CS6700 WA2

Nived Narayanan

EP14B035

1 Multi-Arm Bandits

. 1. The agent has 4 degrees of freedom and each can be governed by the 4 bandits.Bandits with payoff value varying with the change in the state,or this could be seen as different bandits at each of the state. So 400 bandits. The bandits initially will be having the same payoff distribution and during the initial stages arms in each states are pulled in random. Once the agent gets to the final position and gets a reward of +1 all of the payoffs of those bandits which made the agent get to the final position are incremented and the action value of pulling those arms are also incremented. UCB algorithm can be used in the bandits to improve the exploration.

2. case 1:If the payoff is a constant value for each arm then it would be a rather easy problem and could proceed with the greedy approach of pulling all the arms once and then choosing the arm thereafter which gives the maximum payoff.
case 2:If the payoffs are given from a distribution whose means are known, then it is a different case. In that case an exploration is needed to find the arm with the highest mean for the payoff distribution. An algorithm which guarantees a better performance that UCB can be devised for achieving the task. The highest mean is given as 4.6,
(i)Pull all the arms once and choose the arm that has the maximum reward.Use a variable count and increment count for that arm if the action value is more than 4.6.
(ii)Pull the chosen arm in step (i),decrease count variable whenever the output is less than 4.6-variance of distribution(or some small value),if the count gets more negative then choose another arm with a UCB approach.
(iii)repeat the process and count will be maximum for the optimal arm.

3. The existing bandit algorithms tries to maximize the value of the actions by including exploration component.They differ because of the difference in the way this exploration is being carried out.Since exploration is not needed in this case these algorithms are not effective. Averaging the action value for each arm and then choosing the arm with the maximum average action value will in turn converge to the optimal solution.

4. Optimistic initial values are a way to improve the exploratory tendency during the initial stages.Even if the actions are non optimal it'll not be judged as a bad action due to the initial value offset. Until there builds up a difference between the optimal action and all other actions will the algorithm become more steady. Until this occurs there will be oscillations and spikes. This method will perform worse in the initial steps when it goes on choosing bad arms.It will have a good performance when it chooses the optimal arm in the beginning itself.

5. $Q_{n+1} = Q_n + \alpha_n(R_n - Q_n)$

$Q_{n+1} = \alpha_n R_n + (1 - \alpha_n)[\alpha_{n-1} R_{n-1} + (1 - \alpha_{n-1} Q_{n-1})]$

$Q_{n+1} = \alpha_n R_n + (1-\alpha_n)\alpha_{n-1} R_{n-1} + ... + (1-\alpha_n)(1-\alpha_{n-1}).....(1-\alpha_1)\alpha_0 Q_0$

$Q_0 = \alpha_0 \prod_{i=1}^{n}(1 - \alpha_i)$