Homework 3 — STAT/CS 387: Data Science II
Multi-armed Bandits

Instructions

Please read these instructions carefully before you begin!

Your writeup

While completing the below problems, prepare a formal writeup that addresses questions given as part of those problems as well as addressing the following points:

- Describe the multi-armed bandit problem in your own words, why it is a good model for dynamic A/B/n testing, and what limitations it may have for practical problems.
- In your simulations, what are the distributions D that you chose for your arms and why did you choose them? What did you do to make the **easy** bandit easy and the **hard** bandit hard? Include plots of the true reward distributions, one for the easy bandit and one for the hard bandit, and refer to them in your answers.
- How should one evaluate the performance and accuracy of bandit strategies? Which of the strategies from Part 2 is the best according to your criteria/metric and why?
- Describe what UCB1 does in your own words. How well does it work? Why does it work as well (or as badly) as it does?

Please show your work! This means (i) provide the code you have produced by placing it in `work/`, (ii) next to any answers in your writeup include a parenthetical statement pointing out where in your code the answer was computed; for example: "(See plot_timeseries.py, lines 100-110.)"

To submit

Name your final writeup `HW03_writeup_[NETID].pdf` (Ex: `HW03_writeup_jbagrow.pdf`). Place the writeup in a directory called `HW03/`, and place your (readable, well-documented) code in `HW03/work/`. Then rename `HW03` to `HW03_[NETID]`, compress (zip) the directory, and upload the `HW03_[NETID].zip` file to Blackboard.

Problem 1. **Simulating MABs**

Implement the MAB model so that you may simulate the problem with algorithms for below. MAB.html is helpful here. To do this, you will need to construct well-documented, readable code that:

- sets up the probability distributions and (true) mean rewards for each of the N arms,
- returns a reward r_i taken from distribution D_i when arm i is played,
- implements **gambles**, the sequential playing of arms over T timesteps,
- computes the **regret** R of a strategy as a function of time over the course of a gamble.

While the *expected* regret discussed in class is a mathematical entity because the expectation is over all possible gambles, here in simulations you can "sample" the expected regret by performing many simulations and averaging R for each one.

- Make sure your code is efficient so that you can average over many gambles.

Distributions of reward Choose probability distributions D for the rewards that have bounded support on the interval $[0, 1]$. A good choice is the Beta distribution, which has two parameters that let you tune the mean reward value.

In your write-up, describe your implementation of MAB simulations and discuss and motivate your choice of reward distribution. You should probably include plots to visualize your reward distributions.

Problem 2. **Strategies for playing MABs**

Implement with your own code the following algorithms/strategies/policies for playing arms during the course of a gamble:

1. Random
2. (Naive) greedy
3. ϵ-first greedy
4. ϵ greedy

These strategies were discussed in class. Some of these strategies require data structures that store the history of rewards received per arm so that you can compute estimates $\hat{r}_i$ of the mean reward $\bar{r}_i$.

- **Note**: While ***you*** have access to the true parameters of the arms, and you need that information to evaluate performance (Part 3), **these *strategies* must not see that information**. They can only have access to the rewards they receive after they play an arm. Otherwise, you are not implementing a realistic model of the uncertainty inherent in dynamic A/B testing or, more generally, reinforcement learning. Put another way, if you already know what version of the website is best, why would you be doing A/B testing in the first place?

In your write-up, describe these strategies, describe and document your code implementation, and discuss how you ensure your strategies are not "cheating" by seeing information they should not have access to.

Problem 3. **Evaluate performance**

Use what you have built in Problem 1 to implement two different five-armed bandits (i.e., $N = 5$):

1. An **easy** bandit, where the true rewards are distinct enough that you expect it is possible to identify the optimal arm;
2. A **hard** bandit, where the true rewards are such that strategies are likely to struggle finding which arm is optimal.

Evaluate the performances of the strategies (Part 3) over the course of gambles of length $T = 1000$. For each strategy, show with plots in your write-up:

1. The regret of an individual gamble (plot $R(t)$ vs. t)
2. The expected regret averaged over 100 gambles (plot average $R(t)$ vs. t)
3. Another *metric* of performance different from regret of your own choosing, also averaged over 100 gambles and as a function of time.

Any time a reward is received, the regret must be accounted for. Otherwise, strategies may be able to "play for free".

Put some thought into what is an appropriate metric to evaluate how efficient/accurate the strategies are. Organize the above plot(s) in a clear, logical, easy-to-follow format. A single plot with 50 curves on it will be too hard to understand. Readability is a grading criterion for this assignment.

In your write-up, describe the second metric you introduced (what it measures and why it is appropriate), and interpret your results and plots to determine what is the best method from Part 2 and why is it the best. Include informative captions with all plots.

Problem 4. **But suddenly, a new contender has emerged**

As per Problem 2, implement the UCB1 algorithm discussed in the reading. Then use the code from Problem 1 to evaluate the performance of this new method as per Problem 3. Make new plots directly comparing this algorithm to the "best" algorithm identified during Problem 3.

Provide a **new section** in your write-up called “UCB1 Performance” presenting the results of Part 4 after your section presenting the results of Part 3.