Tutorial 2: Dynamic Programming

Jacob Green

Abstract

ABSTRACT HERE

1 Bellman's Equation

Recall the following definition from tutorial 1.

Definition 1. *(discrete MDP) A discrete Markov decision process is a tuple* $(\mathcal{S}, \mathcal{A}, \mathbb{T}, \mathbb{T}_0, R, \gamma, N)$ *where*

- $\mathcal{S}$ *and* $\mathcal{A}$ *are sets, named the state and action spaces respectively.*
- $\mathbb{T}(s'|a,s) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ *is a conditional distribution, named the transition distribution, describing the dynamics of state-action updates.*
- $\mathbb{T}(s) : \mathcal{S} \to [0,1]$ *is a distribution, named initial state distribution, describing the dynamics of initial state selection.*
- $R(s'|a,s) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ *is a map defining the reward signalled by observing state* s' *from the state-action pair* (s,a)
- $\gamma \in [0,1)$ *denotes the discount rate*
- $N \in \mathbb{N} \cup \{+\infty\}$ *denotes the, possibly infinite, time horizon.*

If $N < +\infty$ *we will call our discrete Markov decision process finite.*

Let $(\mathcal{S}, \mathcal{A}, \mathbb{T}, \mathbb{T}_0, R, \gamma, N)$ be a discrete MDP with $\mathcal{S}$ and $\mathcal{A}$ *discrete*[1]. We defined a *policy* to be a collection of conditional distributions $\pi(a|s) : \mathcal{A} \times \mathcal{S} \to [0,1]$, an episode to be a tuple of the form

$$\tau = (s_0, a_0, s_1, r_0, a_1, \ldots, s_{N-1}, r_{N-2}, a_{N-1}, s_N, r_{N-1})$$

where $r_t := R(s_{t+1}|a_t, s_t)$ and $a_t \sim \pi(\cdot|s_t)$, and the *expected discounted cumulative reward* for the policy π to be

$$J(\pi) := \mathbb{E}_{\tau \sim \mathbb{T}(\cdot\,;\,\pi)} \left[\sum_{t=0}^{N-1} \gamma^t r_t \right]$$

where $\mathbb{T}(\cdot\,;\,\pi) : \mathcal{T} \to [0,1]$ denotes the episode distribution. The goal of reinforcement learning is to find

$$\pi^\star \operatorname*{argmax}_{\pi} J(\pi)$$

In this tutorial, we'll be exploring learning via *dynamic programming*, an umbella term for algorithms that can compute the optimal policy for a *fully specified* MDP. By fully specified, we mean

[1] i.e. $\mathcal{S}$ and $\mathcal{A}$ finite or countable

that our MDP perfectly models our game. In practice, to perfectly specify a game is either (A) impossible (B) computationally infeasible (e.g. large $\mathcal{S}$ or $\mathcal{A}$). That said, DP (dynamic programming) makes for an ideal theoretical basis. All the more practically used algorithms, e.g. Monte Carlo simulation[2], used to find optimal policies in MDPs can be seen as approximations of their dynamic cousins.

We make the following assumption, for reasons that will become clear shortly.

Assumption 1 (null terminal rewards)**.** *If there exists T such that s_T is a terminal state[3], then for all $t > T$ we have $r_t = 0$.*

We begin by exploring two new quantites: the *state-value function* and the *action-value function*. These are defined, respectively, by

$$V^\pi(s) := \mathbb{E}_{\tau \sim \mathbb{T}(\cdot\,;\,\pi)}\left[\sum_{k=0}^{\infty} \gamma^t r_{t+k+1} \;\middle|\; s_t = s\right]$$

$$Q^\pi(s,a) := \mathbb{E}_{\tau \sim \mathbb{T}(\cdot\,;\,\pi)}\left[\sum_{k=0}^{\infty} \gamma^t r_{t+k+1} \;\middle|\; s_t = s, a_t = a\right]$$

Note that these are just our expected discounted cumulative rewards conditioned on the initial state, and initial state-action pair, at time t respectively. By the Markov property and assumption 1, $V^\pi(\cdot)$ and $Q^\pi(\cdot,\cdot)$ do not depend on t, hence the notational abscence.

Let $V^\star(\cdot) := V^{\pi^\star}(\cdot)$ and $Q^\star(\cdot,\cdot) := Q^{\pi^\star}(\cdot,\cdot)$. It is clear

$$V^\star(s) = \max_\pi V^\pi(s) \quad \text{and} \quad Q^\star(s,a) = \max_\pi Q^\pi(s,a)$$

[2]We'll cover this next tutorial

[3]That is, a state s where our game is considered complete. E.G. in chess, this would be a checkmate position. The key idea is that we need not consider our actions beyond such a point.