CS6700 : Reinforcement Learning
Written Assignment #2

Akshit Kumar
EE14B127

17th February 2017

Contents

1 Question 1

1.1 Question

Consider the following grid world task. The environment is a 10 x 10 grid. The aim is to learn a policy to go from the start state to the goal state in the fewest possible steps. The 4 deterministic actions available are to move one step up,down,left or right. Standard grid world dynamics apply. The agent receives a reward of 0 at each time step and 1 when it reaches the goal. There is a discount factor $0 < \gamma < 1$. Formulate this problem as a family of bandit tasks. These tasks are obviously related to one another. Describe the structure of the set up and the rewards associated with each action for each of the tasks, to make it perform similarly to a Q-learning agent.

1.2 Answer

In a grid world setting, the Q-learning problem can be represented pictorially as each grid cell being divided into 4 triangles and each triangle holds the Q-value for that state and action and the Q-learning agent learns the Q-values for each action in that state. Since the reward at each time step is 0 and reward of 1 is only given when the goal is reached, the Q-values for each action in each state will be bounded between 0 and 1. Let us assume that we have solved the grid-world problem using a Q-learning agent and have the Q-values for each action in each state. Now we work backwards, formulating the problem as a family of bandit tasks.

This problem can be modelled as a *contextual bandits problem* where the context is the cell in the 10x10 grid. To each grid cell, we assign a 4-armed bandit where each arm represents an action in the grid world. Pull of an arm signifies which direction the agent chooses to move in - up,down,right or left and for choosing each arm the agent gets a reward which are sampled from a distribution whose expected mean is the Q-value for that action in that state. Since for each bandit, we have a different set of reward distribution(i.e. different expected mean) - they are different bandit problems but are also associated with each other. For example, the reward distribution for the arms in a cell is influenced by the reward distribution of the arms in the neighbouring cells. Hence this qualifies as a contextual bandits problem.

Now for the reward distribution for each action for each task can simply be chosen to be a univariate Gaussian distribution with mean of the distribution to be the same as the Q-values that we get for the action from the Q-learning agent and some common value of variance(say 1) across all the actions. We can even go further and relax this condition of keeping the same values of mean as that of the Q-values and keep arbitrary values of the mean of the Gaussian bounded between 0 and 1 as long as we keep the ordering of the means of the Gaussian in the bandit problem the same as the ordering of the Q-values from Q-learning agent.

The above described setting should perform similar to a Q-learning agent.

2 Question 2

2.1 Question

Consider a bandit problem in which you know the set of expected payoffs for pulling various arms, but you do not know which arm maps to which expected payoff. For example, consider a 5 arm bandit problem and you know that the arms 1 through 5 have payoffs 3.1, 2.3, 4.6, 1.2, 0.9, but not necessarily in that order. Can you design a regret minimizing algorithm that will achieve better bounds than UCB? What makes you believe that it is possible? What parts of the analysis of UCB will you modify to achieve better bounds? Note that I am not asking you for a complete algorithm or analysis, only the intuition.

2.2 Answer

Yes, intuitively it seems possible to design a regret minimizing algorithm that can take advantage of the fact that we know the expected payoff of the best arm. Since we know all the expected payoffs, we also know the expected payoff of the best arm which is simply given by

$$q_*(a^*) = max\{q_*(i) | i \in A\}$$

where $q_*(.)$ denotes the true expectation of the arm, a^* denotes the optimal arm and A denotes the set of all possible actions/arms.

In the UCB algorithm, we don't have the knowledge of expected mean of the best arm. In UCB, we choose arms by being "optimistic" about the estimates of the arm and behaving greedily with the optimistic estimate values. As we sample more of an arm, our confidence in the arm's estimate grows and uncertainty about the estimate decreases. But in-case we knew the expected value of best arm then in that scenario, we can coverge to the estimate of the best arm faster compared to the UCB as now we can just choose arms whose estimates are close to the expected payoff of the best arm and also weigh them by how many times they have been played in the past to avoid being stuck with a suboptimal arm and continue accuring regret because of continuous selection of the suboptimal arm. By doing this we don't care for arms whose estimates is constantly off compared to the expected payoff of the best arm. Since we would converge to the optimal arm in the lesser number of time steps, we could accure less regret overall and hence can get a lower regret bound than because of UCB.

The parts of the UCB analysis that we would need to modify to achieve better bounds is where we are calculating the expectation of $T_i(n)$. Now since we are not sampling all the arms which are "bad" that often now, we have a lower expectation value of such arms and hence a lower expectation of regret.

We can also follow the GCL^* policy [1].

3 Question 3

3.1 Question

Define a bandit set up as follows. At each time instant for each arm of the bandit we sample a reward from some unknown distribution. Now the agent picks an arm. The environment then reveals all the rewards that were chosen. Regret is now defined as the difference between the best arm at that instant and the one chosen summed over all times steps. Would the existing algorithms for bandit problems work well in this setting? Can we do better by taking advantage of the fact that all rewards are revealed? For e.g., exploration is not an issue now, since all arms are revealed at each time step.

3.2 Answer

In this new setting, where all the rewards are revealed at each time step allows us to update the estimates of all arms in just one time step and this makes the UCB algorithm same as a greedy algorithm because the confidence interval term $\sqrt{\frac{2lnN}{T_i(N)}}$ is the same for all the arms and we just end up selecting the action which maximizes $Q_t(.)$ thereby behaving greedily. In the new setting, we have a new notion of regret which is defined as

$$R = \sum_{i=1}^{N} \Delta i$$

where R is the total regret and $\Delta_i = r_i^* - r_{\text{argmax}\, Q_{i-1}(.)}$, r_i^* denotes the best arm at the current step and $\text{argmax}\, Q_{i-1}(.)$ denotes the best arm chosen according to the estimates till one step before the current time step.

Using the "traditional notion of regret", the sequence of $\{\Delta_i\}$ would converge to 0, however in this "new notion of regret", the sequence $\{\Delta_i\}$ need not converge to 0 and is stochastic and hence our solution concept of regret bounds is no longer valid. Since the UCB algorithm and other bandit algorithms are designed to minimize the "traditional notion of regret", they *may not work well* in this new setting with a new definition of regret. If we use the new notion of regret as a performance metric of existing algorithm to "work well" then the answer is *it depends.*

One of the factors on which the performance depends is the *variance of the reward distribution.* If the variance is high, then the sequence $\{\Delta_i\}$ would mostly have non-zero values resulting in accumulating more and more regret over each time step. On the similar lines, if the variance is low, then the sequence $\{\Delta_i\}$ would have a lot more zeros and we won't accrue much regret.
Another factor on which the performance depends is how separated the expected payoff of each arm is. If they are close to each other(for example, the second best arm is really close to the optimal arm), in that case we are bound to make a lot of errors in arm selection whereas if the arms have significant difference amongst the expected payoffs of their respective arms, then we can clearly choose one arm as a single best arm which would lead to less regret being accrued.

This new setting doesn't seem to offer any advantages in optimizing for the new notion of regret apart from the fact the estimates of all the arms can be updated at each time leading to convergence of estimates to their expected payoff values within some ϵ range where ϵ is a really small number, faster compared to the traditional bandits setting wherein the reward for only the pulled arm is revealed. Another advantage is that there is no need of exploration any more as we can just act greedily and converge to the expected payoffs of the arm without wasting steps on the non optimal arm. Also pulling an optimal arm and non optimal arm is now equivalent in terms of rate of learning of the expected payoff values of the arms.

4 Question 4

4.1 Question

The results shown in Figure 2.3 (of course text book uploaded in moodle) should be quite reliable because they are averages over 2000 individual, randomly chosen 10-armed bandit tasks. Why, then, are there oscillations and spikes in the early part of the curve for the optimistic method? In other words, what might make this method perform particularly better or worse, on average, on particular early steps?

4.2 Answer

For the sake of simplicity let us assume that we have 3-armed bandit tasks instead of 10-armed bandit tasks, say each of arms A,B,C (say) have a reward being drawn from a Gaussian Distribution of mean 1,3,2 respectively and each of them having a variance of 0.5(equal variance). Given this underlying distribution of rewards, we know that arm B is the best arm to pull. Let us now use optimistic initial values for selecting the arms and each of the arm estimates being initialsed to a high optimistic value say 10. We break ties by giving preference to the arms index. Lower the arm index, higher its priority. Therefore we choose arm A over arm B over arm C in case of ties.

Since initially the arms have the same optimistic value, we choose arm A and get a reward of say 0.7, then we choose arm B and get a reward of 3.2 and then we choose arm C and get a reward of 1.8. Since all the rewards are less than the optimistic initial values, this will cause the estimates to decrease where the rate of decrease will be proportional to how suboptimal the arm is(ie worse the arm is, more the decrease). In the initial few steps, since only a few arm pulls have been made, it is possible that *decrease in estimate* of arm B is the least *for a lot of time steps in the 2000 independent runs* and hence the agent will pull arm B many times initially. This explains the spike in the early part of the curve where the agent chooses the optimal action with high frequency. Though this argument is for set of 3-armed bandit tasks, it can be extended to a 10-armed bandit task as well. This nature of the oscillations and spikes is artibuted to the fact that the initial estimates are higher than the expected rewards of each arm and the rewards obtained after each action lower the estimate of that arm.

If the initial values are low compared to the expected rewards then if the agent picks a suboptimal arm intially, it would keep on picking that particular suboptimal arm repeatedly because the estimate of that suboptimal arm would be higher than the initial estimate of the optimal arm and the agent would perform *worse* throughout including the early steps.

Similarly because of the initial estimates being set high compared to the expected means of the distribution from which rewards for the respective arms are sampled from has a built in exploration strategy into it which forces the agent to "pull" each arm atleast once and hence this method would perform atleast as good as an ϵ-greedy method in the early steps but this strategy to "explore" would eventually dies out and the agent would just behave greedily. Hence it would perform *better* in the early steps.

5 Question 5

5.1 Question

If the step-size parameters, α_n, are not constant, then the estimate Q_n is a weighted average of previously received rewards with a weighting different from that given by (eq 2.6 of course text book). What is the weighting on each prior reward for the general case, analogous to (eq 2.6 of course text book), in terms of the sequence of step-size parameters?

5.2 Answer

Let us say Q_{k+1} is the estimate at $k+1$ step and Q_1 represent the initial estimate. Let R_i represent rewards at each step i. Assuming a constant step size α, we can say

$$Q_{k+1} = (1-\alpha)^k Q_1 + \sum_{i=1}^{k} \alpha (1-\alpha)^{k-i} R_i$$

But if the step size α is not fixed, then in that case, the weighting on prior rewards changes. Let us denote the step size by α_i for step i. Then we can say

$$\begin{aligned}
Q_{k+1} &= Q_k + \alpha_k (R_k - Q_k) \\
&= \alpha_k R_k + (1-\alpha_k) Q_k \\
&= \alpha_k R_k + (1-\alpha_k)[\alpha_{k-1} R_{k-1} + (1-\alpha_{k-1}) Q_{k-1}] \\
&= \alpha_k R_k + (1-\alpha_k)\alpha_{k-1} R_{k-1} + (1-\alpha_k)(1-\alpha_{k-1})[\alpha_{k-2} R_{k-2} + (1-\alpha_{k-2}) Q_{k-2}] \\
&= \alpha_k R_k + (1-\alpha_k)\alpha_{k-1} R_{k-1} + (1-\alpha_k)(1-\alpha_{k-1})\alpha_{k-2} R_{k-2} + \ldots\ldots. + (1-\alpha_k)(1-\alpha_{k-1})(1-\alpha_{k-2})\ldots..(1-\alpha_1) Q_1 \\
&= \sum_{i=1}^{k} \alpha_i \prod_{j=i+1}^{k} (1-\alpha_j) R_i + \prod_{i=1}^{k} (1-\alpha_i) Q_1
\end{aligned}$$

Now if we choose a sequence of step-size $\{\alpha_n\}$, such that

$$\sum_{i=1}^{k} \alpha_i \prod_{j=i+1}^{k} (1-\alpha_j) + \prod_{i=1}^{k} (1-\alpha_i) = 1$$

then the weight on each prior reward R_i for $i < k$ is given by $\alpha_i \prod_{j=i+1}^{k} (1-\alpha_j)$. Note that the weight given to the reward R_i depends on the step-size at i-th step and step-sizes of future steps. The sequence of step-size $\{\alpha_n\}$ should also satisfy the condition $\sum_{n=1}^{\infty} \alpha_n = 0$ and $\sum_{n=1}^{\infty} \alpha_n^2 < \infty$ to assure convergence with probability 1.

References

[1] Salomon, Antoine and Audibert, Jean-Yves *Deviations of stochastic bandit regret.* In Algorithmic Learning Theory,pp. 159 - 173. Springer, 2011.