Comp 138 RL: Homework 1

Matt McDermott

September 25, 2020

Background

Multi-armed bandit problems describe the class of problems in which an agent must repeatedly pick an action to maximise some type of reward function without having prior knowledge of the true value of the expected reward for each possible action (*Gittins*). The problems get their name from slot machines (known as one armed bandits) because of the common strategies utilized by the player in order to maximize reward per unit input. The applications of multi-armed bandit problems are not limited to casinos- the stock market, governmental resource management and even picking volunteers for medical trials are all systems that can be modeled with the same strategies used in multi-armed bandit simulations. At its core, the multi-armed bandit problem is an optimization of exploration; trying out different options to see which available action has the best reward at the cost of exploitation- performing the best available action and actualizing potential reward.

In the most simple static case, after a large enough number of trials, expected values of rewards for each possible action can be estimated within some margin of error. In the static case, it may make sense to explore as widely as possible at the beginning of a trial until an optimal reward has been identified, then switch to exclusively exploiting the corresponding action. Unfortunately, this strategy does not work in situations in which the reward functions may change over time. These non-stationary cases add another layer of difficulty to the problem and require more advanced techniques to achieve satisfactory results.

Problem Statement

The goal of this exercise was to conduct an experiment to demonstrate the effectiveness of sample-average and weighted-average methods in the context of a nonstationary k-armed bandit problem. Using the "10-armed testbed" described in the text, both sample-average and weighted-average agents were created according to the provided specifications.

Methods

Generally, the predicted reward for any bandit Q is a function of the previous estimate, a step size parameter α, and the reward received on the current action R.

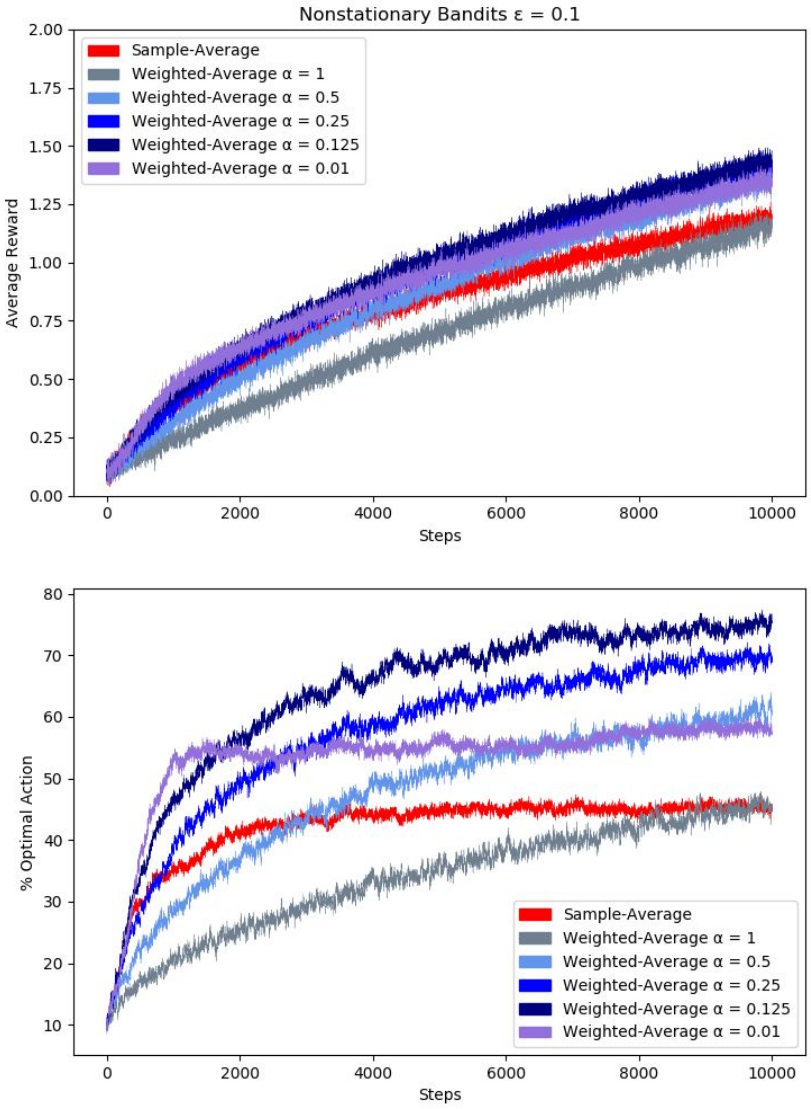
$$Q_{n+1} = Q_n + \alpha[R_n - Q_n]$$

In the case of a sample-average method, α is set as 1/n where n is the number of times the bandit has been chosen. This makes the predicted reward to be an average of all past rewards. Alternatively, setting α to a constant value (0,1] makes the estimated reward to be a weighted average, favoring recent trials more heavily. In the extreme case, setting α to 1 sets the expected reward to be equal to the result of the most recent trial, and setting α to a value close to zero achieves a similar result to that of the sample-average case.

The simulation begins by initializing 10 bandits, each with normally distributed reward functions centered at +0.1 with standard deviations of 1. For each timestep, the reward for each follows a random walk with standard deviation 0.01. The initial estimate of reward for both SA and WA methods was set as 0.5, which causes more exploratory behavior at the beginning of each trial run. For both cases, ε was set to 0.1, meaning that 10% of the time each method would choose a bandit at random. The other 90% of the time, each agent would select whichever bandit had the highest estimated reward. Rewards would be randomly generated for the selected bandit and the results would be recorded and estimates updated before repeating again in the next timestep. This process was repeated for 10,000 timesteps for each trial.

A monte carlo study was constructed and the above simulation was run for 2000 trials at various alpha values. Values of average reward and percent optimal selection were recorded for each trial and averaged across all 2000 trials and plotted.

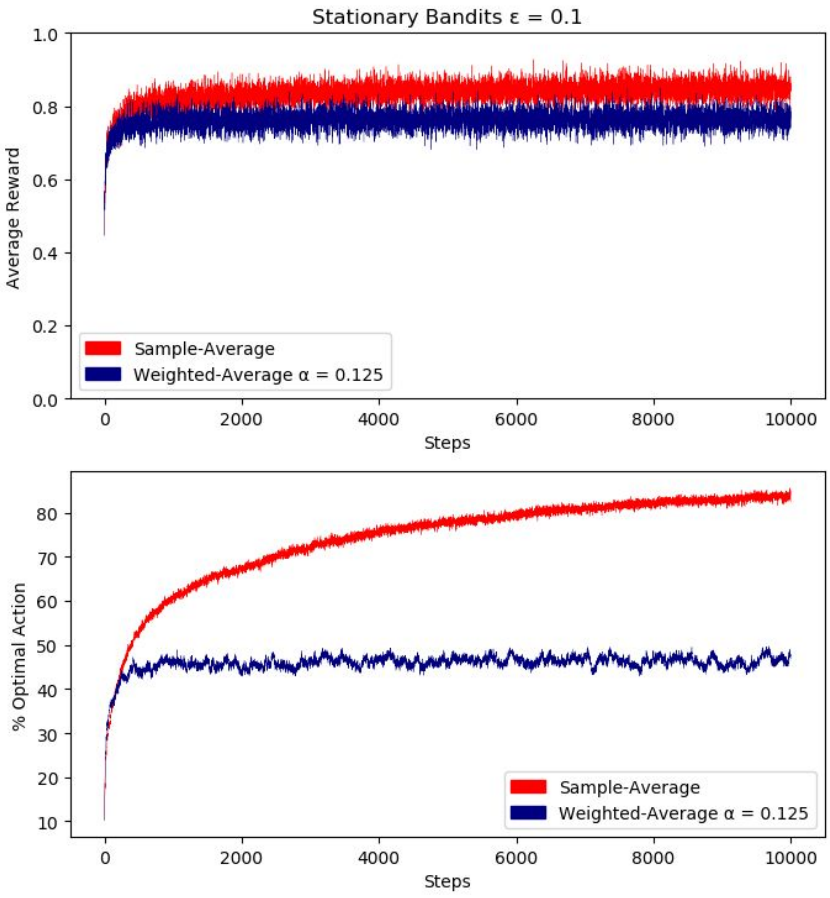
Results



The weighted-average agent with $\alpha = 1$ was consistently the worst performing agent in the experiment with the lowest average reward for all timesteps until just before t = 10,000 where it finally

began to catch up with the sample-average control method. For the first 1500 steps, the $\alpha = 0.01$ agent had the highest percent optimal actions until it reached around 55% at which point it leveled off sharply and was overtaken by the $\alpha = 0.125$ agent and then was passed by the $\alpha = 0.25$ agent around 2500 steps.

Using the most successful agent from the nonstationary experiment, a control run was conducted comparing the $\alpha = 0.125$ weighted-average method against the standard sample-average agent. Rather than beginning with 10 agents with equal reward functions that would diverge randomly throughout the duration of a trial, bandits were given constant rewards varying from (0, 0.1,..., 0.9) that would remain consistent throughout the experiment.



In these trials, the sample-average agents were consistently better with steady state optimal action approaching 90% (a perfect score with $\varepsilon = 0.1$) than their weighted-average counterparts that leveled off at ~45%.

Analysis

Because the average reward consistently increases at any given time step as α decreases from 1 to 0.125 and then decreases again at 0.01, the true best α value for the simulation conducted is likely between 0.01 and 0.125. While the weighted-average agent with $\alpha = 0.125$ produced the best long term results it is important to consider the application of the problem before choosing an approach. For example, if one were to enter a casino with 1000 quarters, it would make the most sense to actually choose the $\alpha = 0.01$ agent because for the duration of the first 1000 trials it actually performs the best out of all available agents.

A higher α value allows the agent to more quickly pick up on changes in the underlying reward functions of the bandits at the cost of potentially misinterpreting noise. Agents that rely heavily on recent rewards are more susceptible to becoming "distracted" by occasional trials that result in rewards that lie more than a couple of standard deviations outside the mean of a bandit's reward distribution. This is exemplified in the control graph where the weighted-average agent is not only picking a lower percentage of optimal actions but the variation in percent optimal actions between subsequent timesteps is much higher than the sample-average agent (meaning the WA agent's plot is much more jagged than the that of the SA agent), and thus it can be assumed that the weighted-average agent is switching back and forth between bandits much more frequently.

Obviously, in the situation above the sample-average agent has a significantly higher average reward so it would make sense as the better agent regardless, however, there are also additional benefits of having a smoother percent optimal selection curve. Generally, a jagged percent optimal selection curve is indicative of not only more frequent changes from the optimal bandit to a suboptimal one, but more changes in bandits overall. While not necessarily important for this problem, there are plenty of real world examples in which there are penalties associated with each change in bandit selection. For example, a bot running a high frequency trading algorithm would be subject to fees for every trade (aka change in bandit), so even if mean percent optimal actions were equal, a sample-average strategy may be more beneficial due to the lower number of overall trades needed.

Conclusion

Intuitively, it makes sense that the sample-average case is not as flexible to changes in the reward functions of the bandits because in a situation with a changing reward function because results from thousands of time steps ago are not guaranteed to be useful in predicting what is about to happen next. Thus, sample-average agents perform much better in situations where reward distributions are held constant, and weighted-average agents perform better in situations with drifting reward distributions. While extrapolation of specific results from this exercise should not be directly applied to other problems, the general trends found in this simulation are likely indicative of the general behaviors of sample-average and weighted-average agents.

References

Sutton, Richard; Barto, Andrew (1998), *Reinforcement Learning*, MIT Press, ISBN 978-0-262-19398-6, archived from the original on 2013-12-11.

Gittins, J. C. (1989), *Multi-armed bandit allocation indices*, Wiley-Interscience Series in Systems and Optimization., Chichester: John Wiley & Sons, Ltd., ISBN 978-0-471-92059-5