Lab Assignment 1: Contextual Bandits

1 Introduction

In the discussions we have had in lecture our agent has not been allowed to use any extra information as it develops a solution to its Multi-armed Bandit choice problem. Contextual bandits (CB) are aimed at giving the agent the ability to incorporate additional information beyond the history of which actions have been tried. There are many CB algorithms that depend on the structure of the context data – in this lab you will focus on building a CB agent using function approximation.

You will work with a simulated scenario where an online platform is recommending articles to users. The goal is to maximize the click-through rate (CTR) by selecting the most relevant article for each user based on the contextual information associated with them. Assume that there are multiple possible actions to take (article recommendations) and that the reward is a click or not.

1.1 The Environment

Reinforcement learning generally involves two components: 1) a simulator that generates environmental data; 2) the algorithm that enables the agent to learn and implements its decisions. In this situation, the simulator must generate contextual data and provide the reward to the agent for showing an article. This environment has the following components:

- A set of actions–we will assume that there are five articles that you could possibly recommend so that $A = \{1, 2, 3, 4, 5\}$ – each of these is a decision to show one of the five articles to a user;
- A set of rewards, $R = \{0, 1\}$ – a reward of 1 is associated with a click;
- A set of features X that describes the context facing the agent each time it needs to make a decision, for example:

t	feature 1	feature 2	feature 3	feature 4	feature 5
1	0	0	3.3502	0	71
2	1	1	1.2263	0	86
3	0	1	2.1973	0	69
4	1	1	5.1876	2	70
5	1	1	2.7689	0	76
⋮	⋮	⋮	⋮	⋮	⋮

Future labs will involve implementing environmental simulators; in this lab I will give you a simple code to generate contextual data.

1.2 The Agent

In general, the agents that we build should include all the information needed to learn and make decisions. In the case of the Contextual Bandit agent in this lab, the agent should include a set of function approximators, one for each of the five types of article $\{f_1, f_2, f_3, f_4, f_5\}$, that measure the probability of a click. It should also include a policy that makes use of the function approximators to decide what action to do.

2 Learn about the environment

1. Download the Contextual_Bandit_Simulator notebook from canvas. This notebook contains a class that implements the components of a CB environment.
2. Set up the class by specifying a dictionary: env_Config = {'n_samples':100000} and passing this into Contextual_Bandit_generator.
3. Experiment with this class by running the step function with randomly generated values for a for 100, 1000, 10000, and 100000 turns.

3 Implement a simple Contextual Bandit

1. **Implement an agent** capable of taking actions in this environment by writing python code that executes the following algorithm:

Algorithm 1 Contextual Bandit

```
Require: $b, T \in \mathbb{N}, \varepsilon \in [0,1]$                    ▷ Set hyperparameters
 1: Init: $f_a$ for all $a \in A$                          ▷ Instantiate function approximators
 2: for $t = 1, 2, 3, \ldots$ do
 3:     Sample a new context $X$ using step()                                  ▷ add data
 4:     if $t \leq T$ then
 5:         Sample $a$ from $A$ uniformly                             ▷ choose a random action
 6:         Sample $r$ using reward($a$)
 7:     else if $t > T$ then
 8:         Sample $a$ from $A$ using $\epsilon$-greedy so that:
                $$a = \begin{cases} \text{any } a \in A & \text{with prob } \varepsilon \\ \text{argmax}_{a \in A} \{f_a(X)\} & \text{with prob } 1 - \varepsilon. \end{cases}$$
 9:         Sample $r$ using reward($a$)
10:         Every $b$ turns refit $f_a$ for all $a \in A$            ▷ update function approximators
11:     end if
12: end for
```

2. For this algorithm you will need to choose T, b, ε, and the model architecture for the function approximators. Use $T = 500$. For the function approximators, use untuned random forest binary classifiers.

3. **Compute average reward** over time for this algorithm running it for 10,000 turns for $b = 100$ or 1000, and $\varepsilon = 0, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5$.

Lab Assignment Report

Write up your analysis in a separate report of no more than 2 pages. Your report should answer the following questions:

1. Define the MAB problem and provide an example where the MAB framework can be applied.

2. In what situations might a purely greedy strategy (always choosing the arm with the highest estimated reward) perform poorly? Provide an example scenario.

3. Consider the performance of the $A/B/n$ testing algorithm (explore then commit). If you were to run this many times with various values of T how often would the algorithm correctly identify the best action and how would this depend on T?

4. How does the Contextual_Bandit_generator class work? Add docstrings and comments to the functions in your code and answer this question in your report.

5. Create a bar chart plot of the rewards for the experiments run in section 2.3. What should you expect the average reward to be in theory? How close do your plots come to this and how does this relate to the number of turns?

6. Explain how you set your contextual bandit agent up. What are the details of how you incorporated the function approximators?

7. Compare the performance of the CB for the different choices of b and ε by plotting the average reward over time that you computed. How do b and ε affect the CB agent?

8. What advantages do you think the CB agent might have over simply treating this as a multiclass supervised machine learning problem?

4 Submission Instructions

- Write up the answers to the questions in a short word document; aim for no more than 2 pages of text and include all graphics generated. Add footnotes identifying which sentence addresses which questions. Write in complete sentences organized into paragraphs – your goal is to explain what you've done and what you've learned to your audience (me!). Include the appropriate plots you've generated as mentioned above. Convert this to pdf and submit it. Submit your .ipynb file and an .html of it as well.

- The grading rubric for this assignment will be available in Canvas.

- NO OTHER SUBMISSION TYPES WILL BE ACCEPTED.

- You are welcome to use generative AI as you code up your contextual bandit. If you do this you must include all prompts used as an appendix in your written report or provide a shareable link to them. Any uncited use of generative AI (i.e. prompts not provided) will be considered plagiarism. You may not use generative AI to write your report.