

MI Sheet 8

Reinforcement Learning

Observe, we have three variants of the Bellman Equation

1. Optimal Utility

$$U(s) = \max_{a \in A(s)} \sum_{s'} P(s'|s, a) [R(s, a, s') + \gamma U(s')]$$

⇒ We use it in **Value Iteration** by writing it for every state then **iteratively updating**.

2. Fixed Policy

$$U^\pi(s) = \sum_{s'} P(s'|s, a) [R(s, a, s') + \gamma U(s')]$$

⇒ Drops the max since a is already decided by the fixed policy π (i.e., $a = \pi(s)$)

⇒ Equivalent to the previous formulation if the policy is optimal

⇒ We use it in **Policy Evaluation** by writing it for each state then **solving the linear system**

⇒ Next step is **Policy Improvement** where we find the policy again with previous formula but using **argmax**

3. Fixed Policy & Fixed Successor

$$U^\pi(s) = [R(s, a, s') + \gamma U(s')]$$

⇒ Also drops the sum over successors since the successor is also known

⇒ Holds exactly if this state under the policy's action always leads to that successor

⇒ We use it in **temporal difference learning**

Passive RL Algorithms

1. Direct Utility Estimation

$$U^\pi(s) = E \left[\sum_{t=0}^{\infty} \gamma^t R(s_t) \right]$$

- ⇒ Estimates the utility of each state by estimating the expected reward-to-go
 - That is, for each state average the rewards-to-go obtained for that state from the given episodes
- ⇒ Ignores Bellman Equation but still guaranteed to converge as trials approach ∞

2. Model-based Learning

$$P(s'|s, a) = \frac{N(s, a, s')}{N(s, a)} \qquad R(s, a, s') = \frac{\sum R(s, a, s')}{N(s, a, s')}$$

- ⇒ Estimates the model of the MDP from the given episodes
- ⇒ Then applies policy evaluation to find the utilities

3. Temporal Difference Learning

$$U^\pi(s) = U^\pi(s) + \alpha ([R(s, a, s') + \gamma U(s')] - U^\pi(s))$$

- ⇒ Applies the shown update for every transition in s, a, r, s'
- ⇒ $\alpha = \frac{1}{t}$ when we are updating for the t_{th} transition unless specified otherwise.

Active RL Algorithms

- No fixed policy and the action is decided by GLIE method such as $\epsilon - greedy$ or *exploration function*
- Defines $Q(s, a)$ which is the true utility from state s given that a is applied
- The following intuitively hold for it

$$U(s) = \max_{a \in A(s)} Q(s, a)$$

$$\pi(s) = \operatorname{argmax}_{a \in A(s)} Q(s, a)$$

- Which can be used to show in the general case:

$$Q(s, a) = \sum_{s'} P(s'|s, a) \left[R(s, a, s') + \gamma \max_{a' \in A(s')} Q(s', a') \right]$$

- And if the successor is known:

$$Q(s, a) = \left[R(s, a, s') + \gamma \max_{a' \in A(s')} Q(s', a') \right]$$

1. Model-based Learning

⇒ Apply passive model-based learning

⇒ Once the parameters are estimated apply value or policy iteration

2. Q-Learning (Off-policy Temporal Difference with Q-function)

$$Q(s, a) = Q(s, a) + \alpha \left(\left[r + \gamma \max_{a' \in A(s')} Q(s', a') \right] - Q(s, a) \right)$$

$$\pi(s) = \operatorname{argmax}_{a \in A(s)} Q(s, a)$$

⇒ Applies the shown update for each real-time transition in s, a, r, s'

⇒ Then updates the policy with the second equation

⇒ $\alpha = \frac{1}{t}$ when we are updating for the t_{th} transition unless specified otherwise.

3. SARSA (On-policy Temporal Difference with Q-function)

$$Q(s, a) = Q(s, a) + \alpha \left([r + \gamma Q(s', a')] - Q(s, a) \right)$$

$$\pi(s) = \operatorname{argmax}_{a \in A(s)} Q(s, a)$$

⇒ Applies the shown update for each real-time transition in s, a, r, s', a'

○ Thus, need to decide a' before the next transition

⇒ Then updates the policy with the second equation

⇒ $\alpha = \frac{1}{t}$ when we are updating for the t_{th} transition unless specified otherwise.

3. Q-Learning by Function Approximation

$$\hat{Q}(s,a) = \sum_{i=1}^K w_i f_i(s,a)$$

$$Q(s,a) = \left[r + \gamma \max_{a \in A(s')} \hat{Q}(s',a') \right]$$

$$w_k = w_k + \alpha \left(Q(s,a) - \hat{Q}(s,a) \right) \cdot f_k(s,a)$$

$$\pi(s) = \operatorname{argmax}_{a \in A(s)} \hat{Q}(s,a)$$

- ⇒ Applies the shown update (third equation for each weight) for each real-time transition in s,a,r,s'
- ⇒ Then updates the policy with the fourth
- ⇒ α is the learning rate

1. Consider the Following Environment



We only know the states, their rewards (via observation, in purple) and a fixed policy. The following episodes were also acquired by acting according to the shown policy in the environment. Assume $\gamma = 1$.

Episode	State Sequence
Episode 1	$s_2 \rightarrow s_1 \rightarrow s_3 \rightarrow s_5 \rightarrow s_7$
Episode 2	$s_1 \rightarrow s_2 \rightarrow s_1 \rightarrow s_3 \rightarrow s_4$
Episode 3	$s_5 \rightarrow s_6 \rightarrow s_7$

Apply all the Passive RL algorithms we covered to find the utility for all states

- Direct Utility Estimation
- Model-based Learning (Adaptive Dynamic Programming)
- Temporal Difference Learning

State	Utility
s_1	
s_2	
s_3	
s_4	
s_5	
s_6	
s_7	
s_8	

1. Direct Utility Estimation

// Observe that all states in any episode have reward -1 except for the last one (generally may need to trace as in the first parenthesis)

State	Utility
s_1	$\frac{[(-1 - 1 - 1 + 10)_1 + (-1 * 4 - 10)_2 + (-1 * 2 - 10)_2]}{3} = -6.33$
s_2	$\frac{[(-1 * 4 + 10)_1 + (-1 * 3 - 10)_2]}{2} = -3.5$
s_3	$\frac{[(-1 * 2 + 10)_1 + (-1 * 1 - 10)_2]}{2} = -1.5$
s_4	$\frac{[(-10)_2]}{1} = -10$
s_5	$\frac{[(-1 * 1 + 10)_1 + (-1 * 2 + 10)_3]}{2} = 8.5$
s_6	$\frac{[(-1 * 1 + 10)_3]}{1} = 9$
s_7	$\frac{[(+10)_3]}{1} = 10$
s_8	—



- This is similar to what we did in the lecture, just average the observed rewards-to-go for each of the states over all the episodes (unless stated otherwise).
- We used subscripts to denote from which episode did we write this reward-to-go but this isn't official notation (its just for clarity).
- As also demonstrated later in the lecture, we can use a running average to solve this. Recall,

$$NewAvg = OldAvg + \alpha(NewSample - OldAvg)$$

- The sample in this case is the reward-to-go, the NewAvg is initially the first sample and then whenever we see a state again, we find its reward-to-go (NewSample) and incorporate it using this equation. If we follow this approach, we can consider episode by episode with state by state in each and apply the described scheme.

2. Model-based Learning (Adaptive Dynamic Programming)

Let’s put this here

Episode	State Sequence
Episode 1	$s_2 \rightarrow s_1 \rightarrow s_3 \rightarrow s_5 \rightarrow s_7$
Episode 2	$s_1 \rightarrow s_2 \rightarrow s_1 \rightarrow s_3 \rightarrow s_4$
Episode 3	$s_5 \rightarrow s_6 \rightarrow s_7$

By counting how many a specific transition occurred over the number of all transitions from the state using the same action we get the following

State	Possible Successors	Transitions	Transition Model
s_1	s_3, s_2	3	$P(s_3 s_1, \pi(s_1)) = \frac{2}{3}$ $P(s_2 s_1, \pi(s_1)) = \frac{1}{3}$
s_2	s_1	2	$P(s_1 s_2, \pi(s_2)) = \frac{2}{2}$
s_3	s_4, s_5	2	$P(s_4 s_3, \pi(s_3)) = \frac{1}{2}$ $P(s_5 s_3, \pi(s_3)) = \frac{1}{2}$
s_4	Exit	1	$P(Exit s_7, \pi(s_7)) = \frac{1}{1}$
s_5	s_6, s_7	2	$P(s_6 s_5, \pi(s_5)) = \frac{1}{2}$ $P(s_7 s_5, \pi(s_5)) = \frac{1}{2}$
s_6	s_7	1	$P(s_7 s_6, \pi(s_6)) = \frac{1}{1}$
s_7	Exit	1	$P(Exit s_4, \pi(s_4)) = \frac{1}{1}$
s_8	—	-	—

Now we have the transition model and the reward model is given. Notice that the Exit transitions were just written to match the lecture but it will be useless in the next step where we apply **policy evaluation as long as we set the terminal states correctly**. In conclusion, you can pretend not seeing them.

Let's apply policy evaluation

$U(S_1) = \frac{2}{3} * (-1 + U(S_3)) + \frac{1}{3} * (-1 + U(S_2))$
$U(S_2) = \frac{2}{2} * (-1 + U(S_1))$
$U(S_3) = \frac{1}{2} * (-1 + U(S_4)) + \frac{1}{2} * (-1 + U(S_5))$
$U(S_4) = -10$
$U(S_5) = \frac{1}{2} * (-1 + U(S_6)) + \frac{1}{2} * (-1 + U(S_7))$
$U(S_6) = \frac{1}{1} * (-1 + U(S_7))$
$U(S_7) = 10$
$U(S_8) = -$

Observe that we used the general form for Bellman Equation although the reward is constant for each state and that it's still correct (but you can factor it out of course)

By plugging with the terminal states and solving the system:

$$U(S_6) = \frac{1}{1} * (-1 + U(S_7)) \rightarrow U(S_6) = 9$$

$$U(S_5) = \frac{1}{2} * (-1 + U(S_6)) + \frac{1}{2} * (-1 + U(S_7)) \rightarrow U(S_5) = 8.5$$

$$U(S_3) = \frac{1}{2} * (-1 + U(S_4)) + \frac{1}{2} * (-1 + U(S_5)) \rightarrow U(S_3) = -1.75$$

$$U(S_2) = \frac{2}{2} * (-1 + U(S_1)) \rightarrow U(S_2) = U(S_1) - 1$$

$$\begin{aligned}
 U(S_1) &= \frac{2}{3} * (-1 + U(S_3)) + \frac{1}{3} * (-1 + U(S_2)) \rightarrow U(S_1) = -1 + \frac{2}{3} * -1.75 + \frac{1}{3} (U(S_1) - 1) \\
 &\rightarrow U(S_1) = -3.75 \text{ and thus } U(S_2) = -4.75
 \end{aligned}$$

The final utilities are hence

$U(S_1)$	$U(S_2)$	$U(S_3)$	$U(S_4)$	$U(S_5)$	$U(S_6)$	$U(S_7)$	$U(S_8)$
-3.75	-4.75	-1.75	-10	8.5	9	10	-

3. Temporal Difference Learning

Let $\alpha=0.1$ and initialize U as 0 (for nonterminal states) and solve.

$$U(s_i) = U(s_i) + \alpha([R(s,a,s') + \gamma U(s')] - U(s_i))$$

Episode	State Sequence
Episode 1	$s_2 \rightarrow s_1 \rightarrow s_3 \rightarrow s_5 \rightarrow s_7$
Episode 2	$s_1 \rightarrow s_2 \rightarrow s_1 \rightarrow s_3 \rightarrow s_4$
Episode 3	$s_5 \rightarrow s_6 \rightarrow s_7$

Utilities

	$U(S_1)$	$U(S_2)$	$U(S_3)$	$U(S_4)$	$U(S_5)$	$U(S_6)$	$U(S_7)$	$U(S_8)$
Initial	0	0	0	-10	0	0	10	-
Ep. 1	-0.1	-0.1	-0.1		0.9			
Ep. 2	-0.2 -0.29	-0.21	-1.19					
					-0.71	0.9		

// Whenever a temporal difference update changes some utility, we forget about the previous one and write the new one (this is what further rows below initial are for).

Looping over transitions in episode 1

$s_2 \rightarrow s_1$	$U(s_2) = U(s_2) + \alpha([R(s,a,s') + \gamma U(s_1)] - U(s_2)) = 0 + 0.1([-1 + 1 * 0] - 0) = -0.1$
$s_1 \rightarrow s_3$	$U(s_1) = 0 + 0.1(-1 + 1 * 0 - 0) = -0.1$
$s_3 \rightarrow s_5$	$U(s_3) = 0 + 0.1(-1 + 1 * 0 - 0) = -0.1$
$s_5 \rightarrow s_7$	$U(s_5) = 0 + 0.1(-1 + 1 * 10 - 0) = 0.9$

Looping over transitions in episode 2











$s_1 \rightarrow s_2$	$U(s_1) = -0.1 + 0.1(-1 + 1 * -0.1 - -0.1) = -0.2$
$s_2 \rightarrow s_1$	$U(s_2) = -0.1 + 0.1(-1 + 1 * -0.2 - -0.1) = -0.21$
$s_1 \rightarrow s_3$	$U(s_1) = -0.2 + 0.1(-1 + 1 * -0.1 - -0.2) = -0.29$
$s_3 \rightarrow s_4$	$U(s_3) = -0.1 + 0.1(-1 + 1 * -10 - -0.1) = -1.19$

Looping over transitions in episode 3

$s_5 \rightarrow s_6$	$U(s_5) = 0.9 + 0.1(-1 + 1 * 0 - 0.9) = 0.71$
$s_6 \rightarrow s_7$	$U(s_6) = 0 + 0.1(-1 + 1 * 10 - 0) = 0.9$

	$U(s_1)$	$U(s_2)$	$U(s_3)$	$U(s_4)$	$U(s_5)$	$U(s_6)$	$U(s_7)$	$U(s_8)$
Initial	-0.29	-0.21	-1.19	-10	0.71	0.9	10	-









We are done with Passive RL! Here is another example that’s just as easy to test your skills:

3				+1
2				-1
1				
	1	2	3	4

Episode	State Sequence
Episode 1	$(1,1) \rightarrow (1,2) \rightarrow (1,3) \rightarrow (2,3) \rightarrow (3,3) \rightarrow (4,3)$
Episode 2	$(3,2) \rightarrow (3,3) \rightarrow (3,2) \rightarrow (3,3) \rightarrow (4,3)$
Episode 3	$(2,3) \rightarrow (3,3) \rightarrow (3,2) \rightarrow (4,2)$

Let the reward for all nonterminal states be zero and apply the three algorithms like we did above. Check with TA solution if needed [26, 37] in the slides.

Now let’s apply active RL algorithms on the environment we considered. This time, no fixed policy or transitions are given. We will have to learn the former and interact with the environment for the latter

							
-1	-1	-1	-10	-1	-1	+10	-10

- 1. Q-Learning
- 2. SARSA
- 3. Q-Learning with Function Approximation

Recall, in active RL we act in the environment to get a dataset of transitions

- This interaction captures the stochasticity in the environment; when you are at s and do action a the environment is what decides s' and r accordingly.
- Since we don't have the environment physically, we will simulate that we do using the following transition model
 - Going left always succeeds
 - Going right takes you one step or two steps right with probability 0.8 and 0.2 respectively.
 - We should take there into account when deciding on s' and r accordingly

To choose actions, it's required to do pure exploitation. When it results in a tie, we should just go left.

In any of the algorithms above, we should end up with Q for all possible s, a pairs. The corresponding policy (which is why we are learning Q) is the result of argmaxing over the actions.

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
$Q(S, L)$	0	0	0	0	0	0	0	0
$Q(S, R)$	0	0	0		0	0		
$\pi(S)$	L	L	L	E	L	L	E	E

Initially, they will be all zero for the Q s and all L for the π due to the ties.

1. Q-Learning ($\alpha = 0.1$)

$$Q(s, a) = Q(s, a) + \alpha ([r + \gamma \max(Q(s', L), Q(s', R))] - Q(s, a))$$

Now we will repeatedly generate a transition, apply the Q-learning update then update the table.

Recall that the initial state is S_2 and that L always succeeds so the first transition must be $S_2, L, -1, S_1$. Then in the next transition we will start from S_1 , see which of $Q(s_1, L)$ and $Q(s_2, L)$ is bigger to decide the action then apply it and so on.

If we ever need to go right, then 1 in every 5 times it should be two rather than one step.

Every time we compute any Q, the previous one should be overwritten in the Q-table. One table should be enough for this, but each change is recorded in a separate table here for clarity.

Transition (<i>s, a, r, s'</i>)	Update	Q-Table																								
<i>S</i> ₂ , <i>L</i> , −1, <i>S</i> ₁	$Q(s_2, L) = 0 + 0.1(-1 + 1 * \max(0,0) - 0) = -0.1$	<table><tr><td></td><td><i>S</i>₁</td><td><i>S</i>₂</td><td><i>S</i>₃</td><td><i>S</i>₅</td><td><i>S</i>₆</td></tr><tr><td><i>Q(S, L)</i></td><td>0</td><td>-0.1</td><td>0</td><td>0</td><td>0</td></tr><tr><td><i>Q(S, R)</i></td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td><i>π(S)</i></td><td>L</td><td>R</td><td>L</td><td>L</td><td>L</td></tr></table>		<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆	<i>Q(S, L)</i>	0	-0.1	0	0	0	<i>Q(S, R)</i>	0	0	0	0	0	<i>π(S)</i>	L	R	L	L	L
	<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆																					
<i>Q(S, L)</i>	0	-0.1	0	0	0																					
<i>Q(S, R)</i>	0	0	0	0	0																					
<i>π(S)</i>	L	R	L	L	L																					
<i>S</i> ₁ , <i>L</i> , −1, <i>S</i> ₁	$Q(s_1, L) = 0 + 0.1(-1 + 1 * \max(0,0) - 0) = -0.1$	<table><tr><td></td><td><i>S</i>₁</td><td><i>S</i>₂</td><td><i>S</i>₃</td><td><i>S</i>₅</td><td><i>S</i>₆</td></tr><tr><td><i>Q(S, L)</i></td><td>-0.1</td><td>-0.1</td><td>0</td><td>0</td><td>0</td></tr><tr><td><i>Q(S, R)</i></td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td><i>π(S)</i></td><td>R</td><td>R</td><td>L</td><td>L</td><td>L</td></tr></table>		<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆	<i>Q(S, L)</i>	-0.1	-0.1	0	0	0	<i>Q(S, R)</i>	0	0	0	0	0	<i>π(S)</i>	R	R	L	L	L
	<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆																					
<i>Q(S, L)</i>	-0.1	-0.1	0	0	0																					
<i>Q(S, R)</i>	0	0	0	0	0																					
<i>π(S)</i>	R	R	L	L	L																					
<i>S</i> ₁ , <i>R</i> , −1, <i>S</i> ₂	$Q(s_1, R) = 0 + 0.1(-1 + 1 * \max(-0.1,0) - 0) = -0.1$	<table><tr><td></td><td><i>S</i>₁</td><td><i>S</i>₂</td><td><i>S</i>₃</td><td><i>S</i>₅</td><td><i>S</i>₆</td></tr><tr><td><i>Q(S, L)</i></td><td>-0.1</td><td>-0.1</td><td>0</td><td>0</td><td>0</td></tr><tr><td><i>Q(S, R)</i></td><td>-0.1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td><i>π(S)</i></td><td>L</td><td>R</td><td>L</td><td>L</td><td>L</td></tr></table>		<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆	<i>Q(S, L)</i>	-0.1	-0.1	0	0	0	<i>Q(S, R)</i>	-0.1	0	0	0	0	<i>π(S)</i>	L	R	L	L	L
	<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆																					
<i>Q(S, L)</i>	-0.1	-0.1	0	0	0																					
<i>Q(S, R)</i>	-0.1	0	0	0	0																					
<i>π(S)</i>	L	R	L	L	L																					
<i>S</i> ₂ , <i>R</i> , −1, <i>S</i> ₃	$Q(s_2, R) = 0 + 0.1(-1 + 1 * \max(0,0) - 0) = -0.1$	<table><tr><td></td><td><i>S</i>₁</td><td><i>S</i>₂</td><td><i>S</i>₃</td><td><i>S</i>₅</td><td><i>S</i>₆</td></tr><tr><td><i>Q(S, L)</i></td><td>-0.1</td><td>-0.1</td><td>0</td><td>0</td><td>0</td></tr><tr><td><i>Q(S, R)</i></td><td>-0.1</td><td>-0.1</td><td>0</td><td>0</td><td>0</td></tr><tr><td><i>π(S)</i></td><td>L</td><td>L</td><td>L</td><td>L</td><td>L</td></tr></table>		<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆	<i>Q(S, L)</i>	-0.1	-0.1	0	0	0	<i>Q(S, R)</i>	-0.1	-0.1	0	0	0	<i>π(S)</i>	L	L	L	L	L
	<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆																					
<i>Q(S, L)</i>	-0.1	-0.1	0	0	0																					
<i>Q(S, R)</i>	-0.1	-0.1	0	0	0																					
<i>π(S)</i>	L	L	L	L	L																					
<i>S</i> ₃ , <i>L</i> , −1, <i>S</i> ₂	$Q(s_3, L) = 0 + 0.1(-1 + 1 * \max(-0.1, -0.1) - 0) = -0.11$	<table><tr><td></td><td><i>S</i>₁</td><td><i>S</i>₂</td><td><i>S</i>₃</td><td><i>S</i>₅</td><td><i>S</i>₆</td></tr><tr><td><i>Q(S, L)</i></td><td>-0.1</td><td>-0.1</td><td>-0.11</td><td>0</td><td>0</td></tr><tr><td><i>Q(S, R)</i></td><td>-0.1</td><td>-0.1</td><td>0</td><td>0</td><td>0</td></tr><tr><td><i>π(S)</i></td><td>L</td><td>L</td><td>R</td><td>L</td><td>L</td></tr></table>		<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆	<i>Q(S, L)</i>	-0.1	-0.1	-0.11	0	0	<i>Q(S, R)</i>	-0.1	-0.1	0	0	0	<i>π(S)</i>	L	L	R	L	L
	<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆																					
<i>Q(S, L)</i>	-0.1	-0.1	-0.11	0	0																					
<i>Q(S, R)</i>	-0.1	-0.1	0	0	0																					
<i>π(S)</i>	L	L	R	L	L																					
<i>S</i> ₂ , <i>L</i> , −1, <i>S</i> ₁	$Q(s_2, L) = -0.1 + 0.1(-1 + 1 * \max(-0.1, -0.1) - -0.1) = -0.2$	<table><tr><td></td><td><i>S</i>₁</td><td><i>S</i>₂</td><td><i>S</i>₃</td><td><i>S</i>₅</td><td><i>S</i>₆</td></tr><tr><td><i>Q(S, L)</i></td><td>-0.1</td><td>-0.2</td><td>-0.11</td><td>0</td><td>0</td></tr><tr><td><i>Q(S, R)</i></td><td>-0.1</td><td>-0.1</td><td>0</td><td>0</td><td>0</td></tr><tr><td><i>π(S)</i></td><td>L</td><td>R</td><td>R</td><td>L</td><td>L</td></tr></table>		<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆	<i>Q(S, L)</i>	-0.1	-0.2	-0.11	0	0	<i>Q(S, R)</i>	-0.1	-0.1	0	0	0	<i>π(S)</i>	L	R	R	L	L
	<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆																					
<i>Q(S, L)</i>	-0.1	-0.2	-0.11	0	0																					
<i>Q(S, R)</i>	-0.1	-0.1	0	0	0																					
<i>π(S)</i>	L	R	R	L	L																					
<i>S</i> ₁ , <i>L</i> , −1, <i>S</i> ₁	$Q(s_1, L) = -0.1 + 0.1(-1 + 1 * \max(-0.1, -0.1) - -0.1) = -0.2$	<table><tr><td></td><td><i>S</i>₁</td><td><i>S</i>₂</td><td><i>S</i>₃</td><td><i>S</i>₅</td><td><i>S</i>₆</td></tr><tr><td><i>Q(S, L)</i></td><td>-0.2</td><td>-0.2</td><td>-0.11</td><td>0</td><td>0</td></tr><tr><td><i>Q(S, R)</i></td><td>-0.1</td><td>-0.1</td><td>0</td><td>0</td><td>0</td></tr><tr><td><i>π(S)</i></td><td>R</td><td>R</td><td>R</td><td>L</td><td>L</td></tr></table>		<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆	<i>Q(S, L)</i>	-0.2	-0.2	-0.11	0	0	<i>Q(S, R)</i>	-0.1	-0.1	0	0	0	<i>π(S)</i>	R	R	R	L	L
	<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆																					
<i>Q(S, L)</i>	-0.2	-0.2	-0.11	0	0																					
<i>Q(S, R)</i>	-0.1	-0.1	0	0	0																					
<i>π(S)</i>	R	R	R	L	L																					
<i>S</i> ₁ , <i>R</i> , −1, <i>S</i> ₃	$Q(s_1, R) = -0.1 + 0.1(-1 + 1 * \max(-0.11,0) - -0.1) = -0.19$	<table><tr><td></td><td><i>S</i>₁</td><td><i>S</i>₂</td><td><i>S</i>₃</td><td><i>S</i>₅</td><td><i>S</i>₆</td></tr><tr><td><i>Q(S, L)</i></td><td>-0.2</td><td>-0.2</td><td>-0.11</td><td>0</td><td>0</td></tr><tr><td><i>Q(S, R)</i></td><td>-0.19</td><td>-0.1</td><td>0</td><td>0</td><td>0</td></tr><tr><td><i>π(S)</i></td><td>R</td><td>R</td><td>R</td><td>L</td><td>L</td></tr></table>		<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆	<i>Q(S, L)</i>	-0.2	-0.2	-0.11	0	0	<i>Q(S, R)</i>	-0.19	-0.1	0	0	0	<i>π(S)</i>	R	R	R	L	L
	<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆																					
<i>Q(S, L)</i>	-0.2	-0.2	-0.11	0	0																					
<i>Q(S, R)</i>	-0.19	-0.1	0	0	0																					
<i>π(S)</i>	R	R	R	L	L																					
<i>S</i> ₃ , <i>R</i> , −1, <i>S</i> ₅	$Q(s_3, R) = 0 + 0.1(-1 + 1 * \max(0,0) - 0) = -0.1$	<table><tr><td></td><td><i>S</i>₁</td><td><i>S</i>₂</td><td><i>S</i>₃</td><td><i>S</i>₅</td><td><i>S</i>₆</td></tr><tr><td><i>Q(S, L)</i></td><td>-0.2</td><td>-0.2</td><td>-0.11</td><td>0</td><td>0</td></tr><tr><td><i>Q(S, R)</i></td><td>-0.19</td><td>-0.1</td><td>-0.1</td><td>0</td><td>0</td></tr><tr><td><i>π(S)</i></td><td>R</td><td>R</td><td>R</td><td>L</td><td>L</td></tr></table>		<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆	<i>Q(S, L)</i>	-0.2	-0.2	-0.11	0	0	<i>Q(S, R)</i>	-0.19	-0.1	-0.1	0	0	<i>π(S)</i>	R	R	R	L	L
	<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆																					
<i>Q(S, L)</i>	-0.2	-0.2	-0.11	0	0																					
<i>Q(S, R)</i>	-0.19	-0.1	-0.1	0	0																					
<i>π(S)</i>	R	R	R	L	L																					
<i>S</i> ₅ , <i>L</i> , −1, <i>S</i> ₄	Set $Q(s_4, Exit) = -10$ $Q(s_5, L) = 0 + 0.1(-1 + 1 * -10 - 0) = -1.1$	<table><tr><td></td><td><i>S</i>₁</td><td><i>S</i>₂</td><td><i>S</i>₃</td><td><i>S</i>₅</td><td><i>S</i>₆</td></tr><tr><td><i>Q(S, L)</i></td><td>-0.2</td><td>-0.2</td><td>-0.11</td><td>-1.1</td><td>0</td></tr><tr><td><i>Q(S, R)</i></td><td>-0.19</td><td>-0.1</td><td>-0.1</td><td>0</td><td>0</td></tr><tr><td><i>π(S)</i></td><td>R</td><td>R</td><td>R</td><td>R</td><td>L</td></tr></table>		<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆	<i>Q(S, L)</i>	-0.2	-0.2	-0.11	-1.1	0	<i>Q(S, R)</i>	-0.19	-0.1	-0.1	0	0	<i>π(S)</i>	R	R	R	R	L
	<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₅	<i>S</i> ₆																					
<i>Q(S, L)</i>	-0.2	-0.2	-0.11	-1.1	0																					
<i>Q(S, R)</i>	-0.19	-0.1	-0.1	0	0																					
<i>π(S)</i>	R	R	R	R	L																					

Final Policy Learnt

	<i>S</i> ₁	<i>S</i> ₂	<i>S</i> ₃	<i>S</i> ₄	<i>S</i> ₅	<i>S</i> ₆	<i>S</i> ₇	<i>S</i> ₈
<i>π(S)</i>	R	R	R	Exit	R	L	Exit	Exit

2. SARSA ($\alpha = 0.1$)

$$Q(s,a) = Q(s,a) + \alpha([r + \gamma Q(s',a')] - Q(s,a))$$

- Recall, Q-Learning is off-policy learning
 - The update uses the Q-function of the best next action (i.e., $\max_{a \in A(s')} Q(s',a')$) even if another action will be chosen
- SARSA is on-policy learning
 - Let's see what next action will be chosen then use its Q-function
 - This means that
 - We pick the action a at the state s and execute it to get a transition s,a,r,s'
 - Before doing the update, we pick the action a' at the observed s'
 - The action that will be picked for the next transition is a'

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
$Q(S,L)$	0	0	0	0	0	0	0	0
$Q(S,R)$	0	0	0		0	0		
$\pi(S)$	L	L	L	Exit	L	L	Exit	Exit

For instance, below in the first iteration. We picked L since this is what our policy told us to do for S_2 , we executed it and got to S_1 and chose the action for it according to the policy which was L again. Instead of using $\max(Q(s',L), Q(s',R))$ in the update, we just used $Q(s',L)$.

In the next iteration, we picked action L which was chosen in the previous iteration, got to S_1 and chose the action for it according to the policy which was L again.

Transition (s, a, r, s', a)	Update	Q-Table																								
$S_2, L, -1, S_1, L$ $Q(s_1, L) = 0$	$Q(s_2, L) = 0 + 0.1(-1 + 1 * 0 - 0) = -0.1$	<table><tr><td></td><td>S_1</td><td>S_2</td><td>S_3</td><td>S_5</td><td>S_6</td></tr><tr><td>$Q(S, L)$</td><td>0</td><td>-0.1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>$Q(S, R)$</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>$\pi(S)$</td><td>L</td><td>R</td><td>L</td><td>L</td><td>L</td></tr></table>		S_1	S_2	S_3	S_5	S_6	$Q(S, L)$	0	-0.1	0	0	0	$Q(S, R)$	0	0	0	0	0	$\pi(S)$	L	R	L	L	L
	S_1	S_2	S_3	S_5	S_6																					
$Q(S, L)$	0	-0.1	0	0	0																					
$Q(S, R)$	0	0	0	0	0																					
$\pi(S)$	L	R	L	L	L																					
$S_1, L, -1, S_1, L$ $Q(s_1, L) = 0$	$Q(s_1, L) = 0 + 0.1(-1 + 1 * 0 - 0) = -0.1$	<table><tr><td></td><td>S_1</td><td>S_2</td><td>S_3</td><td>S_5</td><td>S_6</td></tr><tr><td>$Q(S, L)$</td><td>-0.1</td><td>-0.1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>$Q(S, R)$</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>$\pi(S)$</td><td>R</td><td>R</td><td>L</td><td>L</td><td>L</td></tr></table>		S_1	S_2	S_3	S_5	S_6	$Q(S, L)$	-0.1	-0.1	0	0	0	$Q(S, R)$	0	0	0	0	0	$\pi(S)$	R	R	L	L	L
	S_1	S_2	S_3	S_5	S_6																					
$Q(S, L)$	-0.1	-0.1	0	0	0																					
$Q(S, R)$	0	0	0	0	0																					
$\pi(S)$	R	R	L	L	L																					
$S_1, L, -1, S_1, R$ $Q(s_1, R) = 0$	$Q(s_1, L) = -0.1 + 0.1(-1 + 1 * 0 - -0.1) = -0.19$	<table><tr><td></td><td>S_1</td><td>S_2</td><td>S_3</td><td>S_5</td><td>S_6</td></tr><tr><td>$Q(S, L)$</td><td>-0.19</td><td>-0.1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>$Q(S, R)$</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>$\pi(S)$</td><td>R</td><td>R</td><td>L</td><td>L</td><td>L</td></tr></table>		S_1	S_2	S_3	S_5	S_6	$Q(S, L)$	-0.19	-0.1	0	0	0	$Q(S, R)$	0	0	0	0	0	$\pi(S)$	R	R	L	L	L
	S_1	S_2	S_3	S_5	S_6																					
$Q(S, L)$	-0.19	-0.1	0	0	0																					
$Q(S, R)$	0	0	0	0	0																					
$\pi(S)$	R	R	L	L	L																					

<div><div>$S_1, R, -1, S_2, R$</div><div>$Q(s_2, R) = 0$</div></div>	<div>$Q(s_1, R) = 0 + 0.1(-1 + 1 * 0 - 0) = -0.1$</div>	<table><tr><td></td><td>S_1</td><td>S_2</td><td>S_3</td><td>S_5</td><td>S_6</td></tr><tr><td>$Q(S, L)$</td><td>-0.19</td><td>-0.1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>$Q(S, R)$</td><td>-0.1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr><tr><td>$\pi(S)$</td><td>R</td><td>R</td><td>L</td><td>L</td><td>L</td></tr></table>		S_1	S_2	S_3	S_5	S_6	$Q(S, L)$	-0.19	-0.1	0	0	0	$Q(S, R)$	-0.1	0	0	0	0	$\pi(S)$	R	R	L	L	L
	S_1	S_2	S_3	S_5	S_6																					
$Q(S, L)$	-0.19	-0.1	0	0	0																					
$Q(S, R)$	-0.1	0	0	0	0																					
$\pi(S)$	R	R	L	L	L																					
<div><div>$S_2, R, -1, S_3, L$</div><div>$Q(s_3, L) = 0$</div></div>	<div>$Q(s_2, R) = 0 + 0.1(-1 + 1 * 0 - 0) = -0.1$</div>	<table><tr><td></td><td>S_1</td><td>S_2</td><td>S_3</td><td>S_5</td><td>S_6</td></tr><tr><td>$Q(S, L)$</td><td>-0.19</td><td>-0.1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>$Q(S, R)$</td><td>-0.1</td><td>-0.1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>$\pi(S)$</td><td>R</td><td>L</td><td>L</td><td>L</td><td>L</td></tr></table>		S_1	S_2	S_3	S_5	S_6	$Q(S, L)$	-0.19	-0.1	0	0	0	$Q(S, R)$	-0.1	-0.1	0	0	0	$\pi(S)$	R	L	L	L	L
	S_1	S_2	S_3	S_5	S_6																					
$Q(S, L)$	-0.19	-0.1	0	0	0																					
$Q(S, R)$	-0.1	-0.1	0	0	0																					
$\pi(S)$	R	L	L	L	L																					
<div><div>$S_3, L, -1, S_2, L$</div><div>$Q(s_2, L) = -0.1$</div></div>	<div>$Q(s_3, L) = 0 + 0.1(-1 + 1 * -0.1 - 0) = -0.11$</div>	<table><tr><td></td><td>S_1</td><td>S_2</td><td>S_3</td><td>S_5</td><td>S_6</td></tr><tr><td>$Q(S, L)$</td><td>-0.19</td><td>-0.1</td><td>-0.11</td><td>0</td><td>0</td></tr><tr><td>$Q(S, R)$</td><td>-0.1</td><td>-0.1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>$\pi(S)$</td><td>R</td><td>L</td><td>R</td><td>L</td><td>L</td></tr></table>		S_1	S_2	S_3	S_5	S_6	$Q(S, L)$	-0.19	-0.1	-0.11	0	0	$Q(S, R)$	-0.1	-0.1	0	0	0	$\pi(S)$	R	L	R	L	L
	S_1	S_2	S_3	S_5	S_6																					
$Q(S, L)$	-0.19	-0.1	-0.11	0	0																					
$Q(S, R)$	-0.1	-0.1	0	0	0																					
$\pi(S)$	R	L	R	L	L																					
<div><div>$S_2, L, -1, S_1, R$</div><div>$Q(s_1, R) = -0.1$</div></div>	<div>$Q(s_2, L) - 0.1 + 0.1(-1 + 1 * -0.1 - -0.1) = -0.2$</div>	<table><tr><td></td><td>S_1</td><td>S_2</td><td>S_3</td><td>S_5</td><td>S_6</td></tr><tr><td>$Q(S, L)$</td><td>-0.19</td><td>-0.2</td><td>-0.11</td><td>0</td><td>0</td></tr><tr><td>$Q(S, R)$</td><td>-0.1</td><td>-0.1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>$\pi(S)$</td><td>R</td><td>R</td><td>R</td><td>L</td><td>L</td></tr></table>		S_1	S_2	S_3	S_5	S_6	$Q(S, L)$	-0.19	-0.2	-0.11	0	0	$Q(S, R)$	-0.1	-0.1	0	0	0	$\pi(S)$	R	R	R	L	L
	S_1	S_2	S_3	S_5	S_6																					
$Q(S, L)$	-0.19	-0.2	-0.11	0	0																					
$Q(S, R)$	-0.1	-0.1	0	0	0																					
$\pi(S)$	R	R	R	L	L																					
<div><div>$S_1, R, -1, S_3, R$</div><div>$Q(s_3, R) = 0$</div></div>	<div>$Q(s_1, R) = -0.1 + 0.1(-1 + 1 * 0 - -0.1) = -0.19$</div>	<table><tr><td></td><td>S_1</td><td>S_2</td><td>S_3</td><td>S_5</td><td>S_6</td></tr><tr><td>$Q(S, L)$</td><td>-0.19</td><td>-0.2</td><td>-0.11</td><td>0</td><td>0</td></tr><tr><td>$Q(S, R)$</td><td>-0.19</td><td>-0.1</td><td>0</td><td>0</td><td>0</td></tr><tr><td>$\pi(S)$</td><td>L</td><td>R</td><td>R</td><td>L</td><td>L</td></tr></table>		S_1	S_2	S_3	S_5	S_6	$Q(S, L)$	-0.19	-0.2	-0.11	0	0	$Q(S, R)$	-0.19	-0.1	0	0	0	$\pi(S)$	L	R	R	L	L
	S_1	S_2	S_3	S_5	S_6																					
$Q(S, L)$	-0.19	-0.2	-0.11	0	0																					
$Q(S, R)$	-0.19	-0.1	0	0	0																					
$\pi(S)$	L	R	R	L	L																					
<div><div>$S_3, R, -1, S_4, E$</div><div>$Q(s_4, E) = -10$</div></div>	<div>Set $Q(s_4, Exit) = -10$ $Q(s_3, R) = 0 + 0.1(-1 + 1 * -10 - 0) = -1.1$</div>	<table><tr><td></td><td>S_1</td><td>S_2</td><td>S_3</td><td>S_5</td><td>S_6</td></tr><tr><td>$Q(S, L)$</td><td>-0.19</td><td>-0.2</td><td>-0.11</td><td>0</td><td>0</td></tr><tr><td>$Q(S, R)$</td><td>-0.19</td><td>-0.1</td><td>-1.1</td><td>0</td><td>0</td></tr><tr><td>$\pi(S)$</td><td>L</td><td>R</td><td>L</td><td>L</td><td>L</td></tr></table>		S_1	S_2	S_3	S_5	S_6	$Q(S, L)$	-0.19	-0.2	-0.11	0	0	$Q(S, R)$	-0.19	-0.1	-1.1	0	0	$\pi(S)$	L	R	L	L	L
	S_1	S_2	S_3	S_5	S_6																					
$Q(S, L)$	-0.19	-0.2	-0.11	0	0																					
$Q(S, R)$	-0.19	-0.1	-1.1	0	0																					
$\pi(S)$	L	R	L	L	L																					

Final Policy Learnt

	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
$\pi(S)$	L	R	L	Exit	L	L	Exit	Exit

3. Approximate Q-Learning ($\alpha = 0.1$)

Recall that we had,

$$\hat{Q}(s,a) = \sum_{i=1}^K w_i f_i(s,a) \qquad Q(s,a) = \left[r + \max_{a \in A(s')} \hat{Q}(s',a')\right]$$

$$E(s,a) = Q(s,a) - \hat{Q}(s,a)$$

$$w_k = w_k + \alpha * E(s,a). f_k(s,a)$$

Now it's typically not easy to find features that use the action, if we decide to let our features be only a function of the state we run into a problem since some s, a pairs are good and some are bad for the true Q-function. The solution is:

- Learn a separate Q-function for each action
 - Now its okay if the features involve the state only as each action can adjust weights differently depending how good or bad is that s, a pair.
- Now the true Q is a piece-wise defined function, depending on the given action it will plug in one of the Qs that were learnt

Consider Approximate Q-learning with one feature which is the state’s index and another feature that’s always 1 (bias). This yields the following

$$\hat{Q}(s,L) = w_{1L} * s + w_{0L} \qquad \hat{Q}(s,R) = w_{1R} * s + w_{0R}$$

$$\text{Since } Q(s,a) = r + \max\left(\hat{Q}(s',L), \hat{Q}(s',R)\right)$$

The update becomes

$$\text{if } a = L \left\{ \begin{array}{l} E(s,L) = r + \max\left(\hat{Q}(s',L), \hat{Q}(s',R)\right) - \hat{Q}(s,L) \\ w_{1L} = w_{1L} + \alpha * E(s,L).s \\ w_{0L} = w_{0L} + \alpha * E(s,L).1 \end{array} \right.$$

Will drop the hat on Q in next page

$$\text{if } a = R \left\{ \begin{array}{l} E(s,R) = r + \max\left(\hat{Q}(s',L), \hat{Q}(s',R)\right) - \hat{Q}(s,R) \\ w_{1R} = w_{1R} + \alpha * E(s,R).s \\ w_{0R} = w_{0R} + \alpha * E(s,R).1 \end{array} \right.$$

This will be our initial Q-table, we won’t be evaluating each Q for all states when a weight update occurs, since that seems so exhaustive. Alternatively, to decide the action for the next transition we will evaluate both Qs on demand and most of the time it will be easily done in your brain.

	w_1	w_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
$Q(S,L)$	0	0	0	0	0	0	0	0	0	0
$Q(S,R)$	0	0	0	0	0		0	0		
$\pi(S)$			L	L	L	Exit	L	L	Exit	Exit

Transition (s, a, r, s')	Update										
<div>Pick L</div> <div>$S_2, L, -1, S_1$</div>	$E = r + \max(w_{1L} * s' + w_{0L}, w_{1R} * s' + w_{0R}) - (w_{1L} * s + w_{0L})$ $= -1 + \max(0 * 1 + 0, 0 * 1 + 0) - (0 * 2 + 0) = -1$ $w_{1L} = w_{1L} + \alpha * E, s = 0 + 0.1 * -1 * 2 = -0.2$ $w_{0L} = w_{0L} + \alpha * E = 0 + 0.1 * -1 = -0.1$	<table> <tr><td></td><td>w_1</td><td>w_0</td></tr> <tr><td>$Q(S, L)$</td><td>-0.2</td><td>-0.1</td></tr> <tr><td>$Q(S, R)$</td><td>0</td><td>0</td></tr> </table>		w_1	w_0	$Q(S, L)$	-0.2	-0.1	$Q(S, R)$	0	0
	w_1	w_0									
$Q(S, L)$	-0.2	-0.1									
$Q(S, R)$	0	0									
<div>Pick R</div> <div>$S_1, R, -1, S_2$</div>	$E = -1 + \max(-0.2 * 2 + -0.1, 0 * 2 + 0) - (0 * 1 + 0) = -1$ $w_{1R} = 0 + 0.1 * -1 * 1 = -0.1$ $w_{0R} = 0 + 0.1 * -1 = -0.1$	<table> <tr><td></td><td>w_1</td><td>w_0</td></tr> <tr><td>$Q(S, L)$</td><td>-0.2</td><td>-0.1</td></tr> <tr><td>$Q(S, R)$</td><td>-0.1</td><td>-0.1</td></tr> </table>		w_1	w_0	$Q(S, L)$	-0.2	-0.1	$Q(S, R)$	-0.1	-0.1
	w_1	w_0									
$Q(S, L)$	-0.2	-0.1									
$Q(S, R)$	-0.1	-0.1									
<div>Pick R</div> <div>$S_2, R, -1, S_3$</div>	$E = -1 + \max(-0.2 * 3 + -0.1, -0.1 * 3 + -0.1) - (-0.1 * 2 + -0.1) = -1.1$ $w_{1R} = -0.1 + 0.1 * -1.1 * 2 = -0.32$ $w_{0R} = -0.1 + 0.1 * -1.1 = -0.21$	<table> <tr><td></td><td>w_1</td><td>w_0</td></tr> <tr><td>$Q(S, L)$</td><td>-0.2</td><td>-0.1</td></tr> <tr><td>$Q(S, R)$</td><td>-0.32</td><td>-0.21</td></tr> </table>		w_1	w_0	$Q(S, L)$	-0.2	-0.1	$Q(S, R)$	-0.32	-0.21
	w_1	w_0									
$Q(S, L)$	-0.2	-0.1									
$Q(S, R)$	-0.32	-0.21									
<div>Pick L</div> <div>$S_3, L, -1, S_2$</div>	$E = -1 + \max(-0.2 * 2 + -0.1, -0.32 * 2 + -0.21) - (-0.2 * 3 + -0.1) = -0.8$ $w_{1L} = -0.2 + 0.1 * -0.8 * 3 = -0.44$ $w_{0L} = -0.1 + 0.1 * -0.8 = -0.18$	<table> <tr><td></td><td>w_1</td><td>w_0</td></tr> <tr><td>$Q(S, L)$</td><td>-0.44</td><td>-0.18</td></tr> <tr><td>$Q(S, R)$</td><td>-0.32</td><td>-0.21</td></tr> </table>		w_1	w_0	$Q(S, L)$	-0.44	-0.18	$Q(S, R)$	-0.32	-0.21
	w_1	w_0									
$Q(S, L)$	-0.44	-0.18									
$Q(S, R)$	-0.32	-0.21									
<div>Pick R</div> <div>$S_2, R, -1, S_4$</div>	$E = -1 + \max(-0.44 * 4 + -0.18, -0.32 * 4 + -0.21) - (-0.32 * 2 + -0.21) = -1.64$ $w_{1R} = -0.32 + 0.1 * -1.64 * 2 = -0.648$ $w_{0R} = -0.21 + 0.1 * -1.64 = -0.374$	<table> <tr><td></td><td>w_1</td><td>w_0</td></tr> <tr><td>$Q(S, L)$</td><td>-0.44</td><td>-0.18</td></tr> <tr><td>$Q(S, R)$</td><td>-0.65</td><td>-0.37</td></tr> </table>		w_1	w_0	$Q(S, L)$	-0.44	-0.18	$Q(S, R)$	-0.65	-0.37
	w_1	w_0									
$Q(S, L)$	-0.44	-0.18									
$Q(S, R)$	-0.65	-0.37									
<div>$S_4, Exit, 10$</div>	$E = -10 + 0 + -(-0.44 * 4 + -0.18) = -8.06$ $w_{1L} = -0.44 + 0.1 * -8.06 * 4 = -3.664$ $w_{0L} = -0.18 + 0.1 * -8.06 = -0.986$ $E = -10 + 0 + -(-0.65 * 4 + -0.37) = -7.03$ $w_{1R} = -0.65 + 0.1 * -7.03 * 4 = -3.46$ $w_{0R} = -0.37 + 0.1 * -7.03 = -1.073$	<table> <tr><td></td><td>w_1</td><td>w_0</td></tr> <tr><td>$Q(S, L)$</td><td>-3.66</td><td>-0.99</td></tr> <tr><td>$Q(S, R)$</td><td>-3.46</td><td>-1.07</td></tr> </table>		w_1	w_0	$Q(S, L)$	-3.66	-0.99	$Q(S, R)$	-3.46	-1.07
	w_1	w_0									
$Q(S, L)$	-3.66	-0.99									
$Q(S, R)$	-3.46	-1.07									

Observe that for the last transition, we ideally would have a third version of the Q function for the exit action and would have updated its weights. The max term in E in this case becomes zero by Bellman equation since an agent gets no more reward (i.e., 0) once they reach the terminal state and perform the Exit action.

Since we didn't have a separate Q function for the Exit action, we resorted to the acceptable alternative of doing the update to both L and R.

Now to get the final policy, plug in each Q for all S from 1 to 8

	w_1	w_0	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8
$Q(S, L)$	-3.66	-0.99	-4.65	-8.31	-11.97	-15.6	-19.3	-23	-26.6	-30.3
$Q(S, R)$	-3.46	-1.07	-4.53	-7.99	-11.45	-14.9	-18.4	-21.8	-25.3	-28.7
$\pi(S)$			R	R	R	Exit	R	R	Exit	Exit

Although the Q functions assign values for invalid s, a pairs such as Left and Right for terminal states, we can ignore them since we know they are invalid, the function assigns values to states based on how similar they are.

Consider TD Learning with Function Approximation to Learn the Utilities of Different States. Write the Parameter Updates if the Utility will be Approximated through the following Function

$$\widehat{U}(x, y) = \theta_0 + \theta_1x + \theta_2y + \theta_3\sqrt{(x - x_g)^2 + (y - y_g)^2}$$

Recall,

$$\widehat{Q}(s, a) = \sum_{i=1}^K w_i f_i(s, a) \qquad w_k = w_k + \alpha \left(Q(s, a) - \widehat{Q}(s, a) \right) \cdot f_k(s, a)$$

The same holds for U, so we have $\theta_k = \theta_k + \alpha \left(U(s) - \widehat{U}(s) \right) \cdot f_k(s, a)$ which yields:

$$\theta_0 = \theta_0 + \alpha \left(U(s) - \widehat{U}(s) \right) \cdot 1$$

$$\theta_1 = \theta_1 + \alpha \left(U(s) - \widehat{U}(s) \right) \cdot x$$

$$\theta_2 = \theta_2 + \alpha \left(U(s) - \widehat{U}(s) \right) \cdot y$$

$$\theta_3 = \theta_3 + \alpha \left(U(s) - \widehat{U}(s) \right) \cdot \sqrt{(x - x_g)^2 + (y - y_g)^2}$$

Where $U(s) = [R(s, a, s') + \gamma \widehat{U}(s')]$