Assignment 2

Due: Nov 15, 2023

<u>Bonus Points</u>: There will be 10 bonus points across the questions, where the answers go above and beyond the mandate of the question. For instance, in Q1, you can provide any novel insights on performance of a specific MAB algorithm.

Q1. (Programming Exercise - 10 points)

Consider a six-armed bandit problem, where each arm either gives a reward of 1$ or nothing. The probability of yielding a reward of 1$ for each arm is given by:

Arm 1	Arm 2	Arm 3	Arm 4	Arm 5	Arm 6
0.55	0.45	0.3	0.40	0.35	0.48

Simulate the different action selection algorithms, i.e.,

1. Greedy,
2. Epsilon-Greedy with epsilon = 0.1
3. Softmax
4. Upper confidence bound based action selection.
5. Thompson Sampling

perform over 5000 iterations. Please make assumptions on parameters as required.

Repeat the experiment 100 times for each algorithm. Plot the average reward earned with each of the algorithms over the 5000 iterations.

Q2. (Programming Exercise - 15 points)

Implement the Pong example described in class (Lecture 10) using the Gym or Gymnasium environment using both DQN and Reinforce. In the class, I explained how Reinforce is implemented on Pong. Here the target is for you to implement both DQN and Reinforce using a single hidden layer neural network (for either policy or value). Compare and contrast the training and testing performance of DQN and Reinforce.

Please provide the code and the results.

Note: You are free to study and explore other implementations, but you are expected to write your own code similar to what was described in class.

Q3. (By Hand or Programming – 10 points)

Consider a Restless Multi-Arm Bandit Problem, where there are four arms, all having the same MDP model.

States: 0, 1

Actions: 0 (passive), 1 (active)

Transition Probabilities:

Action = 0	0	1
0	0.7	0.3
1	0.25	0.75

Action = 1	0	1
0	0.2	0.8
1	0.05	0.95

Rewards, R(s): R(0) = 0; R(1) = 1

Horizon is 2 and Discount factor is 1. **Compute Whittle Index for each of the arms at the first-time step** (i.e., one more decision to go after the current decision), if they are in the following states:

Arm 1, State 0

Arm 2, State 1

Arm 3, State 1

Arm 4, State 0

Q4. (Programming – 10 points)

In this question, you are expected to code up the behavior cloning algorithm discussed in class.

Please take the cartpole-v0 example from Gym and find expert trajectories. Please use the DQN algorithm from the StableBaselines library.

For the obtained set of expert trajectories, please train the policy neural network (single hidden layer) as per the Behavioral Cloning algorithm. The obtained policy network is the imitation learning solution.

Plot how well the policy learnt by Behavior Cloning performs across multiple simulations (10) on average for 5 different numbers of expert trajectories.