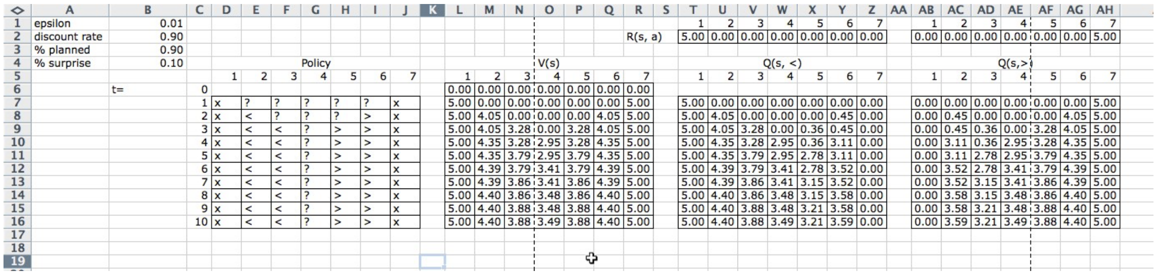
Reinforcement Learning Self Check
605.645 – Artificial Intelligence

The purpose of this self-check is to make sure you understand key concepts for the algorithms presented during the module and to prepare you for the programming assignment. As you work through problems, you should always be thinking “how would I do this in code? What basic data structures would I need? What operations on those basic data structures?”

As noted in the video lectures, it is possible to solve value iteration for a 1-d Markov Decision Process in a spreadsheet by explicitly modeling the π, V, and Q arrays. You have been provided with a starter spreadsheet (Module 11 Self Check.xlsx)

For the self-check, you have been provided with a starter spreadsheet (module4-self-check.xlsx). Fill in the formulas for Value Iteration (stochastic version) so that it looks like the following:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	epsilon	0.01																
2	discount rate	0.90																R(s, a)
3	% planned	0.90																
4	% surprise	0.10					Policy								V(s)			
5				1	2	3	4	5	6	7		1	2	3	4	5	6	7
6		t=	0									0.00	0.00	0.00	0.00	0.00	0.00	0.00
7			1	x	?	?	?	?	?	x		5.00	0.00	0.00	0.00	0.00	0.00	5.00
8			2	x	<	?	?	?	>	x		5.00	4.05	0.00	0.00	0.00	4.05	5.00
9			3	x	<	<	?	>	>	x		5.00	4.05	3.28	0.00	3.28	4.05	5.00
10			4	x	<	<	?	>	>	x		5.00	4.35	3.28	2.95	3.28	4.35	5.00
11			5	x	<	<	?	>	>	x		5.00	4.35	3.79	2.95	3.79	4.35	5.00
12			6	x	<	<	?	>	>	x		5.00	4.39	3.79	3.41	3.79	4.39	5.00
13			7	x	<	<	?	>	>	x		5.00	4.39	3.86	3.41	3.86	4.39	5.00
14			8	x	<	<	?	>	>	x		5.00	4.40	3.86	3.48	3.86	4.40	5.00
15			9	x	<	<	?	>	>	x		5.00	4.40	3.88	3.48	3.88	4.40	5.00
16			10	x	<	<	?	>	>	x		5.00	4.40	3.88	3.49	3.88	4.40	5.00
17																		
18																		
19																		

	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH
1		1	2	3	4	5	6	7		1	2	3	4	5	6	7
2		5.00	0.00	0.00	0.00	0.00	0.00	0.00		0.00	0.00	0.00	0.00	0.00	0.00	5.00
3																
4					Q(s, <)								Q(s,>			
5		1	2	3	4	5	6	7		1	2	3	4	5	6	7
6																
7		5.00	0.00	0.00	0.00	0.00	0.00	0.00		0.00	0.00	0.00	0.00	0.00	0.00	5.00
8		5.00	4.05	0.00	0.00	0.00	0.45	0.00		0.00	0.45	0.00	0.00	0.00	4.05	5.00
9		5.00	4.05	3.28	0.00	0.36	0.45	0.00		0.00	0.45	0.36	0.00	3.28	4.05	5.00
10		5.00	4.35	3.28	2.95	0.36	3.11	0.00		0.00	3.11	0.36	2.95	3.28	4.35	5.00
11		5.00	4.35	3.79	2.95	2.78	3.11	0.00		0.00	3.11	2.78	2.95	3.79	4.35	5.00
12		5.00	4.39	3.79	3.41	2.78	3.52	0.00		0.00	3.52	2.78	3.41	3.79	4.39	5.00
13		5.00	4.39	3.86	3.41	3.15	3.52	0.00		0.00	3.52	3.15	3.41	3.86	4.39	5.00
14		5.00	4.40	3.86	3.48	3.15	3.58	0.00		0.00	3.58	3.15	3.48	3.86	4.40	5.00
15		5.00	4.40	3.88	3.48	3.21	3.58	0.00		0.00	3.58	3.21	3.48	3.88	4.40	5.00
16		5.00	4.40	3.88	3.49	3.21	3.59	0.00		0.00	3.59	3.21	3.49	3.88	4.40	5.00
17																
18																
19																

1. the cell B1 is named “discount_rate”, you can refer to it by that name in cell formulas, for example, “=discount_rate*10”
2. the cell B2 is named “planned”, you can refer to it by that name in formulas.
3. The cell B3 is named “surprise”, you can refer to it by that name in formulas.
4. if(condition,then,else) can be used to test values in cells. It can also be nested (the first one will need an =).
5. =max(*cellref1*, *cellref2*) will return the maximum value.
6. You do not need to fill in formulas for the goal states 1 or 7 except in the case of V(s).
7. Use “<” for “left”, “?” for “pick random” and “>” for “right” in your policy formula.
8. There isn’t a good way to control the number of iterations automatically, so epsilon is there to let you know when you should stop copying.

You should get the same results as above and converge in 10 iterations.

Additional

1. Copy everything into a new Tab, “Right”. Change the rewards so that the agent always goes right.
2. Copy everything into a new Tab, “Left”. Change the rewards so that the agent always goes left except if they’re in the square right next to the goal on the right.
3. Try make a new Tab, “Fourteen”, that increases the size of the world from 7 to 14. Leave the rewards the same values. What happens to the policy? Use your experiments with #1 and #2 to change the rewards so that there is a deterministic policy for the whole world (no “?”).