# TIC TAC TOE

Enemy - X
Agent - O

## Question ①

Let $S$ be the set of states.

∴ Each element of state $S$ is a 3x3 matrix. (represented using list of numpy array)

$$S = \{s_t\}_{t=1}^{n}$$

$$s_t = \begin{bmatrix} s_{00} & s_{01} & s_{02} \\ s_{10} & s_{11} & s_{12} \\ s_{20} & s_{21} & s_{22} \end{bmatrix} \quad \text{such that} \quad s_{ij} = \begin{cases} -1 & \text{enemy move} \\ +1 & \text{agent move} \\ 0 & \text{empty} \end{cases}$$

In addition, for all non-terminal states $\sum_{i=0}^{2} \sum_{j=0}^{2} s_{ij} = -1$

for terminal states
$$\begin{cases} \sum_{i=0}^{2} \sum_{j=0}^{2} s_{ij} = 0 & \text{if agent wins} \\ \sum_{i=0}^{2} \sum_{j=0}^{2} s_{ij} = -1 & \text{if enemy wins or draw.} \end{cases}$$

→ Each state has been also represented in a python string format

String Representation of a random ~~any~~ state = "_ _ _ _ _ X X O _ _"

python string

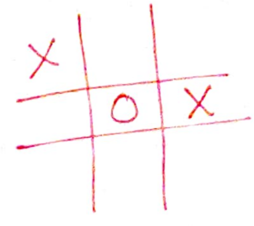
'_' represents empty cell / $s_{ij} = 0$

'x' represents enemy cell / $s_{ij} = -1$

'o' represents agent or player cell / $s_{ij} = 1$

For example,

$$s_t = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 0 \end{bmatrix} = \text{"X - - - O X - - -"} =$$



Action set = {0, 1, 2, 3, 4, 5, 6, 7, 8}

$\underbrace{\text{Action} = i}$ $\quad 0 \leq i \leq 8$ and $i \in \mathbb{Z}$ (set of integers)

↳ corresponds to putting 'o' in the ith location of the string state

For example, Action = 4 ⇒ "-----xox-" (Initial state)

↓ Action = 4

"----oxox-" (state after performing action = 4)

## 5 Terminal States where player/agent wins

$S_t =$

| | | |
|---|---|---|
| 1 | 1 | 1 |
| 0 | -1 | 0 |
| -1 | 0 | -1 |

≡ "ooo-x-x-x"

$S_t =$

| | | |
|---|---|---|
| 1 | 0 | -1 |
| -1 | 1 | 0 |
| -1 | 0 | 1 |

≡ "o-xxo-x-o"

$S_t =$

| | | |
|---|---|---|
| 1 | 1 | 1 |
| -1 | 1 | -1 |
| 0 | -1 | -1 |

≡ "oooxox-xx"

$S_t =$

| | | |
|---|---|---|
| 1 | 0 | -1 |
| 1 | -1 | -1 |
| 1 | 0 | 0 |

≡ "o-xoxxo--"

$S_t =$

| | | |
|---|---|---|
| -1 | 1 | 0 |
| -1 | 1 | -1 |
| 1 | 1 | -1 |

≡ "xo-xoxoox"

Question 5

Initial game state

$S_0$

| | | |
|---|---|---|
| X | | X |
| O | O | |
| X | | |

Reward function

$$R(s) = \begin{cases} +10 & \text{player wins} \\ -10 & \text{player loses} \\ 0 & \text{draw} \\ c & \text{else} \end{cases}$$

Come up with 2 different $c_1$ & $c_2$

$c_1 < 0$, $c_2 > 0$

$$\boxed{\begin{array}{l} X = -1 \\ O = +1 \\ \text{empty} = 0 \end{array}}$$

$$V^{\pi^*}(s_0) = c + \gamma \max_{a \in A(s_0)} \sum_{s'} P(s' | s_0, a) V^{\pi^*}(s')$$

$$= c + \gamma \max [A_1, A_2, A_3, A_4]$$

where
$$A_1 = \sum_{s'} P(s' | s_0, a_{01} = 1) V^{\pi^*}(s')$$
$$A_2 = \sum_{s'} P(s' | s_0, a_{12} = 1) V^{\pi^*}(s')$$
$$A_3 = \sum_{s'} P(s' | s_0, a_{21} = 1) V^{\pi^*}(s')$$
$$A_4 = \sum_{s'} P(s' | s_0, a_{22} = 1) V^{\pi^*}(s')$$

$$A_1 = \frac{1}{3}\left[c + \gamma \max_{a \in A(s_1)} \{P(s_1' | s_1, a_{21} = 1) \cdot 10, P(s_1'' | s_1, a_{22} = 1) \cdot 0\}\right]$$
$$+ \frac{1}{3}\left[c + \gamma \max_{a \in A(s_2)} \{P(s_2' | s_2, a_{12} = 1) \cdot 10, P(s_2'' | s_2, a_{22} = 1) \cdot 0\}\right]$$
$$+ \frac{1}{3}\left[c + \gamma \max_{a \in A(s_3)} \{P(s_3' | s_3, a_{12} = 1) \cdot 10, P(s_3'' | s_3, a_{21} = 1) \cdot 10\}\right]$$

$$= \frac{1}{3}\left[c + \gamma \max(1 \times 10, 1 \times 0)\right] + \frac{1}{3}\left[c + \gamma \max(1 \times 10, 1 \times 0)\right]$$
$$+ \frac{1}{3}\left[c + \gamma \max(1 \times 10, 1 \times 10)\right].$$

$$= \frac{1}{3}(c + 10\gamma) + \frac{1}{3}(c + 10\gamma) + \frac{1}{3}(c + 10\gamma) = (c + 10\gamma)$$

$$A_2 = \frac{1}{3} \times 10 + \frac{1}{3} \times 10 + \frac{1}{3} \times 10 = 10$$

$$A_3 = \frac{1}{3}[-10] + \frac{1}{3}\left[c + \gamma \max\left(1 \times 10,\ 1 \times -10\right)\right] + \frac{1}{3}\left[c + \gamma \max\left(1 \times 10,\ 1 \times 10\right)\right]$$

$$= -\frac{10}{3} + \frac{1}{3}(c + 10\gamma) + \frac{1}{3}(c + 10\gamma)$$

$$= \frac{2}{3}(c + 10\gamma) - \frac{10}{3}$$

$$A_4 = \frac{1}{3}[-10] + \frac{1}{3}\left[c + \gamma \max\left(1 \times 0,\ 1 \times -10\right)\right] + \frac{1}{3}\left[c + \gamma \max\left(1 \times 10,\ 1 \times 0\right)\right]$$

$$= -\frac{10}{3} + \frac{1}{3}c + \frac{1}{3}(c + 10\gamma) = \frac{2c}{3} + \frac{10\gamma}{3} - \frac{10}{3}$$

Now

$$V^{\pi^*}(s_0) = c + \gamma \max(A_1, A_2, A_3, A_4)$$

$$= c + \gamma \max\left[c + 10\gamma,\ 10,\ \frac{2c}{3} + \frac{20\gamma}{3} - \frac{10}{3},\ \frac{2c}{3} + \frac{10\gamma}{3} - \frac{10}{3}\right]$$

Using $\gamma = 0.9$

$$V^{\pi^*}(s_0) = c + \frac{9}{10} \max\left[c + 9,\ 10,\ \frac{2c}{3} + \frac{8}{3},\ \frac{2c}{3} - \frac{1}{3}\right]$$

Let $c_1 = -1 < 0$ and $c_2 = 2 > 0$

$$V^{\pi^*}(s_0)\big|_{c=c_1} = c_1 + \frac{9}{10} \max\left[8,\ 10,\ 2,\ -1\right]$$

corresponds to 2nd action $a_{12} = 1$ at $s_0$.

$$V^{\pi^*}(s_0)\big|_{c=c_2} = c_2 + \frac{9}{10} \max\left[11,\ 10,\ 4,\ 1\right]$$

corresponds to 1st action $a_{01} = 1$ at $s_0$.

Choosing $c_1 = -1$ and $c_2 = 2$ ~~achieves~~ achieves the desired purpose.. and optimal decision/action changes from $a_{12}=1$ to $a_{01}=1$

**Under $c_1 = -1$**

Optimal action at $s_0$ $a_{12}=1$

| | | |
|---|---|---|
| X | | X |
| O | O | O |
| X | | |

**Under $c_2 = 2$**

Optimal action at $s_0$ $a_{01}=1$

| | | |
|---|---|---|
| X | O | X |
| O | O | |
| X | | |

Now,
$$V^{\pi^*}(s_0) = c + \gamma \max_{a \in A(s_0)} \sum_{s'} P(s'|s_0,a)\, V^{\pi^*}(s')$$

$$\therefore V^{\pi^*}(s_0) = \max_{a \in A(s_0)} \underbrace{\left[c + \gamma \sum_{s'} P(s'|s_0,a)\, V^{\pi^*}(s')\right]}_{Q(s_0,a)}$$

**Under $c = c_1 = -1$**

$$Q(s_0, a_{01}=1) = c_1 + \gamma \cdot 8 = 6.2$$

$$Q(s_0, a_{12}=1) = c_1 + \gamma \cdot 10 = 8$$

$$Q(s_0, a_{21}=1) = c_1 + \gamma \cdot \left(\frac{2c_1}{3} + \frac{8}{3}\right) = 0.8$$

$$Q(s_0, a_{22}=1) = c_1 + \gamma \left(\frac{2c_1}{3} - \frac{1}{3}\right) = -1.9$$

**Under $c = c_2 = 2$**

$$Q(s_0, a_{01}=1) = c_2 + \gamma \cdot (c_2 + 9) = 11.9$$

$$Q(s_0, a_{12}=1) = c_2 + \gamma \cdot 10 = 11$$

$$Q(s_0, a_{21}=1) = c_2 + \gamma \left(\frac{2c_2}{3} + \frac{8}{3}\right) = 5.6$$

$$Q(s_0, a_{22}=1) = c_2 + \gamma \left(\frac{2c_2}{3} - \frac{1}{3}\right) = 2.9$$

Question 6

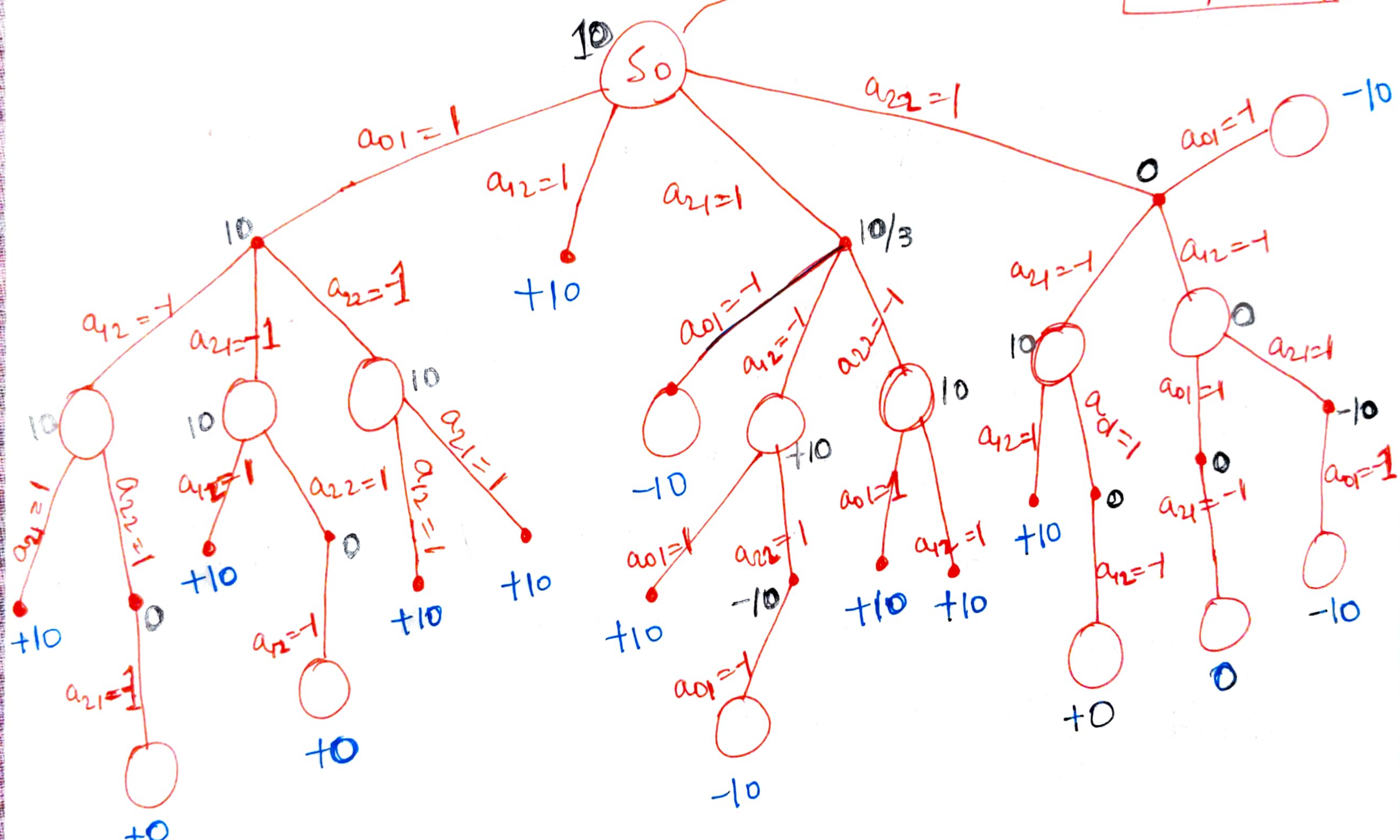
$$a_{ij} = \begin{cases} +1 & \text{represents O (circle)} \\ -1 & \text{represents X (cross)} \\ 0 & \text{represents empty cell} \end{cases}$$

Game state in figure 1

$S_0 =$

| | | |
|---|---|---|
| X | | X |
| O | O | |
| X | | |

| | | |
|---|---|---|
| -1 | 0 | -1 |
| 1 | 1 | 0 |
| -1 | 0 | 0 |




→ Using Expectimax, the player at $S_0$ sees 4 options with respective scores $-10$, $10$, $\frac{10}{3}$, $0$. Therefore, the max player at $S_0$ chooses the path with highest expected score $= +10$.

→ Optimal Actions at $S_0$ = Either of $a_{01} = 1$ or $a_{12} = 1$

In string format representation,

Optimal Actions are $a_{01} = 1$ (or $a = 1$)

and $a_{12} = 1$ (or $a = 5$)

Reason for change of ~~reward~~ optimal action based on reward definition

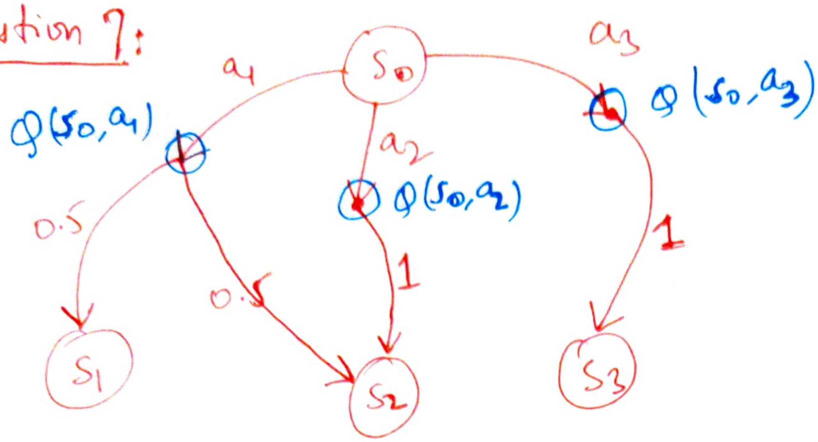
When we assign a low reward value to states, the optimal action should be to finish the game as soon as possible and get to the terminal state. On the other hand, when reward values are high, the optimal actions ~~will~~ may shift towards collecting more non-terminal high reward values before finishing the game.

For example, in Question 5, when $G = 7$, optimal action is $a_{02} = 1$ at $s_0$ in which case the player wins immediately. But when $G = 2$, the optimal action changes to $a_{01} = 1$ at $s_0$. Which does not guarantee a win (either future win or draw). but more expected discounted reward sum compared to the action $a_{12} = 1$.

## ② Function Approximation in RL

$\gamma = 0.5$

**Question 7:**




$S_i = (i+1) \qquad a_i = i$

$\phi(s, a) = \alpha s + \beta a$

$\phi(s_0, a_1) = \alpha + \beta$ —— ①

$\phi(s_0, a_2) = \alpha + 2\beta$ —— ②

$\phi(s_0, a_3) = \alpha + 3\beta$ —— ③

$s_1, s_2, s_3$ — terminal states

$P(s_1 | s_0, a_1) = 0.5 \qquad P(s_2 | s_0, a_2) = 1$

$P(s_2 | s_0, a_1) = 0.5 \qquad P(s_3 | s_0, a_3) = 1$

Let $R(s_0) = R_0$, $R(s_1) = R_1$
Let $R(s_2) = R_2$, $R(s_3) = R_3$
} unknowns.

$$\phi(s_0, a_1) = R(s_0) + \gamma \sum_{s'} P(s' | s_0, a_1)\, V(s') = R_0 + \gamma \left[ P(s_1 | s_0, a_1) V(s_1) + P(s_2 | s_0, a_1) V(s_2) \right]$$

$$= R_0 + \gamma \left[ \frac{1}{2} V(s_1) + \frac{1}{2} V(s_2) \right] = R_0 + \frac{1}{4} \left[ V(s_1) + V(s_2) \right]$$

Since $s_1, s_2, s_3$ are terminal states, $V(s_1) = R_1 \quad V(s_2) = R_2 \quad V(s_3) = R_3$

$\phi(s_0, a_1) = R_0 + \frac{1}{4}[R_1 + R_2]$. —— ④

$\phi(s_0, a_2) = R(s_0) + \gamma \cdot P(s_2 | s_0, a_2)\, V(s_2) = R_0 + \frac{1}{2} R_2$ —— ⑤

$\phi(s_0, a_3) = R(s_0) + \gamma\, P(s_3 | s_0, a_3)\, V(s_3) = R_0 + \frac{1}{2} R_3$. —— ⑥

From ① – ④, ② – ⑤, ③ – ⑥ (In order for linear $\phi$-function to correctly represent $\phi$-values at $s_0$)

$$\alpha + \beta = R_0 + \frac{R_1 + R_2}{4}, \quad \alpha + 2\beta = R_0 + \frac{R_2}{2}, \quad \alpha + 3\beta = R_0 + \frac{R_3}{2}.$$

We have 3 equations and 4 unknowns $(R_0, R_1, R_2, R_3)$.

$$\beta = \frac{R_2 - R_1}{4} = \frac{R_3 - R_2}{2}$$

$$\alpha = R_0 + \frac{R_1}{2}$$

$$\Rightarrow \boxed{R_1 = 2(\alpha - R_0)}$$

$$\Rightarrow R_2 = (4\beta + R_1)$$

$$= (4\beta + 2\alpha - 2R_0)$$

$$\boxed{R_2 = 2(\alpha + 2\beta - R_0)}$$

$$R_3 = 2\beta + R_2$$

$$= 2\beta + 2\alpha + 4\beta - 2R_0$$

$$\boxed{R_3 = 2(\alpha + 3\beta - R_0)}.$$

$$\left.\begin{aligned} R(s_0) &= R_0 \\ R(s_1) &= 2(\alpha - R_0) \\ R(s_2) &= 2(\alpha + 2\beta - R_0) \\ R(s_3) &= 2(\alpha + 3\beta - R_0). \end{aligned}\right\}$$

where $\alpha, \beta \in R$ are weight parameters that can be arbitrarily chosen.

Let $\boxed{\alpha = 2 \text{ and } \beta = 3. \text{ and } R_0 = 1}$.

| | |
|---|---|
| $R(s_0) = 1$ | $R(s_2) = 14$ |
| $R(s_1) = 2$ | $R(s_3) = 20$ |

Q-value function

$$Q(s_0, a_1) = \alpha + \beta = R_0 + \frac{1}{4}[R_1 + R_2] = 5$$

$$Q(s_0, a_2) = \alpha + 2\beta = R_0 + \frac{R_2}{2} = 8$$

$$Q(s_0, a_3) = \alpha + 3\beta = R_0 + \frac{R_3}{2} = 11$$

The above reward function can correctly represents all Q-values by the linear function $Q(s,a) = \alpha s + \beta a$ at state $s_0$.

② Goal: To design a reward function s.t. $Q(s,a) = \alpha s + \beta a$ does not work...

Question 7.2

$$Q(s,a) = \alpha s + \beta a \qquad \text{①}$$

Cannot represent all Q-values at $s_0$ for any $\alpha, \beta \in \mathbb{R}$

| | $Q^L(s,a)$ Linear function | $Q^T(s,a)$ True Value |
|---|---|---|
| $Q(s_0,a_1)$ | $\alpha+\beta$ | $R_0 + \frac{R_1+R_2}{4}$ |
| $Q(s_0,a_2)$ | $\alpha+2\beta$ | $R_0 + \frac{R_2}{2}$ |
| $Q(s_0,a_3)$ | $\alpha+3\beta$ | $R_0 + \frac{R_3}{2}$ |

$Q^T(s,a)$ : True value computed using reward

$Q^L(s,a)$ : computed using linear function

Let reward function be $R(s) = s$

Thus $R(s_0) = 1$ $\quad R(s_1) = 2$ $\quad R(s_2) = 3$ $\quad R(s_3) = 4$.

Then

$$Q^L(s_0,a_1) = \alpha+\beta = \frac{9}{4} = Q^T(s_0,a_1)$$
$$Q^L(s_0,a_2) = \alpha+2\beta = \frac{5}{2} = Q^T(s_0,a_2)$$
$$Q^L(s_0,a_3) = \alpha+3\beta = 3 = Q^T(s_0,a_3)$$

3 eqns. 2 unknowns

$(\alpha,\beta) = \left(\frac{3}{2}, \frac{1}{2}\right) \Rightarrow \frac{3}{2}+\frac{1}{2} = 2 \neq \frac{9}{4}$. (does not satisfy 1st equation)

$(\alpha,\beta) = \left(2, \frac{1}{4}\right) \Rightarrow 2+\frac{3}{4} = \frac{11}{4} \neq 3$

→ Thus, for $R(s) = s$, the above set of linear equations does not have any solution.

→ For any choice of $(\alpha,\beta)$, the linear function $Q^L(s,a) = \alpha s + \beta a$ does not represent all the true values of $Q^T(s,a)$, if $R(s) = s$.

$Q^L(s_0, a_1) = \alpha + \beta$ $\qquad Q^T(s_0, a_1) = \frac{9}{4}$ ——— 1

$Q^L(s_0, a_2) = \alpha + 2\beta$ $\qquad Q^T(s_0, a_2) = \frac{5}{2}$ ——— 2

$Q^L(s_0, a_3) = \alpha + 3\beta$ $\qquad Q^T(s_0, a_3) = 3$ ——— 3

Solving 1 & 2

$\alpha + \beta = \frac{9}{4}$

$\alpha + 2\beta = 5/2$

$\Downarrow$

$\alpha = 2, \quad \beta = \frac{1}{4}$

$\Downarrow$

putting into the 3 equation.

$\alpha + 3\beta = \frac{11}{4} \neq 3 = Q^T(s_0, a_3)$

Solving 2 & 3

$\alpha + 2\beta = \frac{5}{2}$

$\alpha + 3\beta = 3.$

$\Downarrow$

$\alpha = \frac{3}{2} \quad \beta = \frac{1}{2}$

$\Downarrow$

putting into the 1 equation..

$\alpha + \beta = 2 \neq \frac{9}{4} = Q^T(s_0, a_1).$

Similarly solving 1 & 3, $\alpha = \frac{15}{8}, \beta = \frac{3}{8}$ and putting into 2 eqn.

$\alpha + 2\beta = \frac{21}{8} \neq \frac{5}{2} = Q^T(s_0, a_2).$

$\therefore$ Hence for $R(s) = s$, the above set of linear equations 1, 2 & 3 does not have any solution. That is, the function $Q(s,a) = \alpha s + \beta a$ cannot represent all true $Q$-values at $s_0$ for any choice of $\alpha$ & $\beta$.

7.3 Reward function for previous question $\boxed{R(s) = s}$

$R(s_0) = R_0 = 1$

$R(s_1) = R_1 = 2$

$R(s_2) = R_2 = 3$

$R(s_3) = R_3 = 4$

True Value

$Q^T(s, a)$

$$Q^T(s_0, a_1) = R_0 + \frac{R_1 + R_2}{4} = \frac{9}{4}$$

$$Q^T(s_0, a_2) = R_0 + \frac{R_2}{2} = \frac{5}{2} = \frac{10}{4}$$

$$Q^T(s_0, a_3) = R_0 + \frac{R_3}{2} = 3 = \frac{12}{4}$$

**Goal:** Design a function approximation $Q(s,a)$ s.t all Q-values at $s_0$ can be correctly represented.

Let $\boxed{Q(s,a) = \frac{9}{4}s + \frac{1}{4}f(a)}$

s.t.

$$Q(s_0, a_1) = \frac{9}{4} + \frac{1}{4}f(a_1) = \frac{9}{4} \Rightarrow f(a_1) = 0$$

$$Q(s_0, a_2) = \frac{9}{4} + \frac{1}{4}f(a_2) = 10/4 \Rightarrow f(a_2) = 1$$

$$Q(s_0, a_3) = \frac{9}{4} + \frac{1}{4}f(a_3) = 12/4 \Rightarrow f(a_3) = 3$$

Need to design $f(a)$ s.t. $f(1) = 0$ $f(2) = 1$ $f(3) = 3$

Let $f(a) = pa^2 + qa + r$

such that

$$\left.\begin{aligned} f(1) &= p + q + r = 0 \\ f(2) &= 4p + 2q + r = 1 \\ f(3) &= 9p + 3q + r = 3 \end{aligned}\right\} \Rightarrow \text{Unique solution for } p, q, r.$$

$$p = \frac{1}{2} \quad q = -\frac{1}{2} \quad r = 0$$

$f(a) = pa^2 + qa + r$

$= \frac{1}{2}a^2 - \frac{1}{2}a = \frac{1}{2}a(a-1)$

Thus, $Q(s,a) = \frac{9}{4}s + \frac{1}{4}f(a)$

$$\boxed{Q(s,a) = \frac{9}{4}s + \frac{1}{8}a(a-1)}$$

For $R(s) = s$, $Q(s,a) = \alpha s + \beta a$ cannot correctly represent all Q-values at $s_0$ but $Q(s,a) = \frac{9}{4}s + \frac{1}{8}a(a-1)$ can correctly represent all Q-values at $s_0$.

# CSE257_A3_Blackjack

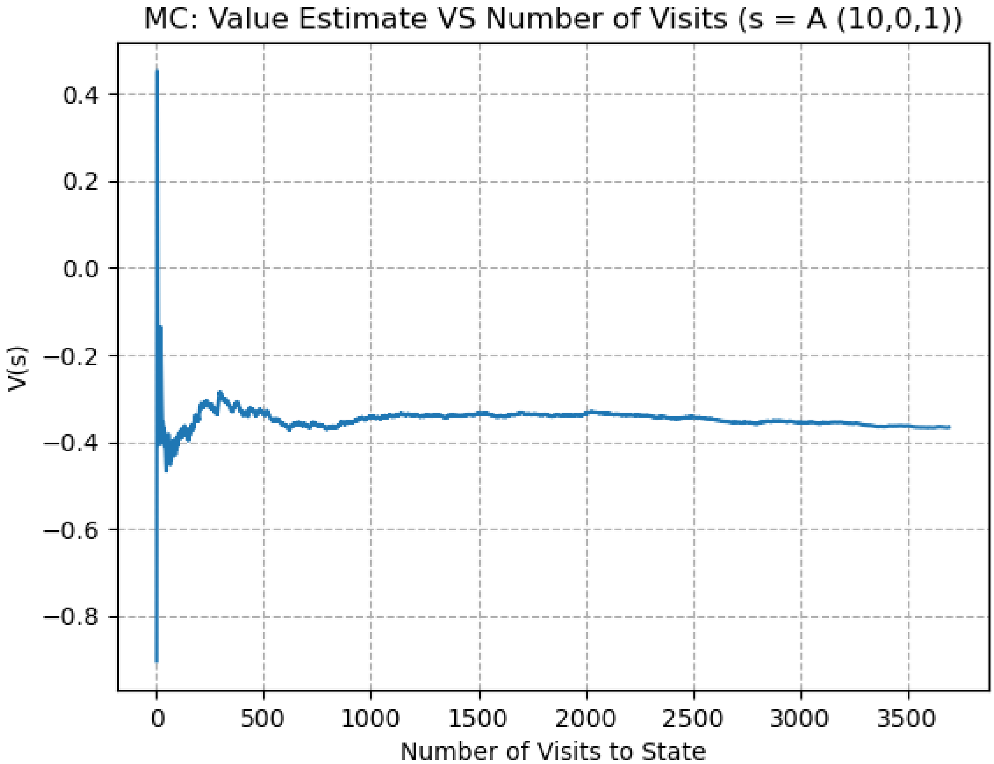
December 5, 2021

### 0.0.1 Question 8 (3 Extra Points):

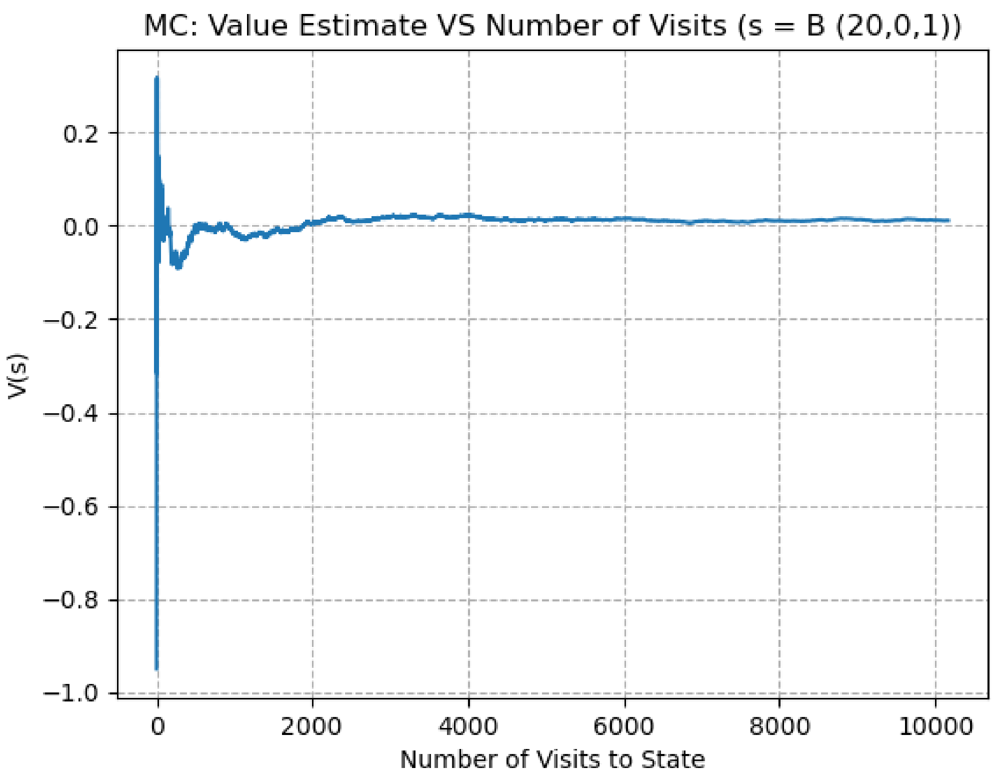
Select two game states, A and B: In State A the player's sum of cards is 10, and in State B the sum of cards is 20. Plot how the value estimate of the each state changes over the number of visits to the state until the values convergence, under Monte Carlo policy evaluation and Temporal-Difference policy evaluation, respectively; so 4 plots in total.

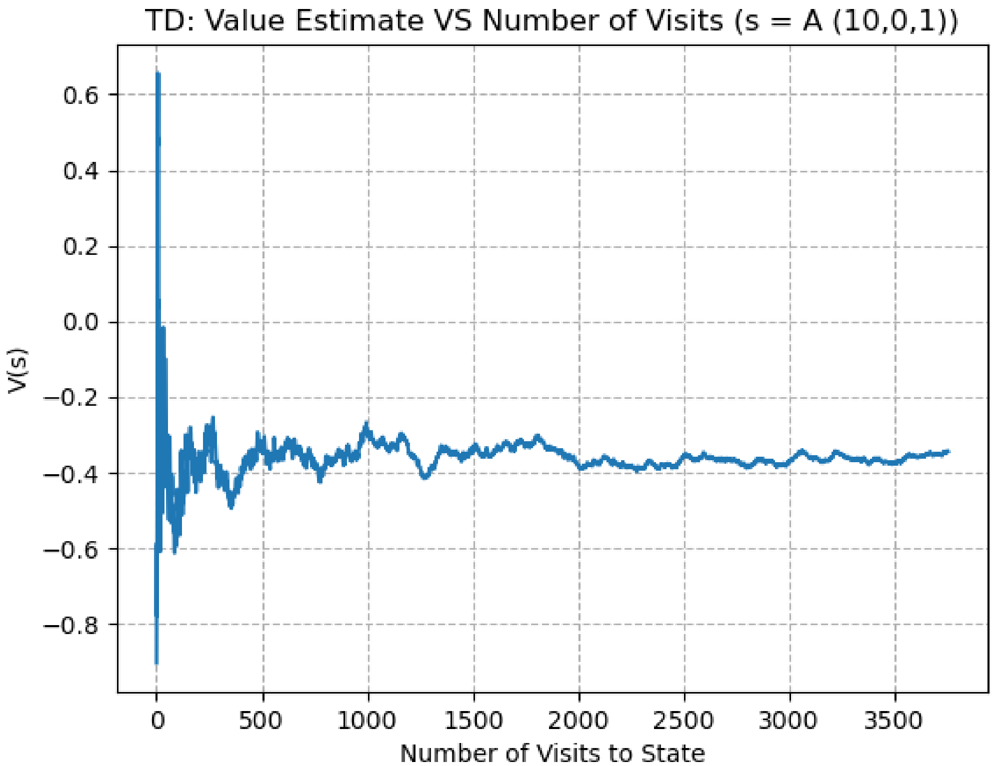
```
(base) siriusA@Barkat-MacAir : ~/Desktop/UCSD/courses/Fall_2021/CSE257/assignments/assignment3/blackjack-main
$ python3 main.py -t 3
  MC 1000000/1000000
++++ PASSED MC with 0 wrong values
  TD 1000000/1000000
++++ PASSED TD with 0 wrong values
  Q 1000000/1000000
++++ PASSED Q-Learning with 0 wrong values
```

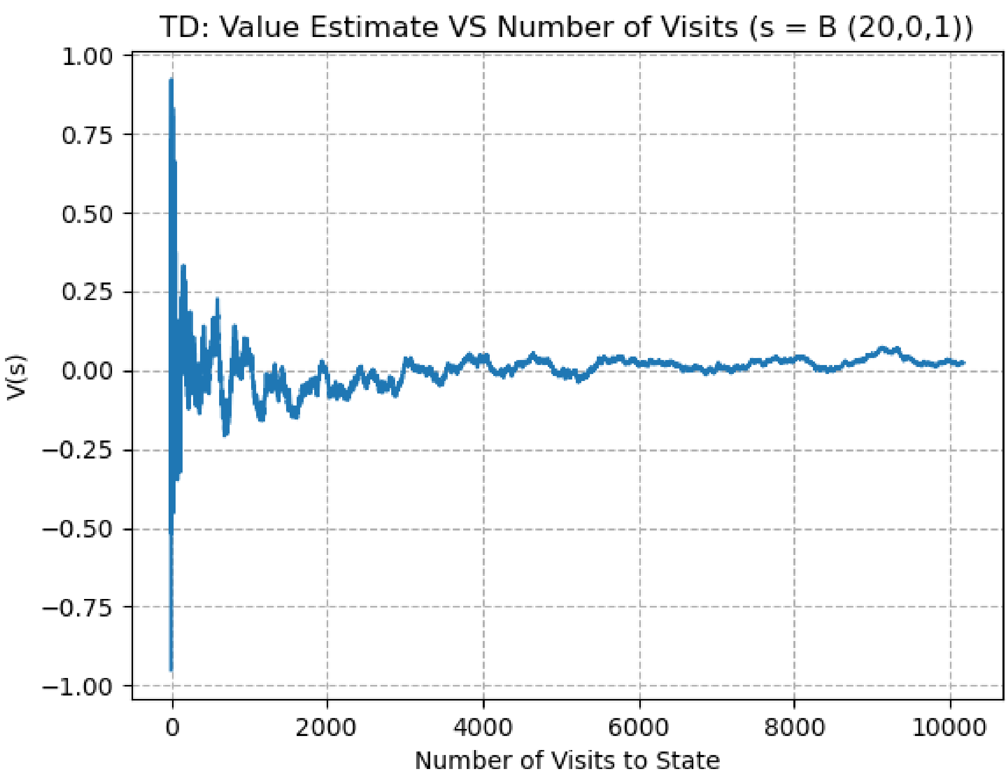
The Monte-Carlo values over the number of visits are stable whereas the Temporal-Difference state values are more fluctuating as compared to Monte-Carlo. This is because, in Monte-Carlo, each update to the state values happens after taking the expectation over large number of simulation sequences. On the other hand, in Temporal difference, each update happens with every incoming


MC: Value Estimate VS Number of Visits (s = A (10,0,1))
V(s)
Number of Visits to State


sample.

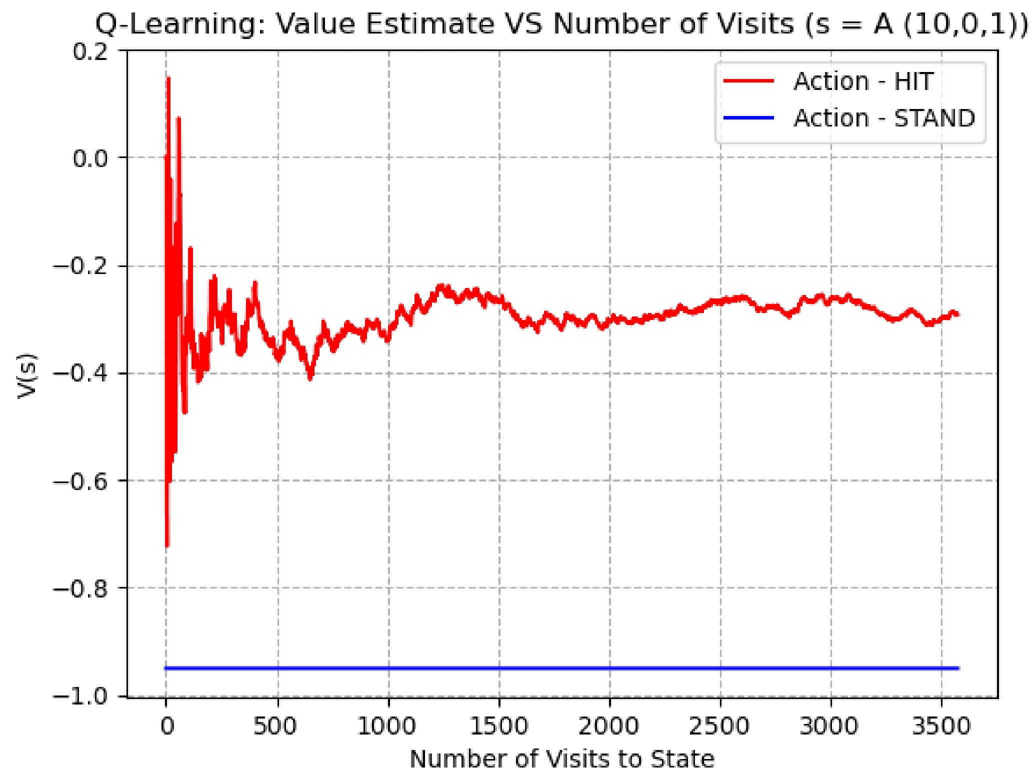

MC: Value Estimate VS Number of Visits (s = B (20,0,1))
V(s)
Number of Visits to State

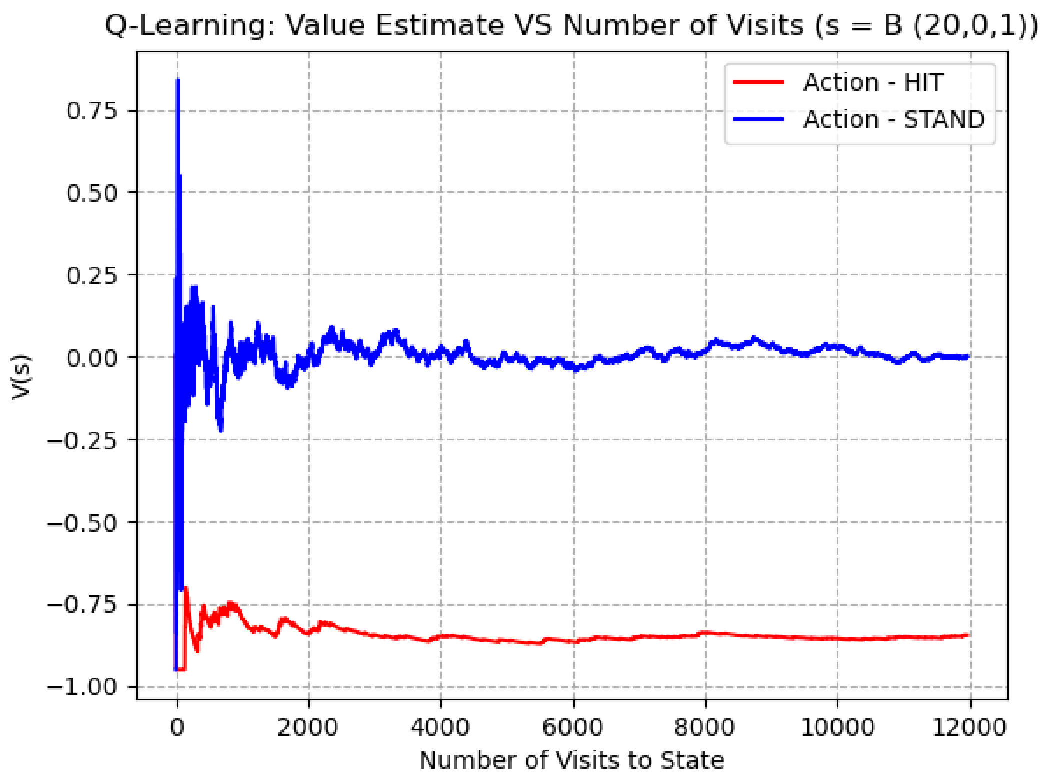

TD: Value Estimate VS Number of Visits (s = A (10,0,1))
V(s)
Number of Visits to State


TD: Value Estimate VS Number of Visits (s = B (20,0,1))
V(s)
Number of Visits to State

### 0.0.2 Question 9 (3 Extra Points):

Perform Q-learning and plot how the Q-value changes over the number of visits to each action for the same two game states you selected above, until you have run Q-learning for long enough so that the Q-value converges at least on some action for each state (note: a very bad action may receive a small number of visits, so this requirement is saying you only need to wait till the better action has been visited enough times so that the Q-value of it stabilizes).


Q-Learning: Value Estimate VS Number of Visits (s = A (10,0,1))
Action - HIT
Action - STAND
V(s)
Number of Visits to State
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
0
500
1000
1500
2000
2500
3000
3500


Q-Learning: Value Estimate VS Number of Visits (s = B (20,0,1))
Action - HIT
Action - STAND
V(s)
Number of Visits to State
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
2000
4000
6000
8000
10000
12000

Also plot the cumulative winning rate over the number of plays in the game: for every n number of plays (x-axis), show the ratio w/n (y-axis) where w is the total number of wins out of the n plays.

For plotting the cumulative win-rate vs number of game plays, I have set the `self.autoQL = True` and `self.autoPlay = True`. Then, I am running sufficiently large number of iterations (=5000). In each iteration, the Q-Learning algorithm (50 simulations) runs, which updates the Q-values of the states followed by the game playing function. The cumulative winning rate stabilizes at around ~ 41 %.