Homework 3

Oct 12, 2022

In this homework assignment you are going to implement the k-bandit algorithm using both ϵ-greedy algorithm and upper confidence bound algorithm.

Task 1 - Make One-step Decision Using ϵ-greedy

The formula is

$$A \leftarrow \begin{cases} \arg\max_a Q(a), \ \textit{with probability } 1-\epsilon \ (\textit{break ties randomly}) \\ \textit{a random action, with probability } \epsilon \end{cases}$$

You are supposed to complete the **e_greedy** function in the *kBandit.py* file. In this function, you choose an action following the ϵ-greedy algorithm.

There are two input parameters of the function **e_greedy**. (1) Q - A dictionary. The keys are the possible actions. The values are the average reward you got when taking the action. (2) ϵ - a scalar between 0 and 1.

The return value of the function **e_greedy** is a scalar. It represents the action which you are about to take if you follow the ϵ-greedy algorithm.

Task 2 - Make One-step Decision Using Upper Confidence Bound

The formula is

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

You are supposed to complete the **upperConfidenceBound** function in the *kBandit.py* file. In this function, you choose an action following the Upper Confidence Bound algorithm.

There are two input parameters of the function **upperConfidenceBound**. (1) Q - A dictionary. The keys are the possible actions. The values are the average reward you got when taking the action. (2) N - A dictionary. The keys are the possible actions. The values are the number of times you took the action. (3) c - A scalar.

The return value of the function **upperConfidenceBound** is a scalar. It represents the action you are taking if you follow the Upper Confidence Bound algorithm.

Task 3 - Update Q and N

You are supposed to complete the **updateQN** function in the *kBandit.py* file. In this function you update the Q and N dictionary.

There are four input parameters of the function **updateQN**. (1) action - A scalar indicating which action you took. (2) reward - A scalar indicating the reward you got from taking the action (3) Q - A dictionary. The keys are the possible actions. The values are the average reward you got when taking the action. (4) N - A dictionary. The keys are the possible actions. The values are the number of times you took the action.

The return value of the function **updateQN** is a tuple containing the updated Q and N.

Tip: Make copies of the dictionaries so that you do not change the initial Q and N.

Task 4 - Make Multi-step Decisions

You are supposed to complete the **decideMultipleSteps** function in the *kBandit.py* file. In this function you iterate through your one-step decision making process for *maxStep* times.

There are five input parameters of the function **decideMultipleSteps**. (1) Q - A dictionary. The keys are the possible actions. The values are the average reward you got when taking the action. (2) N - A dictionary. The keys are the possible actions. The values are the number of times you took the action. (3) policy - A function showing what policy you are using. (4) bandit - A function that gives you the reward of your sample. (5) maxStep - A scalar showing how many steps you have.

The return value of the function **decideMultipleSteps** is a dictionary containing keys "Q", "N", and "actionReward". The values of the keys "Q" and "N" are the updated dictionaries Q and N. The value of the "actionReward" is a **list of tuples** (action, reward) that records your action and reward of each step.

Test examples

For this problem, your code should print out a figure like Figure 2.2 upper panel on page 23 in *Reinforcement Learning: An Introduction (second edition)*. It will not be exactly the same because we do not use the same setting.

Submission

Please submit a completed *kBandit_YourLastName_YourFirstName.py* file on Canvas before due. **The due date and time of this homework assignment is Tuesday, 10/18/2022 11:59pm**.