# A Tutorial on the Bellman Equations for Simple Reinforcement Learning Problems

Abraham Nunes[1,2]

[1] *Department of Psychiatry, Dalhousie University*
[2] *Hierarchical Anticipatory Learning Lab, Faculty of Computer Science, Dalhousie University*

March 21, 2016

## Markov Decision Process

A Markov decision process (MDP) has the following elements:

- A set of states, $s \in \mathcal{S}$
- A set of actions which depend on the current state, $a \in \mathcal{A}(s)$
- A policy which maps from state to action, $\pi(s) \in \mathcal{A}(s)$
- State transition probabilities, $\mathcal{T}(s'|s, a)$
- A reward function $\mathcal{R}(s'|s, a)$. We can also denote the reward received at the current time step as $r$

The agent operating within such a process will generally accumulate value at each state according to behaviour under some policy. This value function is typically denoted as $V^{\pi}(s)$, but for state-action pairs is typically denoted as $\mathcal{Q}^{\pi}(s, a)$.

The goal of the agent solving the MDP is to find the optimal policy such that total future value is maximized. The value functions under the optimal policy are typically denoted as either $V^*(s)$ or $\mathcal{Q}^*(s, a)$.

Over time, the reward obtained will accumulate. At the $k^{th}$ time step, the reward is $\gamma^k V$, where $0 < \gamma < 1$ represents a discount factor. One can liken this to a measure of impulsivity or impatience, in behavioural terms.

## Understanding the Bellman Equations

If you were to measure the value of the current state you are in, how would you do this? The intuitive way would simply be to tally the value of any rewarding properties obtained at the present instant. This, however, misses an important fact: that the current state partially determines which states one can end up in later. As such, we can expand on the previous point by stating that the value of a current state would be the sum of the reward received in the moment, plus the total value of all future rewards expected as a result. This is closer to the true answer, but we must add one final touch: the future is worth less than the present (on account of the uncertainty of the future), so we must *discount* future rewards when adding them to the current reward. Adding current reward to a discounted total future reward results in what business-folk call the *net present value.*

You might have noticed a problem: we can't know the future, especially *far* into the future. This is where the Bellman equations become interesting due to the property of *recursion.* Recall that the value of the present state represents the net present value of all states in the future. This necessarily means that the value of the next state, $s'$, represents the net present value of all future rewards thereafter. If we replace the prime (') notation with subscripts denoting the time step (i.e. $s_0$ is the initial state, $s_2$ is the state at time step 2, etc.), we can see this recursion in action (Equation 1).

$$V(s_0) = r_0 + \gamma \sum_{s_1 \in \mathcal{S}} \mathcal{T}(s_1|s_0, a_0)\mathcal{R}(s_1|s_0, a_0) + \gamma^2 \sum_{s_2 \in \mathcal{S}} \mathcal{T}(s_2|s_1, a_1)\mathcal{R}(s_2|s_1, a_1) + \cdots \tag{1}$$

We can summarize Equation 1 as follows:

$$\begin{aligned} V(s) &= \sum_{t=0}^{T} \gamma^t \sum_{s_{t+1} \in \mathcal{S}} \mathcal{T}(s_{t+1}|s_t, a_t)\mathcal{R}(s_{t+1}|s_t, a_t) \\ &= \sum_{t=0}^{T} \gamma^t \left\langle \mathcal{R}(s_{t+1}|s_t, a_t) \right\rangle_{\mathcal{T}}, \end{aligned}$$

where the angled brackets $\langle \cdot \rangle_{\mathcal{T}}$ denote an expectation under probability measure $\mathcal{T}$. As such,

$$V(s_t) = r_t + \gamma \sum_{s_{t+1} \in \mathcal{S}} \mathcal{T}(s_{t+1}|s_t, a_t)V(s_{t+1}). \tag{2}$$

Bellman was concerned with finding the *optimal policy*, which in plain language means choosing the best possible action at each given state, where ”best possible action” means the action that maximizes total future reward. The optimal control policy is thus

$$\pi^*(s_t) = \arg\max_{a_t} \left[ \sum_{s_{t+1} \in \mathcal{S}} \mathcal{T}(s_{t+1}|s_t, a_t)V^*(s_{t+1}) \right], \tag{3}$$

and it serves to maximize the value $V^*(s_t)$ at the present state:

$$V^*(s_t) = r_t + \max_{a_t} \left[ \gamma \sum_{s_{t+1} \in \mathcal{S}} \mathcal{T}(s_{t+1}|s_t, a_t)V^*(s_{t+1}) \right] \tag{4}$$

Typically—except for a few exceptions—this must be solved numerically.