

Week 7: Bellman's Equations

Saturday, March 19, 2022 10:41 AM

Trajectory:

→ time function

→ graph

↳ includes $r(t)$ and $a(t)$ (reward and actions as time functions)

The graph of the world \neq the graph of a trajectory

→ if $\gamma = 0$ we are ignoring all future rewards, and we are only considering the current transition's reward.

→ $\gamma < 1$ so that it converges

For deterministic worlds: $r_t + \gamma [f_R(\bar{T})]$

Average Accumulated Reward $V(s) = \overline{f_{RA}(\bar{T})}_{s(0)=s}$ } The average of the acc reward of all trajectories that start in state s

→ Why average?

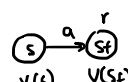
$\overline{f_{RA}(\bar{T})}$

In non-deterministic worlds, we need to calculate the average reward. From this case, we can derive the deterministic case: the average in a deterministic world will always be the same.

Since in non-deterministic worlds, trajectories can be infinite. → how do we calculate $f_{RA}(\bar{T})$

Bellman

MT Deterministic

$V(s) = \overline{f_{RA}(\bar{T})}$  $\left. \begin{aligned} V(s) &= r + V(sf) \\ &\quad \text{reward from that on} \\ &\quad \text{↳ immediate reward} \end{aligned} \right\} V(s) \text{ is now written in terms of } V(sf) \text{ (next state)}$
 $\Rightarrow f_R(s, a, sf) + \gamma V(sf)$

Since it is the same average because it is deterministic

$= f_R(s, a, f_{MT}(s, a)) + \gamma V(f_{MT}(s, a))$
 $f_{MT} = \text{Transition Model function: } sf = f_{MT}(s, a)$

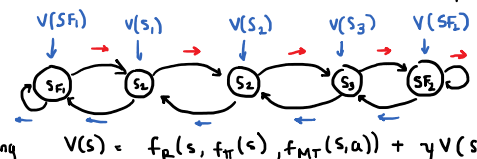
If we consider $a = f_\pi(s)$

$V(s) = f_R(s, f_\pi(s), f_{MT}(s, f_\pi(s))) + \gamma V(f_{MT}(s, f_\pi(s)))$
 Which is now a function only depending sf on s .

We have, in a world with N states, we have N equations in the form:

$V(s)$ = written in $V(sf)$ → Bellman's eq

Thus we have a linear system of equations with $V(sf)$ as the unknown.

Example:  $f_\pi(s) = s_1 \rightarrow s_2 \rightarrow s_3 \rightarrow sf_2$ $f_R(sf) = \begin{bmatrix} sf_1 & -10 \\ s_1 & 0 \\ s_2 & -0.4 \\ s_3 & -0.9 \\ sf_2 & 10 \end{bmatrix}$
 Considering $V(s) = f_R(s, f_\pi(s), f_{MT}(s, a)) + \gamma V(sf)$
 ↳ immediate reward after the transition

(given by politic)

$$V(S_1) = -0.4 + \gamma V(S_2)$$

Now we write $V(S_2)$ as reward function from S_2 on, according to the politic:

$$\begin{aligned} (2) \quad V(S_2) &= -0.4 + \gamma V(S_3) \\ (3) \quad V(S_3) &= 10 + \gamma V(SF_2) \\ (4) \quad (SF_1) \quad V(SF_1) &= -10 + \gamma V(SF_2) \\ (5) \quad V(SF_2) &= 10 + \gamma V(SF_2) \end{aligned} \quad \left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \begin{array}{l} \text{all these are calculated} \\ \text{for getting } f_{AA} \end{array}$$

with $\gamma = 0.9$

Thus we have a linear system of equations (5 eq)

$$\begin{aligned} (5) \quad V(SF_2) &= 10 + \gamma V(SF_2) \\ V(SF_2) - \gamma V(SF_2) &= 10 \\ (1 - \gamma) V(SF_2) &= 10 \\ V(SF_2) &= 10 \\ V(SF_2) &= \frac{10}{1 - \gamma} = \frac{10}{1 - 0.9} = \frac{10}{0.1} = \boxed{100} \end{aligned}$$

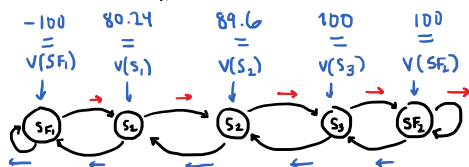
$$\begin{aligned} (4) \quad V(SF_1) &= -10 + \gamma V(SF_2) \\ [1 - \gamma] V(SF_1) &= -10 \\ V(SF_1) &= \frac{-10}{1 - \gamma} = \frac{-10}{1 - 0.9} = \frac{-10}{0.1} = \boxed{-100} \end{aligned}$$

$$\begin{aligned} (3) \quad V(S_3) &= 10 + \gamma V(SF_2) \\ V(S_3) &= 10 + (0.9)(100) \\ V(S_3) &= 10 + 90 \\ \boxed{V(S_3) = 100} &\rightarrow \text{you can use the geometric series to solve this and get the same 100} \end{aligned}$$

$$\begin{aligned} (2) \quad V(S_2) &= -0.4 + \gamma V(S_3) & (1) \quad V(S_1) &= -0.4 + \gamma V(S_2) \\ V(S_2) &= -0.4 + (0.9)(100) & V(S_1) &= -0.4 + (0.9)(89.6) \\ V(S_2) &= -0.4 + 90 & V(S_1) &= \boxed{80.24} \\ \boxed{V(S_2) = 89.6} & & & \end{aligned}$$

exam: April 2nd

Now, we can say,



The politic originally is to begin in S_1 and move until SF_2 , and with the results given, the agent coincidentally has to move to the right (biggest reward)

Now, for the nondeterministic world, where the trajectories can be infinite. Their average value converges, but how to find it?

Bellman's Equation, nondeterministic

$$V(s) = \overline{f_{AA}(T)} \Big|_{s(0)=s} \quad \left. \begin{array}{l} \text{given that} \\ \text{the average } f_{AA} \text{ of all trajectories} \\ \text{that start in state } s \end{array} \right\}$$

Using the definition of the sum this sum is = $\overline{f_{AA} + \gamma V(S_t)}$

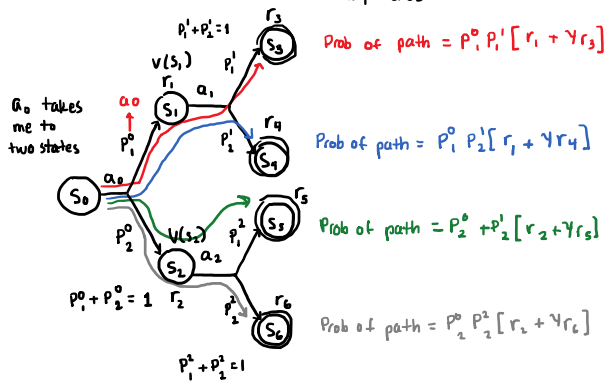
$$= \overline{f_{AA}(s, a, S_t) + \gamma V(S_t)} = \sum_{S_t \in S} p_{NT}(S_t | s, a) [f_{AA}(s, a, S_t) + \gamma V(S_t)]$$

with $a = f_{\pi}(s)$

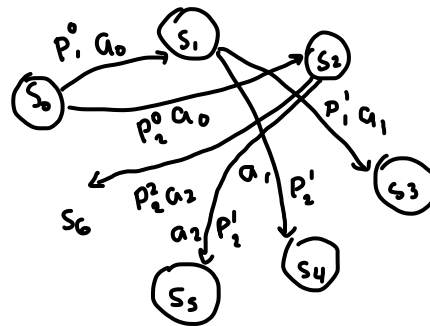
Example

Let's imagine we start in S_0
Assume in final states, it ends
r.

Let's imagine we start in S_0
Assume in final states, it ends



We got
a total of
4 trajectories
to final states



$$V(S_0) = p_1^0 p_1^1 [r_1 + \gamma r_3] + p_1^0 p_2^1 [r_1 + \gamma r_4] + p_2^0 p_2^1 [r_2 + \gamma r_5] + p_2^0 p_2^2 [r_2 + \gamma r_6]$$

Let's write this in Bellman's form:

$$= p_1^0 [p_1^1 [r_1 + \gamma r_3] + p_2^1 [r_1 + \gamma r_4]] + p_2^0 [p_2^1 [r_2 + \gamma r_5] + p_2^2 [r_2 + \gamma r_6]]$$

$$= p_1^0 \left[\underbrace{(p_1^1 + p_2^1)}_1 r_1 + \gamma (p_1^1 r_3 + p_2^1 r_4) \right] + p_2^0 \left[\underbrace{(p_2^1 + p_2^2)}_1 r_2 + \gamma (p_2^1 r_5 + p_2^2 r_6) \right]$$

$$= p_1^0 r_1 + \gamma p_1^0 \underbrace{(p_1^1 r_3 + p_2^1 r_4)}_{V(S_1)} + p_2^0 r_2 + \gamma p_2^0 \underbrace{(p_2^1 r_5 + p_2^2 r_6)}_{V(S_2)}$$

$V(S_1)$ = all reward from S_1 till the end
 $V(S_2)$ = all reward from S_2 till the end

$$= p_1^0 r_1 + \gamma p_1^0 V(S_1) + p_2^0 r_2 + \gamma p_2^0 V(S_2)$$

$$= p_1^0 [r_1 + \gamma V(S_1)] + p_2^0 [r_2 + \gamma V(S_2)]$$

$$= P_{MT}(S_1 | S_0, a_0) [r_1 + \gamma V(S_1)] + P_{MT}(S_2 | S_0, a_0) [r_2 + \gamma V(S_2)]$$

In sum form, we arrive at a Bellman's form,

$$V(S_0) = \sum_{i=1}^2 P_{MT}(S_i | S_0, a_0) [r_i + \gamma V(S_i)]$$

where a 's are given by a policy

We have an equation per state, and this forms a linear system of equations where V is the unknown

Two unknowns,

$a = \pi(s)$ is the policy we want the agent to learn, and thus in real cases we need to solve this through approximations (Fixed Point Iteration)

Keep in mind:
 $S_t \sim P_{MT}(S_t | S_{t-1}, a)$
 $p_i^0 = P_{MT}(S_i | S_0, a_0)$
 $p_i^1 = P_{MT}(S_i | S_1, a_1)$

for this world, we would need 6 eqs at this