Reinforcement Learning
Written Assignment #2

Author:
CS17S011 Ajay Kumar Pandey

Submitted to-
Assoc. Professor: B.Ravindran

May 18, 2020

1 Solution: 1

1.1 Part (a)

Eligibility traces combine both Frequency heuristic and Recency heuristic.
Backward view formula for computing linear type of eligibility traces:
$E_0(s) = 0$
$E_t(s) = E_{t-1}(s) - \gamma\lambda + 1(s_t = s)$

1.2 Part (b)

For replacing eligibility traces,
$E_t(s) = e_{t-1}(s) - \gamma\lambda \quad \text{if } s \neq s_t$
$E_t(s) = 1 \quad \text{if } s = s_t$

1.3 part (c)

In geometric decrement the state eligibility decreases very fast as it is decreasing geometrically hence we will not get state instance for very long time, whereas in case of linear decrement it last for some more time.

2 Solution: 2

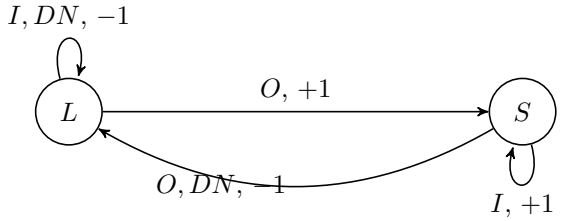
2.1 Part (a)

State-space $S = \{laughter\,(L),\, silent\,(S)\}$

Actions $A = \{playing\,organ\,(O),\, lighting\,incense\,(I)\, Do\,Nothing\,(DN)\}$

Discount factor $\gamma = 0.9$.

The state transition diagram is given below.



2.2 Part (b)

2.2.1 Policy iteration

- Let us assume an initial estimate of value function, $V_0 = \{L : 0, S : 0\}$.
- Initial Policy given as, $\pi_0 = \{L : I,\, S : I\}$.
- $p_{\pi_0} = \begin{matrix} 1 & 0 \\ 0 & 1 \end{matrix} \cdot r_{\pi_0} = \begin{matrix} -1 \\ +1 \end{matrix}$ $V_{\pi_0} = (I - \gamma p_{\pi_0})^{-1} r_{\pi_0} = \begin{matrix} -10 \\ 10 \end{matrix}$
  The state value function in policy evaluation step forming a infinite gp (sum of G.P with $a = 1, r = 0.9$).
- Now, $\pi_1(L) = argmax_a\{O :\, 1 + 0.9(10), I : -1 + 0.9(-10)\} = argmax_a\{O : 10, I : -10\} = O$
  $\pi_1(S) = argmax_a\{O :\, -1 + 0.9(-10), I : 1 + 0.9(10)\} = argmax_a\{O : -10, I : 10\} = I$
  Therefore, $\pi_1 = \{L : O,\, S : I\}$.
- $p_{\pi_1} = \begin{matrix} 0 & 1 \\ 0 & 1 \end{matrix} \cdot r_{\pi_1} = \begin{matrix} +1 \\ +1 \end{matrix}$ $V_{\pi_1} = (I - \gamma p_{\pi_1})^{-1} r_{\pi_1} = \begin{matrix} 10 \\ 10 \end{matrix}$
- Now, $\pi_2(L) = argmax_a\{O :\, 1 + 0.9(10), I : -1 + 0.9(10)\} = argmax_a\{O : 10, I : 8\} = O$
  $\pi_2(S) = argmax_a\{O :\, -1 + 0.9(10), I : 1 + 0.9(10)\} = argmax_a\{O : 8, I : 10\} = I$
  Therefore, $\pi_2 = \{L : O,\, S : I\}$.
- $\pi_1 = \pi_2$. Thereforce, policy iteration converged to the optimal policy $\pi^* = \{L : O,\, S : I\}$.
- And optimal value function $V^* = V_{\pi_1} = \{L : 10, S : 10\}$.

2.2.2 Value iteration

- Let us assume an initial estimate of value function, $V_0 = \{L : 0, S : 0\}$.
- Now, $V_1(L) = max_a\{O : 1 + 0.9(0), I : -1 + 0.9(0)\} = 1$.
  $V_1(S) = max_a\{O : -1 + 0.9(0), I : 1 + 0.9(0)\} = 1$.
- $V_2(L) = max_a\{O : 1 + 0.9(1), I : -1 + 0.9(1)\} = 1.9$.
  $V_2(S) = max_a\{O : -1 + 0.9(1), I : 1 + 0.9(1)\} = 1.9$.
- $V_3(L) = max_a\{O : 1 + 0.9(1.9), I : -1 + 0.9(1.9)\} = 2.71$.
  $V_3(S) = max_a\{O : -1 + 0.9(1.9), I : 1 + 0.9(1.9)\} = 2.71$.
- $V_4(L) = max_a\{O : 1 + 0.9(2.71), I : -1 + 0.9(2.71)\} = 3.439$.
  $V_4(S) = max_a\{O : -1 + 0.9(2.71), I : 1 + 0.9(2.71)\} = 3.439$.
- Following the trend, we can observe that V^*converges to $\{L : 10, S : 10\}$ (sum of G.P with $a = 1, r = 0.9$).

2.3 Part (c)

Now we know that $V^* = \{L : 10, S : 10\}$. Knowing this, we can compute $Q^*(s, a)$ with one-step look ahead at V^*.

$Q^*(L, O) = 1 + 0.9V^*(S) = 10$
$Q^*(L, I) = -1 + 0.9V^*(S) = 8$
$Q^*(S, O) = -1 + 0.9V^*(L) = 8$
$Q^*(S, I) = 1 + 0.9V^*(L) = 10$

2.4 Part (d)

At present it is Laughing state, To come to silent state one should follow the optimal policy. Play the organ then house become silent then burn the incense and continue it burning the house will remain silent forever.

3 Solution: 3

Monte-Carlo and Policy gradient will work on Non Markovian problem setup also, whereas TD learning better work on Markovian scenario.
MC must wait until end of episode before return is known. MC only works for episodic (terminating) environments.MC does not exploit Markov property. Usually more effective in non-Markov environments.
TD can learn online after every step.TD can learn from incomplete sequences.TD works in continuing (non-terminating) environments. TD exploits Markov property. Usually more efficient in Markov environments.

4 Solution: 4

For the given grid-world we have, State set $= \{S, P, Q, T1, T2\}$.

		S		
T2	*	P	Q	T1

It is mentioned in the question for terminating on state $T1$ will get reward $+10$ and on terminating in state $T2$ will get $+5$, For state $*$ is a units reward.
Below transition diagram shows all possible actions that can be taken on any state. there are 2 actions left and right in all three states P,Q and $*$ total possibility of policies are 8. Now we calculate value function on all 4 states on 8 policies. choose policy which is better than all.

		$\downarrow$		
$T2$	$\leftarrow$ $\rightarrow$	$\leftarrow$ $\rightarrow$	$\leftarrow$ $\rightarrow$	$T1$

4.1 Possibility 1

		$\downarrow$		
$T2$	$\leftarrow$	$\leftarrow$	$\leftarrow$	$T1$

$S = a\gamma + 5\gamma^2 \quad Q = a\gamma + 5\gamma^2 \quad P = a + 5\gamma \quad * = 5$

4.2 Possibility 2

		↓		
$T2$	←	←	→	$T1$

$S = a\gamma + 5\gamma^2 \quad Q = 10 \quad P = a + 5\gamma \quad * = 5$

4.3 Possibility 3

		↓		
$T2$	←	→	←	$T1$

$S = 0 \quad Q = 0 \quad P = 0 \quad * = 5$

4.4 Possibility 4

		↓		
$T2$	←	→	→	$T1$

$S = 10\gamma^2 \quad Q = 10 \quad P = 10\gamma \quad * = 5$

4.5 Possibility 5

		↓		
$T2$	→	←	←	$T1$

$S = \frac{a\gamma^3}{1-\gamma^2} \quad Q = \frac{a\gamma^3}{1-\gamma^2} \quad P = \frac{a\gamma^2}{1-\gamma^2} \quad * = \frac{a\gamma}{1-\gamma^2}$

4.6 Possibility 6

		↓		
$T2$ →	←	→	$T1$	

$S = \frac{a\gamma^3}{1-\gamma^2} \quad Q = 10 \quad P = \frac{a\gamma^2}{1-\gamma^2} \quad * = \frac{a\gamma}{1-\gamma^2}$

4.7 Possibility 7

		↓		
$T2$	→	→	←	$T1$

$S = 0 \quad Q = 0 \quad P = 0 \quad * = 0$

4.8 Possibility 8

		↓		
$T2$	→	→	→	$T1$

$S = 10\gamma^2 \quad Q = 10 \quad P = 10\gamma \quad * = 10\gamma^2$

Conclusion

For optimal policy π_i , $V_{\pi_i}(s) \geq max\, V_{\pi_j}(s)$ for $i \neq j$ for all states s. Use optimal policy rule for each policy, we get the following results.

- Possibility 1 and 2 can never be optimal for any λ.
- Possibility 4 and 8 will be optimal for a≤ 0.
- Policy 5 and 6 will be optimal for $\lambda > \frac{\sqrt{a^2+400}-a}{20}$.
- Policy 3 and 7 can never be optimal for any λ.

5 Solution: 5

Egocentric: Self to neighbor, Represents the location of agent in state space relative to its neighbor only.

- Assume state configuration in some group repeated in the state space then in this scenario Egocentric RL agent take advantage to learn the problem faster,
- In egocentric state representation memory space can also be saved as the state space will reduced.
- Where as similar scenario may be confusing also to the agent. Because it only relate things to its neighbor it don't know what will happen after some steps.
- In Egocentric representation, Generalization is more difficult as compare to normal RL state space scenario.

6 Solution: 6

There are some drawbacks in DP algorithm as when number of states are increases its complexity increases too much. To overcome this problem we use RTDP Algorithm which give preference to frequently used states. If we take advantage of symmetries ,it will account into less computation for solving the problem.
In RTDP we need a technique for sampling strategy on the basis of symmetry which take less time and tends to better algorithm than exists.

Algorithm

We make groups of similar states based on the symmetries.

- Initialize $Q(s_t, a_t) = 0 \; \forall$ s $\forall$ a, for time t.
- Sample a trajectory following a policy π.
- At each time step t (s_t, a_t) in the trajectory:
  - if (s_t, a_t) seen earlier that is exist in any symmetric group. we leave it and start the computation from fresh (s_{t+1}, a_{t+1})
  - else sample it till we get terminal state.
- Update all the states that are in symmetric group with the same value. $Q(s_t, a_t) = Q(s'_t, a'_t)$ if s and s' are in same symmetric group.
- Other states have already there unique values from sampling. $Q(s, a) = R(s, a) + \Sigma\gamma P'(s, a, s')max_a{}'(s', a')$

Advantages

- If RTDP need MDP in memory it will take less memory as we group out all symmetric states.
- It is faster also as the state space is reduced.
- This algorithm converge faster also as the update step, updates similar state at same time step.

7 Solution: 7

We will treat it as separate M MDP problems with each of size K.

7.1 Problem representation

$S = s_1, s_2, ..., s_m$ where s_i represent state space of i^th MDP.
$A = a_1, a_2, ..., a_m$ where a_i represent action set of i^th MDP. Similarly Rewards and probability also defined. The set is defined S,A,R for all M problems.

8 Solution: 8

8.1 Part (a)

For the calculation of state values in First visit Monte Carlo we take first time vist of state in any episode and calculate its return in that episode. state values .

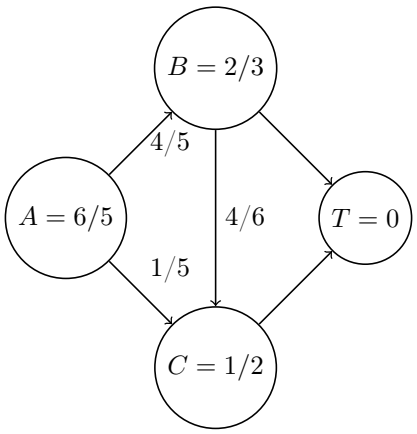
$$V(A) = \frac{(3) + (1) + (0) + (1) + (1)}{5} = 6/5$$

$$V(B) = \frac{(2) + (0) + (1) + (0) + (0) + (1)}{6} = 4/6$$

$$V(C) = \frac{(1) + (0) + (0) + (0) + (1) + (1)}{6} = 3/6$$

8.2 Part (b)

The state-transition diagram is shown below.State include there reward and on arrows transition probability mentioned.



For calculating transition probabilities consider all episodes and average the no of transitions A->B , A->C, B->C, B->T, C->T. For reward calculation consider episode states reward and average them.

8.3 Part (c)

Batch $TD(0)$ converged state values are calculated from MDP. BY solving simultaneous state value equations of each state. $V(T) = 0$. Based on transition probabilities,

$V(A) = 6/5 + 4/5V(B) + 1/5V(C)$

$V(B) = 2/3 + 4/6V(C)$

$V(C) = 1/2$

Solving the above system of equations, we get,

State s	Value $V(s)$
A	21/10
B	1
C	1/2
T	0

9 Solution: 9

Yes we can learn the value function of an arbitrary policy using the optimal policy. We use SARSA or Monte Carlo as these are on-policy method and generate sample trajectory from them to calculate arbitrary policy.

- Be sure Q-values that are estimated must belong to arbitrary policy. For this we can use importance sampling to weigh the updates.
- This method have high variance so we can use bias in important sampling to reduce it.
- Optimal policy should cover the arbitrary police then only we can learn the value function.
- Optimal policy must be stochastic wherever arbitrary policy is stochastic.

10 Solution: 10

Reference: RL course Book

In question it is asked for a bandit problem in which the parameters on which the policy depends are the preference of the actions and the action selection probabilities which are determined by following softmax relationship-

$$\pi_t(a_j) = \frac{e^{p_t(a_j)}}{\sum_{i=1}^{n} e^{p_t(a_i)}}$$

where n is the total number of actions and $p_t(a)$ is the preference value of action a at time t.
The REINFORCE update equation is:

$$\theta_{t+1} \leftarrow \theta_t + \alpha_t(r_t - b_t)\nabla_{p_t} log(\pi_t(a_t))\frac{\partial p_t}{\partial \theta_t}$$

where relation of b_t and b_{t+1} is given by $b_{t+1} = b_t + \beta(r_t - b_t)$.

solving both the equation we get,

$$\theta_{t+1} \leftarrow \theta_t + \alpha_t(r_t - b_t)\{1 - \pi_t(a_t)\}\frac{\partial p_t}{\partial \theta_t}$$

11 Solution 11

Reference: RL course book

Now for the same bandit problem we have parameters as the mean μ and variance σ^2 of the normal distribution. Action are chosen according these parameters and the baseline is considered here $b_t = 0$..

$$\pi_t(a; \mu_t, \sigma_t) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(a-\mu_t)^2}{\sigma_t^2}}$$

The REINFORCE update equation is:

$$\theta_{t+1} \leftarrow \theta_t + \alpha_t(r_t - b_t)\nabla_{\theta_t} log(\pi_t(a_t))$$

After considering θ_t as $\begin{matrix}\mu_t \\ \sigma_t\end{matrix}$.
We get separate equation for mean $\mu_t t + 1$ and for σ_{t+1}

$$\mu_{t+1} \leftarrow \mu_t + \alpha_t r_t(a_t - \mu_t)$$

$$\sigma_{t+1} \leftarrow \sigma_t + \frac{\alpha_t r_t\{(a_t - \mu_t)^2 - \sigma_t^2\}}{\sigma_t}$$