Comparison of Bandit Algorithms

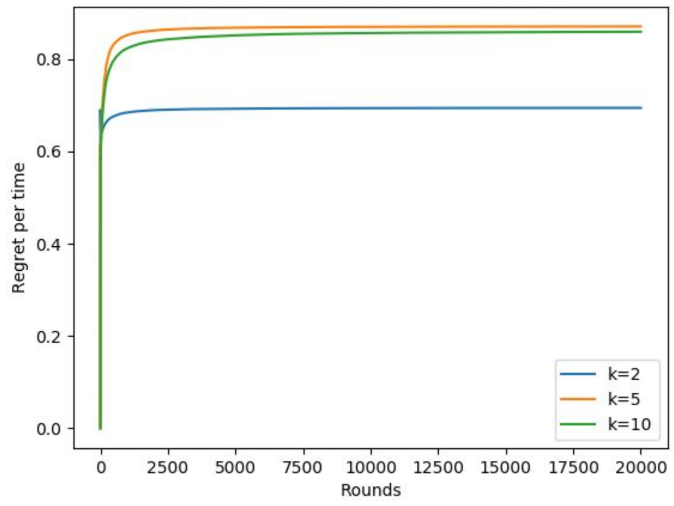
3rd September 2019

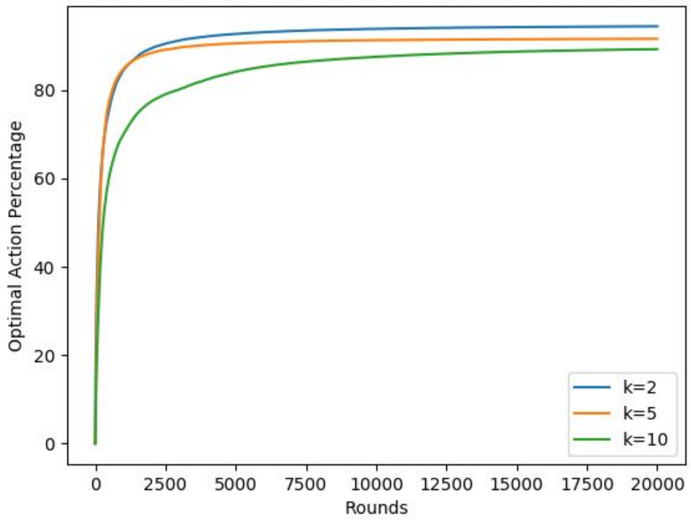
Group 4
Shruti Patel (160010002)
Mayur Atkade (160010011)
Ketan Karnakota (160010031)
Samveed Desai (160020003)
Harini Dandu (160020032)

EPSILON GREEDY

A common approach to balancing the exploitation-exploration trade off is the *epsilon-* or *e-greedy* algorithm. Greedy here means what you probably think it does. After an initial period of exploration (for example 1000 trials), the algorithm greedily exploits the best option *k*, *e* percent of the time. For example, if we set *e*=0.05, the algorithm will exploit the best variant 95% of the time and will explore random alternatives 5% of the time.

CASE : WHEN EPSILON = 0.1

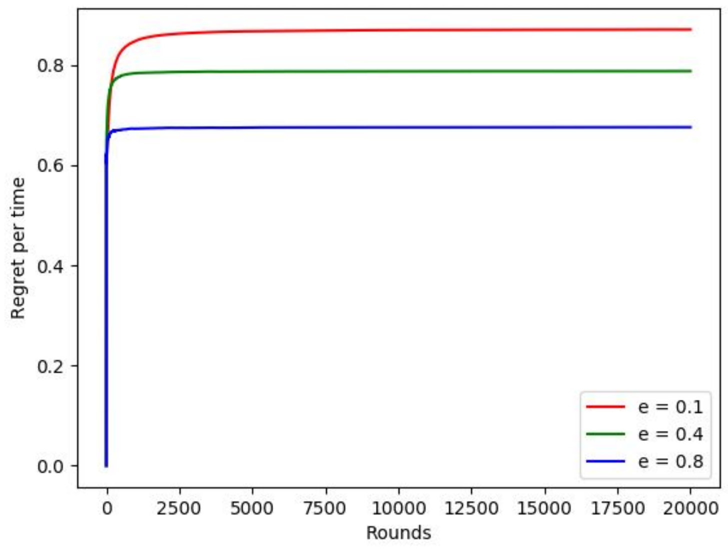


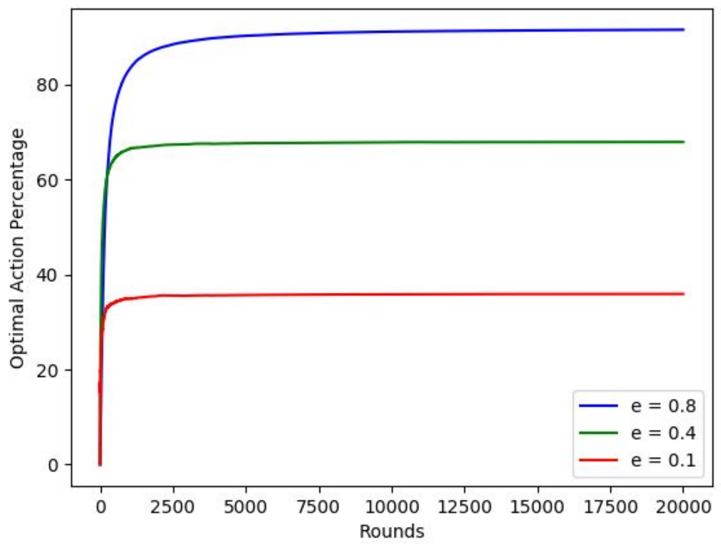


Observations:

- The regret per time is proportional to the time. The algorithms gives a very high regret value. However, the algorithm gives lesser regret than greedy algorithm.
- The optimal action percentage increases with time and gradually becomes constant as it constantly pulls the same arm .

CASE : WHEN NUMBER OF ARMS = 5



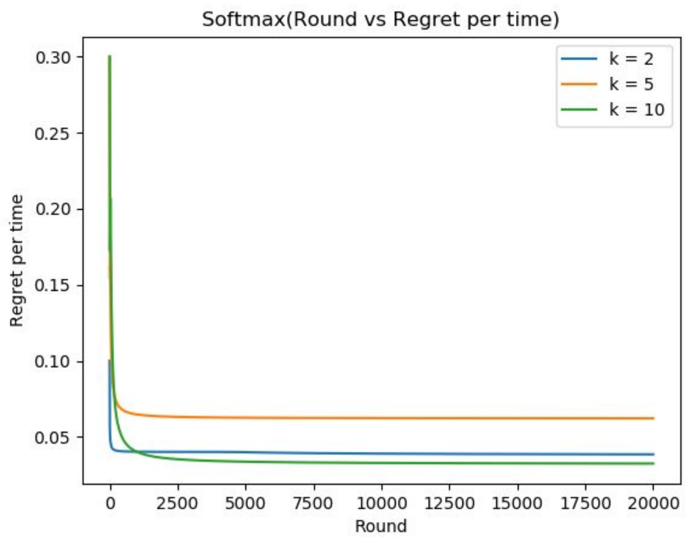


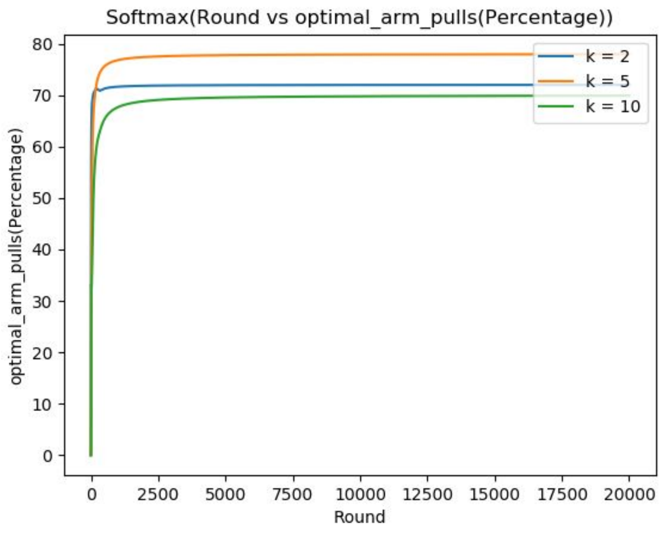
Observations :

- With increase in epsilon, the greediness decreases. Hence, the chances of picking the optimal arm increases. So, as epsilon increases, the regret per time decreases.
- Since for e=0.1, the regret per time is more than that of e=0.4 and e=0.8, the percentage of optimal pulls is lesser. However, the percentage of optimal pulls increases with time. After 20,000 runs, the percentage of optimal pulls is only about 30% for e=0.1.

SOFTMAX

The Softmax policy converts the estimated arm rewards into probabilities then randomly samples from the resultant distribution.





Observations:

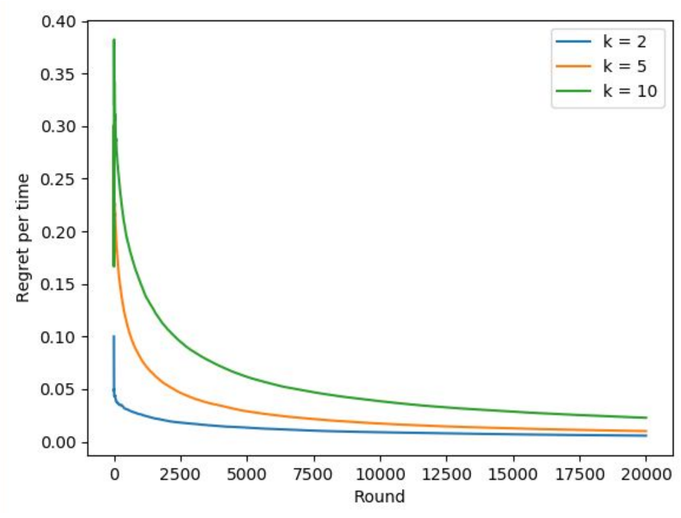
- Softmax chooses the arm stochastically. It incurs logarithmic regret. For, t=20000, regret is proportional to log(T) = log(20000) = 4.3 (The no.of times suboptimal arm is pulled).
- The percentage of optimal pulls is increases with time. After 20,000 runs, approximately 80% of the times, optimal arm is chosen.

UCB

Upper Confidence Bound (UCB) algorithm is based on the principle of optimism in the face of uncertainty. In other words, the more uncertain we are about an arm, the more important it becomes to explore that arm.

The intuitive reason this works is that when acting optimistically in this way, one of two things happen:

- optimism is justified and we get a positive reward which is the objective ultimately
- the optimism was not justified. In this case, we play an arm that we believed might give a large reward when in fact it does not. If this happens sufficiently often, then we will learn what is the true payoff of this action and not choose it in the future.



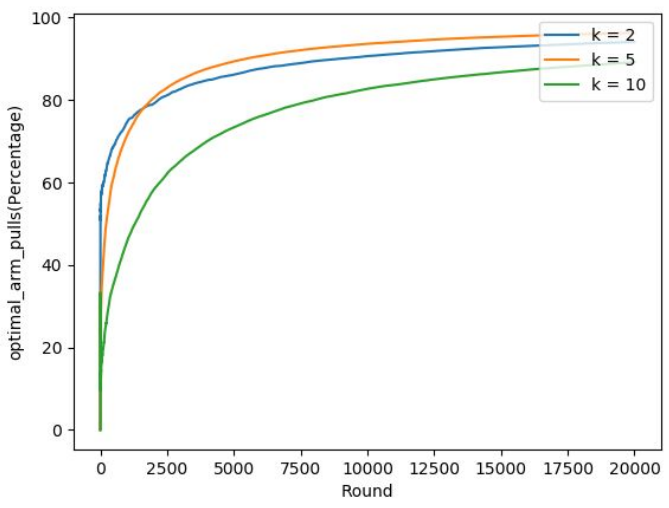


Fig1: UCB (Round vs Regret per time)

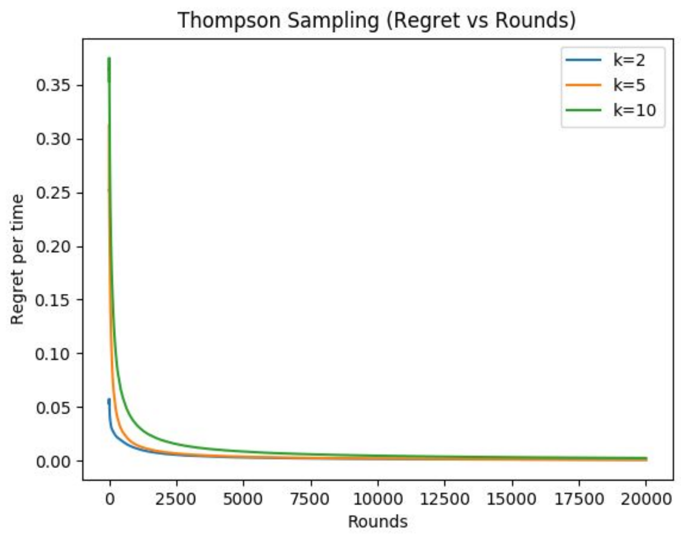
Fig2: UCB (Round vs Optimal Arm Pull (%)

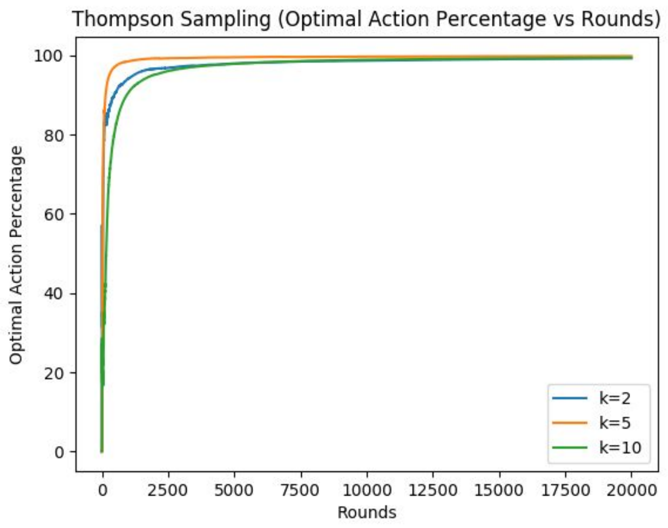
Observations:

- For rounds = 20000. Regret is proportional to log(20000)= 4.3 (from Fig1). The regret is order of log(no.of rounds). Which is approximately the no. of times a suboptimal arm is pulled at round = 20000 (from Figure 2).
- So, the regret is proportional to logT (no.of times suboptimal arm is pulled)
- As the algorithm estimates the optimal arm correctly with increase in no.of rounds, the no. of times suboptimal pulled is decreased.

THOMPSON SAMPLING

Thompson Sampling is an algorithm that follows exploration and exploitation to maximize the cumulative rewards obtained by performing an action. An action is performed multiple times which is called exploration and based on the results obtained from the actions, either rewards or penalties, further actions are performed with the goal to maximize the reward which is called exploitation. In other words, new choices are explored to maximize rewards while exploiting the already explored choices.





Observations:

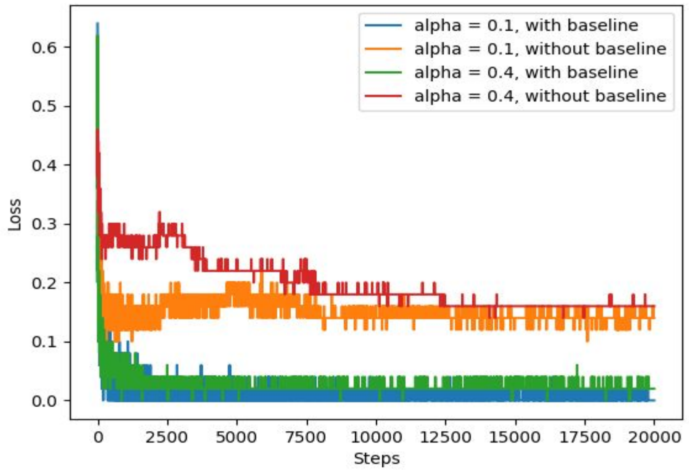
- The regret is of the order O(log T). Therefore, the number of times suboptimal arm is chosen decreases with time.

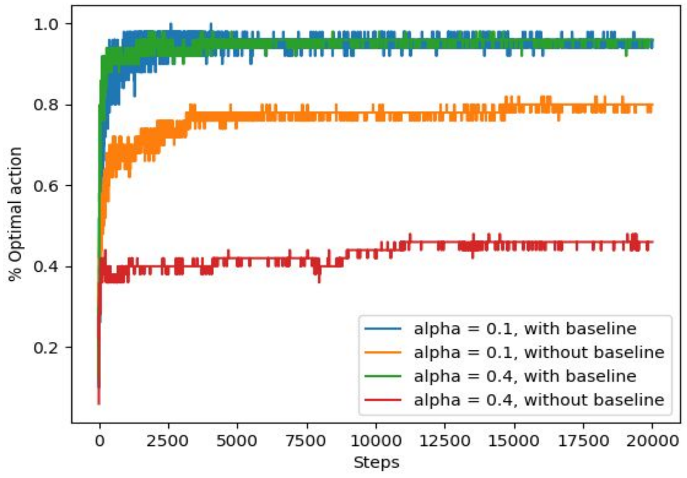
- The percentage of times the optimal action is chosen increases with time. It implies that the algorithm learns as time progresses and eventually it chooses the optimal arm with a very high probability. After 20,000 runs, the percentage

REINFORCE

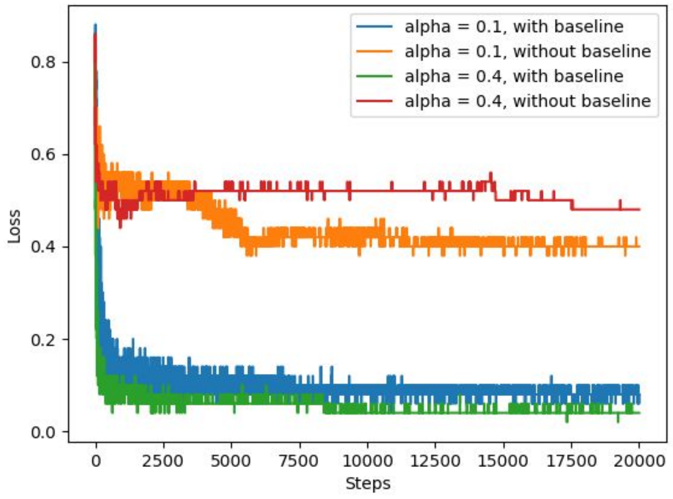
Reinforcement comparison methods [Sutton and Barto (1998)] are similar to pursuit methods, in that they maintain a distribution over actions which is not computed directly from the empirical means. These methods also maintain an average expected reward r(t). The probability of selecting an arm is computed by comparing its empirical mean with r(t). The probability will be increased if it is above average, and decreased otherwise. Intuitively, this scheme is designed to account for cases in which arms have very similar value.

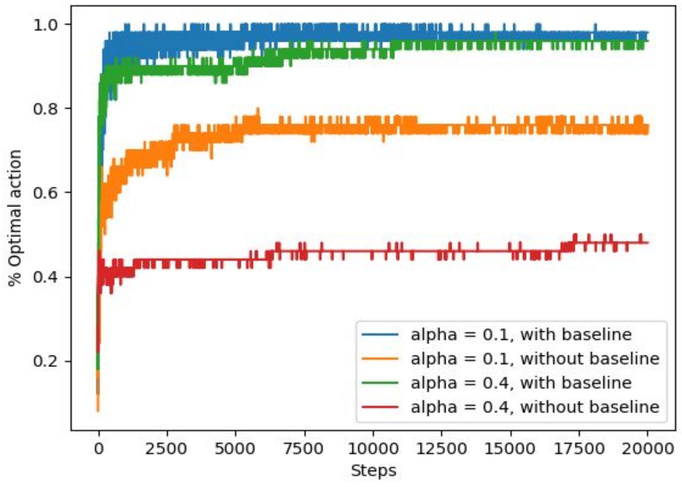
For k=2



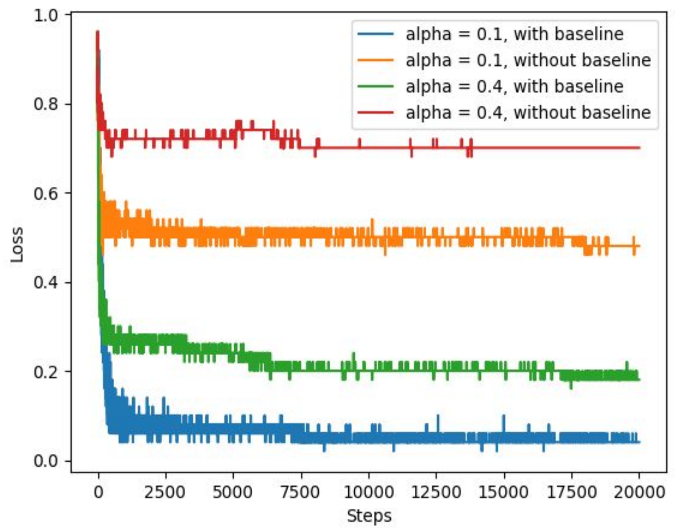


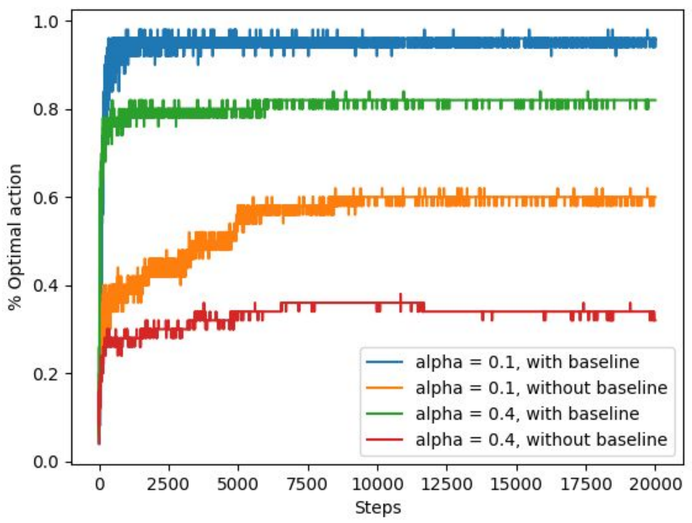
For k = 5





For k = 10

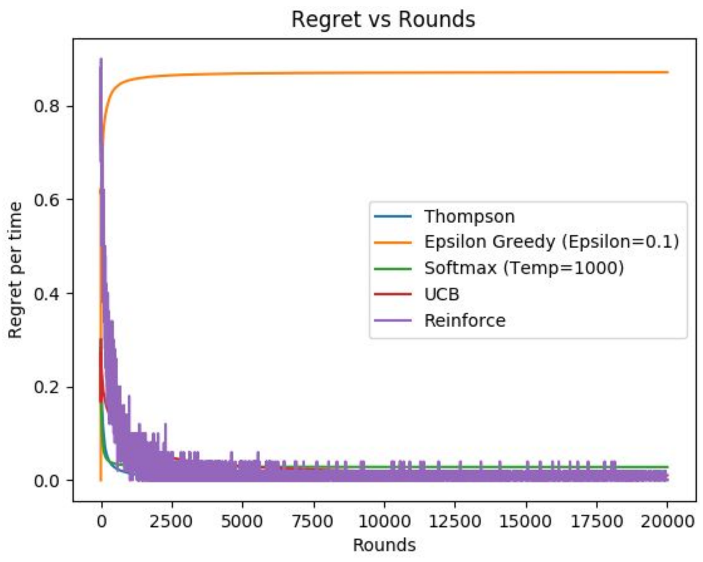


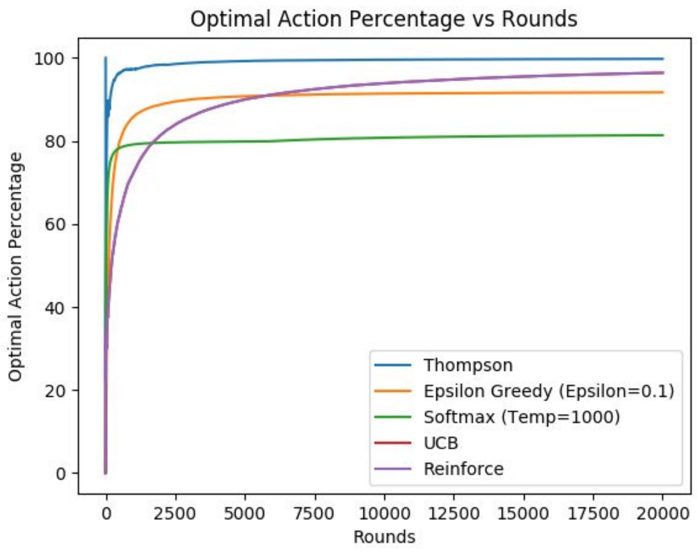


Observations:

- This is a Policy based Algorithm, which doesn't work accurately for Bandit Problems but works pretty well for Reinforcement Learning Problems
- The use of the baseline is to reduce the wide/large changes in the overall results(in this case, rewards) and basically, normalize it, by subtracting the mean of all the rewards from the individual rewards. As we see above, all the metrics with inclusion of the baseline, show better results than those without it.
- According to us, alpha is how fast the value of the theta, gets updates. As we have higher values of alpha, the theta gets updates faster and accurately and thus, reaches the optimum metrics(minimal loss and better optimal action) faster.

COMPARISON OF DIFFERENT ALGORITHMS :





- e - greedy chooses arms greedily most of the times,it can get stuck exploiting a suboptimal variant, so the regret incurred is very high when compared to other algorithms. Softmax chooses the arms proportional to their average reward. So, the regret incurred is less when compared to e-greedy. Whereas, the regret incurred in UCB is even less when compared to Softmax as it adds an extra term (square root(2*lnt/ Ni)) which decreases the frequency of arm pulled with increase in no.of times it has been picked. so all the arms are chosen at least once eventually.
- Thompson sampling gives very low regret and high percentage of optimal pulls. Thompson sampling is the gives the best results in value based methods.
- Reinforce algorithm also gives results similar to thompson sampling as it incurs approximately zero regret. With time, the algorithm learns the reward of arms. In each round, the algorithm modifies its parameters such that the average reward is high for the optimal arm.