Dilan Dinushka
School of Computing and Information Systems
Singapore Management University

November 17, 2024

ASSIGNMENT — 2

Question 1

The related python implementations are provided separately.

(1)

Based on 1000 independent trials with each trial having 1000 product recommendation rounds, **Thompson Sampling method** achieved the highest number of average sales.

(2)

Compared to non-exploration methods like greedy method and softmax method, all the exploration/exploitation methods achieved better selling frequencies. Below table shows the average results of each method for 1000 trials while recommending 1000 items within each.

Method	Average Recommendations
Greedy	572.851
Softmax	563.968
ϵ- Greedy	769.464
Upper Confidence Bound	790.763
Thompson Sampling	812.275

(3)

For a real store setting where users preferences might change with time, it is important to keep the exploration aspect of the MAB arm selection process more active. Therefore, it would be better to utilize Thompson sampling method which consider the successes and failures of recommendations. By using this method, it is possible to adapt to shifting user preferences since each such change would impact the underline Beta distributions of Thompson sampling method through alpha (num of successes) and beta(num of failures) values.

Question 2

The related python implementations are provided separately.

When comparing the DQN and Reinforce algorithm, the Policy gradient models seems to take a considerable amount of time before start learning the playing patterns for the given environment. This might be due to the credit assignment problem arising in the long sequence of events (trajectory) in the Pong environment(ie: Hitting the ball require to check the other players action and ball position from a somewhat early step). We can observe this behaviour by comparing the training rewards graph provided below.

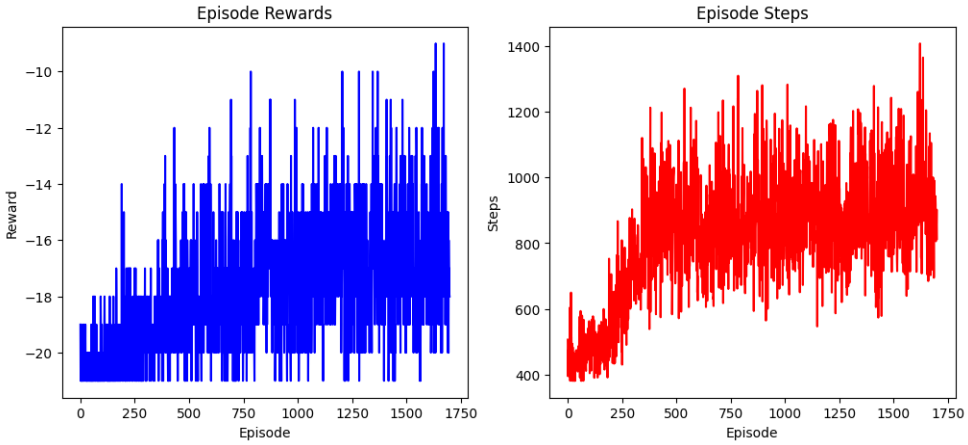

Figure 1: Training of DQN based model

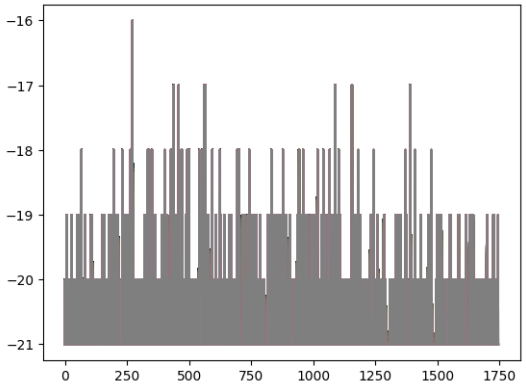

Figure 2: Training of Policy Gradient based model

Question 3

In the given problem, the MDP of the Arms be defined as follows.

$$\begin{aligned} \text{States} &\longrightarrow 0, 1 \\ \text{Actions} &\longrightarrow 0 \text{ (passive)}, 1 \text{ (active)} \\ \text{Rewards} &\longrightarrow R(0) = 0 \ \& \ R(1) = 1 \\ \text{Discount Factor} &\longrightarrow 1 \\ \text{Horizon} &\longrightarrow 2 \end{aligned}$$

The transition probabilities are as below.
For Passive action;

	0	1
0	0.6	0.4
1	0.3	0.7

For Active action;

	0	1
0	0.4	0.6
1	0.15	0.85

Let $V_t(s)$ represent the expected maximum reward from state s at time t.
Since the event horizon is at timestep 2, Values should be able to take the full reward values;

$$V_2(0) = 0$$
$$V_2(1) = 1$$

At $t = 1$;

$$V_1(s) = max[R(s) + \Sigma_{s'} P(s'|s, a)V_2(s')]$$

To get the $V_1(s)$ value, we need to consider the $Q_1(s, a)$ values of at timestep 1. Therefore using

$$Q_1(s, a) = R(s) + \Sigma_{s'} P(s'|s, a)V_2(s')$$

For state 0,

for action 0
$$Q_1(0, 0) = 0 + 0.6 * 0 + 0.4 * 1$$
$$Q_1(0, 0) = 0.4$$

for action 1
$$Q_1(0, 1) = 0 + 0.4 * 0 + 0.6 * 1$$
$$Q_1(0, 1) = 0.6$$

Similarly for state 1,

for action 0
$$Q_1(1, 0) = 1 + 0.3 * 0 + 0.7 * 1$$
$$Q_1(1, 0) = 1.7$$

for action 1
$$Q_1(1, 1) = 1 + 0.15 * 0 + 0.85 * 1$$
$$Q_1(1, 1) = 1.85$$

Whittle index is the smallest subsidiary required to make the passive value optimal compared to the active value.

$$Q(s, 0) + W = Q(s, 1)$$

Using above for each of the states;

for state 0
$$Q(0, 0) + W = Q(0, 1)$$
$$W = 0.6 - 0.4$$
$$W = 0.2$$

for state 1
$$Q(1, 0) + W = Q(1, 1)$$
$$W = 1.85 - 1.7$$
$$W = 0.15$$

Therefore results for each arms are;

- Arm-1 $\longrightarrow$ 0.15

- Arm-2 $\longrightarrow$ 0.2
- Arm-3 $\longrightarrow$ 0.15
- Arm-4 $\longrightarrow$ 0.2

Question 4

All the related code files are given separately.

Environment Building

During the environment building I assumed the following conditions.

- For the Environment Repair shop is fixed. (resetting the environment does not change the location)
- Monsters have fixed set of locations to be spawned. They spawn in this location with a predefined probability.
- Monsters cant spawn in location where Repair shop is held or prize is spawned.

Therefore all the experiments were done on same environment (means monster spawning locations, repair shop location are fixed.). Some environment parameters are as follows.

- Prize spawning probability: 0.5
- Monster spawning probability: 0.7
- Number of possible monster spawn locations: 5

Below figure include a sample output from the simulator.

```
Top Right of the grid considered as (0,0) coordinate in the renders!

Initial Observation: (0, 0, 0, 0)
Action: Up, Observation: (0, 0, 0, 0), Reward: -1
A R . . P
. . . . .
. . . . .
. . . . .
. . . . .

Action: Right, Observation: (1, 0, 0, 0), Reward: 0
. R M . P
. . . . .
. . . . .
. . . M .
M M . . .

Action: Right, Observation: (2, 0, 0, 1), Reward: -10
. R A . P
. . . . .
. . . . M
. . . M .
M M . . .

Action: Right, Observation: (3, 0, 0, 1), Reward: -10
. R M A P
. . . . .
. . . . M
. . . M .
M M . . .

Action: Down, Observation: (3, 1, 0, 1), Reward: -10
. R M . P
. . . A .
. . . . M
. . . M .
M M . . .
```

Figure 3: Example simulation output from the environment

Q-Learning with Simulator

In the given environment the state space include (5 x 5 x 5 x 2) values with each of such state having 4 values corresponding to the actions possible by the agent.

The learning parameters for the given problem was defined as below.

- Learning rate: 0.001
- Discount Factor: 0.99
- Epsilon: 1
- Epsilon Decay: 0.999
- Epsilon min: 0.01

During the training process approximate reward per episode was **278.03696**.

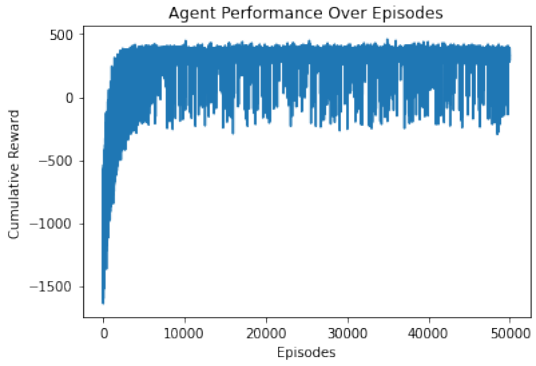

Figure 4: Q-Learning agent reward history

While testing agent was able to obtain **318.0** reward per episode on average.

During the training agent seems to avoid the positions where monsters can be spawned unless it is required to pass through such place to obtain the prizes.

Approximate Q-Learning

The features defined for approximate Q-Learning is based on the agents view of world. Therefore, we cannot utilize the distance to Repair Station, Monster Location etc. since only way for agent to learn about these is by exploration.

Therefore the used features for agent were as follows.

- Distance to prize
- Coordinates of prize location
- prize x, y coordinates
- Damaged Status
- Distance to mid point
- Agent x, y coordinates
- Agent x, y coordinates after taking the considering action

Below is the comparison between approximate and standard Q-Learning processes.

It should be noted that proper design of features is required for better training of approximate Q-Learning mechanism. Otherwise it would perform worse than the standard method.

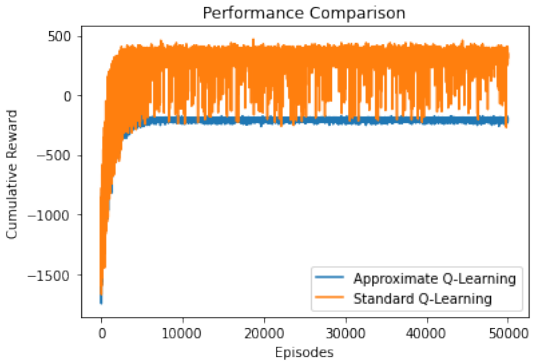

Figure 5: Q-Learning agent reward history

Question 5

The given player expected rewards are as follows.

	A	B
A	5, 5	0, 7
B	7, 0	1, 1

And the given partial potential function is as follows.

	A	B
A	0	x
B	x	3

According to the potential function definition;

$$\phi(a_i^{'}, a_{-i}) - \phi(a_i^{''}, a_{-i}) = u_i(a_i^{'}, a_{-i}) - u_i(a_i^{''}, a_{-i})$$

Therefore, by applying above function to given game we can calculate the x value. Let the player 2's action be B. Then,

$$\begin{aligned} \phi(A, B) - \phi(B, B) &= u_i(A, B) - u_i(B, B) \\ 0 - 1 &= x - 3 \\ x &= 2 \end{aligned}$$

We can verify the answer by letting the player 2's action to be A. Then,

$$\begin{aligned} \phi(A, A) - \phi(B, A) &= u_i(A, A) - u_i(B, A) \\ 5 - 7 &= 0 - x \\ x &= 2 \end{aligned}$$

Therefore, when the value of $x = 2$, the potential function can replace the rewards distribution.

Submitted by Dilan Dinushka on November 17, 2024.