Week 10: V(s) and Q(s,a) for Bellman's Optimality

Saturday, April 9, 2022 9:51 AM

Bellman's Optimality Equations are defined in two ways: $V(s)$ and $Q(s,a)$ functions.
Both give us a set of equations for $V(s)$ and Q, respectively. In essence, they're the same.

→ $V(s)$ is a vector (one variable function)
→ $Q(s,a)$ is a matrix (two variable function)

	Q(s,a)		
	a1	a2	a(m=3)
s1	Q(s1,a1)	Q(s1,a2)	Q(s1,a3)
s2	Q(s2,a1)	Q(s2,a2)	Q(s2,a3)
s3	Q(s3,a1)	Q(s3,a2)	Q(s3,a3)
s4	Q(s4,a1)	Q(s4,a2)	Q(s4,a3)
s(n=5)	Q(s5,a1)	Q(s5,a2)	Q(s5,a3)

the max Q of each row, goes to $V(s)$ vector and the $Q(s,a)$ action is the action of $V(s_i)$ ⇒ to know the action is to know Q matrix

$Q(s3,a1)$ is the accumulated reward of all trajectories that start in s3 and execute a1 on that s3 initial state

Iteración de Valor

$$V(s) \leftarrow \max_a \left[\sum_{s_f \in S} P_{M_T}(s_f|s,a)[f_R(s,a,s_f) + \gamma V(s_f)] \right]$$

Desconocidos

$$Q(s,a) \leftarrow \sum_{s_f \in S} P_{M_T}(s_f|s,a)\left[f_R(s,a,s_f) + \gamma \max_{a_f}[Q(s_f,a_f)]\right]$$

Desconocidos

The bellman equations will discover the politic: set of actions.
↳ the optimal politic
and thus $Q(s,a)$ too

→ Since $V(s)$ involves max() function, the solution cannot be analytical: we will use a numerical method, Value Iteration.
For this, both the Transition Model and reward function must be known.

Taking the example world:

→ The world contains the set of states $S = \{s_1, s_2, s_3, sF_1, sF_2\}$ where s_1 = initial state and, sF_1 and sF_2 are final states:

sF_1	s_1	s_2	s_3	sF_2

→ The world has the following set of Actions $A = \{\rightarrow, \leftarrow\}$, where:
- $\rightarrow$ = agent moves to left, one cell
- $\leftarrow$ = agent moves to right, one cell

} deterministic example

→ The reward function $f_R(s,a,s_f) = f_R(s_f)$ only depends on the state that the agent arrives to.

$$f_R(s_f) = \begin{array}{c} sF_1 \\ s_1 \\ s_2 \\ s_3 \\ sF_2 \end{array} \begin{bmatrix} -10 \\ 0 \\ -0.4 \\ -0.4 \\ 10 \end{bmatrix}$$

In the excel file, choose the sheet WITHOUT $F_\pi(s)$:
we have the calculation with $V(s)$ Value Iteration, under Bellman's Optimal Politic

Calculation with V(s) Value Iteration, under Bellman's Optimal Politic

		a1=←	a2=→		
	sF1	s1	s2	s3	sF2
fR(s)=	-10	0	-0.04	-0.04	10
input V(s)=	0	0	0	0	0
V(s)=	0	-0.04	0	10	10
γ=	0.9				

Bellman's Optimality Eq. for V(S):

$$V(s) \leftarrow \max_a \left[\sum_{s_f \in S} P_{M_T}(s_f|s,a)[f_R(s,a,s_f) + \gamma V(s_f)] \right] \quad (1)$$

→ Reward func.

→ V(S) values: since it's a vector, we have a value per state

→ Init with zeroes.

$P_{MT} = 1$ for s_1, $\emptyset$ for rest

here, we write for each state the Bellman's Optimality Equation:

Formally

$$V(s) \leftarrow \max_a \left[\sum_{s_f \in S} P_{M_T}(s_f|s,a)[f_R(s,a,s_f) + \gamma V(s_f)] \right] \Rightarrow \text{=MAX(E4+\$E\$10*E6, F4+\$E\$10*F6)}$$

One sum per action → Since it is deterministic, the sum Σ disappears, and thus we only got the $f_R(s_f) + \gamma V(s_f)$ term for each action.

1 for current state, $\emptyset$ for the rest

maximum

deterministic

max of two terms: one for each action ($a_1 = \leftarrow$, $a_2 = \rightarrow$)

for $a_1 = \leftarrow$ in the cell:

$(-10_1 + 0.9 * \emptyset_1,\ \emptyset_2 + 0.9 * \emptyset_2)$

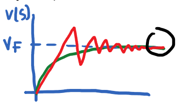
$a_1 = \leftarrow$ $a_2 = \rightarrow$

immediate reward after transition with a_i

1^{st} iteration result: copy it and paste nums → w/o format in above V(s) (input)

→ all 5 V(S) must converge at the same iteration

we say Convergence when $\Delta V(s)$ is very small → we decide how small = 0.1 or 0.001 or 0.0001



Here we are looking for the Optimal Politic, and we will do so from these V(S) values when converged: V(s) values' approximation, but how do we know the action?

↳ each V(S) was calculated as the max(a_1, a_2) thus if we write each of the two:

cell =MAX(E4+\$E\$10*E6, F4+\$E\$10*F6)

$a_1 \leftarrow$ → a_2

→ we need to write how the cell decided so that we can deduce the action.

55.542887	x→ a_2	72.826047						
	55.542887	x→ a_2	80.918447					
		72.826047	x→ a_2	89.954447				
			80.918447	x→ a_2	99.994447			
				89.954447	x→ a_2	99.994447		

are the same from Q table

The solution to Bellman's Opt Eq are the 5 V(S) values, from which we can deduce the Optimal Politic.

↳ Optimal Politic: if agent started on $s_1, s_2, s_3, \dots$ it needs to move according to this politic: $a_2 \rightarrow$

gives the maximum possible reward on each state.

Calculation with Q(s,a) Value Iteration, under Bellman's Optimal Politic

	move ←	a1=←	a2=→ move →		move → but frontier	
	sF1	s1	s2	s3	sF2	
fR(s)=	-10	0	-0.04	-0.04	10	
input	a2=→	a2=→	a2=→	a2=→	a2=→	
Q(s,a1=←)=	0	0	0	0	0	a1=←
Q(s,a2=→)=	0	0	0	0	0	a2=→
formulas						
Q(s,a1=←)=	-10	-10	0	-0.04	-0.04	a1=← → Eqs for $a_1 = \leftarrow$
Q(s,a2=→)=	0	-0.04	-0.04	10	10	a2=→ → Eqs for $a_2 = \rightarrow$
γ=	0.9					

MAX

MAX

→ reward func.

→ This politic is discovered at the end ↳ Bellman's Equations outcome!

Whereas here, it is a matrix: → two rows (actions) and 5 columns (states)

↳ Init with zeroes

↳ In this matrix we write the Bellman's Optimality Equations (10)

this max will tell us which action for the final Optimal Politic

For Q(s,a) this is the Bellman's Opt Equation System

$$Q(s,a) \leftarrow \sum_{s_f \in S} P_{M_T}(s_f|s,a) \left[f_R(s,a,s_f) + \gamma \max_{a_f}[Q(s_f,a_f)] \right]$$

→ we will have 10 equations looking like this

↳ Since its the same deterministic problem, the sum and P_{MT} dissapear.

cell row is $a_1 = \leftarrow$

For this cell we have:

gamma $Q(SF_1, \leftarrow)$ $Q(SF_1, \leftarrow)$

$$Q(s,a) \leftarrow \cancel{\sum_{s_f \in S} P_{M_T}(s_f|s,a)} \underbrace{[f_R(s,a,s_f)}_{\text{immediate reward}} + \gamma \max_{a_f}[Q(s_f,a_f)]] \Rightarrow \text{=L4+\$L\$12*MAX(L6,L7)}$$

get

$$-10 + 0.9 * \max_{a_f}(\emptyset \quad \emptyset)$$

$\boxed{\text{cell}}$ → represents the value of the average accumulated reward given that we started on state s, and took (on that state), action a: the average acc. reward of all trayectories that start on s_1 (it's column name) and execute $a_1 = \leftarrow$ (row name) on that s_1 state they started with, and from then on, they do whatever.

→ Iterate as before: put the formula matrix's results into the input matrix values and so on.

→ Not all systems of Equations converge under Value Iteration, but Bellman's Optimality Equation Systems always converge (either $V(S)$ or $Q(s,a)$ versions).

→ What determines the Optimal Politic is the Reward function $f_R(s,a,s_f)$

→ Both $V(S)$ and $Q(S,a)$ give the <u>same</u> <u>output</u>:

V(s)=	71.8618263	79.9542263	88.9902263	99.0302263	99.0302263	
Q(s,a1=←)=	54.5786663	54.5786663	71.8618263	79.9542263	88.9902263	a1=←
Q(s,a2=→)=	71.8618263	79.9542263	88.9902263	99.0302263	99.0302263	a2=→

V(S) will take the max Q value, so in order examples V(S) will have other Q row value.

→ Q needs more memory: $V(S)$ calculates the same but when <u>needed</u>.

→ For $V(S)$ and $Q(s,a)$ we need P_{MT} and f_R to be known.