Bellman's Equations

Saturday, March 19, 2022 10:41 AM

Trayectory:

→ time function

→ graph

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

The graph of the world ≠ the graph of a trayectory

→ 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: $r_1 + \gamma[f_R(\bar{\tau})]$

Average Accumulated Reward

$V(s) = \overline{f_{RA}(\tau)}_{s(0)=s}$ } The average of the acc reward of all trayectories that start in state s

→ Why average? $\overline{f_{RA}(\tau)}$

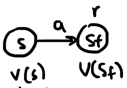
In nondeterministic worlds, we need to calculate the average reward. From this case, we can derive the deterministic case: the average in a deterministic world will always be the same.

Since in non deterministic worlds, trayectories can be infinite. → How do we calculate $f_{RA}(\tau)$

Bellman

M_T Deterministic

$V(s) = \overline{f_{RA}(\tau)}$



$V(s) = r + V(S_f)$ → reward from that on } $V(S)$ is now written in terms of $V(S_f)$ (next state)

↳ immediate reward

$= f_R(s, a, s_f) + \gamma V(S_f)$

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} = Transition Model function: $s_f = f_{MT}(s,a)$

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

$$V(s) = f_R(s, f_\pi(s), f_{MT}(s, f_\pi(s))) + \gamma \overbrace{V(\underbrace{f_{MT}(s, f_\pi(s))}_{s_f})}^{V(s_f)}$$

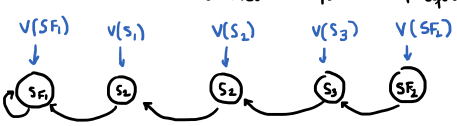
Which is now a function only depending on s.

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

$V(s)$ = writen in $V(S_f)$ → Bellman's eq

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

Example:



$f_\pi(s) =$
SF_1 ←
S_1 →
S_2 →
S_3 →
SF_2 →

Considering $V(s) = \underbrace{f_R(s, f_\pi(s), f_{MT}(s,a))}_{\text{immediate reward after the transition (given by policie)}} + \gamma V(s_f)$

$V(s_1) = -0.4 + \gamma V(s_2)$

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

② $V(S_2) = -0.4 + \gamma V(S_3)$
③ $V(S_3) = 10 + \gamma V(SF_2)$
④ $V(SF_1) = -10 + \gamma V(SF_1)$
⑤ $V(SF_2) = 10 + \gamma V(SF_2)$

} all these are calculated for getting f_{AR}

with $\gamma = 0.9$

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

⑤ $V(SF_2) = 10 + \gamma V(SF_2)$

$V(SF_2) - \gamma V(SF_2) = 10$

$(1-\gamma)\, V(SF_2) = 10$

$V(SF_2) = 10$

$V(SF_2) = \frac{10}{1-\gamma} = \frac{10}{1-0.9} = \frac{10}{0.1} = \boxed{100}$

④ $V(SF_1) = -10 + \gamma V(SF_1)$

$[1-\gamma]\, V(SF_1) = -10$

$V(SF_1) = \frac{-10}{1-\gamma} = \frac{-10}{1-0.9} = \frac{-10}{0.1} = \boxed{100}$

③ $V(S_3) = 10 + \gamma V(SF_2)$

$V(S_3) = 10 + (0.9)(100)$

$V(S_3) = 10 + 90$

$\boxed{V(S_3) = 100}$ ⟶ you can use the geometric series to solve this and get the same 100

② $V(S_2) = -0.4 + \gamma V(S_3)$

$V(S_2) = -0.4 + (0.9)(100)$

$V(S_2) = -0.4 + 90$

$\boxed{V(S_2) = 89.6}$

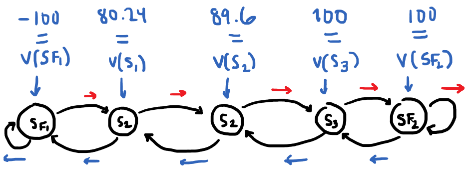
① $V(S_1) = -0.4 + \gamma V(S_2)$

$V(S_1) = -0.4 + (0.9)(89.6)$

$V(S_1) = \boxed{80.24}$

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

$$V(s) = \overline{f_{RA}(\tau)}\Big|_{s(0)=s} \leftarrow \text{given that}$$

} the average f_{AR} of all trajectories that start in start s

Using the definition of the sum

this sum is $= \overline{f_R + \gamma V(s_f)}$

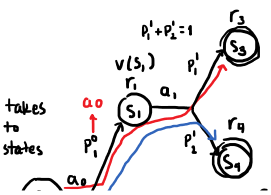
$$= \overline{f_R(s,a,s_f) + \gamma V(s_f)} = \sum_{s_f \in S} P_{MT}(s_f \mid s,a)\left[f_R(s,a,s_f) + \gamma V(s_f)\right]$$

with $a = f_\pi(s)$

Example

Let's imagine we start in S_0

Assume in final states, it ends

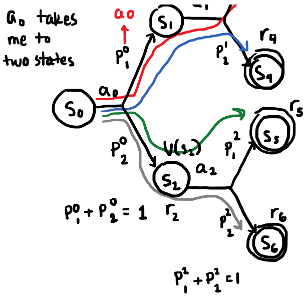


a_0 takes me to two states

Prob of path $= P^0_1 P^1_1 [r_1 + \gamma r_3]$

Prob of path $= P^0_1 P^1_2 [r_1 + \gamma r_4]$

We got a total of



Prob of path $= P_1^0 \, P_2^1 [r_1 + \gamma r_4]$

Prob of path $= P_2^0 + P_2^1 [r_2 + \gamma r_5]$

Prob of path $= P_2^0 \, P_2^2 [r_2 + \gamma r_6]$

We got a total of 4 trajectories to final states

$$
\begin{aligned}
V(S_0) = \; & P_1^0 \, P_1^1 [r_1 + \gamma r_3] + \\
& P_1^0 \, P_2^1 [r_1 + \gamma r_4] + \\
& P_2^0 \, P_2^1 [r_2 + \gamma r_5] + \\
& P_2^0 \, P_2^2 [r_2 + \gamma r_6]
\end{aligned}
$$

Let's write this in Bellman's form:

$$
\begin{aligned}
= \; & P_1^0 \Big[P_1^1 [r_1 + \gamma r_3] + P_2^1 [r_1 + \gamma r_4] \Big] + \\
& P_2^0 \Big[P_2^1 [r_2 + \gamma r_5] + P_2^2 [r_2 + \gamma r_6] \Big]
\end{aligned}
$$

$$
\begin{aligned}
= \; & P_1^0 \Big[\big(\underbrace{P_1^1 + P_2^1}_{1} \big) r_1 + \gamma \big(P_1^1 r_3 + P_2^1 r_4 \big) \Big] + \\
& P_2^0 \Big[\big(\underbrace{P_1^2 + P_2^2}_{1} \big) r_2 + \gamma \big(P_1^2 r_5 + P_2^2 r_6 \big) \Big]
\end{aligned}
$$

$$
\begin{aligned}
= \; & P_1^0 r_1 + \gamma P_1^0 \overbrace{\left(P_1^1 r_3 + P_2^1 r_4 \right)}^{V(S_1) = \text{acc reward from } S_1 \text{ till the end}} + \\
& P_2^0 r_2 + \gamma P_2^0 \underbrace{\left(P_1^2 r_5 + P_2^2 r_6 \right)}_{V(S_2) = \text{acc reward from } S_2 \text{ till the end}}
\end{aligned}
$$

$$
\begin{aligned}
= \; & P_1^0 r_1 + \gamma P_1^0 \, V(S_1) + \\
& P_2^0 r_2 + \gamma P_2^0 \, V(S_2)
\end{aligned}
$$

Keep in mind:

$S_f \sim P_{MT}(S_f | s, a)$

$P_1^0 = P_{MT}(S_1 | S_0, a_0)$

$P_2^0 = P_{MT}(S_2 | S_0, a_0)$

$$
\begin{aligned}
= \; & P_1^0 \left[r_1 + \gamma V(S_1) \right] + \\
& P_2^0 \left[r_2 + \gamma V(S_2) \right]
\end{aligned}
$$

$$
\begin{aligned}
= \; & P_{MT}(S_1 | S_0, a_0) \left[r_1 + \gamma V(S_1) \right] + \\
& P_{MT}(S_2 | S_0, a_0) \left[r_2 + \gamma V(S_2) \right]
\end{aligned}
$$

for this world, we would need 6 eqs as this

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

$$V(S_0) = \sum_{i=1}^{2} P_{MT}(S_i \mid S_0, a_0) \left[r_2 + \gamma (S_2) \right]$$

↳ where a's are given by a politic

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

Two unknowns, V and a

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