CS6046 Multi-Armed Bandits: Assignment 2 Report

-Akash Reddy A, EE17B001

1 Question 1

FTRL with Quadratic Regularization

- The update rule for FTRL with Quadratic Regularization in the set $\mathbb{R}^d$ is:

$$p_{t+1} = p_t - \eta z_t$$

where z_t is the loss vector generated by the adversary, p_t is the algorithm's guess of vector in $\mathbb{R}^d$, and η is set to be $\frac{B}{\sqrt{2}L\sqrt{T}}$ according to the regret analysis done in class. Here, B is the upper bound on the algorithm vectors $(||p_t||B)$ norms, and L is the upper bound on the adversary loss vector norms $(||z_t|| \leq L)$.

In order to restrict this to the simplex Δ_d, the vector p_{t+1} is projected to the simplex. Therefore,

$$p_{t+1} = \Pi_{\Delta_d}[p_t - \eta z_t]$$

where Π_{Δ_d} is the projection operator onto the simplex: $\Pi_{\Delta_d}(x) = \arg\min_p ||x - p||$.

- The regret plot is shown below:

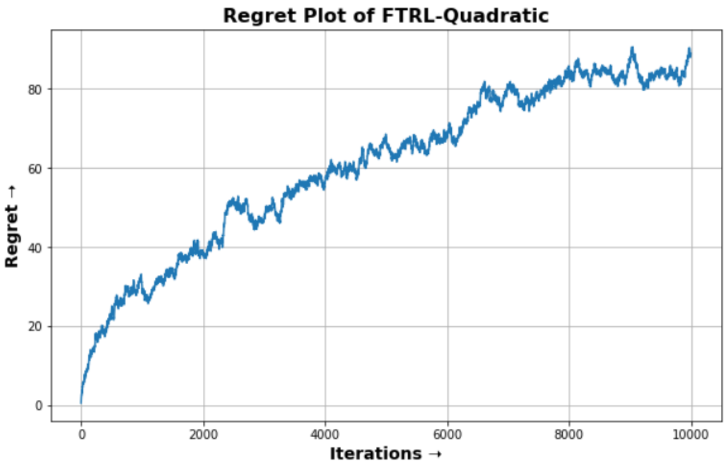

Figure 1: Regret plot for FTRL with Quadratic Regularization on the d-dimensional simplex

This plot is in agreement with the regret bound derived in class $2BL\sqrt{T}$ for the more general $\mathbb{R}^d$ case - basically is proportional to $\sqrt{T}$.

FTRL with Entropic Regularization

- Since FTRL with Entropic Regularization is equivalent to Hedge, the update rule is:

$$w_{t+1} = w_t e^{-\eta z_t}$$
$$p_{t+1} = \frac{w_t}{||w_t||_1}$$

where z_t is the loss vector generated by the adversary, w_t is a weight vector that is proportional to p_t initialised as [1, 1,...., 1], p_t is the algorithm's guess of vector initialised as [1/d, 1/d,...., 1/d], and η

is set to be $\sqrt{\frac{\log d}{TB^2}}$ according to the regret analysis done in class. Here, B is the upper bound on the adversary loss vector norms ($||z_t|| \leq B$).

Since p_t is always normalised with respect to the L1- norm of w_t, it remains in the simplex as time progresses.

- The regret plot is shown below:

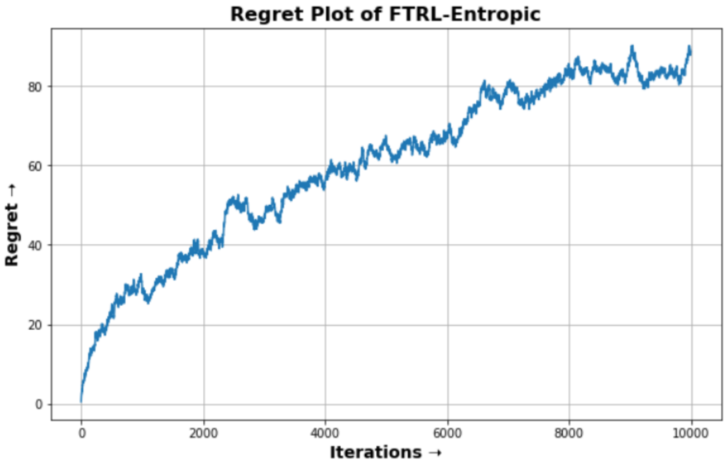

Figure 2: Regret plot for FTRL with Entropic Regularization on the d-dimensional simplex

This plot is in agreement with the regret bound derived in class $O(\sqrt{T \log d})$ - basically is proportional to $\sqrt{T}$.

FTL with Randomization

- A random 1000 dimensional vector R sampled uniformly from [0, $1/\eta$], and the given update rule used for prediction is:

$$p_{t+1} = \arg\min_{p \epsilon \Delta_d} \sum_{i=1}^{T} (p^T(z^i + R))$$

where z_i is the loss vector generated by the adversary at the i^{th} time step, p_t is the algorithm's guess of vector, and η is to be set as $\sqrt{\frac{2}{Td}}$, in accordance with the reference here. Basically, η is set in the same way as in our analyses for FTRL - trying to derive the regret bound in terms of η and then finding the value of η that minimises the regret bound.

- The regret plot is shown below:

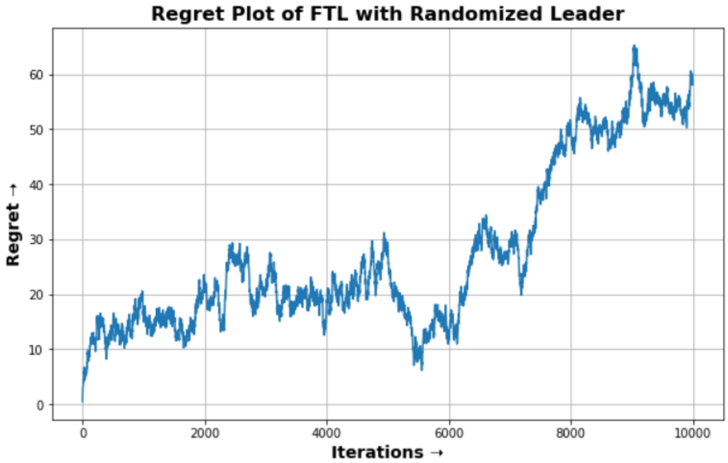

Figure 3: Regret plot for FTL with Randomization on the d-dimensional simplex

This plot is in agreement with the regret bound provided in the reference - the regret is bounded by a constant times $\sqrt{T}$ once again.

Below is a graph showing all three regret plots together. The FTL with Randomization algorithm seems to have a **better** regret bound than FTRL with both types of regularization:

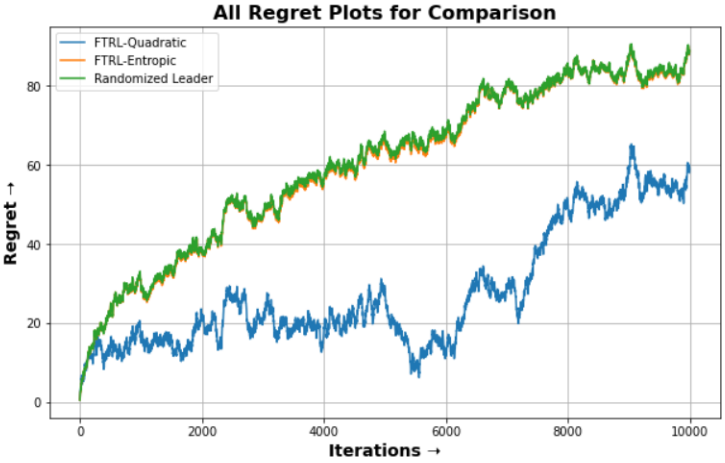

Figure 4: All Regret Plots

2 Question 2

Shown below is the plot of the regret along with the error bars which span ±1 standard deviation. Loss vectors for arms are generated according to 10 beta distributions with parameters $\alpha = [5, 5, 5, 5, 5, 5, 5, 5, 5, 5]$ and $\beta = [5, 5, 5, 5, 5, 5, 5, 5, 5, 10]$. Also, the plot of variance vs number of iterations is shown alongside.

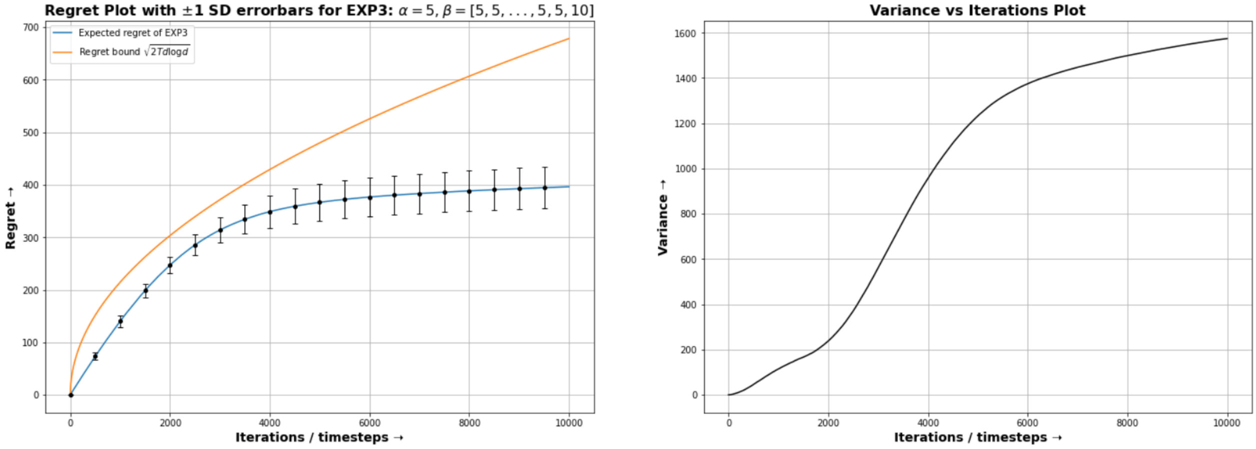

Figure 5: Regret plot for EXP3 with errorbars

- We observe that the regret bound of EXP3 derived in class, $\sqrt{2Td\log d}$, is satisfied by the expected regret curve.
- The errorbars have been plotted as ±stdev, and they themselves grow with increasing iterations, indicating a large variance. It is further made clear from the variance vs iterations plot that the variance becomes very large with time (variance in regret is over 1500 at the end of 10000 iterations). Therefore, it is indeed the case that EXP3 is Mean Wise but Variance Foolish. We miss this aspect of EXP3 in our analysis because we only consider the unbiasedness of the EXP3 estimator of the loss vector. Therefore, we obtain an algorithm that does well in expectation (mean wise) but there is no guarantee on stability (variance foolish). We would need a consistent estimator (not just unbiased) to get a variance wise algorithm too.

- Below are some examples of regret plots after shape parameters are modified:

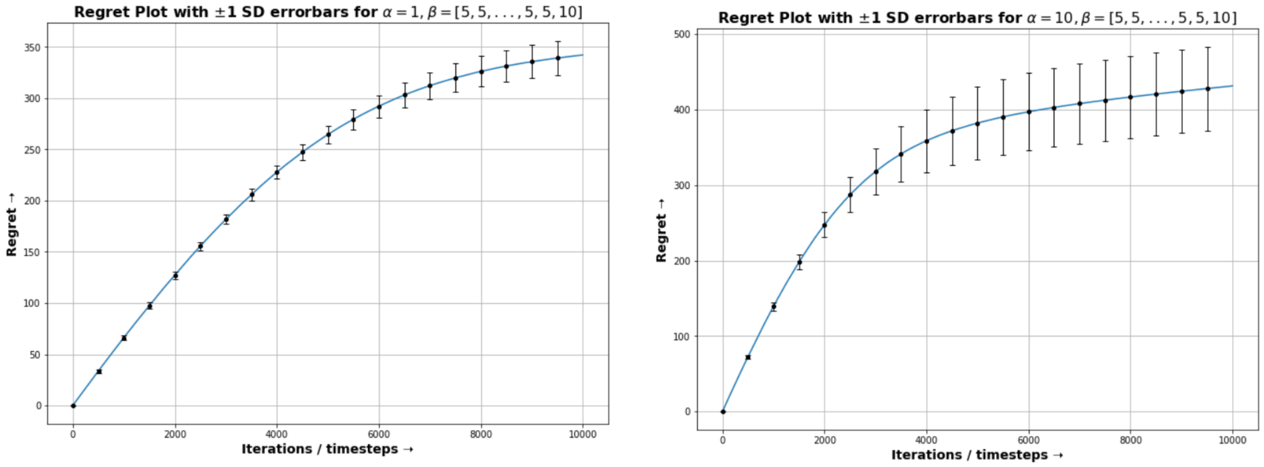

Figure 6: Varying only parameter α of the beta distributions

When α for all distributions is reduced to 1, the variance of EXP3 reduces.

When α for all distributions is increased to 10 however, the variance of EXP3 increases compared to $\alpha = 5$.

This is probably because at $\alpha = 1$, $\beta = 5$ (first 9 arms) and $\beta = 10$ (10th arm) both are quite similar distributions which return both low losses with high probabilities. The mean of the beta distribution $= \frac{\alpha}{\alpha+\beta} = 1/6$ and $1/11$ respectively, which are nearby values. Therefore, there is not much difference in various runs of EXP3. However, at $\alpha = 10$, $\beta = 5$ (first 9 arms) and $\beta = 10$ (10th arm) are quite different distributions. $\beta = 5$ distribution has a significantly higher mean (mean=0.66) than $\beta = 10$ (mean = 0.5). Therefore, in different runs of EXP3 the trajectories of regret can look quite different because the difference in maximum loss and minimum loss across arms tends to be high.

When β for the last arm is reduced to 1 and 5, the variance of EXP3 decreases. However, so does the regret, which is why the errorbars look larger. In the case where β of the last arm is also equal to 5, all arms are the same, therefore the loss of the best arm will not be too far off from the algorithm's choice, which leads to low regret irrespective of what arm is picked.

When β for the last arm is increased to 15, the variance of EXP3 again decreases.

Bonus: EXP3 uses an unbiased estimator to estimate the adversary loss vectors at each round, and accrues a large amount of variance. This algorithm, hence, lies on one extreme of the bias-variance tradeoff. In order to reduce variance compared to EXP3, we can use an algorithm which makes uses of a slightly biased estimator but produces lesser variance in regret.

More specifically, instead of directly using the obtained loss values to produce an unbiased estimator for the loss vector l_t in expectation, we can use an upper confidence bound of this loss to produce the estimator. What this will do is bias the expectation of the loss vector above the true loss vector (biased estimation), and therefore the regret bound will also be less tight. However, the arms with high variance will be weighted very low after a Hedge update, therefore the algorithm will be forced to first select all arms enough times so that variance is drastically decreased.

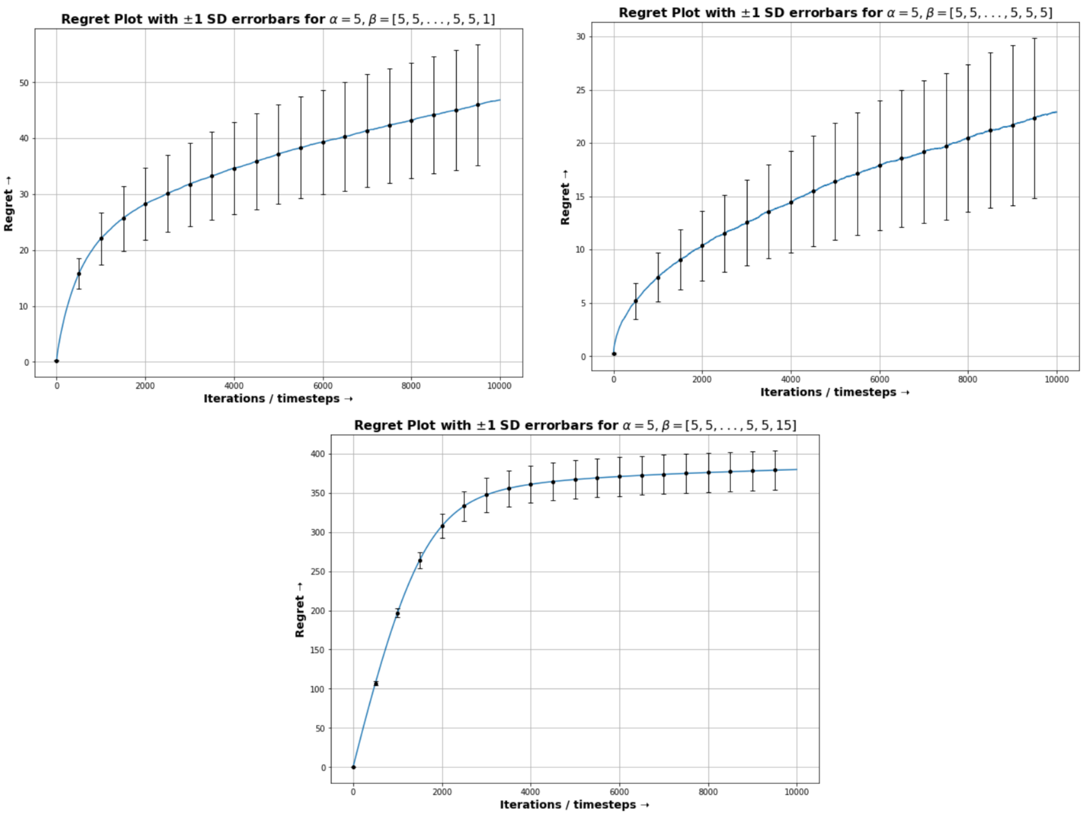

Figure 7: Varying only parameter β of the beta distribution of the LAST arm

3 Question 3

For ϵ-greedy, EXP3, and greedy after round-robin exploration are executed. ϵ-greedy is played with 4 values: $\epsilon = [0.01, 0.03, 0.1, 0.3]$.

Reward vectors for arms are generated according to 10 beta distributions with parameters $\alpha = [5, 5, 5, 5, 5, 5, 5, 5, 5, 5]$ and $\beta = [5, 10, 15, 20, 25, 30, 35, 40, 45, 50]$.

The losses are then calculated as (1-reward), since the losses are in the interval [0,1]. These losses are used for the EXP3 algorithm. Further, the regret for ϵ-greedy is calculated as follows:

$$\begin{aligned}
R &= \text{cumulative loss of algorithm} - \text{cumulative loss of best arm so far} \\
&= \sum_{t=1}^{T}(1 - r_i^t) - \min_{i^*} \sum_{t=1}^{T}(1 - r_{i^*}^t) \\
&= \max_{i^*} \sum_{t=1}^{T}(r_{i^*}^t) - \sum_{t=1}^{T}(r_i^t) \\
&= \text{cumulative reward of best arm so far} - \text{cumulative reward of algorithm}
\end{aligned}$$

Shown below are the regret plots along with the error bars which span ±1 standard deviation for these algorithms.

- ϵ-greedy, especially for higher values of ϵ, seems to suffer linear regret. This is because if the ϵ value is high enough, the sub-optimal arms continue to be picked with probability ϵ in every round even after the algorithm has learnt good estimates of rewards and the optimal arm as well. This leads to a linear accumulation of losses that does not slow down despite sufficient exploration. The $\epsilon = 0.3$ case goes well above EXP3's regret in the first 10000 steps as seen.
- On the other hand, if the exploration in ϵ-greedy is not performed, we can see that the regret still tends to be high and linear, as seen in the round-robin greedy case. Also, the variance for this case is

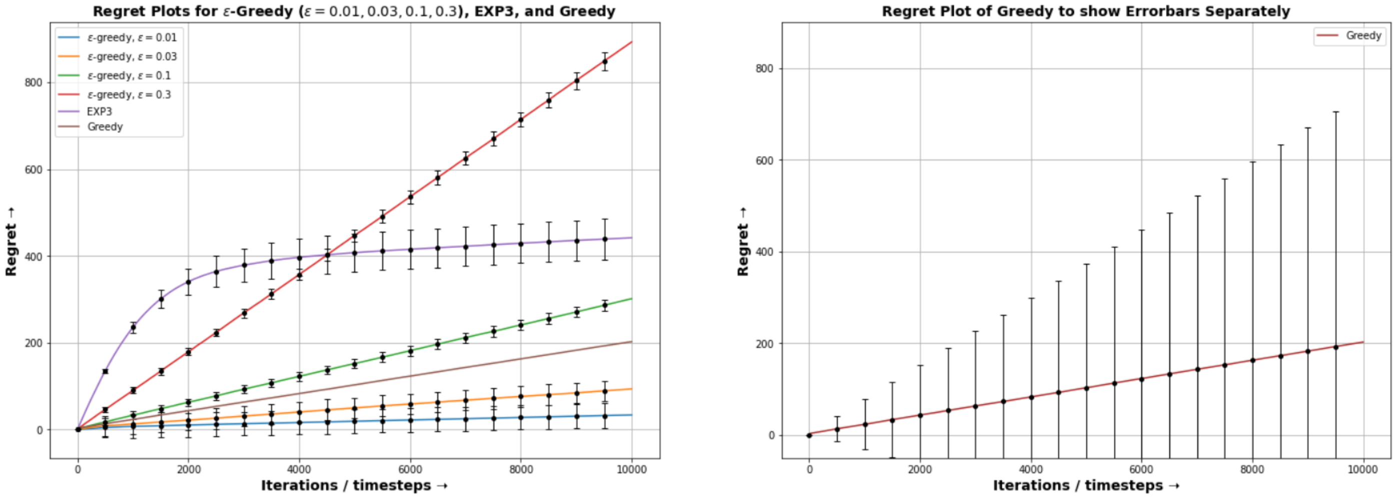

Figure 8: Regret plots for the various algorithms

extremely high (the errorbars are so large that it is clearer to plot them separately). since the greedy algorithm can get stuck on any arm in each run based on just one round of round-robin. Even if the worst arm gives the best reward in the first round, it can easily become our chosen arm for all time steps if the average reward of the worst arm remains higher than all the other arms' first reward (unluckily). Even if this does not happen, the second-worst arm could get stuck as the choice forever, and so on. This leads to a very unstable algorithm, therefore exploration is important.

- With the right choice of exploration probability ϵ, we can obtain a good algorithm. We can see that the $\epsilon = 0.01$ case has low regret, and the slope of regret seems to even decrease a little after the initial exploration stage, when exploitation has been maximised. It looks to be better than EXP3. Perhaps an even lower value of ϵ such as 0.005 will do even better.

Bonus: Regret Analysis for ϵ-Greedy

Let r^*_T denote the actual best reward at time step T, and $\mathbb{E}[r^*_T] = \mu^*_T$ denote its expected value.

Similarly, let $\bar{r}^*_T$ denote the estimated/empirical best reward so far, at time step T, and $\mathbb{E}[\bar{r}^*_T] = \bar{\mu}^*_T$ denote its expected value. Let $\bar{r}^i_T$ denote the empirical reward of any arm i, and $\bar{\mu}^i_T$ denote its mean.

Let R denote the regret of the ϵ-greedy algorithm. The regret at a single time step can be given as:

$$R_T = r^*_T - \bar{r}^*_T$$

if the exploitation is done with probability $1 - \epsilon$, or:

$$R_T = r^*_T - \bar{r}^i_T$$

if the uniform random exploration with probability ϵ/d (because there are d arms) selects arm i.

We shall try to give a guarantee on the expected regret at time step T.

$$\begin{aligned}
\mathbb{E}[R_T] &= \mathbb{E}[(1-\epsilon)[r^*_T - \bar{r}^*_T] + \frac{\epsilon}{d}(r^*_T - \bar{r}^1_T) + \frac{\epsilon}{d}(r^*_T - \bar{r}^2_T) + \ldots] \\
&= \mathbb{E}[(1-\epsilon)[r^*_T - \bar{r}^*_T] + \frac{\epsilon}{d}\sum_{i=1}^{d}(r^*_T - \bar{r}^i_T)] \\
&= (1-\epsilon)[\mathbb{E}[r^*_T] - \mathbb{E}[\bar{r}^*_T]] + \frac{\epsilon}{d}\sum_{i=1}^{d}(\mathbb{E}[r^*_T] - \mathbb{E}[\bar{r}^i_T]) \\
&= (1-\epsilon)[\mu^*_T - \bar{\mu}^*_T] + \frac{\epsilon}{d}\sum_{i=1}^{d}(\mu^*_T - \bar{\mu}^i_T)
\end{aligned}$$

Now, we do not have access to the true expected reward of the true best arm. However, after enough iterations of ϵ-greedy, sufficient exploration has been done such that the empirical average reward of the best arm is close to the true average reward. Therefore, we replace μ_T^* with $\bar{\mu}_T^*$ considering the best case scenario. The first term in the expected regret goes to 0. We have:

$$\mathbb{E}[R_T] = \frac{\epsilon}{d} \sum_{i=1}^{d} (\bar{\mu}_T^* - \bar{\mu}_T^i)$$

Let us define the empirical average reward of the 2nd best arm (sub-optimal arm) as $\bar{\mu}_T^{2nd}$ (smallest difference from the optimal arm reward). Then each term in the above summation is atleast $\bar{\mu}_T^* - \bar{\mu}_T^{2nd}$. Therefore,

$$\begin{aligned} \mathbb{E}[R_T] &\geq \frac{\epsilon}{d} \sum_{i=1}^{d} (\bar{\mu}_T^* - \bar{\mu}_T^{2nd}) \\ &\geq \frac{\epsilon}{d} \cdot d(\bar{\mu}_T^* - \bar{\mu}_T^{2nd}) \\ &\geq \epsilon(\bar{\mu}_T^* - \bar{\mu}_T^{2nd}) \end{aligned}$$

Since $\bar{\mu}_T^* - \bar{\mu}_T^{2nd} > 0$, we have that expected regret $E[R_T] > 0$ at each time step T.

Therefore, the regret of ϵ-greedy with constant ϵ cannot be sub-linear. More intuitively, the fact that ϵ-greedy with constant ϵ keeps on choosing sub-optimal arms even after it has explored enough to identify the optimal one, leads to a constant accumulation of regret.

We can also say that the probability of choosing a sub-optimal arm is very low for very small ϵ, and therefore the probability of super-linear regret is also very small. This explains the linear regret observed in the plots above.

* *