Markov Decision Processes: Bellman Equations and Policy Evaluation

Pierre-Luc Bacon, COMP-767 Reinforcement Learning

January 18, 2018

Returns and Values

In the discounted case, the sum of discounted rewards is called the **return**, a random variable which we write :

$$G_t = R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} ...$$

Here R_t is a random variable denoting the reward accrued at time t. Let's say that the rewards are at most some some constant $R_{\max}$ at all time. What would be the return ?

$$\sum_{t=0}^{\infty} \gamma^t R_{\max} = \frac{R_{\max}}{1-\gamma}$$

since $\sum_{t=0}^{\infty} \gamma^t$ is a geometric series converging to $\frac{1}{1-\gamma}$. Taking $\lim \gamma \to 1$ pushes the denominator to 0 and makes the return unbounded. We will therefore require the discount factor to be strictly smaller than 1: $\gamma \in [0,1)$. Having a discount factor in our definition helps us model our problems more naturally[1], but also plays a practical role of just bounding the returns.

A reward is an immediate quantity while the return is a sum of rewards over an extended period of time. Many of the problems that we will be interested in concern the expectation of the return for a policy in a given MDP. We call the expectation of the return the *value* [2].

In reinforcement learning, the agent might only be able to see this rewards without quite knowing how they were generated in the first place. But when crossing the agent-environment interaction boundary, we discover that the rewards are in fact generated from a *reward function* $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, where $\mathcal{S}$ is a set of states and $\mathcal{A}$ is a set of actions.

The expected return can then be written [3]. as :

$$v_\pi(s) \doteq \mathbb{E}\left[G_0 | S_0 = s\right] = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(S_t, A_t) \,\middle|\, S_0 = s\right]$$

and we use the notation $v_\pi(s)$ to denote the value function for the policy. As a reminder, a randomized (stationary) or stochastic policy π is a distribution over actions given a state. We will write: $\pi(a \mid s)$, meaning $\pi : \mathcal{S} \to \text{Dist}(\mathcal{A})$. If it helps, you can conceptualize π as a

[1] I discussed in class how the discount factor can also be interpreted as a random horizon in an MDP. For more details, see Puterman (1994), 5.3.1.

[2] [...] in a designated state or state-action pair, for a given policy and MDP

[3] Under the random horizon interpretation of the discount factor, we could also write $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(S_t, A_t)\right] = \mathbb{E}\left[\mathbb{E}\left[\sum_{t=0}^{T} r(S_t, A_t)\right]\right]$ where T is now a *stopping time*. In this particular case, T is distributed according to a geometric distribution where $P(T = n) = (1-\gamma)\gamma^{n-1}$. The discount factor then represents the probability of continuation.

large matrix of size $|\mathcal{S}| \times |\mathcal{A}|$ where each row corresponds to a separate distribution over actions (thus summing to 1 across columns). Deterministic policies are also of interest, in which case π is function $\pi : \mathcal{S} \to \mathcal{A}$.

We can write the expectation for the infinite sum of discounted rewards recursively:

$$\begin{aligned}
v_\pi(s) &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(S_t, A_t) \,\middle|\, S_0 = s\right] \\
&= \mathbb{E}\left[r(S_0, A_0) + \sum_{t=1}^{\infty} \gamma^t r(S_t, A_t) \,\middle|\, S_0 = s\right] \\
&= \mathbb{E}\left[r(S_0, A_0) + \sum_{t=0}^{\infty} \gamma^{t+1} r(S_{t+1}, A_{t+1}) \,\middle|\, S_0 = s\right] \\
&= \mathbb{E}\left[r(S_0, A_0) + \gamma \sum_{t=0}^{\infty} \gamma^t r(S_{t+1}, A_{t+1}) \,\middle|\, S_0 = s\right] \\
&= \mathbb{E}\left[r(S_0, A_0) + \gamma v_\pi(S_1) \,|\, S_0 = s\right]
\end{aligned}$$

In this expression, the initial state S_0 is given but we need to marginalize over A_0 and S_1 :

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi\left(a \,|\, s\right) \left(r(s, a) + \gamma \sum_{s'} P\left(s' \,|\, s, a\right) v_\pi(s')\right)$$

The term P in this equation is called the *transition probability matrix* [4] and is conditional distribution over next states given a state and action: $P : \mathcal{S} \times \mathcal{A} \to \text{Dist}(\mathcal{S})$ (therefore $\sum_{s'} P\left(s' \,|\, s, a\right) = 1$).

[4] also called *transition function* or *transition matrix* or *transition kernel* in the continuous case

Markov Decision Process

We have seen all the ingredients for defining a Markov Decision Process (MDP). An finite (state and actions) discounted MDP is a tuple $(\mathcal{S}, \mathcal{A}, r, P, \gamma)$ consisting of a finite set of states, a finite set of actions, a reward function, a transition matrix and a discount factor $\gamma \in [0, 1)$. We assume finite sets for the moment, but we will soon (a month and a half or so) talk about how we can deal with continuous set of states using function approximation and how to handle continuous action spaces with policy gradient methods.

Bellman Equations and Policy Evaluation

The recursive expression that we found above for the value function is in fact a set of $|S|$ linear equations where our *unknown* is v_π:

$$v_\pi(s) = \sum_{a \in \mathcal{A}} \pi\left(a \,|\, s\right) \left(r(s,a) + \gamma \sum_{s'} P\left(s' \,\middle|\, s, a\right) v_\pi(s') \right) \quad . \tag{1}$$

We call these equation**s** the *Bellman equation**s*** in reference to the seminal work of Richard Bellman who invented and coined the term *dynamic programming*.

Since these equations are linear, we can easily solve them by writing them in matrix form. To do so, we define the following quantities:

$$r_\pi(s) \doteq \sum_a \pi\left(a \,|\, s\right) r(s,a) \qquad P_\pi(s,s') \doteq \sum_a \pi\left(a \,|\, s\right) P\left(s' \,\middle|\, s, a\right) \quad ,$$

where $r_\pi \in \mathbb{R}^{|\mathcal{S}|}$ and $P_\pi \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$. We then think of the value function as a vector $v_\pi \in \mathbb{R}^{|\mathcal{S}|}$ which satisfies:

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

This expression is exactly equivalent to (1), but written in matrix form. We then have :

$$\begin{aligned} v_\pi &= r_\pi + \gamma P_\pi v_\pi \\ &\iff \\ v_\pi - \gamma P_\pi v_\pi &= r_\pi \\ (I - \gamma P_\pi) v_\pi &= r_\pi \quad . \end{aligned}$$

We recognize the last line as the familiar form of a good old linear system of equations $Ax = b$. The solution to the Bellman equations is then $v_\pi = (I - \gamma P_\pi)^{-1} r_\pi$. Given a policy and MDP, the problem of finding the value function satifying the Bellman equations is called the *policy evaluation* problem [5].

[5] also called the *prediction* problem in S&B

Convergence and Existence

From our discussion above, we saw that things might go wrong if γ goes to 1 because the return can be upper bounded by $\frac{R_{\max}}{1-\gamma}$.

When translating this discussion to our Bellman equations, this amounts to asking the question : when does the inverse of $I - \gamma P_\pi$ exists ? One way to address this question is with the so-called spectral radius σ of a linear operator : the eigenvalue of maximum magnitude. It can be shown [6] that if $\sigma(\gamma P_\pi) < 1$, then $(I - \gamma P_\pi)^{-1}$ exists

[6] For more details, see Puterman (1994) corollary C.4 in the appendix.

and :

$$(I - \gamma P_\pi)^{-1} = \sum_{t=0}^{\infty} \gamma^t P_\pi^t \ .$$

The series on the RHS is called a Neumann series and is just generalization of the geometric series in the scalar case. Because P_π is a stochastic matrix, $\sum_{s'} P_\pi(s, s') = 1$ for any s and assuming that $\gamma < 1$ we have that:

$$\sigma(\gamma P_\pi) \leq \|\gamma P_\pi\|_\infty < 1 \ .$$

So it's all good, as long as $\gamma \in [0, 1)$!

Matrix powers

To understand the meaning of $(I - \gamma P_\pi)^{-1}$, it's useful to consider its Neumann series:

$$(I - \gamma P_\pi)^{-1} = \sum_{t=0}^{\infty} (\gamma P_\pi)^t$$

In a Markov chain, P^t (the t-th power of P) gives us the distribution over next states t steps into the future. Therefore, if we choose a row s and column s', then $P^t(s, s') = P\left(S_t = s' \mid S_0 = s\right)$: the probability that the system is in state s' t steps into the future if it started from state s.

By taking the sum to infinity in the Neumman series, we are effectively marginalizing over trajectories of all possible length. This quantity is sometimes referred to as the *occupancy measure* over states. But in the discounted case, $\sum_{t=0}^{\infty} (\gamma P_\pi)^t$ does not lead to a distribution over states ! A detail which can be easily forgotten[7] but nevertheless an important one... To see this for ourselves, let's sum over all possible next states:

$$\sum_{s'} \sum_{t=0}^{\infty} \gamma^t P\left(S_t = s' \mid s\right) = \sum_{t=0}^{\infty} \gamma^t \sum_{s'} \cancelto{1}{P\left(S_t = s' \mid s\right)} = \frac{1}{1-\gamma}$$

So we're off by a factor of $\frac{1}{1-\gamma}$ from having a distribution: the rows of $(I - \gamma P_\pi)^{-1}$ are not distributions over next states.

Value Iteration

One way to visualize what the Bellman equations *mean* is using a *backup diagram*: a graphical depiction of how values are propagated when computing a *Bellman backup*.

Equipped with the Bellman equations (and keeping in mind the backup diagram), we can devise an iterative algorithm for policy

[7] We often make this mistake in policy gradient methods and refer to $\sum_{t=0}^{\infty} (\gamma P_\pi)^t$ as a stationary distribution. It's not the case !

Figure 1: Richard Bellman laid out the foundations for the MDP formalism RAND corporation in the 50s. He coined the term *dynamic programming*, which has an interested story that you can read about here `https://www.jstor.org/stable/3088448`

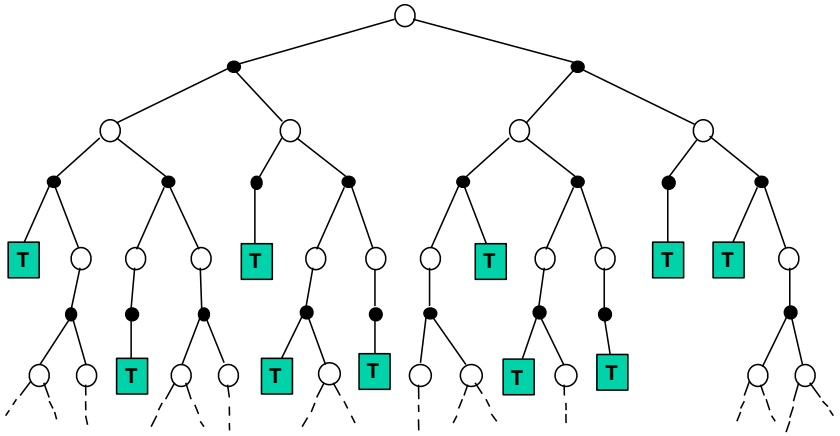

Figure 2: A backup diagram is a depiction of the computation carried out in a Bellman backup/sweep. An open circle means a state and a solid black circle is an action. Square boxes represent terminal states. Figure taken from the S&B 2017 textbook

evaluation called value iteration:

$$v_\pi^{(k+1)}(s) = \sum_a \pi\left(a \,|\, s\right) \left(r(s,a) + \gamma \sum_{s'} P\left(s' \,\middle|\, s, a\right) v_\pi^{(k)}(s') \right) \quad ,$$

or in matrix form:

$$v_\pi^{(k+1)} = r_\pi + \gamma P_\pi v_\pi^{(k)} \quad .$$

This algorithm is guaranteed to converge once again only if $\sigma(\gamma P_\pi)$ (the spectral radius) is strictly smaller than 1.