RBE595 - Week 2 Assignment

Keith Chester

Due date: January 22, 2023

Question 1

What is the benefit of incremental update for action-value function, versus non-incremental?

The benefit of this approach is it's faster (you're only doing one computation after each step instead of many after each step) and it saves space in memory.

Question 2

Why balancing exploration/exploitation is important? Name a few ways to balance exploration/exploitation in multi-armed bandits.

Consider two scenarios - both extremes:

If we focus entirely on exploitation, then we act only on the information we presume to know. Thus we underperform, as we possibly never find the true best action. If the value of reward for a given action changes over time, we may never discover the change.

Alternatively, if we focus entirely on exploration, we may learn what the best choice is, but our insistence on constantly exploring for new options prevents us from maximizing our score.

By balancing the two healthily, we can try to act greedy enough of the time to reap a high reward, but still leave the possibility of exploration to discover if there are better approaches for our agent.

A few ways that we can balance exploration / exploitation in multi-armed bandits problems are:

- Epsilon-Greedy - exploration based on random chance, otherwise greedy (take the best expected reward)
- Gradient Bandit - to encourage the agent to utilize exploration, applies a numerical preference in the direction of actions which exceed a given reward baseline
- Upper Confidence Bound - utilizes probability to introduce uncertainty in exploitation of expected rewards
- Optimistic Initial Value - utilizes early exploration to find an optimistic target

Question 3

3- In the equation belowwhat is the Target?

$$NewEstimate = OldEstimate + StepSize * [Target - OldEstimate] \tag{1}$$

In this equation, Target is the ground truth that we are comparing against; ie the actual discovered reward.

For instance, if I suspect that a given action will yield reward of 1, but find when choosing it I receive a reward of 2, with a step size of 0.1, we'd find:

$$NewEstimate = 1 + 0.1 * (2 - 1) = 1.1 \tag{2}$$

This approach allows us to incrementally change the reward estimate slowly, such that we avoid large swings in changing our estimate but ultimately find the correct average reward of the action.

Question 4

What is the purpose of using Upper Confidence Bound (UCB)?

UCB introduces a new method of utilizing exploration. Consider our implemntation for the week 2 programming assignment, wherein we mdae an ϵ greedy approach where one randomly decides to explore, and then chooses with equal probability one of the other action. While this approach works, it can underperform for two reasons:

- When we explore, we choose with an even probability; thus we are routinely choosing actions with low return of rewards compared to other, known good rewards
- If the rewards are not drastically changing to some rule over time, our need to explore diminishes in time, yet we continue to explore at the same probability.

To this end, UCB addresses both of these issues. As time goes on, the chances of exploration of other actions diminishes. Likewise, when we do explore, we do not choose a new action with a uniform distribution but rather weight the our random choice by our knowledge of its rewards.

Question 5

Why do you think in Gradient Bandit Algorithm, we defined a soft-max distribution to choose the actions from, rather than just choosing action?

We do this to weight our choice, such that we still have a random chance to choose any possible action, but the probability of that chance triggering is based entirely on our presumed knowledge of its potential reward. Thus, we are exploring, but in a manner that still aims to maximize our reward.

Question 6

Read the article below and summarize what you learned in a paragraph: https://www.spotx.tv/resources/blog/developer-blog/introduction-to-multi-armed-bandits-with-applications-in-digital-advertising/

The article discusses the application of the k-Arm Bandit problem (called Multi-Arm Bandit, or MAB, in the article) in the context of online advertising, where you have a global marketplace of consumers to barrage with your ads and you are trying to maximize clickthrough rate. The problem is that you have a large collection of ads to show, but are ultimately unsure which ones are best. Normally consumer-facing changes like this are A/B tested, but this doesn't work at scale due to the need for a human-in-the-loop to make decisions and it increases the amount of time you're performing at a sub-par click through rate while you do your initial testing. Instead, Epsilon-Greedy is proposed and demonstrated as an approach to find the best performing ads in a rapid manner. Then the article discusses the Thompson Sampling approach, which utilizes a Bayesian probability distribution to determine the relative likelihood of a click-through rate (our reward) for a given ad. Finally, the article begins talking about Regret, a measure of what we missed out by not choosing the best option available to us. We can measure the regret of ϵ-greedy and Thompson sampling to determine which approach is better for a given timeline of exploration/exploitation. Thompson Sampling vastly outperforms ϵ-greedy in minimizing this regret metric.