2024-1 Reinforcement Learning - Home work 1

1- Gittin Index : Measure of the reward that can be achieved by a sequence of actions from the present state onwords with the probability that it will be terminated in the future.

Gittin Index theorem helps on deciding a strategy of explore exploit trade-off by measuring the reward of certain action to give the best reward or discounted reward.

Thompson Sampling: Thompson sampling is an algorithm for online decision problems where actions are taken sequentially in a manner that must balance between exploitting what is known to maximize immediate performance and investing to accumulate new information that may improve future performance. It's particularly useful in situations where the decision has the outcomes of uncertainty.

Thompson sampling tends to explore actions with uncertainty outcomes while exploitting actions that are likely to yield high rewards, by balancing the trade off of exploration and exploitation.

2- In the first 10 steps, the agents has explored the available actions which are the 10 actions. The effect of selecting 10 actions is that the action will be guaranteed to be picked at least once in the first 10 steps.

Given the parameter $c > 0$, the agent is likely to prioritize the action with the highest return from the first sample. As of result, it may spike the reward/performance. As the agent try to do the other action, the agent will be much more forces / to do the explore option making it a sudden drop and try to get more reward as time moves on.

3. $Pr\{A_t = a\} = \dfrac{e^{H_t(a)}}{\sum_{b=1}^{k} e^{H_t(b)}}$ sigmoid function: $\dfrac{1}{1+e^{-x}}$

$$= \frac{e^{H_t(a)}}{e^{H_t(a)} + e^{H_t(b)}} = \frac{1}{1 + \frac{e^{H_t(b)}}{e^{H_t(a)}}}$$

$$= \frac{1}{1 + e^{H_t(b) - H_t(a)}}$$

if $x = H_t(b) - H_t(a)$.

then $= \dfrac{1}{1+e^{-x}}$ #

4. Prove that:

$H_{t+1}(A_t) = H_t(A_t) + a(R_t - \bar{R}_t)(1 - \pi_t(A_t))$ &

$H_{t+1}(a) = H_t(a) - a(R_t - \bar{R}_t)\pi_t(a)$ $a \neq A_t$

$H_{t+1}(a) = H_t(a) + a\dfrac{\partial \mathbb{E}[R_t]}{\partial H_t(a)}$ $\mathbb{E}[R_t] = \sum_x \pi_t(x)\, q_*(x)$

$$= H_t(a) + a\left(\sum_x \left(\frac{\partial}{\partial H_t(a)}\left[\pi_t(x) \cdot q_*(x)\right]\right)\right)$$

$$= H_t(a) + a\left(\sum_x \left(q_*(x)\frac{\partial \pi_t(x)}{\partial H_t(a)}\right)\right)$$

B_t: Baseline (we add Baseline to compare the current and previous value).

$$= H_t(a) + a\left(\sum_x \left(q_*(x) - B_t\right)\frac{\partial \pi_t(x)}{\partial H_t(a)}\right)$$

$$= H_t(a) + a\left(\sum_x \frac{\left(\pi_t(x) \cdot (q_*(x) - B_t)\frac{\partial \pi_t(x)}{\partial H_t(a)}\right)}{\pi_t(x)}\right)$$

$$= H_t(a) + a\,\mathbb{E}\left[(q_*(A_t) - B_t)\frac{\partial \pi_t(A_t)}{\partial H_t(a)} \Big/ \pi_t(A_t)\right]$$

$$= H_t(a) + a\,\mathbb{E}\left[(R_t - \bar{R}_t)\frac{\partial \pi_t(A_t)}{\partial H_t(a)} \Big/ \pi_t(A_t)\right].$$

$$\frac{\partial \pi_t(x)}{\partial H_t(a)} = \frac{\partial}{\partial H_t(a)} \pi_t(x).$$

$$= \frac{\partial}{\partial H_t(a)} \left[\frac{e^{H_t(x)}}{\sum_{y=1}^{n} e^{H_t(y)}} \right]$$

$$f'(x) = \frac{g'(x) h(x) - g(x) h'(x)}{h(x)^2}$$

$$= \frac{\frac{\partial e^{H_t(x)}}{\partial H_t(y)} \sum_{y=1}^{n} e^{H_t(y)} - e^{H_t(x)} \cdot e^{H_t(a)}}{\left(\sum_{y=1}^{n} e^{H_t(y)}\right)^2} \qquad \Rightarrow \frac{\partial e^{H_t(x)}}{\partial H_t(y)} = \mathbb{1}_{a=x} e^{H_t(x)}$$

$$= \frac{\mathbb{1}_{a=x} e^{H_t(x)} \cdot \sum_{y=1}^{n} e^{H_t(y)} - e^{H_t(x)} \cdot e^{H_t(a)}}{\left(\sum_{y=1}^{n} e^{H_t(y)}\right)^2}$$

$$= \frac{\mathbb{1}_{a=x} e^{H_t(x)} \cdot \sum_{y=1}^{n} e^{H_t(y)}}{\left(\sum_{y=1}^{n} e^{H_t(y)}\right)^2} - \frac{e^{H_t(x)} e^{H_t(a)}}{\left(\sum_{y=1}^{n} e^{H_t(y)}\right)^2}$$

$$= \mathbb{1}_{a=x} \pi_t(x) - \pi_t(x) \pi_t(a)$$

$$= \pi_t(x) \left(\mathbb{1}_{a=x} - \pi_t(a)\right).$$

$$H_{t+1}(a) = H_t(a) + \alpha \left[\frac{(R_t - \bar{R}_t) \cancel{(\pi_t(x))} (\mathbb{1}_{a=x} - \pi_t(a))}{\cancel{(\pi_t(x))}} \right]$$

$$= H_t(a) + \alpha \left[(R_t - \bar{R}_t)(\mathbb{1}_{a=x} - \pi_t(a)) \right]$$

If $\mathbb{1}_{a=x} = 1$, then $a = x$. | else $= 0$.

$$H_{t+1}(A_t) = H_t(A_t) + \alpha (R_t - \bar{R}_t)(1 - \pi_t(A_t))$$

$$H_{t+1}(a) = H_t(a) + \alpha (R_t - \bar{R}_t)(-\pi_t(a))$$

$$= H_t(a) - \alpha (R_t - \bar{R}_t)(\pi_t(a))$$

5a- add l_1

$$H_{t+1}(a) = H_t(a) + \alpha \frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)}$$

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \frac{\partial \left(\sum_x \pi_t(x) \cdot q_*(x)\right)}{\partial H_t(a)} - \lambda \sum [H_t(a)] \frac{\partial}{\partial H_t(a)}$$

$$= \underbrace{\frac{\partial \left(\sum_x \pi_t(x) \cdot q_*(x)\right)}{\partial H_t(a)}}_{\mathrm{I}} - \underbrace{\lambda \, \mathrm{sign}(H_t(a))}_{\mathrm{II}}.$$

I

$$= \frac{\partial \left(\sum_x \pi_t(x) . q_*(x)\right)}{\partial H_t(a)}.$$

$$= H_t(a) + a\left(\sum_x \left(q_*(x) - B_t\right) \frac{\partial \pi_t(x)}{\partial H_t(a)}\right)$$

$$= H_t(a) + \frac{a\left(\sum_x \left(\pi_t(x)\right)\left(q_*(x) - B_t\right) \frac{\partial \pi_t(x)}{\partial H_t(a)}\right)}{\pi_t(x)}$$

$$= H_t(a) + a\,\mathbb{E}\left[(R_t - \bar{R}_t) \frac{\partial \pi_t(A_t)}{\partial H_t(a)} \Big/ \pi_t(A_t)\right]$$

$$= H_t(a) + a\left[(R_t - \bar{R}_t)\left(\mathbb{1}_{a=x} - \pi_t(a)\right) - \lambda \operatorname{sign}(H_t(a))\right]$$

If $\mathbb{1}_{a=x} = 1$, then $a = x$.

$$H_{t+1}(A_t) = H_t(A_t) + a(R_t - \bar{R}_t)(1 - \pi_t(A_t)) - \lambda \operatorname{sign}(H_t(a))$$

else $=0$.

$$H_{t+1}(a) = H_t(a) + a(R_t - \bar{R}_t)(-\pi_t(a)) - \lambda \operatorname{sign}(H_t(a)).$$

$$= H_t(a) - a(R_t - \bar{R}_t)(\pi_t(a) - \lambda \operatorname{sign}(H_t(a)).$$

5b. add l_2.

$$H_{t+1}(a) = H_t(a) + a\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)}.$$

$$\frac{\partial \mathbb{E}[R_t]}{\partial H_t(a)} = \frac{\partial \left(\sum_x \pi_t(x) . q_*(x)\right)}{\partial H_t(a)} - \lambda\left(\sum [H_t(a)] \frac{\partial}{\partial H_t(a)}\right)^2$$

$$= \underbrace{\frac{\partial \left(\sum_x \pi_t(x) \cdot q_*(x)\right)}{\partial H_t(a)}}_{\mathrm{I}} - \underbrace{2\lambda \operatorname{sign}(H_t(a))}_{\mathrm{II}}.$$

I

$$= \frac{\partial \left(\sum_x \pi_t(x) . q_*(x)\right)}{\partial H_t(a)}.$$

$$= H_t(a) + a\left(\sum_x \left(q_*(x) - B_t\right) \frac{\partial \pi_t(x)}{\partial H_t(a)}\right).$$

$$= H_t(a) + \frac{a\left(\sum_x \left(\pi_t(x)\right)\left(q_*(x) - B_t\right) \frac{\partial \pi_t(x)}{\partial H_t(a)}\right)}{\pi_t(x)}.$$

$$= H_t(a) + a\,\mathbb{E}\left[(R_t - \bar{R}_t) \frac{\partial \pi_t(A_t)}{\partial H_t(a)} \Big/ \pi_t(A_t)\right].$$

$$= H_t(a) + a\,[(R_t - \bar{R}_t)(\mathbb{1}_{a=t} - \pi_t(a)) - 2\lambda\,\mathrm{sign}(H_t(a))].$$

If $\mathbb{1}_{a=x} = 1$, then $a = x$.

$$H_{t+1}(A_t) = H_t(A_t) + a(R_t - \bar{R}_t)(1 - \pi_t(A_t)) - 2\lambda\,\mathrm{sign}(H_t(a)).$$

else $= 0$.

$$H_{t+1}(a) = H_t(a) + a(R_t - \bar{R}_t) * (-\pi_t(a)) - 2\lambda\,\mathrm{sign}(H_t(a))$$
$$= H_t(a) - a(R_t - \bar{R}_t)(\pi_t(a)) - 2\lambda\,\mathrm{sign}(H_t(a)).$$

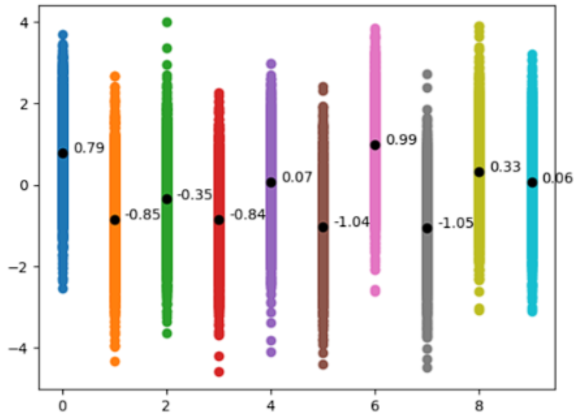
6.

[72]:
```python
bandits = np.arange(0,10)
```

[73]:
```python
mean = 0
std = 1
nArms = 10 #n number of bandits
start_time = time.time()
iterations = 2000 # number of iterations
plays = 1000 #number of plays per iterations


rewards = np.random.normal(mean, std, nArms)
```

[74]:
```python
ran_rewards = np.array([ np.random.normal(rewards,std,iterations) for rewards in rewards]) #2000 interations is used
```

[75]:
```python
for index in bandits :
    plt.scatter(np.full(iterations,index),ran_rewards[index])
    plt.text(index+0.2,rewards[index],str(round(rewards[index],2)))
plt.plot(bandits, rewards,'o', color='black')
plt.show()
```



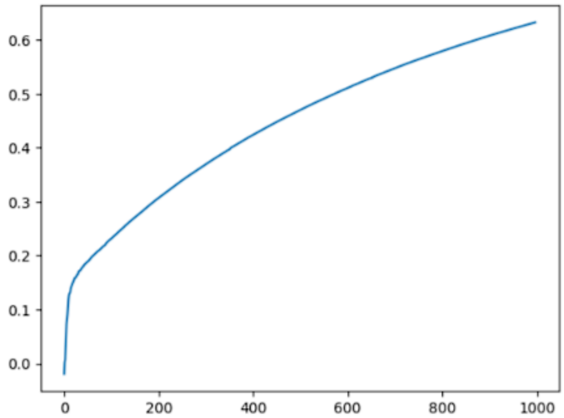
6a)

"A. Implement Greedy Method"

[79]:
```
running_reward_sum = np.copy(reward_estimates) # _per_action_per_bandit

rewards = []
#rewards.append(0)
rewards.append(np.mean(initial_reward_estimates)) # step 1

epsilon = 0.0 # Greedy method
for plays in range(2,plays):
    sum_of_reward = 0
    for problem_index in range(iterations):
        if np.random.random() > epsilon: # Greedy Selection
            maxval = np.amax(reward_estimates[problem_index])
            maxval_indices = np.ravel(np.array(np.where(reward_estimates[problem_index] == maxval)))
            random_choice = np.random.choice( maxval_indices ) # breaking ties randomly
        else :
             random_choice = np.random.randint(nArms)

        #print(str(problem_index),str(step),str(random_choice))
        running_reward_sum[problem_index][random_choice] += np.random.normal(testbed[problem_index][random_choice],1)
        action_count[problem_index][random_choice] += 1
        avg_reward = running_reward_sum[problem_index][random_choice] / action_count[problem_index][random_choice]
        reward_estimates[problem_index][random_choice] = avg_reward

        sum_of_reward += avg_reward

    rewards.append((sum_of_reward)/iterations)
```

[80]:
```
plt.plot(np.arange(plays),rewards)
```

[80]:
```
[<matplotlib.lines.Line2D at 0x28f0cf18730>]
```



6b.

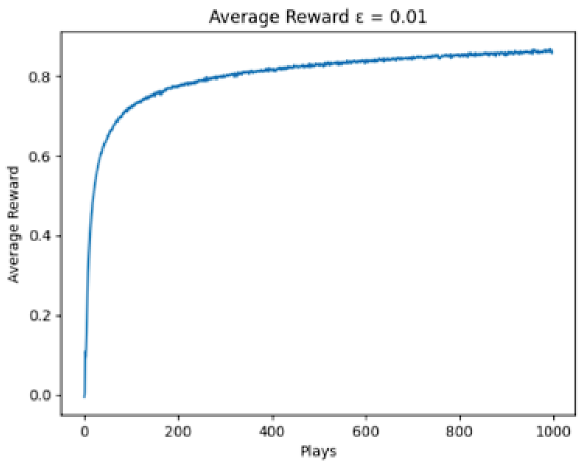
B.Implement Epsilon-Greedy Method

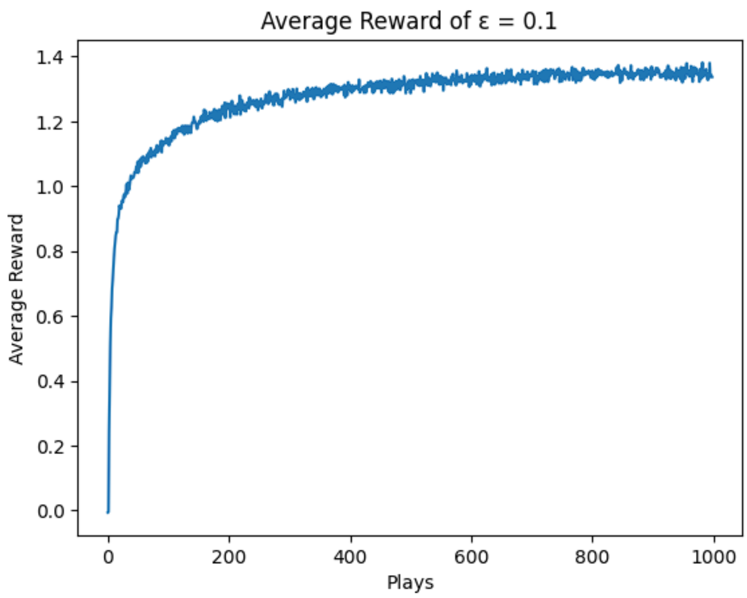
[ ]:

[111]:
```
running_reward_sum = np.copy(reward_estimates) # _per_action_per_bandit

rewards = []
#rewards.append(0)
rewards.append(np.mean(initial_reward_estimates)) # step 1

epsilon = 0.01 # Greedy method
for plays in range(2,plays):
    sum_of_reward = 0
    for problem_index in range(iterations):
        if np.random.random() > epsilon: # Greedy Selection
            maxval = np.amax(reward_estimates[problem_index])
            maxval_indices = np.ravel(np.array(np.where(reward_estimates[problem_index] == maxval)))
            random_choice = np.random.choice( maxval_indices ) # breaking ties randomly
        else :
             random_choice = np.random.randint(nArms)

        #print(str(problem_index),str(step),str(random_choice))
        running_reward_sum[problem_index][random_choice] += np.random.normal(testbed[problem_index][random_choice],1)
        action_count[problem_index][random_choice] += 1
        avg_reward = running_reward_sum[problem_index][random_choice] / action_count[problem_index][random_choice]
        reward_estimates[problem_index][random_choice] = avg_reward

        sum_of_reward += avg_reward

    rewards.append((sum_of_reward)/iterations)
```

[112]:
```
plt.xlabel('Plays')
plt.ylabel('Average Reward')
plt.title('Average Reward ε = 0.01')
plt.plot(np.arange(plays),rewards, label=f"Epsilon = {epsilon}")
```

[112]:
```
[<matplotlib.lines.Line2D at 0x28f0d59a9a0>]
```

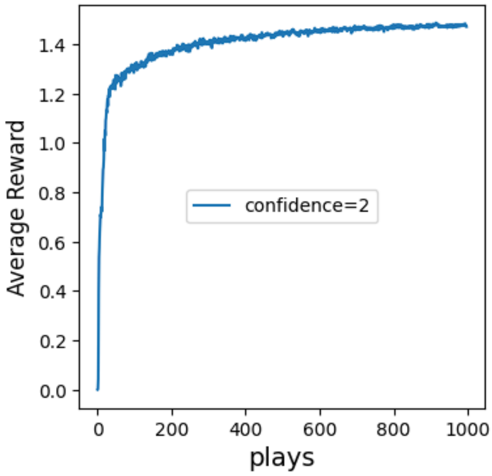




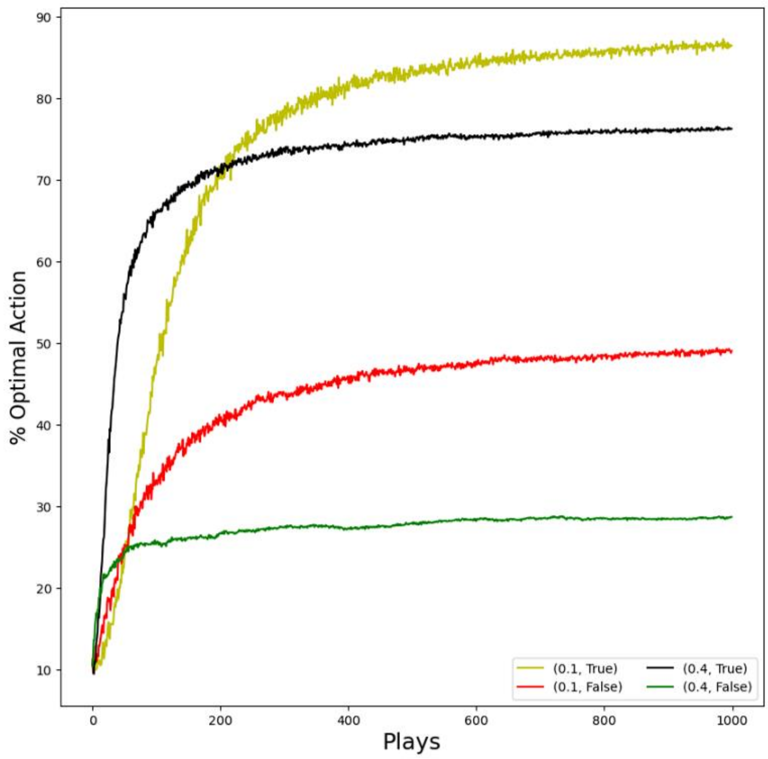
6c.

```
[126]: ucb_result = []
       ucb_c(plays, iterations, nArms, 0 , 2,  testbed, initial_reward_estimates, ucb_result)
```

```
[130]: plt.figure(figsize=(4,4))
       plt.xlabel('plays', fontsize=14)
       plt.ylabel('Average Reward', fontsize=12)
       plt.plot(np.arange(plays), ucb_result[0], label=("confidence=2"))
       plt.legend(loc='center', ncol=2)
       plt.show()
```

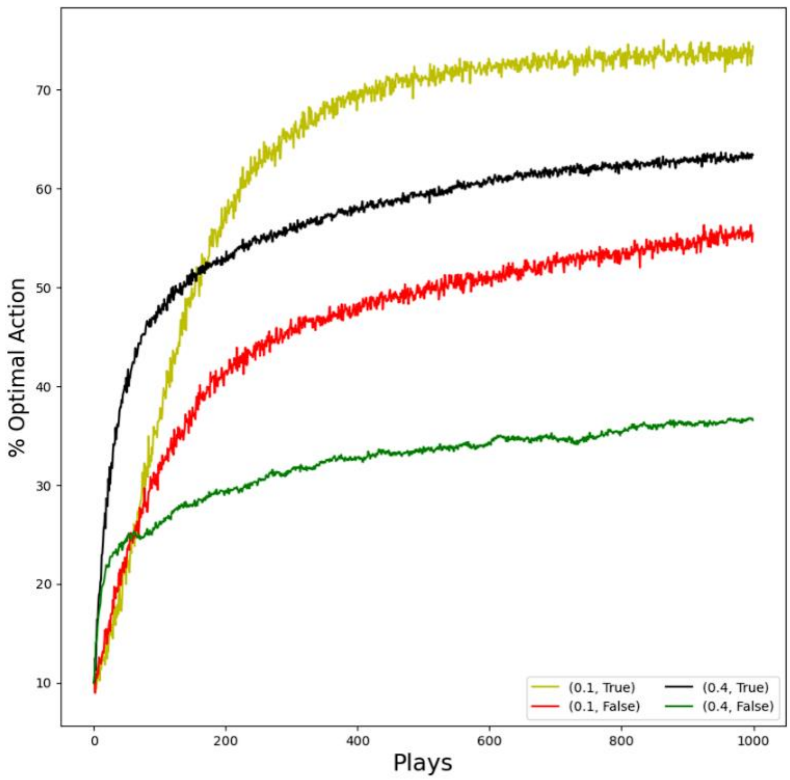


6d.



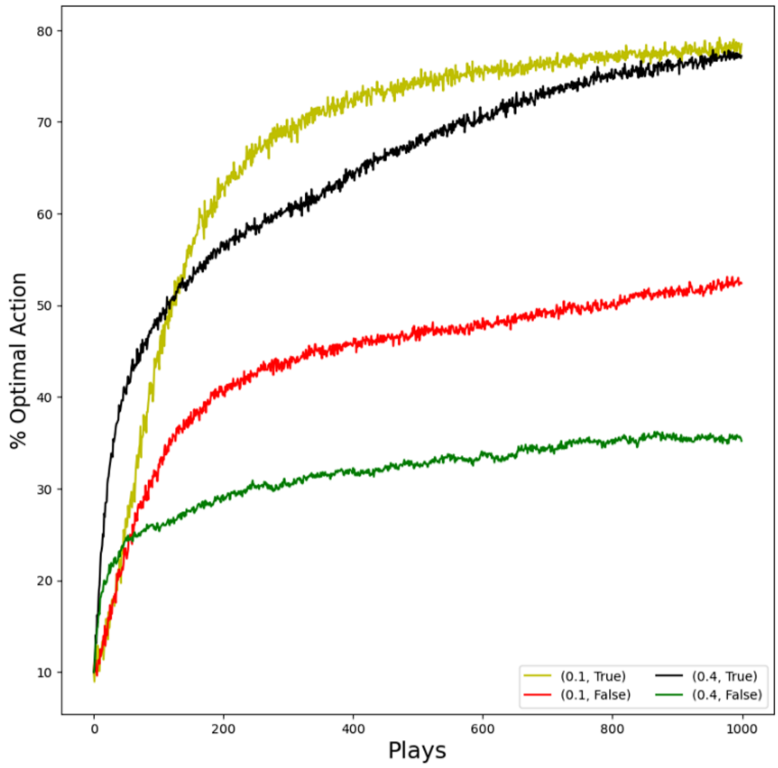
90
80
70
60
50
40
30
20
10
% Optimal Action
0
200
400
600
800
1000
Plays
(0.1, True)
(0.1, False)
(0.4, True)
(0.4, False)

6e.



70
60
50
40
30
20
10
% Optimal Action
0
200
400
600
800
1000
Plays
(0.1, True)
(0.1, False)
(0.4, True)
(0.4, False)

6f.



80
70
60
50
40
30
20
10
% Optimal Action
0
200
400
600
800
1000
Plays
(0.1, True)
(0.1, False)
(0.4, True)
(0.4, False)

7. $G_t = \gamma G + R_{t+1}$ $\quad R_1 = -1,\ R_2 = 2,\ R_3 = 6,\ R_4 = 3,\ R_5 = 2.$

$G_5 = 0$ $\quad T = 5,\ \gamma = 0{,}5$

$G_4 = 0{,}5 \cdot G_5 + R_5$
$= (0{,}5 \cdot 0) + 2$
$= 2.$

$G_3 = 0{,}5 \cdot G_4 + R_4$
$= (0{,}5 \cdot 2) + 3$
$= 4$

$G_2 = 0{,}5 \cdot G_3 + R_3$
$= (0{,}5 \cdot 3) + 6$
$= 7{,}5$

$G_1 = 0{,}5 \cdot G_2 + R_2$
$= (0{,}5 \cdot 7{,}5) + 2 = 5{,}75$

$G_0 = 0{,}5 \cdot G_1 + R_1$
$= (0{,}5 \cdot 5{,}75) + (-1)$
$= 1{,}875$

8. $\gamma = 0{,}9$ $\quad R_1 = 2.$ Infinite Number of terms : $G_t = \sum_{k=0}^{\infty} \gamma^k = \dfrac{1}{1-\gamma}$

$1_s = 1$ time step.

$G_t = \sum_{k=0}^{\infty} \gamma^k = \dfrac{1}{1-0{,}9}$

$= \dfrac{1}{0{,}1} \ (7_1)$

$= \dfrac{7}{0{,}1}$

$G_1 = \dfrac{7}{0{,}1} + 7 = 77$

$G_0 = \dfrac{7}{0{,}1} + 2 = 72.$

9. $G_t = \sum_{k=0}^{\infty} \gamma^k = \dfrac{1}{1-\gamma}$

$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots$ (1).

$\gamma G_t = \gamma R_{t+1} + \gamma^2 R_{t+2} + \gamma^3 R_{t+3} + \dots \gamma^{t+1} R_{t+n}.$

If $R = r.$

$\gamma G_t = \gamma r + \gamma^2 r + \gamma^3 r + \dots \gamma^{k+1} r$ (2)

$G_t - \gamma G_t = r - \gamma^{k+1} r$

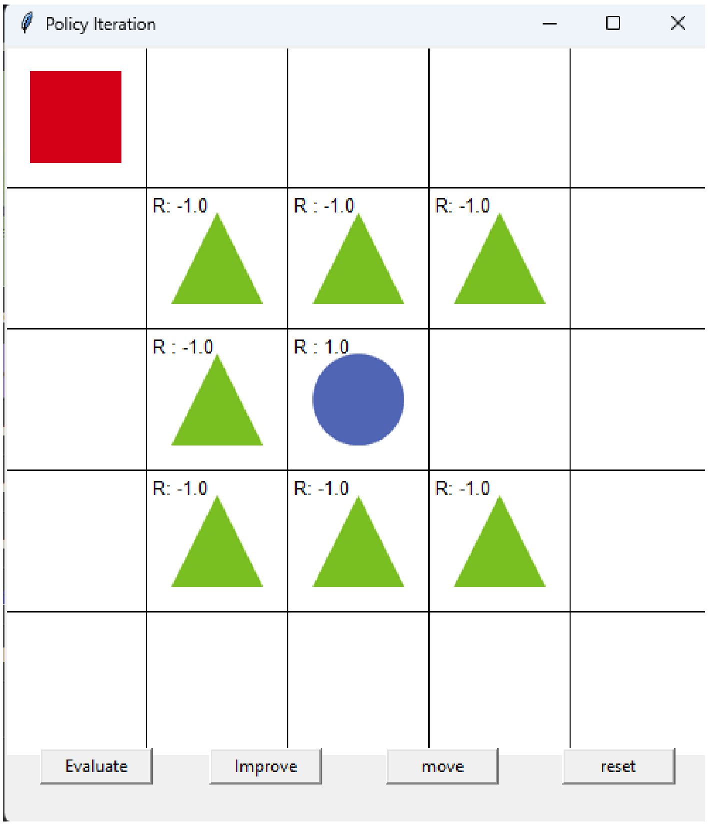
$G_t(1-\gamma) = r(1-\gamma^{k+1})$

$G_t = \dfrac{r(1-\gamma^{k+1})}{(1-\gamma)}$

If $k \to \infty$ $\quad G_t = \dfrac{r}{1-\gamma}$ $\quad R_t = r = 1.$

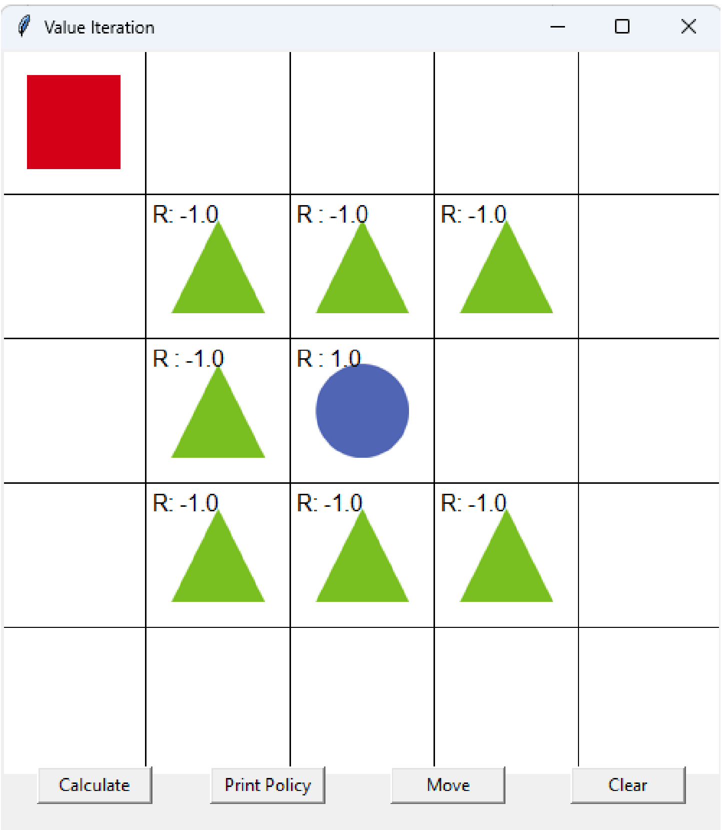
$= \dfrac{1}{1-\gamma}$

10. b



Policy Iteration
R: -1.0
R : -1.0
R: -1.0
R : -1.0
R : 1.0
R: -1.0
R: -1.0
R: -1.0
Evaluate
Improve
move
reset

10. d



Value Iteration
R: -1.0
R : -1.0
R: -1.0
R : -1.0
R : 1.0
R: -1.0
R: -1.0
R: -1.0
Calculate
Print Policy
Move
Clear

11. Reward Hypothesis can de describe as maximizing the satisfaction from interecting a certain action in the encvironment. Reward can be obtain from every action as a feedback to the agent. The goal from this hypothesis is to maximizing the accumulated reward, although the current or immediate reward may be small. The agent must balance between short term benefits and long terms objectives. Agent must navigate this trade-off, which often prioritizing actions that may yield smaller immediate reward, but lead to longer cumulative returns or reward in the future.