Advanced topics in Machine Learning
Assignment 6

Mariuta Nicolae (rqt629) October 25, 2015

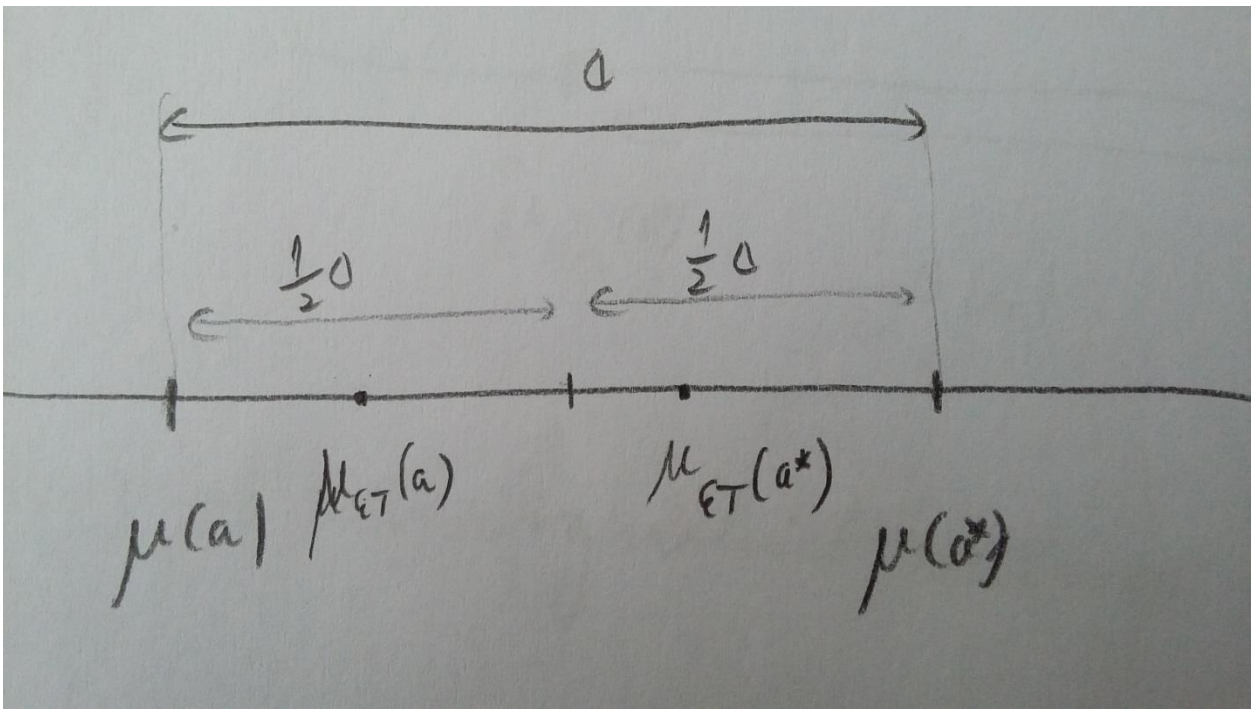
Question 1

I think that for this algorithm, I have to make a derivation much like the analysis we did at lecture for the discussion on Exploration-Exploitation trade-off. In our case, we will have the initial period of time where the empirical estimate is too far from the expected reward.

So we will have a period of time $T\varepsilon$ for the exploration and a period of time $T(1-\varepsilon)$ for exploitation when our algorithm will have more accurate results so I will have to find a good value for ε.

First of all, I will take the assumption that K =2 as suggested in the guidelines as it can make my derivation easier to calculate. I will also assume that we have a fixed time T.

As suggested in the guidelines, I will also have that $\mu(a)$ to be the expected reward for action a and the empirical estimate $\hat{\mu}(a)$. I will also define that a^* to be the optimal action. As told in guidelines, having more optimal action will be better for our algorithm because, for example at K=2 we will have that both actions will be optimal and we get maximum reward no matter what is the choice and the same will be for K>2 as we will have higher chance to have maximum reward even for randomly chosen action. We also have $\Delta(a) = \mu(a^*) - \mu(a)$. If we also set that the confidence bounds to be $\frac{1}{2}\Delta(a)$ we then have something like this:



So we will have the regret for this algorithm:

$$E[R_T] \leq \frac{1}{2}\Delta(a)\varepsilon T + \ \delta(\varepsilon)\Delta(a)(1-\varepsilon)T \leq \frac{1}{2}\Delta\varepsilon T + \delta(\varepsilon)\Delta(a)T$$

We know that the algorithm FTL plays $a \ \neq a^*$ on the round t for which $\hat{\mu}_{t-1}(a) \geq \hat{\mu}_{t-1}(a^*)$ and I will try to analyze how often this may happen:

$$\delta(\varepsilon) = P\{\hat{\mu}_{\varepsilon T}(a) \geq \hat{\mu}_{\varepsilon T}(a^*)\} \leq P\left\{\hat{\mu}_{\varepsilon T}(a) \geq \ \mu(a) + \frac{1}{2}\Delta\right\} + \ P\left\{\hat{\mu}_{\varepsilon T}(a^*) \geq \ \mu(a^*) + \frac{1}{2}\Delta\right\}$$

Now I think here I have to use for both parts the form of Hoeffding's inequality:

$P\left\{\hat{\mu}_t(a) - \ \mu(a) \geq \sqrt{\frac{\ln\left(\frac{1}{\delta_t}\right)}{2t}}\right\} \leq \ \delta_t$ from where we have with high probability $1-\delta_t$ that :

$$\hat{\mu}_t(a) - \ \mu(a) \leq \sqrt{\frac{\ln\left(\frac{1}{\delta_t}\right)}{2t}}$$

By applying this confidence bound, I will have that:

$$\delta(\varepsilon) = \sqrt{\frac{\ln\left(\frac{1}{\delta_{\varepsilon t}}\right)}{2t}} + \sqrt{\frac{\ln\left(\frac{1}{\delta_{\varepsilon t}}\right)}{2t}} = 2\sqrt{\frac{\ln\left(\frac{1}{\delta_{\varepsilon t}}\right)}{2t}}$$

So the regret will be:

$$E[R_T] \leq \frac{1}{2}\Delta\varepsilon T + 2\Delta T\sqrt{\frac{\ln\left(\frac{1}{\delta_{\varepsilon t}}\right)}{2t}}$$

Now I think I should set δ_t so the confidence interval is bound to $\frac{1}{2}\Delta$ but I do not know how to make this replacement so I will keep using $\delta_{\varepsilon t}$.

Now for the approximation of :

$$\frac{1}{2}\Delta\varepsilon T = 2\Delta T\sqrt{\frac{\ln\left(\frac{1}{\delta_{\varepsilon t}}\right)}{2t}}$$

By doing the reduction, there it will be:

$$\frac{1}{2}\varepsilon = 2\sqrt{\frac{\ln\left(\frac{1}{\delta_{\varepsilon t}}\right)}{2t}}$$

By raising both sides at power of 2:

$$\frac{1}{4}\varepsilon^2 = 4\frac{\ln\left(\frac{1}{\delta_{\varepsilon t}}\right)}{2t}$$

So I have the approximation for ε:

$$\varepsilon = 4\sqrt{\frac{\ln\left(\frac{1}{\delta_{\varepsilon t}}\right)}{2t}}$$

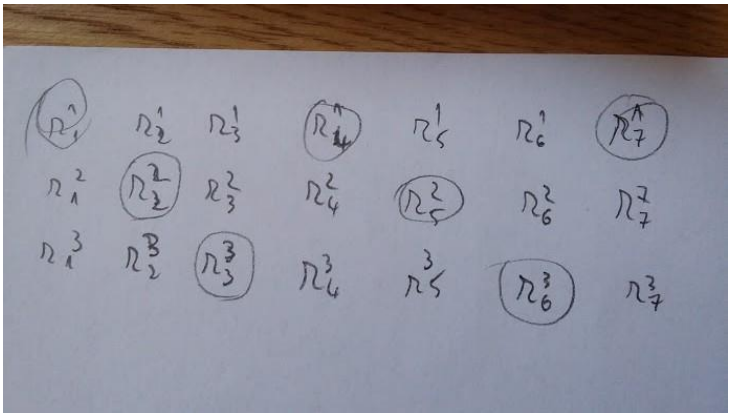
The expected regret becomes:

$$E[R_T] \leq \frac{1}{2}\Delta\, 4\sqrt{\frac{\ln\left(\frac{1}{\delta_{\varepsilon t}}\right)}{2t}}T + 2\Delta T\sqrt{\frac{\ln\left(\frac{1}{\delta_{\varepsilon t}}\right)}{2t}} = 4\Delta T\sqrt{\frac{\ln\left(\frac{1}{\delta_{\varepsilon t}}\right)}{2t}}$$

It does not look like the result from the guidelines but I think it is because I did not manage to find the δ_t for our confidence bound and I should have also calculated an approximation for the number of times we have $\hat{\mu}_{t-1}(a) \geq \hat{\mu}_{t-1}(a^*)$ and then I could obtain a better approximation of the bound.

Question 2

For this game, since we do not know the rewards for each action as for the previous algorithm, I think we have to observe the reward for each action during K moves, and play the action that has the highest reward. The observation of actions would look like this for K=3:



And with this strategy to make the observation, I make at each t the action that gives the best $\hat{\mu}_t(a)$. It will have a big disadvantage since the confidence intervals will be out of control for a longer time than we had at previous question so we will need longer exploration time but I did not manage to figure a more efficient strategy for playing this game. I think we can also use the advantages of UCB algorithm and make something like this:

Multiarmed bandit game

for t = K+1, K+2,….

Play $A_t = argmax_a \hat{\mu}_{t-1}(a) + \frac{3lnt}{2N_{t-1}(a)}$

Observe $r_t^{t\%k}$

end for

I think that there is also the option to just pick $argmax_a \hat{\mu}_{t-1}(a)$ at each step as we did at the previous exercise but I think it is better to use the algorithm as I have written above.

Question 3

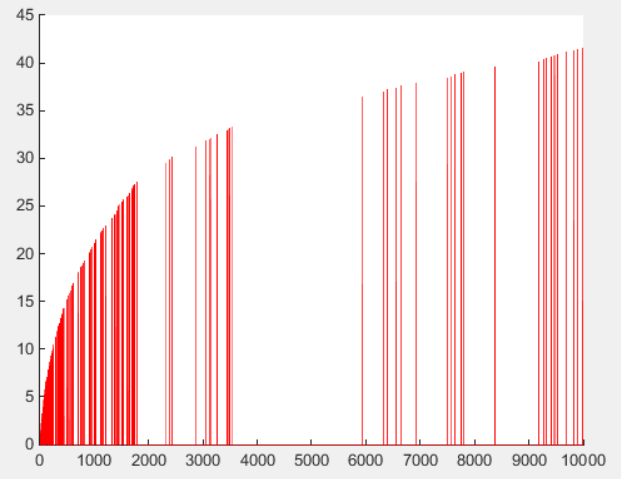
For this game, I will choose to do the same as for the previous exercise, but observe each 2 consecutive rewards this time. By doing this, the exploitation time will be halved and the algorithm should perform better. Each 2k loops we will have a full read for the rewards at each arm and for a large value of T, it should result in good performance.

Question 4

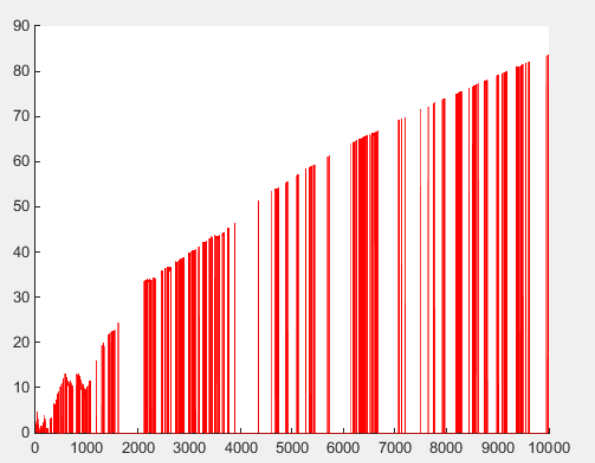
For the implementation of UCB algorithm, I started by generating the parameters: bias for the optimal arm, the bias for the other arms, created arrays to keep the empirical loss, the number of times each arm was played and also an array to store the regret after each t.

In the for loop I started by generating the random values for the reward of each arm according with their biases, and found the value for $argmax_a \hat{\mu}_{t-1}(a) + \frac{3lnt}{2N_{t-1}(a)}$, played the resulting optimal arm and then calculated the regret as described in the question text.

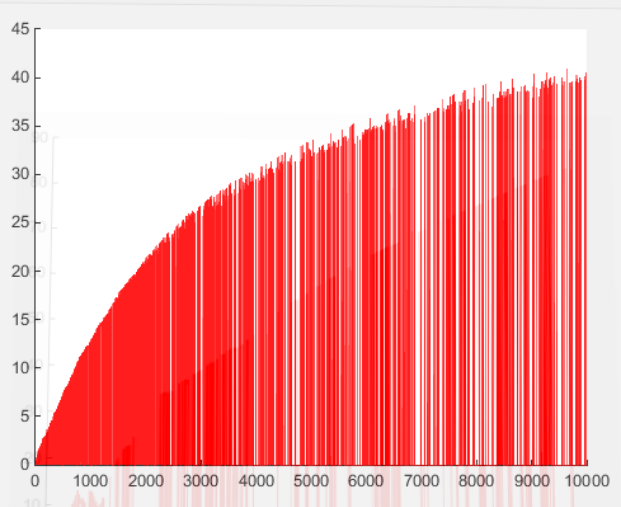
After these, I plotted the resulting regret. For k =2 , bias of optimal arm 0.5 and for the other arms 0.5 – ¼ , I obtained:



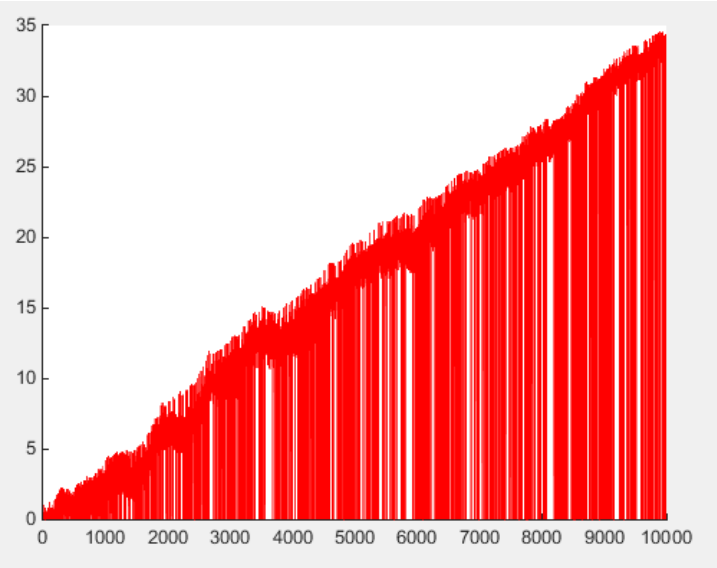
For k = 2, bias for main arm 0.5 and for the other arms 0.5 – 1/16:



For k = 16, bias for main arm 0.5 and for the other arms 0.5 – 1/4:

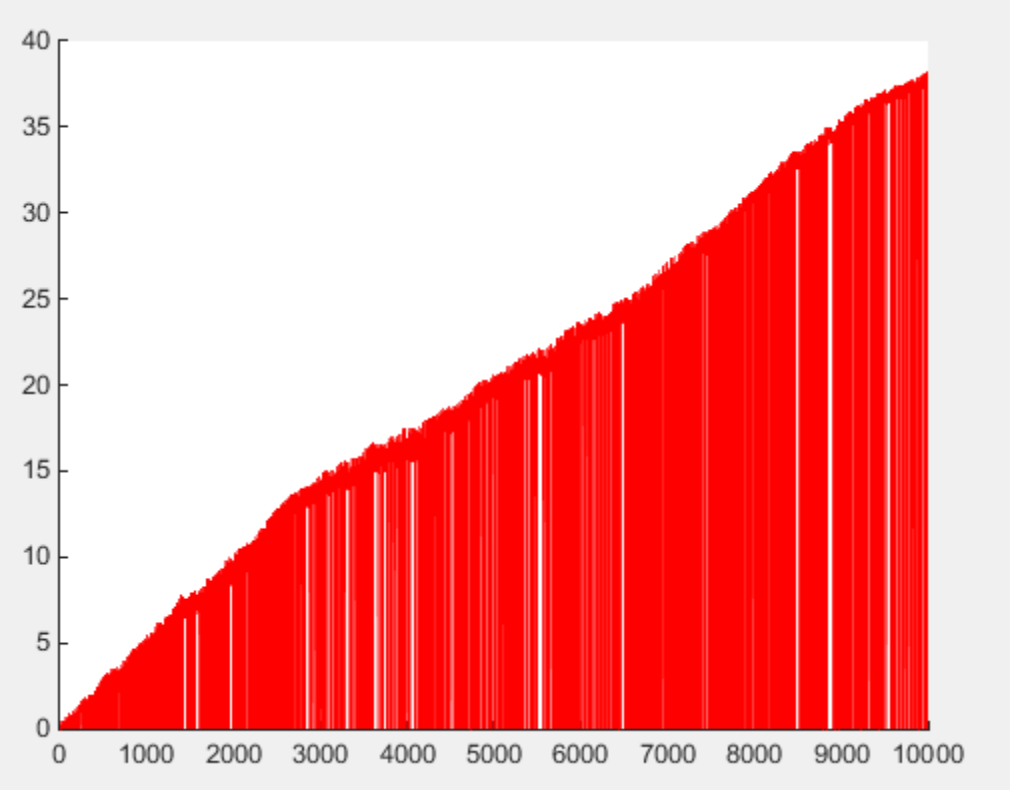


For k = 16, bias for main arm 0.5 and for the other arms 0.5 – 1/16:



With these results, I see a higher regret for having a higher number of arms and also higher regret if the difference between the bias of main arm and the bias of the other arms is smaller.

I also tried for k = 16, bias for main arm 0.9 and for the other arms 0.9 – 1/16:



Unfortunately, I did not understand the algorithm EXP3 and got stuck with his implementation. I tried with same parameters that I used for UCB, then calculated inside the for loops the value of

$$\forall a: p_t(a) = \frac{e^{-\eta_t \hat{L}_{t-1}(a)}}{\sum_{a'} e^{-\eta_t \hat{L}_{t-1}(a')}}$$

But I did not figure how to sample A_t according to p_t because we now have k arms and I did not know how to choose the value to play since I obtained k probabilities for each arm a.

If I could manage to implement this algorithm also I guess I could obtain more meaningful plot by comparing with the performance of UCB algorithm.