb{R} \mid a = p_0 \cdot k,\ k \in \mathbb{N}_0,\ p_0 \in P_0,\ a \le \max\{W_0\}\}$$
$$E_0 := \{0\} \times P_0 \times W_0, \quad D_0 := E_0 \times A_0$$

For $n = 1, 2, 3$ **we define inductively:**

- $P_n := \{p \in \mathbb{R}_+ \mid p = (1+R)p_{n-1},\ p_{n-1} \in P_{n-1},\ R \in \{d, u\}\}$ *(possible asset prices)*
- $W_n := \{w_n \in \mathbb{R} \mid w_n = w_{n-1} - a_{n-1} + a_{n-1}(1+R),\ R \in \{d,u\}, ((n-1, p_{n-1}, w_{n-1}), a_{n-1}) \in D_{n-1}\}$ *(possible wealth)*
- $E_n := \mathbb{Z} \times P_n \times W_n$ *(state space for $t = n$)*
- $A_n := \{a \in \mathbb{R}_+ \mid a = p_{n-1} \cdot k,\ k \in \mathbb{N}_0,\ p_{n-1} \in P_{n-1}$ and $a \le \max\{W_{n-1}\}\}$ *(action space for $t = n$)*
- $D_n := \{(n, p_n, w_n, a) \in E_n \times A_n \mid a \le w_n\}$ *(valid state-action pairs)*

For A_3, D_3, we make an exception. We set

$$A_3 := \{\emptyset\} \quad \text{and} \quad D_3 := E_3 \times A_3.$$

This will be useful to have an infinite time horizon needed in the definition of MDP.

- The state space is now defined as: $E := E_0 \cup E_1 \cup E_2 \cup E_3 \cup \{(t, \top)\}$
- Action space: $A := A_0 \cup \cdots \cup A_3$
- Valid state-action pairs: $D := D_0 \cup D_1 \cup D_2 \cup D_3 \cup \{(t, \top)\}$
- The transition probability function p is defined as: $p\left((n+1, p', w') \mid (n, p, w), a\right) = 0.5$ if $p' = (1+R)p$, $w' = w + a \cdot R$, $R \in \{d, u\}, n \in \{0, 1, 2\}, p \in P_n,$ and $w \in W_n$. Further,

  $$p\left((t, \top) \mid (n, p, w), a\right) = 1 \quad \text{for } n = 3,\ p, w, a \text{ arbitrary.}$$

  and

  $$p\left((t, \top) \mid (t, \top)\right) = 1.$$

  All other probabilities are 0.
- Reward function r is defined as:

  $$r((n, p, w), a) = \log(w) \quad \text{if } n = 3.$$
- Discount factor: $\gamma = 1$, as it isn't needed.

Given this setting and a policy π, we get a MDP

$$\{S_k\}_{k \in \mathbb{N}_0}, \quad \{A_k\}_{k \in \mathbb{N}_0} \quad \text{with expected return}$$

$$\mathbb{E}^\pi \left[\sum_{k=0}^{\infty} \gamma^k\, r(S_k, A_k) \right] = \mathbb{E}^\pi \left[\log\left(\widetilde{W}_{3-t} \right) \right],$$

with $S_k = (t_k, \widetilde{P}_k, \widetilde{W}_k)$ and $s = (t, p, w) \in \mathcal{S}$ denotes an arbitrary start-state.

□

4.2 Exercise 5

Exercise 4.5. *Let (S, A, D, p, r, γ) be a Markov Decision Model, π a policy and μ a probability measure on S. From the Ionescu-Tulcea Theorem for Markov Decision Models, let $\mathbb{P}^{\pi}_{\mu}$ be the unique probability measure on $(\Omega, \mathcal{A})$, with the property*

$$\mathbb{P}^{\pi}_{\mu}[B \times \bigtimes_{k=n+1}^{\infty} (S \times A)] = \sum_{(s_0,a_0,\dots,s_n,a_n)\in B} \mu(\{s_0\})\pi(a_0 \mid s_0) \prod_{k=1}^{n} p(s_k \mid s_{k-1}, a_{k-1})\pi(a_k \mid s_k),$$

for all $B \subseteq (S \times A)^{n+1}$ and $n \in \mathbb{N}_0$.

Prove that $\mathbb{P}^{\pi}_{\mu}$ satisfies the following properties:

1. For all $n \in \mathbb{N}_0$ and $(s, a) \in S \times A$,

$$\mathbb{P}^{\pi}_{\mu}[A_n = a \mid S_n = s] = \pi(a \mid s).$$

2. For all $n \in \mathbb{N}_0$ and $(s_k, a_k)_{k=0,\dots,n+1} \in (S \times A)^{n+2}$,

$$\begin{aligned}\mathbb{P}^{\pi}_{\mu}[S_{n+1} = s_{n+1}, A_{n+1} = a_{n+1} \mid S_0 = s_0, A_0 = a_0, \dots, S_n = s_n, A_n = a_n] \\ = \mathbb{P}^{\pi}_{\mu}[S_{n+1} = s_{n+1}, A_{n+1} = a_{n+1} \mid S_n = s_n, A_n = a_n].\end{aligned}$$

Exercise 4.6. *(Bellman equation) In the setting of exercise class 2 task 4, show that for any deterministic policy $\pi : S \to A$, $t = 0, 1, 2$ and $s = (t, p, w) \in S$, the following equation holds:*

$$V^{\pi}(t, p, w) = \gamma \sum_{(t+1,p',w')\in S} p(t + 1, p', w' \mid s, \pi(s))V^{\pi}(t + 1, p', w').$$

Compute for all $s = (3, p, w) \in S$ the value $V^{\pi}(s)$.

4.3 Exercise 6

Exercise 4.7. *(Bellman optimality equation in the optimal investment problem) Consider the setting of the optimal investment problem discussed in exercise class 2 task 4. Let $W : S \to \mathbb{R}$ be a function defined as*

$$W(\dagger) := 0, \quad W(T, p, w) := \log(w) \quad \text{and} \quad W(t, p, w) := \max_{a\in A(t,p,w)} \sum_{s':=(t+1,p',w')\in S} p(s' \mid (t, p, w), a)V(s'),$$

for any $(t, p, w) \in S$ with $t < T$.

Show that $W = V$, where V is the optimal value function, and that policies $\pi^* \in \Pi_d$, for which for all $s = (t, p, w) \in S$, with $t < T$, an action $a(s) \in A$ exists with

$$\pi^*(a(s) \mid s) = 1 \quad \text{and} \quad a(s) \in \arg\max_{a\in A(s)} \sum_{s':=(t+1,p',w')\in S} p(s' \mid s, a)V(s'),$$

it holds that π^* is optimal for all s.

Hint: *Argue analogously to the proof of Theorem 2.9. Start by showing that $W(\dagger) = V(\dagger)$, continue by showing that $W(T, \cdot) = V(T, \cdot)$. Finally, perform a backwards induction, to show that the policy is optimal and that $V = W$.*

4.4 Exercise 7

Exercise 4.8. *(Existence of a fixed point) Let $\gamma < 1$ and define an operator T acting on functions $w : S \to \mathbb{R}$ by*

$$T[w](s) := \max_{a\in A(s)} \left[r(s, a) + \gamma \sum_{s'\in S} p(s' \mid s, a)w(s') \right], \quad \text{for } s \in S.$$

Show that T has a unique fixed point $W : S \to \mathbb{R}$, that is, $W(s) = T[W](s)$ for all $s \in S$.

Hint: *Use Banach's fixed point theorem with the supremum norm on the space of functions $w : S \to \mathbb{R}$.*

4.5 Exercise 8

Exercise 4.9. *Let $\gamma < 1$, π an arbitrary policy and define an operator T^{π} acting on functions $w : S \to \mathbb{R}$ by*

$$T^{\pi}[w](s) := \sum_{s'\in S, a\in A} [r(s, a) + \gamma w(s')] \pi(a \mid s)p(s' \mid s, a), \quad \text{for } s \in S.$$

Show that T^{π} has a unique fixed point $W : S \to \mathbb{R}$, that is, $W(s) = T^{\pi}[W](s)$ for all $s \in S$.

Exercise 4.10. *Let $M = (S, A, D, p, r, \gamma)$ be a Markov Decision Model, with $\gamma < 1$. Show that there exists a Markov Decision Model $\widetilde{M} = (S, A, D, \widetilde{p}, \widetilde{r}, \gamma)$ such that*

$$\sup_{\pi \in \Pi^M \ \epsilon\text{-}soft} V^\pi(s) = \sup_{\pi \in \Pi^{\widetilde{M}}} V^\pi(s), \quad \textit{for all } s \in S,$$

where Π^M and $\Pi^{\widetilde{M}}$ denote the set of all policies in the Markov Decision Models M and $\widetilde{M}$.

Hints: *Start by defining a new transition probability function $\widetilde{p}$ that incorporates the ϵ-softness into the new MDM and adjust the reward function. Show that there exists a transformation of the policies such that the value function remains invariant with respect to these changes. For the last point, use the results of exercise 1.*

Definition 4.11. *(ϵ-soft optimal) An ϵ-soft policy π^* is called ϵ-soft optimal if*

$$V^{\pi^*}(s) = \sup_{\pi \ \epsilon\text{-}soft} V^\pi(s) =: \widetilde{V}^*(s), \quad \textit{for all } s \in S.$$

Exercise 4.12. *Let $\gamma < 1$ and π_0 be an arbitrary ϵ-soft policy and $\{\pi_n\}_{n \in \mathbb{N}} \subseteq \Pi$ be a sequence of ϵ-soft policies, where π_n is chosen to be ϵ-greedy with respect to $Q^{\pi^{n-1}}$, for all $n > 0$. Show that for some $N \in \mathbb{N}$, for all $m \geq N$, the policy π_m is ϵ-soft optimal.*

Hint: *Use exercise 4.10.*

4.6 Exercise 10

Exercise 4.13. *Consider the setting of the optimal investment problem discussed in Exercise Class 2, **Task 4**. Implement a script in Python that approximates the optimal strategy in the optimal investment problem using the Q-Learning algorithm (the pseudocode can be found in the Handwritten Lecture Notes 10). Visualise the learned portfolio allocations after 50,000, 500,000 and 5,000,000 episodes, as shown in Figure 1.6 in the lecture notes (see also exercise class 6).*

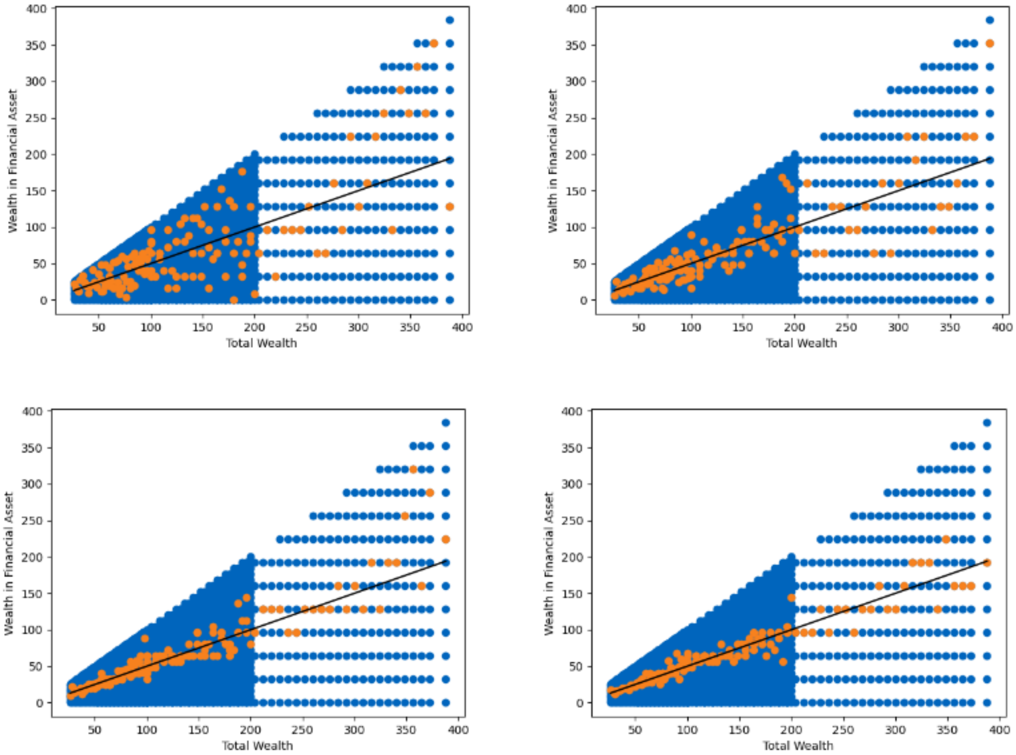


Figure 1.6: Learned portfolio allocations after 50,000 episodes (top left), 500,000 episodes (top right), 5,000,000 episodes (bottom left), and 50,000,000 episodes (bottom right).

*Use only the Python packages **NumPy** and **Matplotlib** and work alone on this task. Make sure that the 3 figures are reproducible, i.e. fix the NumPy seed before running the code (see here).*

Submit the Python file (either `.py` or `.ipynb` file) and additionally the 3 figures as PNG files (use the command `matplotlib.pyplot.savefig`) to the submission folder on the Mathematics of Reinforcement Learning Moodle website.

5 Appendix

Theorem 5.1 (Law of large numbers)**.** *For a random variable X, suppose that $\{x_i\}_{i=1}^n$ are some i.i.d. samples. Let*

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

be the average of the samples. Then,

$$\mathbb{E}[\bar{x}] = \mathbb{E}[X],$$

$$\mathrm{var}[\bar{x}] = \frac{1}{n}\mathrm{var}[X].$$

The above two equations indicate that $\bar{x}$ is an unbiased estimate of $\mathbb{E}[X]$, and its variance decreases to zero as n increases to infinity.

Proof. First,

$$\mathbb{E}[\bar{x}] = \mathbb{E}\left[\sum_{i=1}^{n} x_i/n\right] = \sum_{i=1}^{n}\mathbb{E}[x_i]/n = \mathbb{E}[X],$$

where the last equality is due to the fact that the samples are *identically distributed* (that is, $\mathbb{E}[x_i] = \mathbb{E}[X]$).

Second,

$$\mathrm{var}(\bar{x}) = \mathrm{var}\left(\sum_{i=1}^{n} x_i/n\right) = \sum_{i=1}^{n}\mathrm{var}[x_i]/n^2 = (n\cdot\mathrm{var}[X])/n^2 = \mathrm{var}[X]/n,$$

where the second equality is due to the fact that the samples are *independent*, and the third equality is a result of the samples being *identically distributed* (that is, $\mathrm{var}[x_i] = \mathrm{var}[X]$).

□