Indian Institute of Technology, Kanpur
Introduction to Reinforcercment Learning

Instructors: David Silver
Authors: Gurpreet Singh
Email: guggu@iitk.ac.in

SCRIBE

2

Markov Decision Processes

1. Introduction

Markov Decision Processes (MDPs) formally describe an environment in a reinforcement learning problem. We assume the environment to be fully observable *i.e.* the current state completely characterizes the environement. Although most RL problems are fully observable, it is also possible to convert partially observable problems to MDPs. A special case of MDPs with one state are bandits.

2. Markov Process

A *Markov Process* is a memoryless random process *i.e.* a sequence of random states $S_1, S_2 \dots$ which satisfy the Markov Property. The markov property is defined as follows.

Definition 2.1. A state S_t is Markov if and only if

$$\mathbb{P}\left[S_{t+1} \,\middle|\, S_1, S_2 \dots S_t \right] \quad = \quad \mathbb{P}\left[S_{t+1} \,\middle|\, S_t \right]$$

Another component of the markov process is the *state transition probability matrix*. This is the transition matrix which defines the probability of transition from one state to another. Suppose if we represent this transition matrix using $\mathcal{P}$, then $\mathcal{P}_{S,\overline{S}} = \mathbb{P}\left[S_{t+1} = \overline{S} \,\middle|\, S_t = S \right]$.

We can now formally define a Markov Process as follows

Definition 2.2. A *Markov Process* (or *Markov Chain*) is a tuple $(\mathcal{S}, \mathcal{P})$ where $\mathcal{S}$ is a (finite) set of states and $\mathcal{P}$ is the state transition probability matrix.

3. Markov Reward Processes

We extend this definition to Markov Reward Processes (MRPs). This is essentially a Markov Chain with additional values defined with each state.

Definition 2.3. A *Markov Reward Process* is a tuple $(\mathcal{S}, \mathcal{P}, \mathcal{R}, \gamma)$ where $\mathcal{S}$ and $\mathcal{P}$ are the same as in a Markov Process, $\mathcal{R}(s) = \mathbb{E}\left[R_{t+1} \,\middle|\, S_t = s \right]$ is the *reward function*, and $\gamma \in (0, 1)$ is the discount factor.

The total reward is then defined as the sum of rewards obtained at all data points. We define another quantity known as the discounted reward from time-step t.

Definition 2.4. The *return* G_t is the *total discounted reward* from time-step t, where G_t is defined as follows

$$G_t \quad = \quad R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots$$

The goal of an RL problem is to maximize the *return*.

Why Discount? One reason to use the discount factor is to model the higher uncertainty of inference in the farther future. The simpler and the main reason is that using discount rewards is mathematicallo more convenient. This also avoids infinite returns in cyclic Markov Processes. Moreover, animals and humans show more preference for immediate reward, therefore using discount rewards is cognitively justified.

Definition 2.5. The *state value function* $v(s)$ of an MRP is the the expected return starting from state s.

$$v(s) \quad = \quad \mathbb{E}\left[G_t \,\middle|\, S_t = s \right]$$

Note. The return G_t is a random variable, and the value function is a scalar as defined in definition 2.5

3.1. The Bellman Equation

We can write the value function using dynamic programming by rewriting the equation given in definition 2.5 as follows

$$\begin{aligned} v(s) &= \mathbb{E}\left[G_t \,\middle|\, S_t = s \right] \\ &= \mathop{\mathbb{E}}_{S_{t+1}, S_{t+2}, \dots} \left[R_{t+1} + \gamma \left(R_{t+2} + \gamma R_{t+3} + \dots \right) \,\middle|\, S_t = s \right] \\ &= \mathop{\mathbb{E}}_{S_{t+1}} \left[R_{t+1} + \gamma v(S_{t+1}) \,\middle|\, S_t = s \right] \end{aligned}$$

We can represent this more concisely in matrix form. Consider $\mathbf{v} = [\, v(1), v(2) \dots v(n) \,]^{\mathrm{T}}$, $\boldsymbol{\mathcal{R}} = [\, \mathcal{R}(1), \mathcal{R}(2) \dots \mathcal{R}(n) \,]^{\mathrm{T}}$ and $\mathcal{P}$ is the transition matrix.

$$\mathbf{v} \quad = \quad \boldsymbol{\mathcal{R}} + \gamma \mathcal{P} \mathbf{v}$$

Therefore, we can solve this as

$$\mathbf{v} \quad = \quad (1 - \gamma \mathcal{P})^{-1} \boldsymbol{\mathcal{R}}$$

This is not always practically possible since the inversion of a matrix is $\mathcal{O}\left(n^3\right)$ where n is the number of states. In order to tackle this, we use three different methods

- Dynamic Programming
- Monte-Carlo Evaluation
- Temporal-Difference Learning

4. Markov Decision Process

We extend the MRP to include actions or decisions. It is an environment in which all states follow the Markov property. In this case, we say that the transition probability matrix depends on the action taken.

Definition 2.6 (Markov Decision Process)**.** A *Markov Decision Process* is a tuple $(\mathcal{S}, \mathcal{P}, \mathcal{A}, \mathcal{R}, \gamma)$ where $\mathcal{A}$ is a finite set of actions and the transition matrix is given as $\mathcal{P}^a_{s,} = \mathbb{P}\left[S_{t+1} = \middle| S_t = s, A_t = a \right]$

The behaviour of agents is decided by policies. These are formally defined as below.

Definition 2.7 (Policy)**.** A *policy* π is a distribution over actions given states,

$$\pi(a \,|\, s) \quad = \quad \mathbb{P}\left[A_t = a \,|\, S_t = s \right]$$

We do not use deterministic policies instead of stochastic policies to allow exploration. Also, note that the policy depends only on the current state and not the time-step.

<++>