Part 2 - Project report

Årne, Runar and Lars

February 2, 2024

1 Introduction

For this project we will be looking into using MAB algorithms to learn which of a set of medicines are the most efficient. To do this we have a set of environments which model these hypothetical medicines which we will attempt to run multiple MAB algorithms on. In doing this we hope to find out which algorithms perform best in each environment. We will be using the algorithms ϵ-greedy, decaying ϵ-greedy and the UCB algorithm.

2 Results

For the first environment with medicines A,B and C we ran 20 experiments of 1000 episodes for each of our three algorithms. Initially the hyperparameters were set to $\epsilon = 0.3$, $\alpha = 0.99$ and c = 1. The left plot on figure 1 shows the mean cumulative regret for each algorithm.
We can see that the normal Epsilon-greedy model (blue) has linear regret, whilst both the Decaying epsilon-greedy model (orange) and the UCB model (green) have logarithmic asymptotic regret. This is easily explained by their exploration behaviour, as the normal Epsilon-greedy model continues exploring at the same rate all the time, regardless of how many episodes it has been running for. The Decaying epsilon-greedy model and the UCB model both gradually stop exploring the more sure they are that they've correctly sampled the expected reward of each action. This matches the asymptotic behaviour presented in class. For the second environment we have the same medicines as in the first with an additional medicine D. This MAB problem is solved in the exact same way as the other problem, however with 4 possible actions each episode (instead of 3). First we would need to update the expected value list to contain 4 values, instead of 3 (1 per medicine). However, there is actually no need in starting from scratch! All 3 different algorithms are based on estimating the expected reward per action (i.e. per medicine). By adding a new medicine we do not change the other three medicines, thus their expected rewards stays the same. We can thus

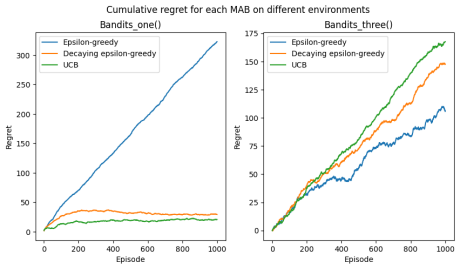

Figure 1: Mean cumulative regret for each algorithm for bandits_one and bandits_three

continue training the other models from where they are at now. However, some models (like the decaying epsilon-greedy) changes how much they explore based on how many episodes they are trained on. Therefore, adding another medicine 1000 episodes in might mean that the chance of exploring this new medicine is practically 0. In this case, it would probably be best to restart from scratch (or reset the epsilon value back to a reasonable place). In conclusion, it depends on which algorithm is being used, but one should not be forced to restart training from scratch.
Next we consider a completely new environment bandits_three, which has three new medicines X,Y and Z. Like we did in bandits_one and bandits_three we run 20 experiments with 1000 episodes for each algorithms. Then in figure 1 we can compare the cumulative regret with bandits_one.
From the plots we can see that Bandits_one follows the expected behaviour of each algorithm, whilst all algorithms behave noisy and linear on Bandits_tree. Comparing the two plots it is easy to see that our algorithms performed way worse on Bandits_three than on Bandits_one. Therefore, we can conclude that Bandits_one was the easiest MAB to solve. Inspecting how the different environments are set up we can understand why. Every reward in Bandits_one has a variance of 1, in Bandits_three, the third medicine has a reward with variance of 3. This means that the reward from medicine Z of Bandits_three could vary wildly compared to every medicine in Bandits_one. This also means that Bandits_three will be harder to solve, as it is harder to compute a representative expectation value.

Finally we consider an environment bandits_four with the same medicines as bandits_three and a gene G which affects the efficiency of each medicine. We train two bandits, one for G = 0 and one for G = 1 using the UCB algorithm. When inspecting the different learned optimal actions, we see that the bandit with gene = 0 flips between choosing action 1 and action 2. This obviously makes sense, as they have very similar means, and action 2 has higher variance. However, bandit gene = 1 consistently learns the actual optimal action 0. For the Bandits_three environment, all models predict that either 0 or 2 is the optimal action, where as the actual optimal action is 2. Thus, they share the optimal action for Bandits_four with gene = 0, but for gene = 1 they differ (as here the optimal action is 0). The reason they differ is because for Bandits_three, the different medicines have other mean values than what the medicines for each gene has in Bandits_four. However, in both Bandits_three and Bandits_four with gene = 0, the action with the highest mean is action 2. If we consider G as a continuous variable between 0 and 1 rather than a binary 0 or 1 the situation changes slightly. We need to take into account the continuous nature of the variable when estimating the expected value. We can see that the response of the three drugs are linear with respect to the continuous gene value. As $E[X|Gene = 0] = 1$ and $E[X|Gene = 1] = 3$, we can reasonably expect that $E[X|Gene = 0.5]$ can be linearly interpolated from the above values. Doing this we estimate that:

$$E[X|Gene = 0.5] = (E[X|Gene = 0] + E[X|Gene = 1])/2 = (1 + 3)/2 = 2$$