Intro to RL and Bandits

Ajinkya Ambatwar
EE16B104
Dept. Of Electrical Engineering

February 18, 2019

1. Tic-tac toe has a very limited and small set of possible states. So learning through previous experience is not so complicated task for the RL agent.

   (a) Now the agent has to make sure that it visits every possible state often in order to learn the dynamics of early winning. Also if it does an exploring start, it will have a benefit of making prediction with a set of statistically similar position. Now speaking of statistically similar, the agent can easily learn such states if we simplify the definition of **'State'** and **'Action'**. Since by this the agent can look for multiple statistically similar states though they might not look the same as per the board position. This will lessen the dimension of the state and the agent will work with smaller state space by taking the advantage of the similarity.

   (b) So if our opponent is not taking the advantage of symmetry then it is recommended for the agent not to go for symmetry. As we have seen that the symmetry based method will reduce the state space and also the number of parameters defining the state. So if the opponent is not the one seeking for symmetry and goes by seeking the actual set of parameters defining a set(and hence works with a larger state space) the agent should also do the same.

   (c) So symmetrically equivalent positions should have the same value if we are considering the system to be fully Markov(as is the case most of the times).

2. If the leaning agent starts playing against itself, initially both of them will start taking very random moves that probaly won't make any sense as they will be doing an exploratory start. But later as they go on playing against each other they might come up with a strategy that even the real human players are not recommended to take and might even win with that policy. So it might learn a totally different policy. But the learning duration will be a lot. Also there is a possibility that the agent might develope a strategy that will facilitate by winning itself. Hence it will oscillate between a "Good" and a "Bad" move. Hence it might not even learn based on the sustained policy.

3. If the agent always learns to play completely greedy, it will definitely lose the benefit of exploration. This will result in the agent getting stuck at

the local optima which in most of the cases will be a suboptimum point. Now if the Q for all the action is initialized with same constant(say zero) and the first action is determined, then with a completely greedy policy the agent will go on just exploiting the same action again and again(if the initial reward it gets is positive) and if the initial action chosen is not the "best action" it will saturate at some lesser value of average reward far from the optimal reward that it coulf have received. So compared to non-greedy(or $\epsilon - greedy$) player, initially it might do well as it will choose the greedy actions and take over the other player, but in long term(say after 10 trials) it will start performing poor and fall behind the non-greedy player.

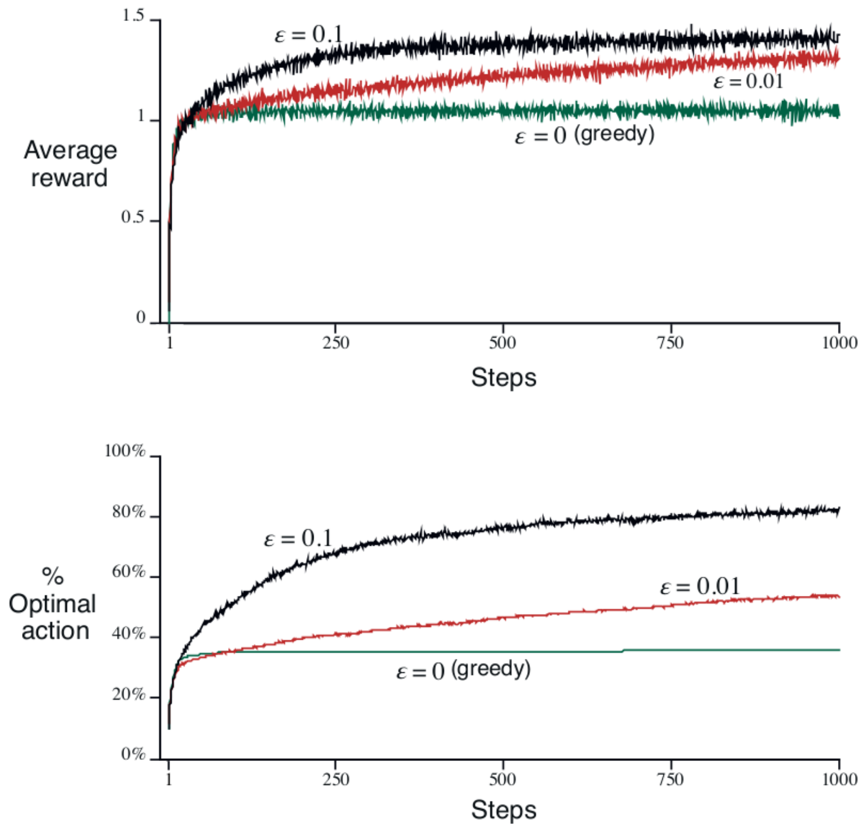

Figure 1: Greedy vs $\epsilon - greedy$ policy

4. In the optimistic start, the average Q values for all the arms is initialized with some larger value, larger than the reward supposed to be received after every action. So because of this, the agent will do more initial exploration automatically. So in this process, if somehow it comes across the *better* choices of actions initially, the average Q value for those actions(say a_{better}), $Q(a_{better})$will be magnified which will emphasis the agent to play more of such actions.This results in up spikes. But as the agent is still do-

ing exploration, it is expected to come across some worse action choices(say a_{worse}), hence the Q values for those actions($Q(a_{worse})$) will be diminished which will result in initial poor play and hence will show down spikes.

5. In such a setup, it is clear that the actual reward vector is changing at every time step and the regret is defined based on the actual reward rather than the long term return. So this can be assumed as a single, nonstationary k-armed bandit problem and we can try to solve it using one of the non-stationary bandit methods(say by keeping α constant). But unless the the true action value(here the revealed rewards at each time steps) change slowly, these methods will not work well. So the necessary condition for this to work is that the variance of the unknown distribution should be less. As mentioned, unless we are sure that the rewards are not changing heavily, we can benefit ourselves with the fact that the rewards are revealed and go for the greedy action every time. And suppose the rewards are changing heavily, then in this case it would be recommended to learn a policy. Then in such a case if the some distinctive clue is provided to how the rewards will be at given time step compared to other time steps, then after sufficient trials a policy can be learned to associate the nature of sampled rewards with the current state of the task. This kind of learning is also called as "***associative mapping***".

6. The idea should be

   (a) Initially take some trials with the normal UCB algorithm method.

   (b) If the implemetation is devised properly the action values are expected to converge to their true means after infinite iterations.

   (c) Say after first K iterations, we will have some estimate of μ for each action with some uncertainty. Now as we already know the true values for each action, we will pair the estimate with that true value which lies in the $\epsilon-$range with probability

   $$P[|\mu - Q_K| \leq \epsilon]$$

   which should be the largest in order to find the correct pair. So now the true value and the estimate pairs are formed correctly with higher probability.

   (d) Now as we know the true value and estimate pairs, now we can set the bounds as follows

      i. Let ϵ_s be the required certainty.

      ii. Now as per the chernoff inequality

      $$P[|\mu - Q_n| \leq \epsilon_s] \geq exp(-2\epsilon_s^2 n)$$

      iii. With this we can ascertain the value n for more certain bounds.

7. The derivation is as follows -

$$\rho_{t+1}(a) = \rho_t(a) + \beta\frac{\partial E(R_t)}{\partial \rho_t(a)}$$

where $E[R_t] = \sum_b \pi_t(b)q_*(b)$. Now

$$\begin{aligned}
\frac{\partial E[R_t]}{\partial \rho_t(a)} &= \frac{\partial}{\partial \rho_t(a)}[\sum \pi_t(b)q_*(b)_b] \\
&= \sum_b q_*(b)\frac{\partial \pi_t(b)}{\partial \rho_t(a)} \\
&= \sum_b [q_*(b) - X_t]\frac{\partial \pi_t(b)}{\partial \rho_t(a)}
\end{aligned}$$

Where X_t is a scalar and $\sum_b \frac{\partial \pi_t(b)}{\partial \rho_t(a)} = 0$, hence inclusion of X_t won't make any difference. Now when $\rho_t(a)$ is changed,some actions'probabilities go up while some others go down but the sum of all the probabilities must be 1 hence the sum of partial derivative is 0.

$$\begin{aligned}
\frac{\partial E[R_t]}{\partial \rho_t(a)} &= \sum_b \pi_t(b)(q_*(b) - X_t)\frac{\partial \pi_t(b)}{\partial \rho_t(a)}/\partial \pi_t(b) \\
&= E[(q_*(A_t) - X_t)\frac{\partial \pi_t(A_t)}{\partial \rho_t(a)}/\partial \pi_t(A_t)] \qquad (1) \\
&= E[(R_t - \bar{R}_t)\frac{\partial \pi_t(A_t)}{\partial \rho_t(a)}/\partial \pi_t(A_t)] \qquad (2)
\end{aligned}$$

Where here we have chosen $X_t = \bar{R}_t$(baseline) and substituted R_t for $q_*(A_t)$ which is valid because $E[R_t|A_t] = q_*(A_t)$. Now

$$\begin{aligned}
\frac{\partial \pi_t(b)}{\partial \rho_t(a)} &= \frac{\partial}{\partial \rho_t(a)}\pi_t(b) \\
&= \frac{\partial}{\partial \rho_t(a)}[\frac{exp(\rho_t(b))}{\sum_{c=1}^n exp(\rho_t(c))}] \\
&= \frac{\frac{\partial exp(\rho_t(b)}{\partial \rho_t(a)} \sum_{c=1}^k exp(\rho_t(c)) - exp(\rho_t(b)\frac{\partial \sum_{c=1}^k exp(\rho_t(c))}{\partial \rho_t(a)}}{(\sum_{c=1}^n exp(\rho_t(c)))^2} \\
&= \frac{1_{a=b}exp(\rho_t(b))\sum_{c=1}^k exp(\rho_t(c)) - exp(\rho_t(b))exp(\rho_t(a))}{(\sum_{c=1}^n exp(\rho_t(c)))^2} \\
&= 1_{a=b}\pi_t(b) - \pi_t(b)\pi_t(a) \\
&= \pi_t(b)(1_{a=b} - \pi_t(a)) \qquad (3)
\end{aligned}$$

Now substituting (3) in (2), we get

$$\frac{\partial E[R_t]}{\partial \rho_t(a)} = E[(R_t - \bar{R}_t)(1_{a=A_t} - \pi_t(a))] \qquad (4)$$

Now for each time step update we shall take sample of (4) and substitute it in the original equation, which gives

$$\rho_{t+1}(a) = \rho_t(a) + \beta(R_t - \bar{R}_t)(1_{a=A_t} - \pi_t(a))$$

$\bar{R}_t$ is the baseline

8. Assuming the parameters to be μ and σ, the policy π is

$$\pi(a|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp(\frac{-(a-\mu)^2}{2\sigma^2}) \tag{5}$$

Now for the mean we can write

$$\begin{aligned}\frac{\partial ln(\pi(a_t|\mu_t,\sigma_t)}{\partial \mu_t} &= \frac{\partial}{\partial \mu_t}\{-\frac{(a_t-\mu_t)^2}{2\sigma_t^2}\} \\ &= \frac{a_t-\mu_t}{\sigma_t^2}\end{aligned}$$

Similarly for the variance we can write,

$$\begin{aligned}\frac{\partial ln(\pi(a_t|\mu_t,\sigma_t)}{\partial \sigma_t} &= \frac{\partial}{\partial \sigma_t}\{-ln(\sqrt{2\pi\sigma_t^2})\} + \frac{\partial}{\partial \sigma_t}\{-\frac{(a_t-\mu_t)^2}{2\sigma_t^2}\} \\ &= -\frac{1}{\sigma_t} + \frac{(a_t-\mu_t)^2}{\sigma_t^3} \\ &= \frac{1}{\sigma_t}\{(\frac{a_t-\mu_t}{\sigma_t})^2 - 1\}\end{aligned}$$

Hence the updates will be

$$\begin{aligned}\mu_{t+1} &= \mu_t + \beta(R_t - \bar{R_t})\frac{\partial ln(\pi(a_t|\mu_t,\sigma_t)}{\partial \mu_t} \\ &= \mu_t + \beta R_t \frac{(a_t-\mu_t)}{\sigma_t^2} \end{aligned} \tag{6}$$

and

$$\begin{aligned}\sigma_{t+1} &= \sigma_t + \beta(R_t - \bar{R_t})\frac{\partial ln(\pi(a_t|\mu_t,\sigma_t)}{\partial \sigma_t} \\ &= \sigma_t + \beta\frac{R_t}{\sigma_t}\{(\frac{a_t-\mu_t}{\sigma_t})^2 - 1\}\end{aligned}$$