1. Introductions

PDF Version

Introduction

Hello everyone and welcome to CMPUT 605: Theoretical Foundations of Reinforcement Learning at the University of Alberta. We are very excited to be teaching this course and hope that you are excited to journey with us through reinforcement learning theory. The course will cover two sub-topics of RL theory: (1) Planning, and (2) Online RL:

- Planning refers to the problem of computing plans, or policies, or just action by interacting with some model.
- Online RL refers to the problem of coming up with actions that maximize total reward while interacting with an environment.

In all of these subproblems, we will use Markov Decision Processes, to describe how either the simulation models, or the environments work. Thus, we start by introducing the formal definition of a Markov Decision Process (MDP).

Markov Decision Process

A Markov Decision Process is a mathematical model for modelling sequential decision making in an environment that undergoes stochastic transitions. An MDP consists of the following elements: states, actions, rules of stochastic transitions between states, rewards, and an objective, which we take for now to be the discounted total expected reward, or return.

States are considered to be primitive thus we do not explicitly define what they are. The set of states will be denoted by $\mathcal{S}$. Actions are also primitive and their set is denoted by $\mathcal{A}$. For simplicity, we assume that both sets are finite. We will let the number of states be denoted by S, and similarly, we let the number of actions be denoted by A. Stochastic transitions between states s and s' are the result of choosing some action a in a given state. For a fixed state s and action a, the probabilities of landing in the various states s' is collected into a probability vector, which is denoted by $P_a(s)$. To minimize clutter, by slightly abusing notation, we will write $P_a(s, s')$ as the $s' \in \mathcal{S}$ component of this probability vector. This is the probability that the process will transition into state s', when in state s it takes action a.

Rewards are scalars and the reward incurred as a result of taking action a in state s is denoted by $r_a(s)$. Since the number of states and actions are finite, there is no loss in generality by assuming that all the rewards belong to the $[0,1]$ interval. Taking action A_t at time step t gives rise to an infinitely long trajectory of state-action pairs $S_0, A_0, S_1, A_1, \ldots$: here, S_{t+1} is the state that results from taking action A_t in time step $t \geq 0$ and the assumption is that as long as A_t is chosen based on the "past" only, the distribution of S_{t+1} given $S_0, A_0, \ldots, S_t, A_t$ is solely determined by $P_{A_t}(S_t)$, and, in particular, $\mathbb{P}$-almost surely,

$$\mathbb{P}(S_{t+1} = s | S_0, A_0, \ldots, S_t, A_t) = P_{A_t}(S_t, s)\,. \tag{1}$$

The objective is to find a way of choosing the actions that result in the largest possible return along the trajectories that arise. The return along a trajectory is defined as

$$R = r_{A_0}(S_0) + \gamma r_{A_1}(S_1) + \gamma^2 r_{A_2}(S_2) + \cdots + \gamma^t r_{A_t}(S_t) + \ldots$$

where $\gamma \in [0,1)$ is the discount factor. Formally, a (discounted) MDP will thus be described by the 5-tuple $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $P = (P_a(s))_{s,a}$ and $r = (r_a(s))_{s,a}$ collect the transitions and the rewards, respectively.

On Discounting

Note that γ makes it so that the future reward does not matter as much as the present reward. Also, if we truncate the above sum after $H \geq 0$ terms, by our assumption on the rewards, the difference between the return and the truncated return is between zero and

$$\gamma^H \left[r_{A_H}(S_H) + \gamma r_{A_{H+1}}(S_{H+1}) + \ldots \right] \leq \gamma^H \sum_{s \geq 0} \gamma^s = \frac{\gamma^H}{1-\gamma}$$

by using the summation rule for geometric series. Solving for the largest H under which the above upper bound on the difference is below ε, we get that this bound on the difference holds as long as H satisfies

$$H \geq \underbrace{\frac{\ln\left(\frac{1}{\varepsilon(1-\gamma)}\right)}{\ln(1/\gamma)}}_{H^*_{\gamma,\varepsilon}}.$$

For H satisfying this, the return is maximized already when considering only the first H time steps.

Notice that the critical value of H depends on not only ε but also γ. For a fixed ε, this critical value is called the *effective horizon*.

Oftentimes, for the sake of simplicity, we replace $H^*_{\gamma,\varepsilon}$ with the following quantity:

$$H_{\gamma,\varepsilon} := \frac{\ln\left(\frac{1}{\varepsilon(1-\gamma)}\right)}{1-\gamma}\,.$$

(In fact, the literature often calls the latter the effective horizon). This quantity is an upper bound on $H^*_{\gamma,\varepsilon}$. Furthermore, it is not hard to verify that the relative difference between these two quantities is of order $o(1-\gamma)$ as $\gamma \to 1$. Thus, $H^*_{\gamma,\varepsilon}$ behaves the same as $H_{\gamma,\varepsilon}$ up to a first-order approximation as $\gamma \to 1$. Since we are typically interested in this regime (large horizons), there is no loss in switching from $H^*_{\gamma,\varepsilon}$ to $H_{\gamma,\varepsilon}$.

The discounted setting may occasionally feel a bit cringey. Where is the discount factor coming from? One approach is to think about how many time steps in the future we think the optimization should look into for some level of desired accuracy and then work backwards to set γ so that the resulting effective horizon matches our expectation. However, it is more honest to admit that the discounted objective may not faithfully capture the nature of a decision problem. Indeed, there are other objectives that one can consider, such as the finite horizon, undiscounted (or discounted) setting, the infinite horizon setting with no discounting ("total reward"), or the infinite horizon with the average reward. All these have their own pros and cons and we will consider some of these objectives and their relationships in future lectures. For now, we will stick to the discounted objective for pedagogical reasons: the math underlying the discounted objective is simple and elegant. Also, many results transfer to the other settings mentioned, perhaps with some extra conditions, or a little change.

Policies

A policy is a rule that describes how the actions should be taken in light of the past. Here, the past at time step $t \geq 0$ is defined as

$$H_t = (S_0, A_0, \ldots, S_{t-1}, A_{t-1}, S_t)$$

which is the sequence of state-action pairs leading up to the state of the process at the current time step t. We allow policies to randomize. As such, formally, a policy becomes an infinite sequence $\pi = (\pi_t)_{t\geq 0}$ of maps of histories to distributions over actions. For a (finite) set X let $\mathcal{M}_1(X)$ denote the set of probability distributions over X. These probability distributions are uniquely determined by what probability they assign to the individual elements of X. Hence, they will be identified with the probability vectors with $|X|$ components, each component giving the probability of some $x \in X$. If $p \in \mathcal{M}_1(X)$, we use both p_x and $p(x)$ to denote this probability (whichever is more convenient).

With this, we can write that the tth "rule" in π, which will be used in the tth time step to come up with the action for that time step, as

$$\pi_t : \mathcal{H}_t \to \mathcal{M}_1(\mathcal{A})\,,$$

where

$$\mathcal{H}_t = (\mathcal{S} \times \mathcal{A})^{t-1} \times \mathcal{S}\,.$$

Note that $\mathcal{H}_0 = \mathcal{S}$. Intuitively, following a policy π means that in time step $t \geq 0$, the distribution of the action A_t to be chosen for that timestep is $\pi_t(H_t)$: the probability that $A_t = a$ is $\pi_t(H_t)(a)$. Since writing $\pi_t(H_t)(a)$ is quite cumbersome, we abuse notation and will write $\pi_t(a|H_t)$ instead. Thus, when following a policy π, in time step $t \geq 0$ we get that, $\mathbb{P}$-almost surely,

$$\mathbb{P}(A_t = a|H_t) = \pi_t(a|H_t)\,. \tag{2}$$

Initial State Distributions, Distributions over Trajectories

When a policy is *interconnected* with an MDP, the interconnection, together with an initial distribution $\mu \in \mathcal{M}_1(\mathcal{S})$ over the states, uniquely determines a distribution over the infinite-long trajectories

$$T = (\mathcal{S} \times \mathcal{A})^{\mathbb{N}}$$

such that for every time step $t \geq 0$, both (1) and (2) hold, in addition to that

$$\mathbb{P}(S_0 = s) = \mu(s)\,, \qquad s \in \mathcal{S}\,. \tag{3}$$

In fact, this distribution could be over some potentially bigger probability space, in which case uniqueness does not hold. When we want to be specific and take the distribution that is defined over the infinite-long state-action trajectories, we will say that this is the distribution over the *canonical probability space* induced by the interconnection of the policy and the MDP.

To emphasize the dependence of the probability distribution $\mathbb{P}$ on μ and π, we will often use $\mathbb{P}_\mu^\pi$, but we will also take the liberty to drop any of these indices when its identity can be uniquely deduced from the context. When needed, the expectation operator corresponding to $\mathbb{P}$ (or $\mathbb{P}_\mu^\pi$) will be denoted by $\mathbb{E}$ (respectively, $\mathbb{E}_\mu^\pi$).

What is the probability assigned to a trajectory $\tau = (s_0, a_0, s_1, a_1, \ldots) \in T$? Let $h_t = (s_0, a_0, \ldots, s_{t-1}, a_{t-1}, s_t)$. Recall that $H_t = (S_0, A_0, \ldots, S_{t-1}, A_{t-1}, S_t)$. By a repeated application of the chain rule of probabilities, we get

$$
\begin{aligned}
&\mathbb{P}(H_t = h_t) \\
&\quad = \mathbb{P}(S_0 = s_0, A_0 = a_0, S_1 = s_1, \dots, S_t = s_t) \\
&\quad = \mathbb{P}(S_t = s_t | H_{t-1} = h_{t-1}, A_{t-1} = a_{t-1}) \mathbb{P}(H_{t-1} = h_{t-1}, A_{t-1} = a_{t-1}) \\
&\quad = P_{a_{t-1}}(s_{t-1}, s_t) \mathbb{P}(H_{t-1} = h_{t-1}, A_{t-1} = a_{t-1}) && \text{(by (1))} \\
&\quad = P_{a_{t-1}}(s_{t-1}, s_t) \mathbb{P}(A_{t-1} = a_{t-1} | H_{t-1} = h_{t-1}) \mathbb{P}(H_{t-1} = h_{t-1}) \\
&\quad = P_{a_{t-1}}(s_{t-1}, s_t) \pi_{t-1}(a_{t-1} | h_{t-1}) \mathbb{P}(H_{t-1} = h_{t-1}) && \text{(by (2))} \\
&\quad \vdots \\
&\quad = P_{a_{t-1}}(s_{t-1}, s_t) \pi_{t-1}(a_{t-1} | h_{t-1}) \times \cdots \times P_{a_0}(s_0, s_1) \pi_0(a_0 | s_0) \mathbb{P}(S_0 = s_0) \\
&\quad = P_{a_{t-1}}(s_{t-1}, s_t) \pi_{t-1}(a_{t-1} | h_{t-1}) \times \cdots \times P_{a_0}(s_0, s_1) \pi_0(a_0 | s_0) \mu(s_0) \,. && \text{(by (3))}
\end{aligned}
$$

Collecting the terms,

$$\mathbb{P}(H_t = h_t) = \mu_0(s_0) \left\{ \Pi_{i=0}^{t-1} \pi_i(a_i | h_i) \right\} \left\{ \Pi_{i=0}^{t-1} P_{a_i}(s_i, s_{i+1}) \right\}.$$

Similarly,

$$\mathbb{P}(H_t = h_t, A_t = a_t) = \mu_0(s_0) \left\{ \Pi_{i=0}^{t} \pi_i(a_i | h_i) \right\} \left\{ \Pi_{i=0}^{t-1} P_{a_i}(s_i, s_{i+1}) \right\}.$$

Value Functions, the Optimal Value Function and the Objective

The total expected discounted reward, or the expected return of policy π in MDP M when the initial state is sampled from μ is

$$v^\pi(\mu) = \mathbb{E}_\mu^\pi [R] \,.$$

When $\mu = \delta_s$ where δ_s is the "Dirac" probability distribution that puts a point mass at s, we use $v^\pi(s)$ to denote the resulting value. Since this assigns a value to every state, v^π can be viewed as a function assigning a value to every state in $\mathcal{S}$. This function will be called the *value function* of policy π. When the dependence on the MDP is important, we may add "in MDP M" and denote the dependence by introducing an index: v_M^π.

The best possible value in state $s \in \mathcal{S}$ that can be obtained by optimizing over all possible policies is

$$v^*(s) = \sup_\pi v^\pi(s) \,.$$

Then, $v^* : \mathcal{S} \to \mathbb{R}$, viewed as a function, is called the *optimal value function*. A policy is *optimal* in state s if $v^\pi(s) = v^*(s)$. A policy is *uniformly optimal* if it is optimal in every state. In what follows, we will drop uniformly as we will usually be interested in finding uniformly optimal policies.

Given an MDP, we are interested in *efficiently computing* an optimal policy.

Planning=Computation

Computing an optimal policy can be seen as a planning problem: the optimal policy answers the question of how to take actions so that the expected return is maximized. This is also an *algorithmic* problem. The *input*, in the simplest case, is a big table (or a number of tables) that describes the transition probabilities and rewards. The interest is to develop algorithms that read in this table and then as *output* should return a description of an optimal policy. At this stage, it may seem unlikely that an efficient algorithm could do this: in the above unrestricted form, policies have an infinite description. As we shall find out soon though, we will be lucky with finite MDPs as in such MDPs one can always find optimal policies that have a short description. Then, the algorithmic question becomes interesting!

As for any algorithmic problem, the main question is how many elementary computational steps are necessary to solve an MDP? As can be suspected, the number of steps will need to scale with the number of states and actions. Indeed, even the size of the input scales with these. If computation indeed needs to scale with the number of state-action pairs, is there still any reason to consider this problem given that the number of states and actions in MDPs that one typically encounters in practical problems is astronomically large, if not infinite? Yes, there are:

- Not all MDPs are in fact large and it may be useful to know what it takes to "solve" a small MDP. Good solvers for "small" MDPs may serve as benchmarks for solvers developed for the "large MDP" case.
- Even if a problem is large (or infinite), one may be able to approximate it well with a small MDP. Then, a solver for a small MDP may be useful.
- Some ideas and tools developed for this problem also generalize (perhaps) with some twists to the "large" MDP setting.

At this stage, the reader may be wondering about what is meant by "small" and "large"? As a rough guideline, by "small" we mean problems where the tables describing the MDP (and/or policy) comfortably fit in the memory of whatever computer one has access to. Large is everything else.

Miscellaneous Remarks

Probabilities of infinite long trajectories?

Based on the above calculations, one expects that the probability of a trajectory $\tau = (s_0, a_0, s_1, a_1, \dots)$ that never ends is

$$\begin{aligned}\mathbb{P}(S_0 = s_0, A_0 = a_0, S_1 = s_1, A_1 = a_1, \dots) = \mu(s_0) \times \pi_0(a_0|h_0) \times P_{a_0}(s_0, s_1) \\ \times \pi_1(a_1|h_1) \times P_{a_1}(s_1, s_2) \\ \times \cdots \\ \times \pi_t(a_t|h_t) \times P_{a_t}(s_t, s_{t+1}) \\ \times \cdots\end{aligned}$$

where $h_t = (s_0, a_0, \dots, s_{t-1}, a_{t-1}, s_t)$ as before. However, this does not work: if in the trajectory, each action is taken with probability $1/2$ by the policy on the given history, the infinite product on the right-hand side is zero! This should make one pause at least for a moment: how is then $\mathbb{P}$ even *defined*? Does this distribution even exist? If yes, and it assigns zero probability to trajectories like above, could not it be that it assigns zero to all the trajectories of infinite length? In the world of infinite, one must tread carefully! The way out of this conundrum is that we must use *measure theoretic* probabilities, or we need to give up on objects like the return, $R = \sum_{t \geq 0} \gamma^t r_{A_t}(S_t)$, which is defined on trajectories of infinite length. The alternative to measure theoretical probability is to define everything through by taking limits (and always taking expectations over finite-length prefixes of the infinite long trajectories). As this would be quite cumbersome, we will take the measure-theoretic route, which will be explained in the next lecture.

Why Markov?

Equation (1) tells us that the only thing that matters from the history of the process as far as the prediction of the next state is concerned is the last action and the last state. This is known as the Markov property. More generally, Markov chains, which are specific stochastic processes, have a similar property.

Bellman's curse of dimensionality

Richard Bellman, who has made many foundational contributions to the early theory, coined the term the "curse of dimensionality". By this, Bellman meant the following: oftentimes when MDPs are used to model a practical decision making problem, the state space oftentimes takes the product form $\mathcal{S} = \mathcal{S}_1 \times \cdots \times \mathcal{S}_d$ with some $d > 0$. If each set $\mathcal{S}_i$ here has at only two(!) elements, the state space will have at least 2^d elements. This is an exponential growth as a function of d, which is taken as the fundamental scaling quantity. Thus, any algorithm that needs to even just *enumerate* the states in the state space is "cursed" to perform a very lengthy calculation. While we start with considering the case when both the state and the action space are small (as described above), the main focus will be on the case when this is not true anymore. In this way, the problem will be to figure out ways of *breaking the curse*. But just to make things clear, in the worst-case, there is no cure to this curse, as we shall see it soon in a rigorous fashion. Any cure will come by changing the problem, either by changing the objective, or by changing the inputs available, or both.

Actions shared across states?

We described MDPs as if the same set of actions was available in all the states. This may create the (false) impression that action a_1 in state s_1 has something to do with action a_1 in state s_2 (i.e., their rewards, or next state distributions are shared or are similar). Given the MDP definition though, clearly, no such assumptions are made.

In a way, a better way of describing an MDP is using a set Z and an equivalence relation over Z, or, equivalently, the partition induced by it over Z. We should think of Z as the set of possible state-action pairs: The equivalence relation over Z then gives which of these share a common state. Alternatively, if z_1 and z_2 are in the same partition, they share a state, which we can identify with the partition. Then, for every $z \in Z$, the MDP would specify a distribution over the parts of the partition (the "next states") and one should specify a reward. While this description is appealing from a mathematical perspective, it is nonstandard and would make it harder to relate everything to the literature. Furthermore, the description chosen, apart from the inconvenience that one need to forcefully remember that actions do not keep their identity across states, is quite intuitive and compact.

A common variation in the literature, which avoids the "sharing issue" is to assume that every state is equipped with a set $\mathcal{A}(s)$ of actions admissible to the state and these sets are disjoint across the states. This description allows the number of actions to be varied across the states. While this has a minor advantage, our notation is simpler and tends not to lose much in comparison to these more sophisticated alternatives.

Are states observed?

In many practical problems it is not a priori clear whether the problem has a good approximate description as an MDP. One critical aspect that is missing from the MDP description is that the states of the MDP may not be available for measurement and thus the control (the choice of the action) cannot use state information. For now, we push this problem aside, but we shall return to it time-to-time. The reason is that it is best to start with the simpler questions and, at least intuitively, the problem of finding a policy that can use state information feels easier than finding one that cannot even access the state information. First, at least, we should find out what can be done in this case (and how efficiently), hoping that the more complex cases will either be reducible to this case, or will share some common patterns.

On the notation

Why use $r_a(s)$ rather than, say, $r(s,a)$? Or $P_a(s)$, or $P_a(s,s')$ rather than $P(s'|s,a)$? All these notations have pros and cons. None of them is ideal for all purposes. One explanation for using this notation is that later we will replace a with π, where π will be a special policy (a memoryless, or stationary Markov policy). When doing so, the notation of r_π (suppressing s) and P_π (a stochastic matrix!) will be tremendously useful.

A bigger question is why use s for states and a for actions. Is not the answer in the words? Well, people working in control would disagree. They would prefer to use x for state and u for actions,

and I am told by Maxim Raginsky, that these come from Russian abbreviations, so they make at least as much sense as the notation used here. That is, if one speaks Russian (and if not, why not learn it?). Dimitri Bertsekas likes using i, j etc. for states, which seems fine if one has discrete (countable) state spaces.

Stochastic rewards

Some authors (e.g., this author in some of their papers or even in his book) considers rewards which are stochastic. This may matter when the problem is to learn a good policy, or to find a good plan while interacting with a stochastic simulator. However, when it comes to defining the object of computation, we can safely ignore (well-behaved) stochastic rewards. Here, the well-behaved stochastic rewards are those whose conditional expectation given an arbitrary history up to a state s and an action a taken in that state depends only on (s, a). Which is what we start here from.

References

"The" book about MDPs is:

Puterman, Martin L. 2005. Markov Decision Processes (Discrete Stochastic Dynamic Programming). Wiley-Interscience.