# Assignment 1 Report

COMP534 – Machine Learning and Bioinspired Optimisation

**Authors:**

Chaiyatorn Permpornsakul – pscpermp@liverpool.ac.uk - 201768687

Chinwa Chimdi-Ezekwe – sgcchimd@liverpool.ac.uk - 201751919

Mehrnaz Miri – sgmmiri@liverpool.ac.uk - 201729365

Neda Yavari – sgnyavar@liverpool.ac.uk - 201765565

Yukabed Ijadi – f.ijadi@liverpool.ac.uk - 201773485

March 2024

## Problem 1: Results and Discussion

In implementing the Multi-Armed Bandit algorithm, we conducted experiments across three variations of the ε-greedy method, with epsilon values set to 0, 0.1, and 0.01. The aim was to analyse their performance in terms of average rewards and the percentage of optimal actions taken.

The initial setup involved 2000 bandit tasks, running for 1000 steps. We initialised the environment variables with 20 arms and assigned action values to each arm using a normal (Gaussian) distribution with a mean of 0 and a variance of 1.

We employed an incremental technique to update the estimated action values, enabling us to calculate averages of the first k rewards without storing all rewards explicitly. This technique utilises a simple formula to update estimates efficiently.

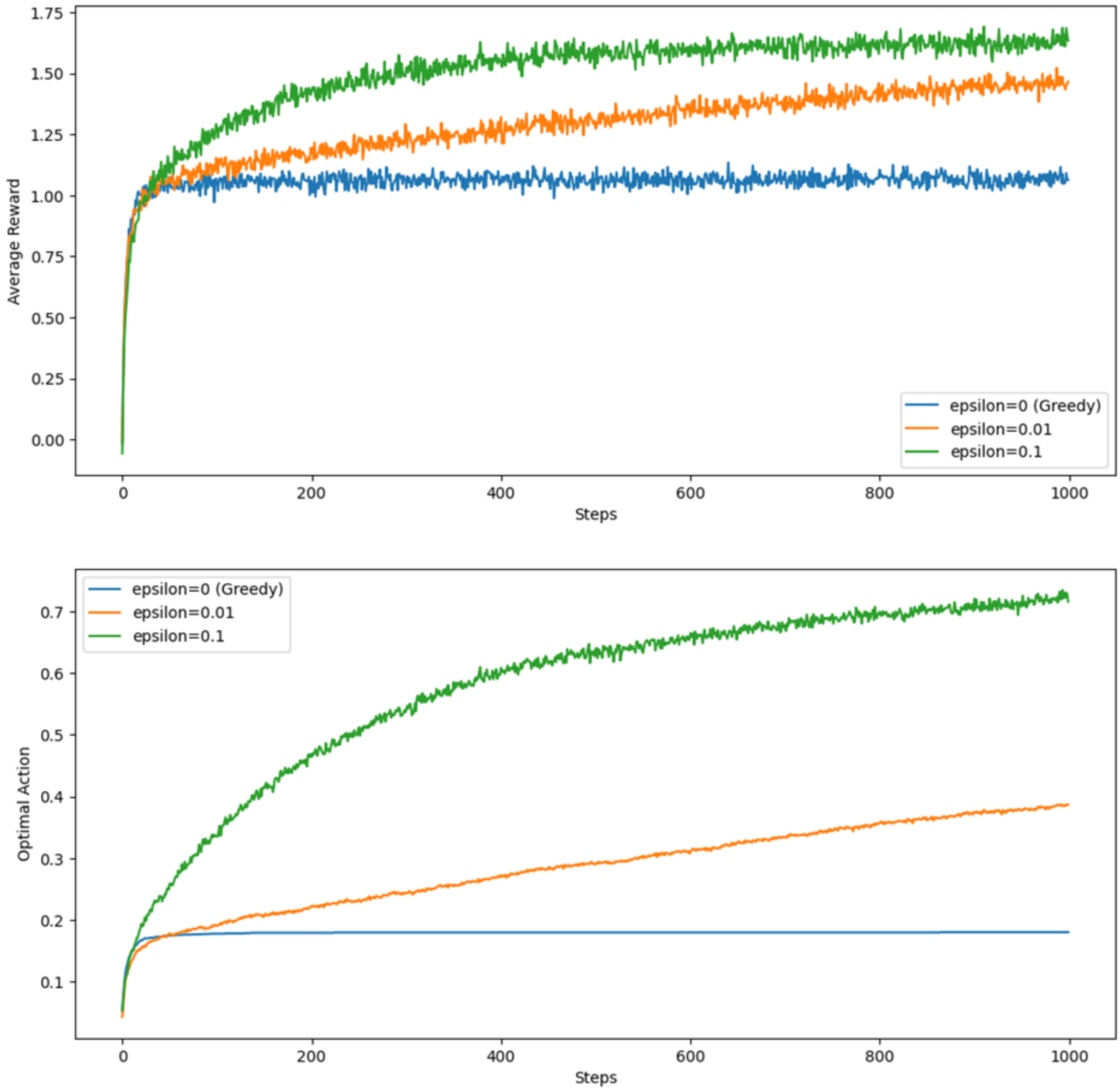

Figure 1: Average performance between the greedy and epsilon(ε) greedy action-value methods on the 20-armed testbed and these data are averages over 2000 tasks.

After running the algorithm, we analysed the results by plotting two graphs: "Average Reward vs Steps" and "Optimal action vs Steps" using the matplotlib library in Python.

The results, represented in Figure 1, reveal distinct performance trends among the different ε-greedy methods. In the "Average Reward vs Steps", the greedy (ε=0) method displays gradually better initially, but it levels off quickly and gains a reward per step of only around 1, compared to the best possible around 1.7. This is expected since a greedy algorithm only exploits known optimal actions and gets stuck performing suboptimal outcomes.

Interestingly, the greedy method initially outperformed the other two ε-greedy methods. However, it eventually plateaued and exhibited the worst long-term performance. This emphasises the importance of exploration in discovering the optimal action over time.

Furthermore, in the second graph, "Optimal Action vs. Steps" the greedy method only found the optimal actions in around 20% of the tasks; besides, the ε-greedy methods perform better. With the ε=0.01 method, it improves slower than ε=0.1, but it performs better on both performance measures. Among these, the ε=0.1 method showed the most promising results, gradually improving over time and eventually outperforming all other methods in terms of both average rewards and the percentage of optimal actions taken.

Overall, these observations underscore the significance of balancing exploration and exploitation in reinforcement learning tasks. While exploiting known optimal actions can yield short-term gains, consistent exploration is crucial for discovering potentially superior strategies and maximising long-term rewards. To conclude, when Epsilon(ε) is bigger, the multi-arm bandit needs to be balanced between exploitation and exploration more and, in the long run, acquire more rewards.

## Problem 2:

When dealing with multi-armed bandits, there are two fundamental strategies for maximising collective rewards: exploration and exploitation. Exploration involves trying out various arms to collect information about the best action to maximise rewards. When exploring, the agent targets arms that are yet to be thoroughly investigated or those whose true reward potential remains uncertain. Its aim is to discover more about the environment and enhance the agent's understanding of which actions produce the highest rewards. This usually involves being more "long-sighted" and sacrificing immediate gains in the interest of gathering insights that can result in more informed decisions in the future. In contrast, exploitation involves selecting known optimal actions based on past outcomes to maximise short-term rewards, reflecting a more "short-sighted" and "greedy" approach focused on immediate gains. For example, in online banner advertising, exploration may involve exhibiting a new advertisement, while exploitation would be prioritise displaying the most successful ones.

Effective multi-armed bandit algorithms require a balance between exploration and exploitation for optimal performance. Overemphasis on exploration can result in suboptimal rewards in the short-term by neglecting already known high-reward actions, while excessive exploitation may lead to premature convergence to suboptimal actions, overlooking potentially high-reward actions that have yet to be explored.

Many algorithms, such as the epsilon-greedy approach mentioned above, have been developed to address the trade-off between exploration and exploitation efficiently in various situations. In the epsilon-greedy algorithm, an option is randomly selected with a probability of epsilon (ε) during exploration, while exploitation focuses on the option that appears to be optimal with a probability of (1- ε). Epsilon represents the likelihood of the agent exploring instead of exploiting. This enables exploration of non-optimal actions while still taking advantage of the currently known optimal ones. Choosing the appropriate value for epsilon is usually based on the exploration-exploitation trade-off. Higher values encourage more exploration, while lower values favour more exploitation, as displayed in the provided figures.

## Problem 3:

Definition: Action-value estimation is the process of evaluating the anticipated reward linked to a particular action. This fundamental concept plays a crucial role in Reinforcement Learning (RL), where agents engage with an environment, make decisions, and subsequently receive feedback based on their chosen actions. By estimating the value of different actions, RL agents can learn to optimise their behaviour and achieve desired outcomes.

Goal: In the context of a multi-armed bandit problem, the objective for a learning agent is to maximise its expected reward across multiple trials. To achieve this, action-value methods come into play. These algorithms define the action-value function Q(a), which represents the expected reward associated with taking a specific action a, given the current state (in this scenario, there exists only one state).

Action Value Function: To estimate the action-value function, a sample-average method is employed. After each trial, the estimated value of Q(a) is updated based on the reward received for that action. The update rule for the action-value function can be expressed as follows:

$$\boldsymbol{Q(k+1) = Q(k) + \frac{1}{k}[R(k) - Q(k)]}$$

In this context, Q(k+1) represents the estimated action-value function for action a in the next selection, while Q(k) corresponds to the current estimate for action a after being selected k times. The reward R(k) is associated with taking action a during the kth trial, and k denotes the total number of times action a has been chosen. The update rule employs a step-size parameter of 1/k, ensuring that the action-value estimates converge to true values as the number of trials approaches infinity. This approach is commonly referred to as the sample-average method.

Action selection in this scenario follows an epsilon-greedy policy. Here, the agent selects the action with the highest estimated value with a probability of 1-epsilon, while it randomly chooses an action with a probability of epsilon. Overall, action-value methods offer a straightforward yet powerful approach to address the multi-armed bandit problem, and they can be further adapted to tackle more intricate reinforcement learning problems.

## Contribution Table

| Name | Student ID | Contribution |
|---|---|---|
| Chaiyatorn Permpornsakul | 201768687 | Problem 1 - Code |
| Chinwa Chimdi-Ezekwe | 201751919 | Problem 2 |
| Mehrnaz Miri | 201729365 | Compilation, overview, formatting |
| Neda Yavari | 201765565 | Problem 3 |
| Yukabed Ijadi | 201773485 | Problem 1 – Discussion and results |

## References

Agrawal, R. (1995) 'Sample mean based index policies with O(log n) regret for the multi-armed bandit problem', Advances in Applied Probability, 27(4), pp. 1054-1078.

Auer, P., Cesa-Bianchi, N. and Fischer, P. (2002) 'Finite-time analysis of the multiarmed bandit problem', Machine Learning, 47(2-3), pp. 235-256.

Berry, D. A. and Fristedt, B. (1985) Bandit problems: Sequential allocation of experiments. Boca Raton, FL: Chapman and Hall.

Bubeck, S. and Cesa-Bianchi, N. (2012) Regret analysis of stochastic and nonstochastic multiarmed bandit problems. Hanover, MA: Now Publishers Inc.

Lattimore, T. and Szepesvári, C. (2020) Bandit algorithms. Cambridge: Cambridge University Press.

Q. Nguyen, N. Teku and T. Bose, (2021) ‘Epsilon Greedy Strategy for Hyper Parameters Tuning of A Neural Network Equalizer,’, 12th International Symposium on Image and Signal Processing and Analysis (ISPA), Zagreb, Croatia, pp. 209-212

Richard S. Sutton, and Andrew G. Barto. (2018) Reinforcement Learning, Second Edition: An Introduction. Bradford Books

Sutton, R. S. and Barto, A. G. (2018) Reinforcement learning: An introduction. Cambridge, MA:MIT Press.