Markov Decision Processes

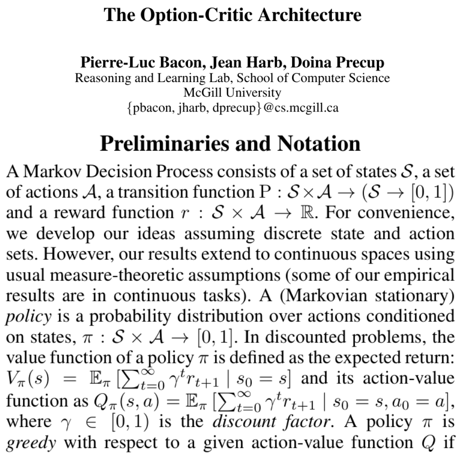
Riccardo Brioschi

21 February 2024

Foundations of Reinforcement Learning

Introduction

Many RL papers contain a background section like the following one:



The Option-Critic Architecture

Pierre-Luc Bacon, Jean Harb, Doina Precup
Reasoning and Learning Lab, School of Computer Science
McGill University
{pbacon, jharb, dprecup}@cs.mcgill.ca

Preliminaries and Notation

A Markov Decision Process consists of a set of states $\mathcal{S}$, a set of actions $\mathcal{A}$, a transition function $\mathrm{P} : \mathcal{S} \times \mathcal{A} \to (\mathcal{S} \to [0, 1])$ and a reward function $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$. For convenience, we develop our ideas assuming discrete state and action sets. However, our results extend to continuous spaces using usual measure-theoretic assumptions (some of our empirical results are in continuous tasks). A (Markovian stationary) *policy* is a probability distribution over actions conditioned on states, $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$. In discounted problems, the value function of a policy π is defined as the expected return: $V_\pi(s) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_0 = s\right]$ and its action-value function as $Q_\pi(s, a) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \mid s_0 = s, a_0 = a\right]$, where $\gamma \in [0, 1)$ is the *discount factor*. A policy π is *greedy* with respect to a given action-value function Q if

In this lecture you will learn

1. What a Markov Decision Process is.
2. How MDPs can be solved with dynamic programming.
3. How future discounted MDPs can be solved with value iteration or policy iteration.

Recommended reading:
Sutton & Barto, Chapters 3 & 4

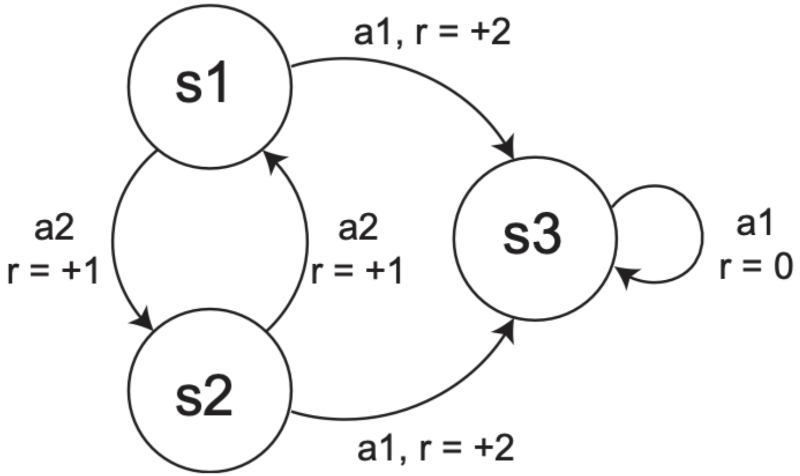
Markov Decision Processes

We define as a **Markov Decision Process (MDP)** a stochastic process characterized by:

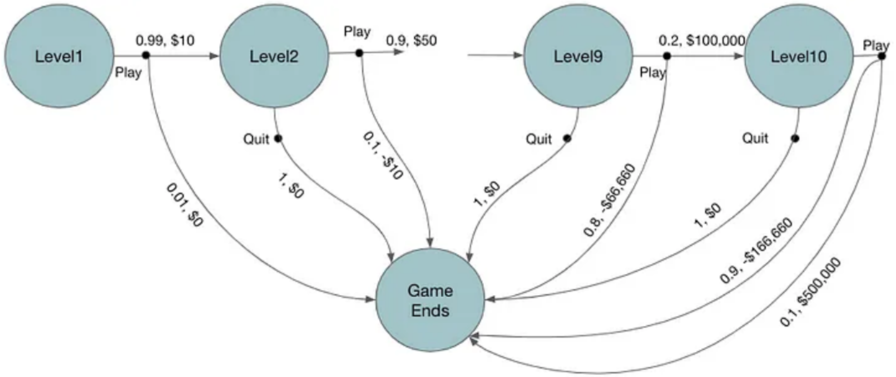
- finite **state space** $\mathcal{S}$ with $|\mathcal{S}| < \infty$,
- finite **action spaces** $\{\mathcal{A}_s | s \in \mathcal{S}\}$ with $|\mathcal{A}_s| < \infty$,
- **transition probabilities** $p^a_{s_i \to s_j} \in [0, 1]$ (alternatively, $P(s_j | s_i, a)$),
- **immediate rewards** $R^a_{s_i \to s_j}$ and $r^a_{s_i} = \sum_{s_j} p^a_{s_i \to s_j} \mathbb{E}[R^a_{s_i \to s_j}] \in \mathbb{R}$,
- **discount factor** $\gamma \in [0, 1]$,
- and **initial state probabilities** $p^{(0)}_{s_i}$.

For a sequence (or trajectory) of state-action-reward tuples, we will use the notation $\tau = (S_0, A_0, R_1, S_1, A_1, \ldots, R_T)$ where $R_i = R^{A_{i-1}}_{S_{i-1} \to S_i}$.

MDPs: An Example



MDPs: An Example



Policies

A policy π selects an action based on the previous history $\tau = (S_{0:T}, A_{0:T})$

- In this course, we only consider **Markov Policies**, i.e. dependent only on the current state
- It is a mapping $\pi : \mathcal{S} \to A$ or $\pi : \mathcal{S} \to \Delta(A)$, $\Delta(\mathcal{A})$ denotes the simplex over A

Families of policies

Deterministic Policy

- Stationary policy $\pi : \mathcal{S} \to \mathcal{A}$, $a_t = \pi(s_t)$
- Markov policy $\pi_t : \mathcal{S} \to \mathcal{A}$, $a_t = \pi_t(s_t)$

Randomized Policy:

- Stationary policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, $a_t \sim \pi(\cdot|s_t)$
- Markov policy $\pi_t : \mathcal{S} \to \Delta(\mathcal{A})$, $a_t \sim \pi_t(\cdot|s_t)$

Returns

Acting on a MDP results in immediate rewards $R^{a}_{s_i \to s_j}$. Accumulating these rewards, we obtain the return.

- **Finite time horizon T**: $\mathbb{E}_{\pi}\left[\sum_{t=1}^{T} \gamma^{(t-1)} R_t\right]$
- **Discounted Reward**: $J(\pi) = \mathbb{E}_{\pi}\left[\sum_{t=1}^{\infty} \gamma^{(t-1)} R_t\right]$

These quantities are obtained averaging the returns over the trajectories obtained moving in the MDP, according to the policy and the transition probabilities of the environment.

Q-Values and V-Values

- It is useful to introduce **Q-values** and **V-values**

$$Q_\gamma^{(T)}(\pi, s, a) = \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{(t-1)} R_t \middle| S_0 = s, A_0 = a\right]$$

$$V_\gamma^{(T)}(\pi, s) = \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{(t-1)} R_t \middle| S_0 = s\right]$$

- These two quantities are strongly related, indeed:

$$V_\gamma^{(T)}(\pi, s) = \sum_{a \in \mathcal{A}_s} \pi(a|s) Q_\gamma^{(T)}(\pi, s, a)$$

- These quantities can also be generalized to the infinite horizon setting

Value Functions and Objectives

The goal is to find a **policy** $\pi^{(t)}(a|s) \in [0,1]$ that maximizes some objective. We define the horizon-T **value function (V-value)**

$$\begin{aligned} V_\gamma^{(T)}(\pi, s) &= \mathrm{E}_\pi \left[\sum_{t=1}^{T} \gamma^{(t-1)} R_t \middle| S_0 = s \right] \\ &= \sum_a \pi^{(T)}(a|s) \left\{ r_s^a + \gamma \sum_{s'} p_{s \to s'}^a V_\gamma^{(T-1)}(\pi, s') \right\} \end{aligned}$$

- **Observation**: The last expression is obtained by recursion, noticing that part of the sum can be expressed as the value function in the state s';
- **Goal**: Find a time-dependent policy π^* maximizing the value function in every state.

The Optimal Policy

Theorem (Bellman's Theorem)

A policy π^* is optimal iff it is greedy with respect to its own value function. In other words, π^* is optimal iff

$$\pi^*(s) = \arg\max_a Q_\gamma(\pi^*, s, a)$$

- This result holds both in the Finite and Infinite Horizon setting;
- **Note**: It avoids optimizing w.r.t. all possible policies;
- **Note**: This policy also maximizes V/Q values in each state.

The Optimal Fixed Horizon Policy

The policy π^* can be found with **Dynamic Programming**.

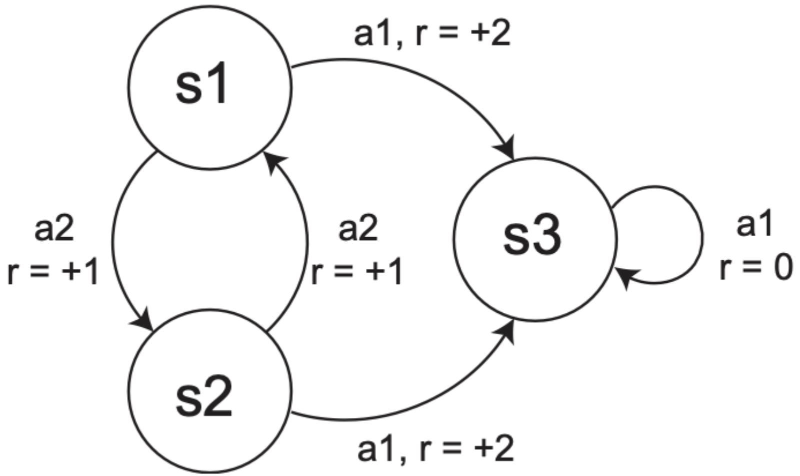
1. The optimal horizon-1 values are $V_\gamma^{(1)}(\pi^*, s) = \max_{a \in \mathcal{A}_s} r_s^a$.
2. The optimal horizon-$(t+1)$ values are

$$V_\gamma^{(t+1)}(\pi^*, s) = \max_{a \in \mathcal{A}_s} Q_\gamma^{(t+1)}(\pi^*, s, a) = \max_{a \in \mathcal{A}_s} \left(r_s^a + \gamma \sum_{s' \in \mathcal{S}} p_{s \to s'}^a V_\gamma^{(t)}(\pi^*, s') \right)$$

This result is based on **Bellman's Principle of Optimality**.

The horizon-T policy is not stationary, in general, i.e. $\pi^{(t)}(a|s) \neq \pi^{(t')}(a|s)$ for $t \neq t'$, but it can be chosen to be deterministic. Any idea why?

Optimal Fixed Horizon Policy: An Example



The Optimal Infinite Horizon Policy

We are now looking for a policy π which maximizes

$$V_\gamma^{(\infty)}(\pi, s) = \mathbb{E}_\pi\left[\sum_{t=1}^{\infty} \gamma^{(t-1)} R_t \middle| S_0 = s\right], \forall s \in S$$

- The optimal policy is now stationary
- As done before, we look for a deterministic policy, for which

$$V_\gamma^{(\infty)}(\pi, s) = \max_{a \in \mathcal{A}_s}\left(r_s^a + \gamma \sum_{s' \in \mathcal{S}} p_{s \to s'}^a V_\gamma^{(\infty)}(\pi, s')\right) \tag{1}$$

Fixed-Point Iterations and Banach's Fixed Point Theorem

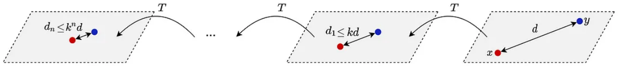
Some equations of the form $x = T(x)$ can be solved with a fixed point iteration:

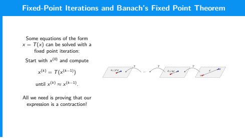
Start with $x^{(0)}$ and compute

$$x^{(k)} = T(x^{(k-1)})$$

until $x^{(k)} \approx x^{(k-1)}$.

All we need is proving that our expression is a contraction!




Fixed-Point Iterations and Banach's Fixed Point Theorem

Some equations of the form $x = T(x)$ can be solved with a fixed point iteration:

Start with $x^{(0)}$ and compute

$$x^{(k)} = T(x^{(k-1)})$$

until $x^{(k)} \approx x^{(k-1)}$.

All we need is proving that our expression is a contraction!

Notes

Let (X, d) be a complete metric space. Then a map $T : X \to X$ is called a **contraction mapping** on X if there exists $q \in [0, 1)$ such that $d(T(x), T(y)) \leq qd(x, y)$ for all $x, y \in X$.

Banach Fixed Point Theorem. Let (X, d) be a non-empty complete metric space with a contraction mapping $T : X \to X$. Then T admits a unique fixed-point x^* in X (i.e. $T(x^*) = x^*$). Furthermore, x^* can be found as follows: start with an arbitrary element $x_0 \in X$ and define a sequence $(x_n)_{n\in\mathbb{N}}$ by $x_n = T(x_{n-1})$ for $n \geq 1$. Then $\lim_{n\to\infty} x_n = x^*$.

Maximizing Future Discounted Values with Dynamic Programming

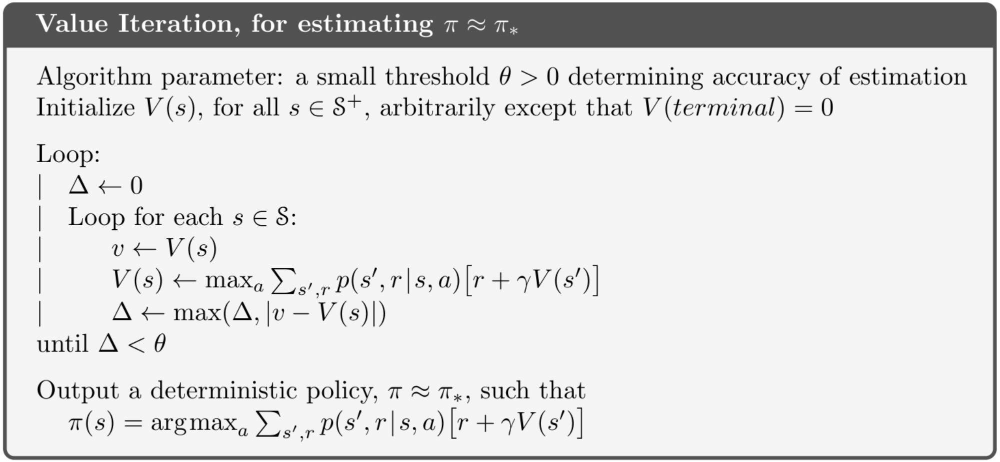
Let us define the mapping (sometimes called **Bellman operator**)

$$T_\gamma : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}, T_\gamma(X)_s = \max_{a \in \mathcal{A}_s} \left(r_s^a + \gamma \sum_{s' \in \mathcal{S}} p_{s \to s'}^a X_{s'} \right). \tag{2}$$

- One can show that the mapping T_γ is a contraction mapping and Banach's fixed point theorem can be applied. Hence, there is a unique fixed point $X^* = T_\gamma(X^*)$.
- The optimal policy is to choose actions in $\arg\max_{a \in \mathcal{A}_s} Q_\gamma^\infty(\pi^*, s, a)$.
- This policy is **stationary** and it can be chosen to be **deterministic**.

Value Iteration

Iteratively compute horizon-t values until $\max_{s\in\mathcal{S}} |V_\gamma^{(t+1)}(\pi^*, s) - V_\gamma^{(t)}(\pi^*, s)| < \theta$, where $\theta > 0$ is some convergence criterion. The optimal stationary policy picks actions in $\arg\max_{a\in\mathcal{A}_s} Q_\gamma^{t^*}(\pi^*, s, a)$, where t^* is the stopping iteration.



Value Iteration, for estimating $\pi \approx \pi_*$

```
Algorithm parameter: a small threshold $\theta > 0$ determining accuracy of estimation
Initialize $V(s)$, for all $s \in \mathcal{S}^+$, arbitrarily except that $V(terminal) = 0$

Loop:
|   $\Delta \leftarrow 0$
|   Loop for each $s \in \mathcal{S}$:
|       $v \leftarrow V(s)$
|       $V(s) \leftarrow \max_a \sum_{s',r} p(s',r|s,a)\big[r + \gamma V(s')\big]$
|       $\Delta \leftarrow \max(\Delta, |v - V(s)|)$
until $\Delta < \theta$

Output a deterministic policy, $\pi \approx \pi_*$, such that
    $\pi(s) = \arg\max_a \sum_{s',r} p(s',r|s,a)\big[r + \gamma V(s')\big]$
```

What is the relationship to Reinforcement Learning?

"Solving" an MDP amounts to a solving an optimal control problem, i.e. finding the optimal policy, where the dynamics is known, i.e. $p^a_{s_i \to s_j}$ and r^a_s are assumed to be known. On the contrary, as we will see, in Reinforcement Learning, one assumes that the dynamics and rewards are unknown.

References

- https://towardsdatascience.com/real-world-applications-of-markov-decision-process-mdp-a39685546026
- Reinforcement Learning: An Introduction, Sutton and Barto
- Reinforcement Learning course (CS-456), EPFL