Assignment1

Bhavna Kandra

February 20, 2021

1 Contents of the code files

MATLAB environment has been used to generate the code for 20-armed bandit problem. Two scripts along with five function files are hereby attached to reproduce the results as discussed in this assignment. A brief up on the scripts/files is enlisted below:

- **multiplebandit.m** This file is the main file which runs through the entire 2000 sets of 20-armed bandit problems. Global variables like ϵ, Q0, variance and c can be tweaked here.
- **singlebandit.m** This file runs up a single bandit algorithm for 2000 time steps.
- **boxmuellernormalm0.m** This function generates a N(0,1) random number using the Box Mueller Method as discussed in the class.
- **boxmuellernormalmusigma.m** This function generates a N(mu,$sigma$) random number with using the N(0,1) random number generated previously.
- **chooseaction.m** This function chooses the next action to be taken using *epsilon*-greedy policy.
- **updateindicator.m** This function maintains and updates chosen action's frequency.
- **assignreward.m** This function assigns rewards as per qstar of the appropriate arm.

2 Analysis of the results

- Plots of both, average reward vs time and percentage optimal action vs time show initial transient fluctuations i.e until 100 time steps or so and then gradually increase with time only to saturate at a particular optimum value. This validates the generic trend observed in multi armed bandit problems: saturation when approaching the optimal trajectory.

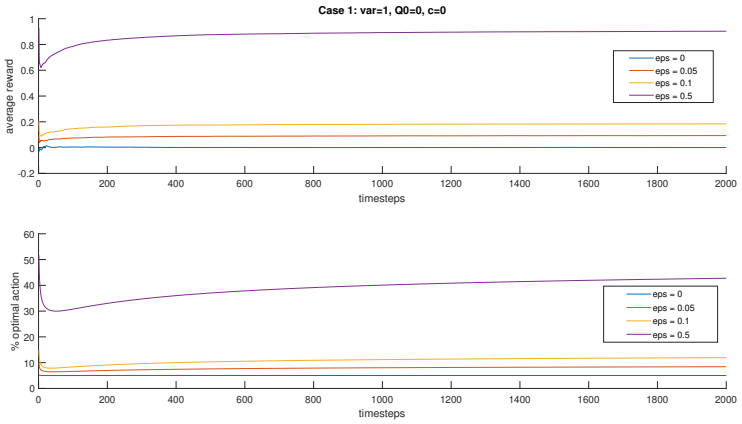

Figure 1: ϵ-greedy algorithm with various ϵ and variance = 1 (case 1)

- Sixteen comparison graphs with varying ϵ values are generated namely from case 1 to case 16, with multiple combinations of variance, Q0 and c. In all these graphs a general trend observed is : upon increasing ϵ from 0 to 0.5, the saturation value for average reward is increasing, which means on exploring more, the algorithm indeed finds better trajectories (with high values and better average rewards) in comparison to a greedy approach (which chooses the immediate best rewarding actions). Hence following an ϵ-greedy approach gives better rewards due to exploration. In fact, in this case $\epsilon = 0.5$ gives the maximum optimal average reward.

  [Please note that all the figures are not attached in this document, only the ones with which some conclusions can be drawn are depicted, otherwise observations are made for all the scenarios and shared separately.]

- Out of the sixteen comparisons, eight of them where variance was 1, had generated higher average reward values, i.e, also depicted in case 1, 3, 5, 7, 9, 11, 13, 15. In these figures Q0 is 0 for four of them and 5 for another four and c is 0, 1, 2 or 5. Consequently, case 2, 4, 6, 8, 10, 12, 14 and 16 have variance 10, and we observe that the saturation value of average reward is low as compared to those of variance 1. This can be reasoned as: higher variance means higher uncertainty in reward allocation with respect to qstar, i.e. the more chance for other arms to get chosen apart from the best arm. Hence this makes the percentage optimal action also less. Figure 1 and 2 show this observation.

  Figure 3 also indicates clear difference between same case scenario with UCB, but with different variances.

- Average reward vs time graph reached the saturation value quite early

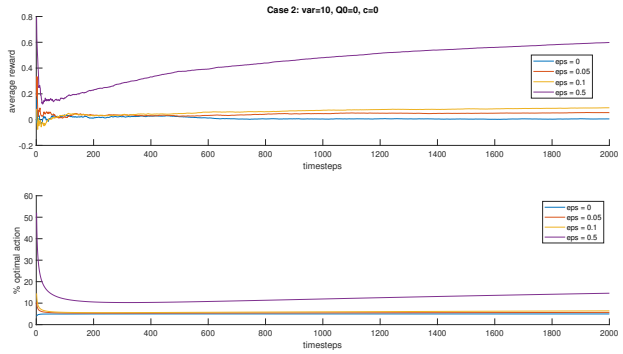

Figure 2: ϵ-greedy algorithm with various ϵ and variance = 10 (case 2)

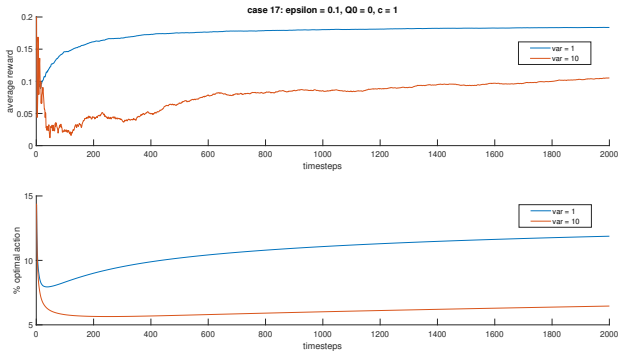

Figure 3: ϵ-greedy algorithm with UCB and different variances (case 17)

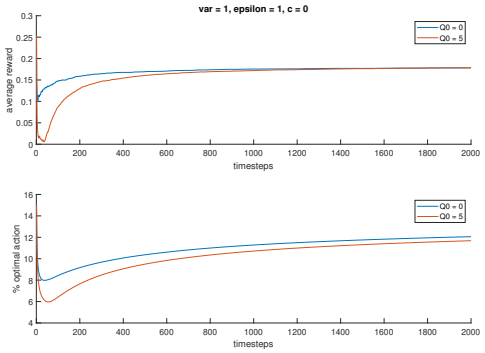

Figure 4: ϵ-greedy algorithm with different initial Q0 (case 23)

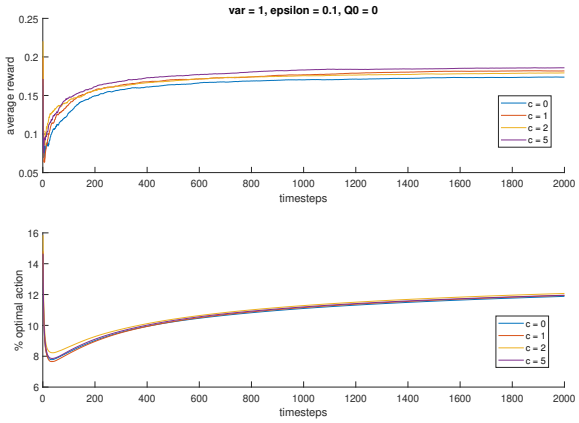

Figure 5: ϵ-greedy algorithm with different c values (case 22)

when variance = 1 whereas it took significant time for the one with higher variance = 10 to reach the saturation value. Which means optimal action is recognised early when the variance is low.

- Percentage optimal action is also higher with lower variance because then the algorithm is not not getting much confused as to which arm to choose to get the optimal reward.
- Comparison on the effect of varying Q0 is also done and depicted in case 23 and figure 4. No significant magnitude change is observed in average reward and percentage optimal action, although one curve reached optimal feature earlier than the other and it depends on the choice of the parameter and its distance from the optimal value. Q0 =5 appears to reach slower than Q0 =1.

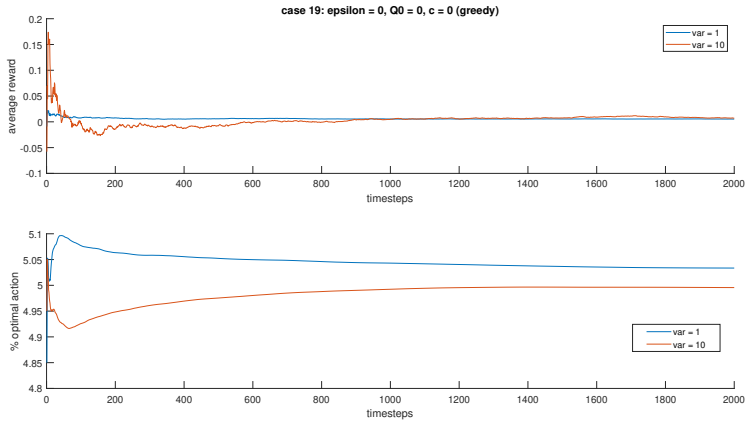

Figure 6: greedy algorithm with different variance (case 19)

- Comparison of various UCB parameters for a greedy approach is done in case 22 (Figure 5). It shows increasing 'c' actually reduces the chances of choosing greedy approach more and hence the average reward received is higher because of more exploration. Same is true for percentage optimal action graph too.
- Lastly, case 19 (Figure 6)shows a pure greedy approach with different variances where lower variance gives higher average reward and percent optimal action. And case 21 (Figure 7) shows a pure greedy approach with different Q0. Q0 = 0 which happens to be near the optimal value fares better.

3 All combination plots

Apart from the figures depicted here, plots have been generated for all combinations of variances, Q0 and c with respect to all ϵ values. Please have a look at them.

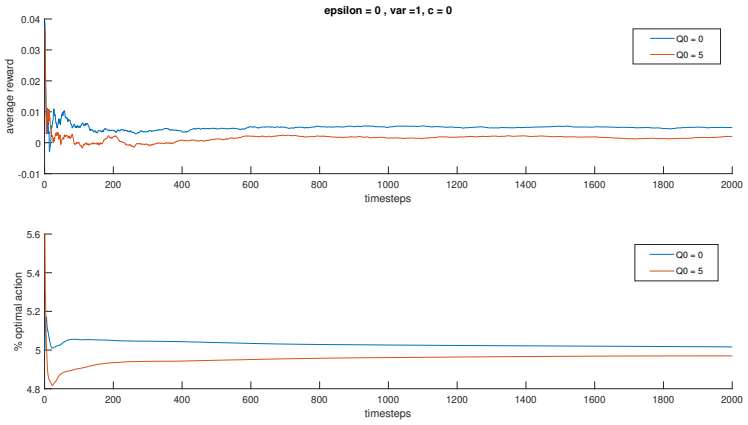

Figure 7: greedy algorithm with different Q0 (case 21)