**COMP532-202324 ASSIGNMENT 1**

Group Member :

Ashutosh Anil Deshpande - sgadeshp@liverpool.ac.uk - 201758974

Abhishek Prabhakar Kalkutki - sgakalku@liverpool.ac.uk - 201749286

Utkarsha Padmakar Salokhe - sgusalok@liverpool.ac.uk - 201669653

Arun

**Problem 1**

**Description of Multi-Armed Bandits**

Many-armed bandits (MAB) is consecutive decision-making employment where an agent has to pick the best action (arm) from a list of options to get the greatest overall reward. Finding the right combination between exploitation (trying different actions to learn their rewards) and exploration (choosing actions believed to yield the highest reward) is tough. Exploitation utilizes that which already exists to make quick money, but study is needed to find the best course of action and avoid taking less-than-ideal ones. Inefficient work could come from excessive research, and better ways to do things could not be found if there is extensive exploitation (Chen *et al.* 2022). Finding the right balance between exploration and exploitation is key to getting the best results on MAB projects. Therefore, agents must carefully split their resources between the two to get the biggest long-term payoff. This balance makes the most of long-term wins in changing situations and ensures good decision-making.

**Evaluation of Epsilon-Greedy Algorithm**

People generally employ the epsilon-greedy approach to find the right balance between taking advantage of chances and searching for novel opportunities when working multi-armed bandit jobs. Initially, the behaviors are picked at random with a chance of epsilon (De Curtò *et al.* 2023). The acts that are most likely to lead to the biggest rewards are then chosen (exploited) with a chance of 1%. This way of doing things allows individuals to look into other choices while still making sure that known high-value actions are used effectively. Expanding epsilon-greedy shifts the balance between exploring and exploiting, which helps people make good decisions when things are unresolved

**Setup for simulation**



**Figure 1: Setup of simulation**

(Source: Evaluated from Google Colab)

The 2000 bandit jobs in the application are done 2000 times using the epsilon-greedy method. Each job has been expected to have10 arms. This setup helps to look into the trade-offs between exploring and taking advantage of several different bandit activities and situations. There are a lot of jobs that give a general idea of the way the algorithm works, and the many runs show the way the method has changed over time. This setup makes it simpler to get an in-depth look at the way the technique works and the way well it adjusts in various bandit situations.

**Evaluation of True Action Values, the Anticipated Action Values, and the Action Counts**



**Figure 2: Value initialization**

(Source: Evaluated from Google Colab)

In the bandit task study, the preliminary distribution of the actual value of each arm has a normal shape, with a mean of 0 and a “standard deviation” of 1. The initial number for all action counts and values that go with them is zero. Once these are set up, the epsilon-greedy technique is used to find out the strength of each arm compared to the others. The system keeps making the agent's decision-making better by changing the anticipated values and action counts on the fly as awards are earned.

**Results and analysis**



**Figure 3: Implementation of bandit algorithm**

(Source: Evaluated from Google Colab)

The present iteration of the epsilon-greedy approach chooses actions sequentially by iterating over each bandit task several times. The action with the highest projected value is either used, or an action with a chance of epsilon is picked at random to look into. The reward is then used to change the expected value of the action that was chosen. The algorithm keeps track of the average payout and the number of times the best course of action is chosen across all bandit tasks, challenges, and iterations so that a full picture of how well it works can be drawn.



**Figure 4: Results visualization**

The chart demonstrates the way the epsilon-greedy technique worked with 2000 bandit jobs and 2000 iterations. The “Average Reward over Plays” plot demonstrates how the average benefit rises over time, and shows the way the application can take benefit of acts with high rewards. The “Percentage of Optimal Action over Plays” plot also goes up as the program learns it stops going up after 2000 trials, at about 65%. These trends show the ability of the algorithm to find the right combination between exploring and taking advantage of opportunities so that the total rewards are as high as possible. Therefore, the results demonstrate that the technique works for balancing the advantages of pursuing new opportunities versus capitalizing on existing ones.

**Problem 2: Explanation of exploration and exploitation for multi-armed bandits**

Agents have to decide which arms to take to get the greatest advantages in multi-armed bandit (MAB) scenarios. The unidentified compensation for each action is selected from a set of potential outcomes. The agent's job is to carefully sample activities to make the most money overall. Exploration highlights the process of trying out various strategies to find possible benefits, which is an important part of MAB. It entails picking things carefully, even if they appear to not like the best idea at first, to find out what benefits they might have (Ehab, 2023). This kind of study is significant because it helps people figure out how to act in better ways, which could lead to bigger benefits in the long run. The limited investigation could result in a faulty evaluation of the environment and missed opportunities for strategic advancement. Exploitation is a method that is used along with research. This entails prioritizing actions that are projected to provide the most immediate benefits, in light of the available information (Elwood *et al.* 2023). Exploitation has significance because it allows the agent to use what they know about the environment and the options they have to make the most money in the short run.

Discovering the ideal balance between exploration and exploitation can be impossible in MAB environments. In recognition of the best advantages, agents have to balance learning new things while utilizing what they currently know (Huang *et al.* 2022). The problem is often dealt with using the epsilon-greedy technique. In epsilon-greedy, the agent employs a two-pronged strategy: it randomly selects an action with a probability epsilon, thus ensuring exploration of all available options, and selects the action with the highest estimated value (exploitation) with a probability of 1-epsilon. The agent can use this two-pronged approach to systematically explore the range of actions while still choosing the ones that are going to provide them with the biggest anticipated rewards (Pokhrel and Mandjes, 2023). The research-to-exploitation ratio can be modified by the agent by altering the value of epsilon. It is suggested to do additional research while looking more closely at all the possible outcomes when the epsilon number is higher.

**Problem 3: Description of Action-Value Approach for Reinforcement Learning**

In the area of reinforcement learning, “action-value methods” find out how well an action works in a certain situation. This is necessary for determining the most significant benefit over time. They are essential for getting up with the most effective methods to do a variety of reinforcement tasks related to learning to get the most out of them.

**1. Basics of Action-Value Methods:**

Action-value techniques find out the worth of an activity by examining its advantages at various phases of its being. The amount appears after working out the anticipated total gain from completing a job and choosing the best course of action (Shen *et al.* 2023). In an alternate situation, it describes the long-term benefits that result from choosing a certain move in an instance.

**2. Sample-Average Method:**

The sample-average technique varies the amount that every action is valuable by taking the average of the advantages that come from each of them. It is an improved action-value method and the projected action value “Qt(a)” is changed in this way for action “a” that took place in state “s” at time step “t”:

“Qt(a) = (R1+R2……+Rn-1+Rn)/n”

where “n” is the number of times action “a” has been taken in state “s” and “R1, R2,.....Rn” are the awards that were given for doing action "a" in status "s" in earlier time steps.

**3. Incremental Update Method:**

The incremental update method is a different way to change action values. After every action, the value of the action is changed to a small portion. In this approach, the number step size referred to as α regulates how quickly the system operates (Sri *et al.* 2023). The equation for the next step in the process for action “a” at time step “t” in state “s” is:

“Qt+1(a) = Qt(a) + α . (Rt - Qt (a))”

Where Qt (a) is the anticipated outcome of action “a” before it is done at time step “t”, Rt is the reward that is given after activity “a” is done at time step “t”, and α is the step size measure (0<α<=1).

**4. Q-Learning:**

Action-value discovery with Q-learning is a well-known technique for reinforcement learning. There is an instance that it opens with and then it finds the action's grade (Q-value) that was thought of for that situation. It records the Q-values for each state-action correspond to in a Q-table. These values are then updated based on the benefits achieved and the expected upcoming rewards. Here is the Q-learning change rule:

“Q(s,a) ← Q(s,a) + α . (R + γ. maxa Q(, a) - Q(s, a))”

Which has Q(s, a) as the Q-value for action “a” in state “s”, R as the prize for action in state, γ as the “discount factor” (0<=γ <1), and maxa Q(, a) as the highest Q-value for the upcoming states.

Action-value techniques are essential for reinforcement education as they find out the amount that various behaviors are valuable in different states. Agents may alter their behavior and values based on what advantages they get from acting in the world. This assists in helping them learn the best rules to adhere to. There are numerous methods used to find the best policies and action values in different reinforcement learning problems. These involve the sample-average method, the gradual update method, and Q-learning.

**Contribution of each student** :

Ashutosh Deshpande - I personally contributed in the coding section to take out expected output successfully and reffered references for theorotical questions.

Abhishek Kalkutki - I worked on theory questions to get

**References**

Chen, X., Kim, M.L., Villar, S.S. And Robertson, D.S., 2022. Some Performance Considerations When Using Multi-Armed Bandit Algorithms In The Presence Of Missing Data. Plos One, 17(9),.

De Curtò, J., De Zarzà, I., Roig, G., Cano, J.C., Manzoni, P. And Calafate, C.T., 2023. Llm-Informed Multi-Armed Bandit Strategies For Non-Stationary Environments. Electronics, 12(13), Pp. 2814.

Ehab, M.M., 2023. Leo Satellite Assisted Uav Distribution Using Combinatorial Bandit With Fairness And Budget Constraints. Plos One, 18(8),.

Elwood, A., Leonardi, M., Mohamed, A. And Rozza, A., 2023. Maximum Entropy Exploration In Contextual Bandits With Neural Networks And Energy Based Models. Entropy, 25(2), Pp. 188.

Huang, C., Bai, H. And Yao, X., 2022. Online Algorithm Configuration For Differential Evolution Algorithm. Applied Intelligence, 52(8), Pp. 9193-9211.

Pokhrel, S.R. And Mandjes, M., 2023. Internet Of Drones: Improving Multipath Tcp Over Wifi With Federated Multi-Armed Bandits For Limitless Connectivity. Drones, 7(1), Pp. 30.

Shen, F., Liu, Z. And Perigo, L., 2023. Strategic Monitoring For Efficient Detection Of Simultaneous Apt Attacks With Limited Resources. International Journal Of Advanced Computer Science And Applications, 14(3),.

Sri, L.K., Prasad, A.V.K. And Bulla, S., 2023. Meta Heuristic Fusion Model For Classification With Modified U-Net-Based Segmentation. International Journal Of Advanced Computer Science And Applications, 14(3),.