Programming Assignment 1

DA5007: Special Topics in ML (Reinforcement Learning)

1 Problem Statement

To understand the effectiveness of the greedy, ϵ greedy, and UCB strategies for selecting actions from a multi-armed bandit problem, this assignment investigates their performance on a classic 10-armed bandit testbed.

10-armed bandit testbed

The 10-armed bandit problem is a special case of the $\mathcal{K}$-armed bandit problem, where $\mathcal{K} = 10$. The agent faces 10 independent actions or "arms," each associated with an unknown reward distribution $\mathbf{R_i}$ for $i = 1, \ldots, 10$. The true mean reward of each arm is denoted by $\mu_i = \mathbb{E}[\mathbf{R_i}]$, which is initially unknown to the agent and fixed (stationary) over time.

- At each time step t, the agent selects an action $a_t \in 1, \ldots, 10$ and receives a reward r_t sampled from the distribution associated with that arm.
- The agent's goal is to maximize the cumulative reward over a fixed horizon T (e.g. $T = 1000$ steps) by balancing exploration (trying out different arms to gain information) and exploitation (choosing the current best-known arm).
- The value or action-value of an arm a is $q_*(a) = \mu_a^*$
- The agent maintains estimates $Q_t(a)$ of $q_*(a)$ based on observed rewards up to time t
- The regret over $\mathbf{T}$ steps is defined as

  $$\Delta_T = T\mu^* - \sum_{t=0}^{T} r_t$$

  where $\mu_* = max_a \mu_a$ is the maximum mean reward.
- The objective is to minimise expected regret, or equivalently, maximise the expected sum of rewards.

2 Tasks

10-Armed Bandit Testbed (5 Marks)

You will simulate a 10-armed bandit problem:

- Each arm a has a true value $q_*(a)$ sampled from a Gaussian distribution with mean zero and variance 1.

- Each time an arm a is selected, the reward is sampled from a Gaussian distribution with mean $q_*(a)$ and unit variance
$$r_a = \mathcal{N}(q_*(a), 1)$$

Using the above bandit problem, generate 2000 independent bandits. Use the sample average method with incremental implementation for learning the action values ($Q(a)$) for each arm. (In the report, share a snapshot of the code for this task as well.)

Greedy and ϵ-greedy Algorithms (30 Marks)

Run the following algorithms on the 10-armed bandit testbed:

- greedy
- ϵ-greedy with $\epsilon = 0.1$ and $\epsilon = 0.01$

For each algorithm, plot the following learning curves averaged over 2000 bandit problems for 1000 timesteps:

1. **Average reward vs. time**
2. **% Optimal action vs. time**

See Figure 1 for a reference for plotting the graphs.

Hint: To compute the average reward at timestep **t**, *sample the reward from all 2000 bandits at time* **t** *and take the mean. The optimal action percentage is computed by checking how often the best arm was selected at each timestep, averaged over all bandits.*

(Include in the report a snapshot of the code showing how rewards and the percentage of optimal actions are computed and averaged over timesteps and across all bandit instances.”)

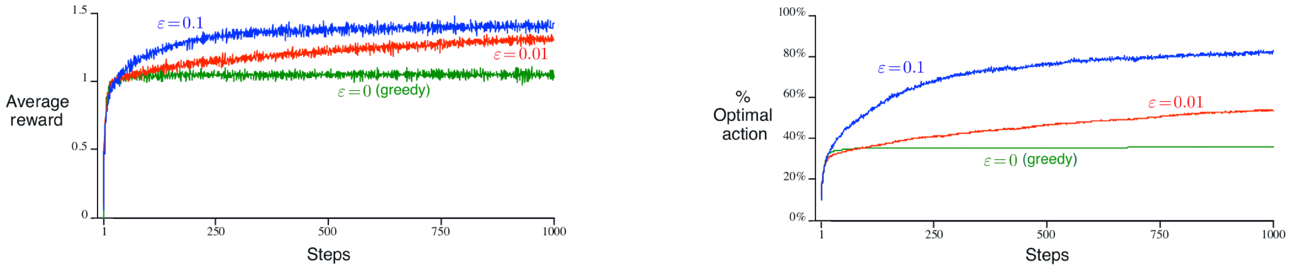

Figure 1: Comparison of greedy and ϵ-greedy algorithms

Upper Confidence Bound (UCB) Algorithm (25 Marks)

Implement the UCB algorithm on the 10-armed bandit testbed. Generate learning curves for **average reward versus time**, averaged over 2000 bandit instances for 1000 timesteps, and compare the results with the ϵ-greedy algorithm for $\epsilon = 0.1$.

For action selection, use the Upper Confidence Bound (UCB) formula instead of the one shown in the lecture slides. Note that the formula in the slides is a special case of this formula when $c = \sqrt{2}$.

$$A_t = arg \max_a \left[Q_t(a) + c\sqrt{\frac{\ln t}{N_t(a)}} \right]$$

See Figure 2 for a reference for plotting the graphs.

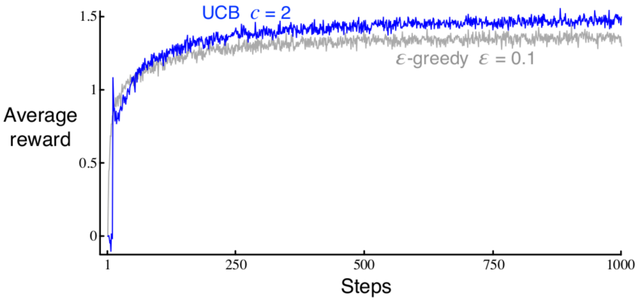

Figure 2: Average performance of UCB action selection on the 10-armed testbed

Analyse the effect of the exploration parameter c by running the UCB algorithm with different values of c, such as $c = 1$, $c = 2$, and $c = \sqrt{2}$. Compare the resulting learning curves (**average reward vs. time**) and discuss how varying c influences exploration and convergence.

Inferences and Conjectures (30 marks)

Write inferences and conjectures from all your experiments and results

Large Bandit Testbed (10 Marks)

Repeat all the above experiments using 10,000-armed bandits. Instead of simulating 2000 independent bandits, you may use a smaller number, such as 200–500 bandits. Plot all the necessary learning curves for the experiments.

3 Submission Instructions

- You are required to create a comprehensive PDF report containing key code snippets, observations, insights, and plots.
- You must also submit the Colab or Jupyter Notebook containing all your experiment code. Ensure the notebooks are self-explanatory with clear markdowns and comments, and include instructions to reproduce your results. We will run your code, which must generate the same plots and charts as shown in the report.
- In your report, include only the important code snippets rather than every line of code.
- Zip your report and code files together and submit the zip file through the portal.
- Submission of both the report and code files is mandatory. Incomplete submissions will not be evaluated and will receive 0 marks.
- Use clear and meaningful code comments for readability, and follow all problem specifications carefully.