CS6700 : Reinforcement Learning
Written Assignment #2

Sai Vinay G
CE17B019
Indian Institute of Technology, Madras

March 18, 2019

Problem 1

(a) Given $\alpha = 1, \gamma = 1$

- **Using batch TD(0) :** We here accumulate all the increments according to TD(0) and update the value of states, using average of all the increments for that particular state, after all the episodes are done.

  The increment of value for each state s_t in a batch is

  $$\frac{1}{n(s_t)}\alpha[r_{t+1} + \gamma V(s_{t+1}) - V(s_t)]$$

  In the batch A appears 3 times, B appears 5 times, C appears 5 times.

  - We initialize the values of each state V(A), V(B), V(C) to 0.
  - **First iteration** : average increment of value for state $A = \frac{1}{3}$, for state $B = \frac{3}{5}$, for state $C = \frac{4}{5}$.
  - Values of states after first iteration update are $V(A) = \frac{1}{3}, V(B) = \frac{3}{5}, V(C) = \frac{4}{5}$
  - **Second iteration** : average increment of value for state $A = \frac{11}{15}$, for state $B = 0$, for state $C = 0$.
  - Values of states after second iteration update are $V(A) = \frac{16}{15}, V(B) = \frac{3}{5}, V(C) = \frac{4}{5}$
  - **Third iteration** : average increment of value for state $A = 0$, for state $B = 0$, for state $C = 0$.
  - Therefore the values of states have converged to $\frac{16}{15}, \frac{3}{5}, \frac{4}{5}$

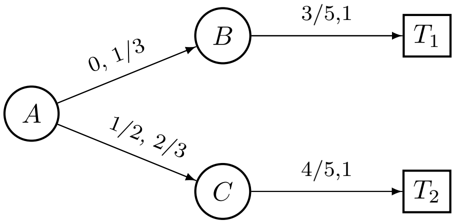
- **Using batch MC :** We here accumulate all the increments according to MC and update the value of states, using average of all the increments for that particular state, after all the episodes are done.

  The increment of value for each state s_t in batch is

  $$\frac{1}{n(s_t)}\alpha[G_t - V(s_t)]$$

- We initialize the values of each state V(A), V(B), V(C) to 0.
- **First iteration** : The average return we get for each states are $\frac{4}{3}, \frac{3}{5}, \frac{4}{5}$
- The values of states after first iteration update are $V(A) = \frac{4}{3}, V(B) = \frac{3}{5}, V(C) = \frac{4}{5}$
- **Second iteration** : average increment of value for state $A = 0$, for state $B = 0$, for state $C = 0$.
- Therefore the values of states have converged to $\frac{4}{3}, \frac{3}{5}, \frac{4}{5}$

(b)



(c) MSE (Mean Squared Error) of the estimates on the train data

TD(0) :
MSE = 0.618
MC :
MSE = 0.550
Based on MSE, Batch Monte carlo is truer to the data than TD(0).

(d) From the model drawn in part (b)
$V(A) = \frac{1}{3}[0 + V(B)] + \frac{2}{3}[\frac{1}{2} + V(C)]$
$V(B) = [\frac{3}{5} + 0]$
$V(C) = [\frac{4}{5} + 0]$
Solving we get $V(A) = \frac{16}{15}, V(B) = \frac{3}{5}, V(C) = \frac{4}{5}$

Same as the once we got in batch TD(0)
Therefore TD(0) is truer to the model.

(e) TD(0) would low error than Monte carlo on future data.
Initially TD(0) has bias decrease asymptotically as the number of samples goes up. Generally, in practice TD methods have shown to converge quickly than MC methods.

Problem 2

(a) Given,
Control actions are delayed i.e, the control agent takes an action on observing the state at time t and The action is applied to the system at time $t + \tau$.
The agent receives a reward at each time step.

In MDP state transition takes place, unit time after an action is applied to the system.i.e, $t(s') = t(a) + 1$, where s' is the state that is obtained by the application of action a on system.

The reward (r) that we get depends on the current state (s'), the action (a) which lead us to this state and the previous state(s).

Here as the actions application is delayed, and the rewards are given at each step,so we get rewards depending on past actions and state transitions.

Let $\tau = k + \beta$, where k is a whole number and $0 < \beta < 1$.
So the state transition takes after $(k+1) + \beta$ units of time from the instant current state is obtained (i.e, $t + (k+1) + \beta$).
So, all the time steps present between t and $t + (k+1) + \beta$ receive the same reward as their is no state transition.

As, the return becomes complex, lets consider $\beta = 0$
Let's say we start at time t. We don't rewards until we make a transition.
Now return will be :

$$G_t = k * R(s_t, a_{t-\tau-1}, s_{t-\tau-1}) + (k+1) * \Big\{\gamma R(s_{t+\tau+1}, a_t, s_t) + \gamma^2 R(s_{t+2(\tau+1)}, a_{t+\tau+1}, s_{t+\tau+1}) + ...\Big\}$$

(b) TD(0) backup equation is:

$$V(s_t) \leftarrow V(s_t) + \alpha\Big((k) * R(s_t, a_{t-\tau-1}, s_{t-\tau-1}) + R(s_{t+\tau+1}, a_t, s_t) + \gamma V(s_{t+\tau+1}) - V(s_t)\Big)$$

Problem 3

Given eligibility trace is truncated after 3 steps and each state is visited only once in a trajectory.

If we consider update of any state, it appears **4 times** in a trajectory. Once in the current step when we are present in the state and in the next 3 states.

We update the value at each time step for all states as:

$$V(s) \leftarrow V(s) + \alpha\gamma e_t(s)$$

Where,

$$e_t(s) = \begin{cases} \gamma\lambda e_{t-1}(s) & \text{if } s \in [s_{t-3}, s_{t-2}, s_{t-1}] \\ 0, & \text{if } s = s_t \\ 1, & \text{otherwise} \end{cases}$$

Consider the total update of state s_t :

$$\begin{aligned} V(s_t) \leftarrow V(s_t) + \alpha\Big\{ & R_t + \gamma V(s_{t+1}) - V(s_t) + \gamma\lambda\{R_{t+1} + \gamma V(s_{t+2}) - V(s_{t+1})\} + (\gamma\lambda)^2\{R_{t+2} + \gamma V(s_{t+3}) - V(s_{t+2})\} \\ & + (\gamma\lambda)^3\{R_{t+3} + \gamma V(s_{t+4}) - V(s_{t+4})\}\Big\} \end{aligned}$$

$$\begin{aligned}V(s_t) \leftarrow V(s_t) + \alpha\Big\{ & R_t + (1-\lambda)\{\gamma V(s_{t+1})\} + \gamma\lambda R_{t+1} + (1-\lambda)\lambda^1\{\gamma^2 V(s_{t+2})\} + (\gamma\lambda)^2 R_{t+2} + (1-\lambda)\lambda^2\{\gamma^3 V(s_{t+3})\} \\ & + (\gamma\lambda)^3 R_{t+3} + (1-\lambda)\lambda^3\{\gamma^4 V(s_{t+4})\}\} + \lambda^4\{\gamma^4 V(s_{t+4})\} - V(s_t)\Big\}\end{aligned}$$

Now to express this in terms of some variant of lambda return, we express R_t, R_{t+1}, R_{t+2} as

$$\begin{aligned}R_t &= R_t(1-\lambda) + R_t(1-\lambda)\lambda + R_t(1-\lambda)\lambda^2 + R_t(1-\lambda)\lambda^3 + R_t\lambda^4 \\ R_{t+1} &= R_{t+1}(1-\lambda) + R_{t+1}(1-\lambda)\lambda + R_{t+1}(1-\lambda)\lambda^2 + R_{t+1}\lambda^3 \\ R_{t+2} &= R_{t+2}(1-\lambda) + R_{t+2}(1-\lambda)\lambda + R_{t+2}\lambda^2\end{aligned}$$

$$\begin{aligned}V(s_t) \leftarrow V(s_t) + \alpha\Bigg\{(1-\lambda)\Big\{ & \{\lambda^0\{R_t + \gamma V(s_{t+1})\} \\ & + \lambda^1\{R_t + \gamma R_{t+1} + \gamma^2 V(s_{t+2})\} \\ & + \lambda^2\{R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \gamma^3 V(s_{t+3})\} \\ & + \lambda^3\{R_t + \gamma R_{t+1} + \gamma^2 R_{t+2}\gamma^3 R_{t+3} + \gamma^4 V(s_{t+4})\}\Big\} \\ & + \lambda^4\{R_t + \gamma R_{t+1} + \gamma^2 R_{t+2} + \gamma^3 R_{t+3} + \gamma^4 V(s_{t+4})\} - V(s_t)\Bigg\}\end{aligned}$$

Therefore Equivalent lambda return for 3 steps truncated case is

$$(G_t^\lambda)_{eq} = (1-\lambda)\sum_{n=1}^{4} \lambda^{n-1} G_{t:t+n} + \lambda^4 G_{t:t+4}$$

In the case where eligibility trace is truncated after n steps

$$(G_t^\lambda)_{eq} = (1-\lambda)\sum_{n=1}^{n+1} \lambda^{n-1} G_{t:t+n} + \lambda^{n+1} G_{t:t+n+1}$$

Problem 4

Given,
The system is not perfectly markov.So, we cannot assume memory less property
Now the transition probabilities and rewards depend on previous actions and previous states.

In this case the transitions and rewards are more constrained, as their is an dependence of transition probability and rewards on all the previous states and actions taken.

Consider for example, we start at s_0, take actions to make trajectories.
As we proceed, in the upcoming states, we are more constrained as the dependence goes on increasing.

So, the exploratory behaviour decreases gradually as we play further i.e, many states which are away from initial states or which not obtained from initial states are explored very less, not don't appear in episodes.

So, TD methods don't perform well on systems which are not perfectly markov.

Problem 5

Given grid-world with 4 states denoted by S,S1,S2,* and 2 terminal states denoted by T1, T2
Reward for terminating in T1 is +10 and for terminating in T2 is +5. Any transition into the state marked has a reward of $a \in \mathbb{R}$. All other transitions have a reward of 0.

		S		
T2	*	S1	S2	T1

Now consider all valid movements at each states.

		↓		
T2	← →	← →	← →	T1

We here neglect the upward movement at S because, it would again reach the same state then the policy won't be optimal in case of discounted reward.

So, in total we have **8 policies** possible, of which we need to find the Blackwell optimal policies.
Consider the policies one by one.

1.

		↓		
T2	←	←	←	T1

We get $V(S) = a\gamma + 5\gamma^2, V(S1) = a + 5\gamma, V(S2) = a\gamma + 5\gamma^2, V(*) = 5$

2.

		↓		
T2	←	→	→	T1

We get $V(S) = 10\gamma^2, V(S1) = 10\gamma, V(S2) = 10, V(*) = 5$

3.

		↓		
T2	→	←	←	T1

We get $V(S) = \frac{a\gamma}{1-\gamma^2}, V(S1) = \frac{a}{1-\gamma^2}, V(S2) = \frac{a\gamma}{1-\gamma^2}, V(*) = \frac{a\gamma}{1-\gamma^2}$

4.

		↓		
T2	→	←	→	T1

We get $V(S) = \frac{a\gamma}{1-\gamma^2}, V(S1) = \frac{a}{1-\gamma^2}, V(S2) = 10, V(*) = \frac{a\gamma}{1-\gamma^2}$

5.

		↓		
T2	←	→	←	T1

We get $V(S) = 0, V(S1) = 0, V(S2) = 0, V(*) = 5$

6.

		↓		
T2	←	←	→	T1

We get $V(S) = a\gamma + 5\gamma^2, V(S1) = a + 5\gamma, V(S2) = 10, V(*) = 5$

7.

		↓		
T2	→	→	→	T1

We get $V(S) = 10\gamma^2, V(S1) = 10\gamma, V(S2) = 10, V(*) = 10\gamma^2$

8.

		↓		
T2	→	→	←	T1

We get $V(S) = 0, V(S1) = 0, V(S2) = 0, V(*) = 0$

Value functions				
	V(S)	V(S1)	V(S2)	V(*)
$\pi 1$	a$\gamma + 5\gamma^2$	a + 5γ	a$\gamma + 5\gamma^2$	5
$\pi 2$	10γ^2	10γ	10	5
$\pi 3$	$\frac{a\gamma}{1-\gamma^2}$	$\frac{a}{1-\gamma^2}$	$\frac{a\gamma}{1-\gamma^2}$	$\frac{a\gamma}{1-\gamma^2}$
$\pi 4$	$\frac{a\gamma}{1-\gamma^2}$	$\frac{a}{1-\gamma^2}$	10	$\frac{a\gamma}{1-\gamma^2}$
$\pi 5$	0	0	0	5
$\pi 6$	a$\gamma + 5\gamma^2$	a + 5γ	10	5
$\pi 7$	10γ^2	10γ	10	10γ^2
$\pi 8$	0	0	0	0

If a policy πi is optimal, then $V_{\pi i}(s) \geq maxV_{\pi j}(s)$

- $\pi 5, \pi 8$ there is no black optimal for any k.
- $\pi 7$ is optimal for $a < \frac{5}{\sqrt{2}}, \gamma > \frac{1}{\sqrt{2}} = k$.
- For $\pi 2$ there is no such k for which it is black optimal, as there is bound for γ.
- $\pi 3$ is optimal for $a > 0, \gamma > \frac{-a+\sqrt{a^2+400}}{20} = k$.
- For $\pi 4$ there is no such k for which it is black optimal.
- For $\pi 1, \pi 6$ there is no black optimal for any k.

Problem 6

Given,
The problem dynamics change after every K steps and the cycle repeats every $M * K$ steps.

So this is like having M different problems with same state set and action set, different transition probabilities and reward functions.

So, to keep track of the problem, we can maintain a variable N, which denotes the numbers steps taken till now.
We can represent this problem as :

$$< S, A, \Pr(s'|s,a), R(s',a,s), \gamma, (N/K)\%M >$$

The $(N/K)\%M$ term, helps us know in which particular dynamics of the system we currently are in (where (N/K) is integer division).

After identifying in which type of system we are, we can have look up tables containing the value functions of all the states. So, we can now work on this problem as normal markov problem.

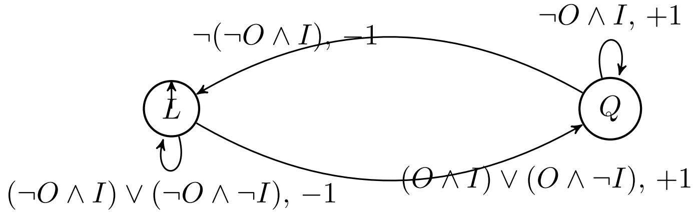
Problem 7

Q-learning is an off policy method, which learns optimal policy, following an exploratory policy for generating trajectories. So, we can use importance sampling to make Q learning on-policy i.e, using target policy as the desired policy.

For learning value function of arbitrary policy while following optimal policy, we take an on-policy method and generate trajectories. Using these trajectories we update the q values of the the arbitrary policy based on weighted estimate of importance sampling and given the condition that the optimal policy must be stochastic where ever arbitrary policy is stochastic or else the importance becomes infinity.

Problem 8

(a) In this problem State set = { Laughter : L , Quiet : Q }
Action set = $\{O \wedge I, O \wedge \neg I, \neg O \wedge I, \neg O \wedge \neg I\}$ where O is playing organ pipe, I is burning incense
Discount factor = 0.9
State transitions and rewards are :


$\neg(\neg O \wedge I)$, -1
$\neg O \wedge I$, $+1$
L
Q
$(\neg O \wedge I) \vee (\neg O \wedge \neg I)$, -1
$(O \wedge I) \vee (O \wedge \neg I)$, $+1$

(b) **Policy iteration :**
Let $\theta = 0.7\ and\ \gamma = 0.9(given)$

Initialize $V(L) = V(Q) = 0 and \pi(L) = \pi(Q) = \neg O \wedge I$

- Evaluation :
  $\Delta = 0$
  $V(L) = P_{LL}^{\pi(L)} * [R_{LL}^{\pi(L)} + V(L)] + P_{LQ}^{\pi(L)} * [R_{LQ}^{\pi(L)} + V(Q)] = 1 * (1 + 0.9 * 0) + 0 = 1$
  $V(Q) = P_{QL}^{\pi(Q)} * [R_{QL}^{\pi(Q)} + V(L)] + P_{QQ}^{\pi(Q)} * [R_{QQ}^{\pi(Q)} + V(Q)] = 0 + 1 * (1 + 0.9 * 0) = +1$
  $\Delta = 1$
  $V(L) = P_{LL}^{\pi(L)} * [R_{LL}^{\pi(L)} + V(L)] + P_{LQ}^{\pi(L)} * [R_{LQ}^{\pi(L)} + V(Q)] = -1.9$
  $V(Q) = P_{QL}^{\pi(Q)} * [R_{QL}^{\pi(Q)} + V(L)] + P_{QQ}^{\pi(Q)} * [R_{QQ}^{\pi(Q)} + V(Q)] = +1.9$
  $\Delta = 0.9$
  $V(L) = P_{LL}^{\pi(L)} * [R_{LL}^{\pi(L)} + V(L)] + P_{LQ}^{\pi(L)} * [R_{LQ}^{\pi(L)} + V(Q)] = -2.71$
  $V(Q) = P_{QL}^{\pi(Q)} * [R_{QL}^{\pi(Q)} + V(L)] + P_{QQ}^{\pi(Q)} * [R_{QQ}^{\pi(Q)} + V(Q)] = +2.71$
  $\Delta = 0.81$
  $V(L) = P_{LL}^{\pi(L)} * [R_{LL}^{\pi(L)} + V(L)] + P_{LQ}^{\pi(L)} * [R_{LQ}^{\pi(L)} + V(Q)] = -3.44$
  $V(Q) = P_{QL}^{\pi(Q)} * [R_{QL}^{\pi(Q)} + V(L)] + P_{QQ}^{\pi(Q)} * [R_{QQ}^{\pi(Q)} + V(Q)] = +3.44$
  $\Delta = 0.729$
  $V(L) = P_{LL}^{\pi(L)} * [R_{LL}^{\pi(L)} + V(L)] + P_{LQ}^{\pi(L)} * [R_{LQ}^{\pi(L)} + V(Q)] = -4.096$
  $V(Q) = P_{QL}^{\pi(Q)} * [R_{QL}^{\pi(Q)} + V(L)] + P_{QQ}^{\pi(Q)} * [R_{QQ}^{\pi(Q)} + V(Q)] = +4.096$
  $\Delta = 0.656$
  Therefore $\Delta < 0.7$ we stop evaluation

- Improvement:
  We see which action from a particular state gives maximum reward.
  $\pi(L) = (O \wedge I) \vee (O \wedge \neg I)$
  $\pi(Q) = \neg O \wedge I$

- Evaluation :
  $\Delta = 0$
  $V(L) = P_{LL}^{\pi(L)} * [R_{LL}^{\pi(L)} + V(L)] + P_{LQ}^{\pi(L)} * [R_{LQ}^{\pi(L)} + V(Q)] = +4.685$
  $V(Q) = P_{QL}^{\pi(Q)} * [R_{QL}^{\pi(Q)} + V(L)] + P_{QQ}^{\pi(Q)} * [R_{QQ}^{\pi(Q)} + V(Q)] = +4.6856$
  $\Delta = 0.5905$
  Therefore $\Delta < 0.7$, we stop evaluation

- Improvement:
  We see which action from a particular state gives maximum reward.
  $\pi(L) = (O \wedge I) \vee (O \wedge \neg I)$
  $\pi(Q) = \neg O \wedge I$

- No change in policy $\implies$ an optimal policy.

Value iteration :
Let $\theta = 0.9$ *and* $\gamma = 0.9(given)$
Initialize $V(L) = V(Q) = 0 and \pi(L) = \pi(Q) = \neg O \wedge I$

$\Delta = 0$
$V(L) = max_a\{P_{LL}^a * [R_{LL}^a + V(L)] + P_{LQ}^a * [R_{LQ}^a + V(Q)]\}$
We get V(L) = +1, for a = $(O \wedge I) \vee (O \wedge \neg I)$
$V(Q) = max_a\{P_{QL}^a * [R_{QL}^a + V(L)] + P_{QQ}^a * [R_{QQ}^a + V(Q)]\}$
We get V(Q) = +1, for a = $\neg O \wedge I$
$\Delta = 1$

$V(L) = max_a\{P^a_{LL} * [R^a_{LL} + V(L)] + P^a_{LQ} * [R^a_{LQ} + V(Q)]\}$
We get V(L) = +1.9, for a = $(O \wedge I) \vee (O \wedge \neg I)$
$V(Q) = max_a\{P^a_{QL} * [R^a_{QL} + V(L)] + P^a_{QQ} * [R^a_{QQ} + V(Q)]\}$
We get V(Q) = +1.9, for a = $\neg O \wedge I$
$\Delta = 0.9$
$V(L) = max_a\{P^a_{LL} * [R^a_{LL} + V(L)] + P^a_{LQ} * [R^a_{LQ} + V(Q)]\}$
We get V(L) = +2.71, for a = $(O \wedge I) \vee (O \wedge \neg I)$
$V(Q) = max_a\{P^a_{QL} * [R^a_{QL} + V(L)] + P^a_{QQ} * [R^a_{QQ} + V(Q)]\}$
We get V(Q) = +2.71, for a = $\neg O \wedge I$
$\Delta = 0.81$

The optimal policy :
$\pi(L) = (O \wedge I) \vee (O \wedge \neg I)$
$\pi(Q) = \neg O \wedge I$

(c) There are two possibilities:
1.Taking optimal action initially and following optimal policy we get $1+\gamma(1)+\gamma^2(1)+... = +1*\frac{1}{1-\gamma} = +10 \quad (\gamma = 0.9)$
2.Taking sub optimal action initially and following optimal policy we get $-1 + \gamma(1) + \gamma^2(1) + ... = -1 + \gamma * \frac{1}{1-\gamma} = -1 + (9) = +8 \quad (\gamma = 0.9)$

Optimal state-action values			
State	Action	Next state	q$^*(s,a)$
L	O $\wedge I$	Q	+10
L	O $\wedge \neg I$	Q	+10
L	$\neg O \wedge I$	L	+8
L	$\neg O \wedge \neg I$	L	+8
Q	O $\wedge I$	L	+8
Q	O $\wedge \neg I$	L	+8
Q	$\neg O \wedge I$	Q	+10
Q	$\neg O \wedge \neg I$	Q	+8

(d) At the wits end, we need to follow the optimal policy to keep the room quiet. So, if the room has laughter, play the organ and if the room is quiet burn the incense and don't play the organ.

References

[1] A. dirk (https://stats.stackexchange.com/users/201613/anne dirk), “When are monte carlo methods preferred over temporal difference ones?” Cross Validated, uRL:https://stats.stackexchange.com/q/336974 (version: 2018-09-23). [Online]. Available: https://stats.stackexchange.com/q/336974

[2] R. S. Sutton, A. G. Barto, *et al.*, *Introduction to reinforcement learning.* MIT press Cambridge, 1998, vol. 135.