Exercise #2

Shrout Oren

EE 048843 - Exploration and Exploitation by Natural and Artificial Agents

May 24, 2020

1 Theoretical

Exercise T1 Regret bound for the ϵ-greedy algorithm (Exercise 1.2 from Slivkins' book)

Consider the ϵ-greedy algorithm below with $\epsilon_t = t^{-\frac{1}{3}}(Klogt)^{\frac{1}{3}}$. Show that $E[R(T)] \leq t^{\frac{2}{3}}\,\mathcal{O}((Klogt)^{\frac{1}{3}})$ for each round t.

Hint: Fix round t and analyze $E[\Delta(at)]$ for this round separately. Set up the "clean event" for rounds 1,...,t much like in Section 1.3.1 (treating t as the time horizon), but also include the number of exploration rounds up to time t.

Epsilon-greedy algorithm
for t = 1,2,... **do**
  Toss a coin with success probability ϵ_t
  if success **then**
    Explore: choose arm uniformly at random
  else
    Exploit: choose the arm with highest empirical reward
  end if
end for

Solution. We start with

$$\begin{aligned} E[R(t)] &= E[R(t)|cleanevent] \cdot p(cleanevent) + E[R(t)|badevent] \cdot p(badevent) \\ &\leq E[R(t)|cleanevent] + t\,\mathcal{O}(t^{-2}) \end{aligned}$$

We used the "clean event" as in section 1.3.1 where we saw that:

$$\text{Clean event}: \quad \varepsilon \stackrel{\triangle}{=} \{\forall a \forall t \;\; |\bar{v}_j(a) - \mu(a)| \leq \rho_t(a)\}$$

And by setting a fixed t we get by define $\rho_t(a) = \sqrt{\frac{2\log t}{n_t(a)}}$ that $p(e) \geq 1 - \frac{2}{t^2}$. Therefore assuming "clean event":

$$\triangle(a_t) \leq \mathcal{O}(\rho_t(a)) = \mathcal{O}\left(\sqrt{\frac{\log t}{n_t(a)}}\right)$$

$$R(t;a) \stackrel{\triangle}{=} \Delta(a_t) n_t(a) \leq \mathcal{O}\left(\sqrt{\log t \; n_t(a)}\right)$$

$$R(t) \leq \sum_a R(t;a) \leq \mathcal{O}\left(\sum_a \sqrt{\log t \; n_t(a)}\right) \stackrel{\star}{\leq} \mathcal{O}\left(K\sqrt{\frac{1}{K}\log t \; \sum_a n_t(a)}\right) = \mathcal{O}\left(\sqrt{Kt\log t}\right)$$

Where at $\star$ we used Jensen inequality. Therefore we get the following

$$\begin{aligned} E[R(t)] &\leq \varepsilon_t \cdot \mathcal{O}(t) + \mathcal{O}(\sqrt{t\log t}) \\ &= \mathcal{O}(t^{\frac{2}{3}}(K\log t)^{\frac{1}{3}})) + \mathcal{O}(t^{\frac{1}{2}}(\log t)^{\frac{1}{2}})) \end{aligned}$$

Where the first term is from explore and the second term from exploit. One may see that we can drop the second term as we have $t^{\frac{2}{3}}$ in the first term and $t^{\frac{1}{2}}$ in the second term. Also we neglect the "bad event" as the "clean event" has a higher bound. Therefore we get

$$E[R(t)] \leq \mathcal{O}(t^{\frac{2}{3}}(K\log t)^{\frac{1}{3}}))$$

Exercise T2 Anytime algorithms (Exercise 1.5 from Slivkins' book)

Take any bandit algorithm $\mathcal{A}$ for fixed time horizon T. Convert it to an algorithm $\mathcal{A}_\infty$ which runs forever, in phases i = 1, 2, 3, ... of 2^i rounds each. In each phase the algorithm A is restarted and run with time horizon 2^i.

State and prove a theorem which converts an instance-independent upper bound on regret for $\mathcal{A}$ into similar bound for $\mathcal{A}_\infty$ (so that this theorem applies to both UCB1 and Explore-first).

Solution. We start by define $T_i = 2^i$ to be the time horizon T at phase i, where $i = 1, 2, 3, ...$ We run the algorithm $\mathcal{A}$ for every i and get algorithm which will be denote as $\mathcal{A}_\infty$, which run for ever.
One may see that for a fixed phase i, T_i is known and therefore algorithm $\mathcal{A}$ is defined. With mentioned above, for UCB1 algorithm we define $\rho_t(a) = \sqrt{\frac{2\log T_i}{n_t(a)}}$ using the reward tape, clean event: $\varepsilon \stackrel{\triangle}{=} \{\forall a \forall t \;\; |\bar{\mu}_t(a) - \mu(a)| \leq \rho_t(a)\}$ and using the recitation notes we get

$$\triangle(a_t) = \mu(a^*) - \mu(a_t) \leq 2\rho_t(a_t) = 2\sqrt{\frac{2\log T_i}{n_t(a_t)}}$$

$$R(T_i;a) \stackrel{\triangle}{=} \Delta(a_t) n_t(a) \leq \mathcal{O}\left(\sqrt{n_t(a)\log T_i}\right)$$

$$
\begin{aligned}
R(T_i) &\leq \sum_a R(T_i; a) \\
&\leq \mathcal{O}\Big(\sum_a \sqrt{\log T_i \; n_t(a)}\Big) \\
&\overset{\star}{\leq} \mathcal{O}\Big(K\sqrt{\frac{1}{K}\log t \; \sum_a n_t(a)}\Big) \\
&= \mathcal{O}\left(\sqrt{KT_i \log T_i}\right)
\end{aligned}
$$

Where at $\star$ we used Jensen inequality. And for Explore-First algorithm we define $\rho(a) = \sqrt{\frac{2\log T_i}{N}}$ and using the "clean event"

$$\sup_a |\bar{\mu}(a) - \mu(a)| \leq \rho(a)$$

we get

$$R(T_i) \leq KN + \mathcal{O}(\sqrt{\frac{\log T_i}{N}}) \cdot \mathcal{O}(T_i)$$

Where the first term is explore and the second is exploit. By balance between the terms we get that

$$
\begin{aligned}
N &= (\frac{T_i}{K})^{\frac{2}{3}} (\log T_i)^{\frac{1}{3}} \\
E[R(T_i)] &\leq \mathcal{O}(T_i^{\frac{2}{3}} (K \log T_i)^{\frac{1}{3}})
\end{aligned}
$$

Therefore we get that the upper bound on the regect for $\mathcal{A}_\infty$ is equal to the one on the regret for algorithm $\mathcal{A}$ where we substitute T by T_i and running for every phase i.

Exercise T3 Bayesian methods

1. Consider the standard Bayesian data generating process. Let $\theta \sim p_0(\theta)$ and $D_n = \{x_1, x_2, ..., x_n\} \overset{iid}{\sim} f(\cdot, \theta)$. Consider the Bayesian objective

$$L\{p(\cdot|D_n); p\} = \iiint \big(f(x;\theta) - p(x|D_n)\big)^2 p(D_n|\theta) p_0(\theta) dD_n d\theta dx$$

Show that the pdf $p(x|D_n)$ minimizing this objective is given by

$$
\begin{aligned}
p(x|D_n) &= \int f(x;\theta) p(\theta|D_n) d\theta \\
p(\theta|D_n) &= \frac{p(D_n|\theta) p_0(\theta)}{p(D_n)}
\end{aligned}
$$

2. Let $D_n = \{x_i\}_{i=1}^n$ be i.i.d Bernoulli random variables, $x_i \sim Ber(\theta)$. Let the prior distribution for $\theta \in (0,1)$ be $Beta(\alpha, \beta)$ where $\alpha > 0$ and $\beta > 0$.

   (a) Compute the optimal Bayes estimator for θ, based on D_n, and compare your results to the maximum likelihood estimator. In particular, compare the estimators for small and large sample sizes.

   (b) Consider the posterior distribution of θ for $(\alpha, \beta) = (1,1)$ and $(\alpha, \beta) = (2,2)$. Consider two observation sets, $D^{(1)}$ with 4 observations of $x_i = 1$ and 1 observation with $x_i = 0$, and $D^{(2)}$ with 40 observations of $x_i = 1$ and 10 observations of $x_i = 0$. Plot the prior, likelihood and posterior pdfs for each of the 4 combinations of prior parameters and data sets. Explain how the sample size and prior parameters affect the results.

Solution. We start by take the derivative of $L\{p(\cdot|D_n)\}$ WRT $p(x|D_n)$ and compare to zero gives us

$$\iiint 2\big(p(x|D_n) - f(x;\theta)\big)p(D_n|\theta)p_0(\theta)dD_n d\theta dx = 0$$

$$\iiint p(x|D_n)p(D_n|\theta)p_0(\theta)dD_n d\theta dx = \iiint f(x;\theta)p(D_n|\theta)p_0(\theta)dD_n d\theta dx$$

Taking the integral over θ on the left term, and using Bayes law on the right term gives us

$$\iint p(x|D_n)p(D_n)dD_n dx = \iint \Big[\int f(x;\theta)p(\theta|D_n)p(D_n)d\theta\Big]dD_n dx$$

Take the derivative WRT D_n and x gives us the following

$$p(x|D_n)p(D_n) = \int f(x;\theta)p(\theta|D_n)p(D_n)d\theta$$

$$p(x|D_n)p(D_n) = p(D_n)\int f(x;\theta)p(\theta|D_n)d\theta$$

$$p(x|D_n) = \int f(x;\theta)p(\theta|D_n)d\theta$$

The optimal Bayes estimator for θ based on D_n using the MSE as risk is given by

$$\hat{\theta} = E[\theta|D_n] = \int (\theta|D_n)d\theta$$

Now using the fact that $\{x_i\}_{i=1}^n$ are iid we get that

$$\begin{aligned}
p(\theta|D_n) &= \prod_{i=1}^n p(\theta|x_i) = \prod_{i=1}^n \frac{p(x_i|\theta)p_0(\theta)}{p(x_i)} \\
&= \prod_{i=1}^n \frac{Ber(x_i|\theta)\cdot Beta(\theta|\alpha,\beta)}{\int Ber(x_i|\theta)\cdot Beta(\theta|\alpha,\beta)d\theta} \\
&\overset{\star}{=} \prod_{i=1}^n \frac{Beta(\theta|\alpha+x_i,\beta+1-x_i)}{\int Beta(\theta|\alpha+x_i,\beta+1-x_i)d\theta} \\
&\overset{\star\star}{=} \prod_{i=1}^n Beta(\theta|\alpha+x_i,\beta+1-x_i)
\end{aligned}$$

Where at $\star\star$ we used the fact that integral over a pdf is one, and in $\star$ we used the following:

$$\begin{aligned}
Ber(x|\theta)\cdot Beta(\theta|\alpha,\beta) &= \theta^x(1-\theta)^{1-x}\frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha,\beta)} \\
&= \frac{\theta^{(\alpha+x)-1}(1-\theta)^{(\beta+1-x)-1}}{B(\alpha,\beta)} \\
&= \frac{B(\alpha+x,\beta+1-x)}{B(\alpha,\beta)}\frac{\theta^{(\alpha+x)-1}(1-\theta)^{(\beta+1-x)-1}}{B(\alpha+x,\beta+1-x)} \\
&= \frac{B(\alpha+x,\beta+1-x)}{B(\alpha,\beta)}Beta(\theta|\alpha+x,\beta+1-x)
\end{aligned}$$

Now we will simplify $p(\theta|D_n)$

$$\begin{aligned}
p(\theta|D_n) &= \prod_{i=1}^n Beta(\theta|\alpha+x_i,\beta+1-x_i) \\
&= \prod_{i=1}^n \frac{\theta^{(\alpha+x_i)-1}(1-\theta)^{(\beta+1-x_i)-1}}{B(\alpha+x_i,\beta+1-x_i)} \\
&= \frac{\theta^{\sum_i\left(\alpha+x_i-1\right)}(1-\theta)^{\sum_i\left(\beta+1-x_i-1\right)}}{\prod_i B(\alpha+x_i,\beta+1-x_i)} \\
&\overset{\star}{=} \frac{\theta^{n\alpha+\sum_i x_i-n}(1-\theta)^{n\beta+n-\sum_i x_i-n}}{B(n\alpha+\sum_i x_i+1-n, n\beta-\sum_i x_i+1)} \\
&= Beta(\theta|n\alpha+\sum_i x_i+1-n, n\beta-\sum_i x_i+1)
\end{aligned}$$

Where at $\star$ we used the definition of $B(\cdot,\cdot)$

$$\begin{aligned}\prod_i B(\alpha+x_i,\beta+1-x_i) &= \prod_i \int_0^1 t^{\alpha+x_i-1}(1-t)^{\beta+1-x_i-1}dt \\ &= \int_0^1 t^{n\alpha+\sum_i x_i-n}(1-t)^{n\beta+n-\sum_i x_i-n}dt \\ &= B(n\alpha+\sum_i x_i+1-n, n\beta-\sum_i x_i+1)\end{aligned}$$

Therefore the optimal Bayes estimator is given by

$$\hat{\theta}_{Bayes} = E_{\theta\sim p(\theta|D_n)}[\theta|D_n] = \frac{n(\alpha-1)+\sum_i x_i+1}{n(\alpha+\beta-1)+2}$$

Note that we used the fact that $E_{\theta\sim Beta(\alpha,\beta)}[\theta] = \frac{\alpha}{\alpha+\beta}$.

Now using MLE will give us

$$\begin{aligned}\hat{\theta} &= \arg\max_\theta p(D_n|\theta) = \arg\max_\theta \prod_{i=1}^n p(x_i|\theta) \\ &= \arg\max_\theta \prod_{i=1}^n \theta^{x_i}(1-\theta)^{1-x_i} \\ &= \arg\max_\theta \sum_{i=1}^n \left[x_i\log(\theta)+(1-x_i)\log(1-\theta)\right]\end{aligned}$$

Take the derivative WRT θ and compare to zero yields

$$\begin{aligned}\sum_{i=1}^n \left[\frac{1}{\theta}x_i - \frac{1}{1-\theta}(1-x_i)\right] &= 0 \\ \sum_{i=1}^n (1-\theta)x_i &= \sum_{i=1}^n \theta(1-x_i) \\ \hat{\theta}_{MLE} &= \frac{1}{n}\sum_{i=1}^n x_i \triangleq \bar{x}_n\end{aligned}$$

In order to compare between the estimators we will look for small and large n. For small n we see that the Bayes estimator uses the prior parameters with the samples to get a better estimation. However for large n the Maximum Likelihood Estimator leans on the mean of the

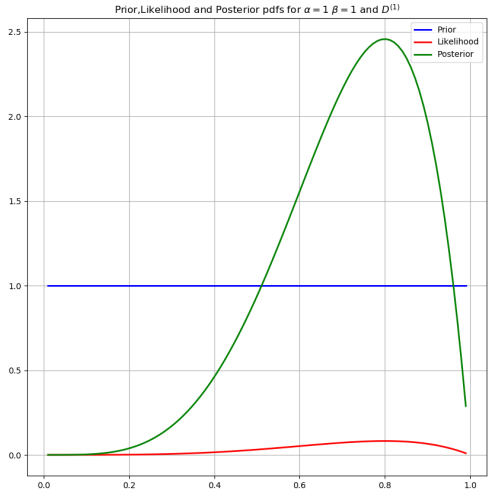
samples $\hat{\theta}_{MLE} = \bar{x}_n \overset{n \to \infty}{\longrightarrow} Ex = \theta$, where the Bayes estimator may get a wrong estimation. For example we can see that by taking $\theta = \beta = 5$ we get

$$\hat{\theta}_{Bayes} = \frac{(\alpha - 1) + \frac{1}{n}\sum_i x_i + \frac{1}{n}}{(\alpha + \beta + 1) + \frac{2}{n}} \overset{n \to \infty}{\longrightarrow} \frac{\alpha - 1 + \theta}{\alpha + \beta - 1} = 1 \neq \theta$$

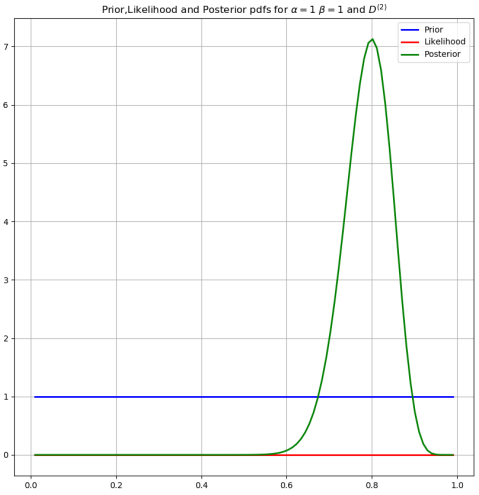
Now we will Consider the posterior distribution of θ for $(\alpha, \beta) = (1, 1)$ and $(\alpha, \beta) = (2, 2)$ with 2 observation sets, $D^{(1)}, D^{(2)}$ as mentioned above. In figure [1] we have all the 4 combinations. One may see that the prior parameters are responsible for for the mean of the posterior which is the Bayes estimator, and the sample size is responsible on the confident of this estimation. The more samples the less is the std of the posterior, the more we are confident of this estimation.

2 Empirical

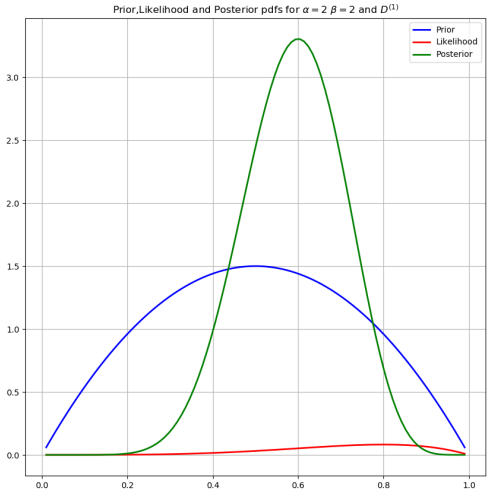
In this Part we are going to compare between the various algorithms of Multi-Armed Bandits. We started by taking the sets $K = \{2, 10, 100\}, \delta = \{0.1, 0.01\}$ and for each case run for $T = 10^6$ and repeated 10 times. Note that due to the complex of these algorithms we decide to run for 10^6 instead of 10^7 (estimation time is 22 hours). In figure [2] we have the value of the mean regret versus time for all the cases of K and δ. One may see that for small value of δ UCB1 method gives poor results, while for large δ and only a few number of arms this algorithm outperform the others. Moreover, for most of the cases Thompson sampling gave the best results. However for large number of arms and small value of δ, which is the hardest case, all methods except for the epsilon-greedy gave poor result. We conclude that for the hard cases, where there is many arms to exploit with small value of difference between the reward of the optimal arm and the sub-optimal arms, we want to keep taking the arm with the maximum empirical mean.
In figure [3] we have the value of the std of the regret versus time for all the cases of K and δ. From this figure we learn how robust each algorithm in each case. We can notice that even so that epsilon-greedy outperform the others in the hardest case ($K = 100,\ \delta = 0.01$) his std has gone through the roof. Therefore we deduce this algorithm is not stable and some simulations gave good results while others gave poor ones. Moreover for the most cases Thompson sampling's std is not worst than the other algorithms. Therefore we conclude that Thompson sampling results are stable. We also see that for large δ and only a few number of arms where UCB1 outperform the others his std is also the lowest and we can understand from that for this case UCB1 is stable and better. Finally we see that for high values of δ UCB-V has high std, while for small value of δ it has the smallest std, we conclude that this method is more stable to use in small values of δ.
In figure [4] we have the mean value of the exploration index versus time for all the cases of K and δ. The exploration index indicates the cumulative measure of the number of times the algorithm selected the non-greedy choice. From this figure we can conclude how choosing the greedy choice effects the mean regret in figure [2]. If we take figure [2] sub-figure (a) for



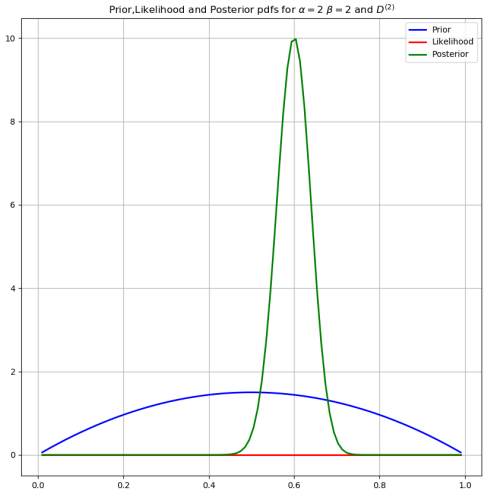
(a) $\alpha = \beta = 1$ and observation set $D^{(1)}$



(b) $\alpha = \beta = 1$ and observation set $D^{(2)}$

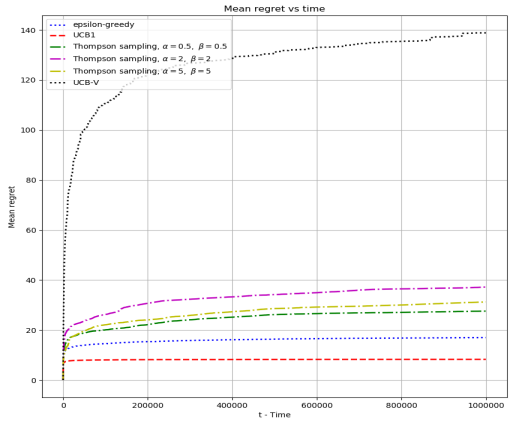


(c) $\alpha = \beta = 1$ and observation set $D^{(1)}$

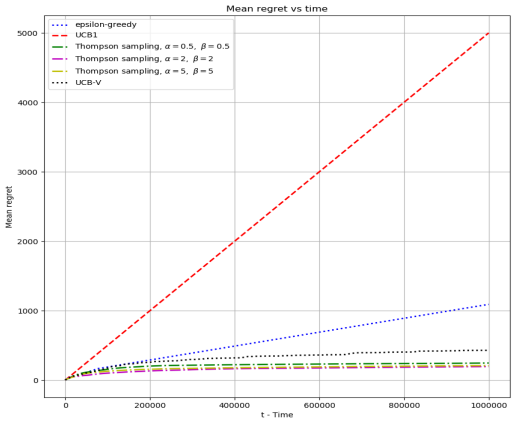


(d) $\alpha = \beta = 2$ and observation set $D^{(2)}$

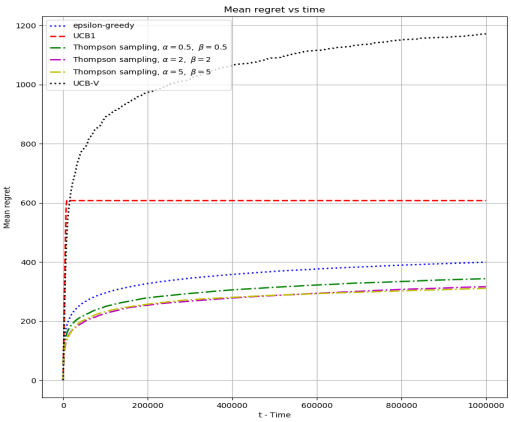
Figure 1: Prior, likelihood and posterior pdfs for each of the 4 combinations of prior parameters and data sets



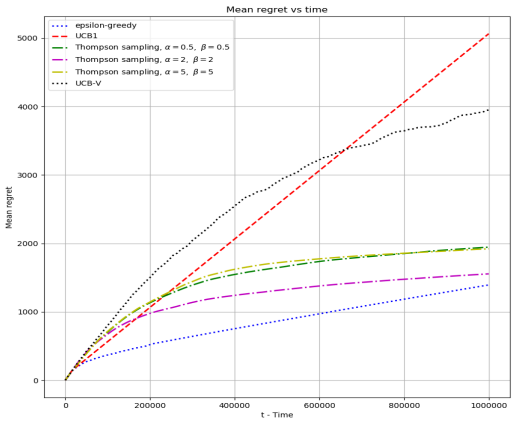
(a) $K = 2, \delta = 0.1$



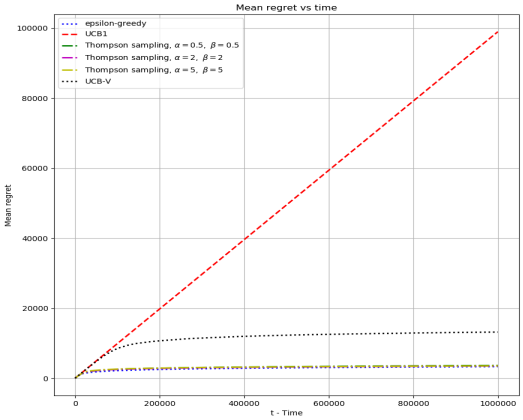
(b) $K = 2, \delta = 0.01$



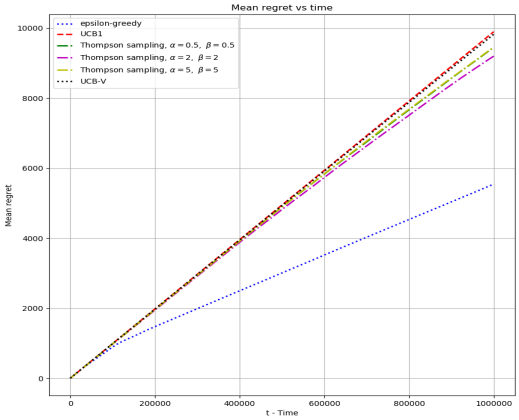
(c) $K = 10, \delta = 0.1$



(d) $K = 10, \delta = 0.01$

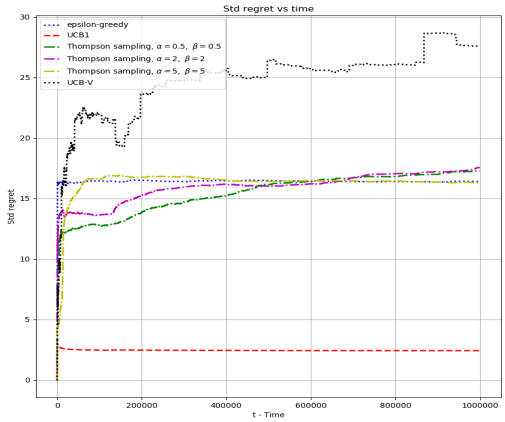


(e) $K = 100, \delta = 0.1$

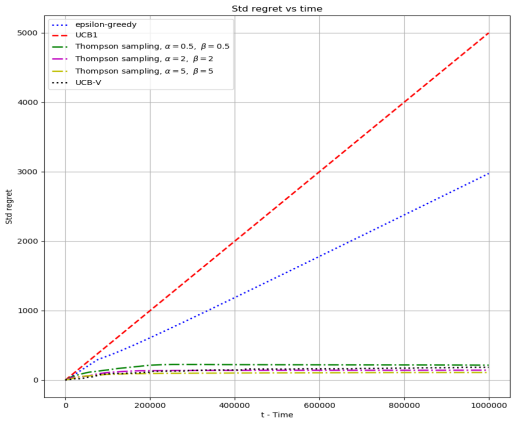


(f) $K = 100, \delta = 0.01$

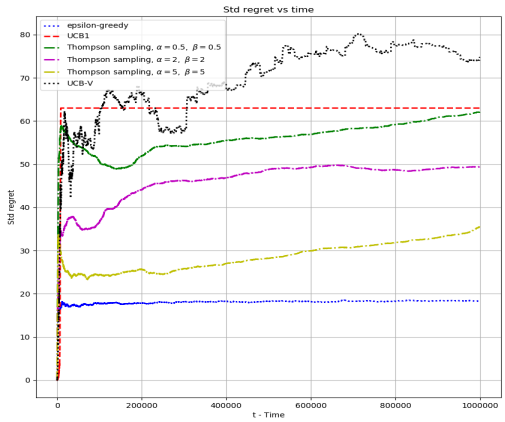
Figure 2: Value of the mean regret versus time for all the cases of $K = \{2, 10, 100\}, \delta = \{0.1, 0.01\}$



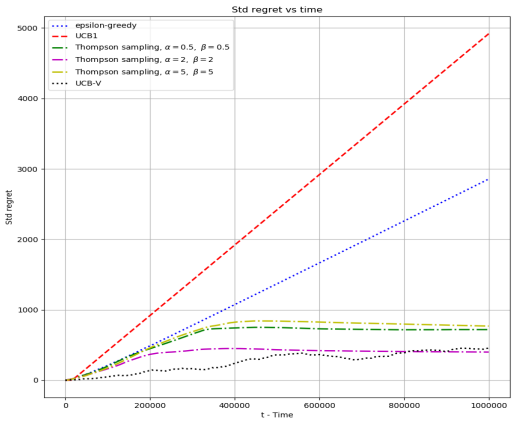
(a) $K = 2, \delta = 0.1$



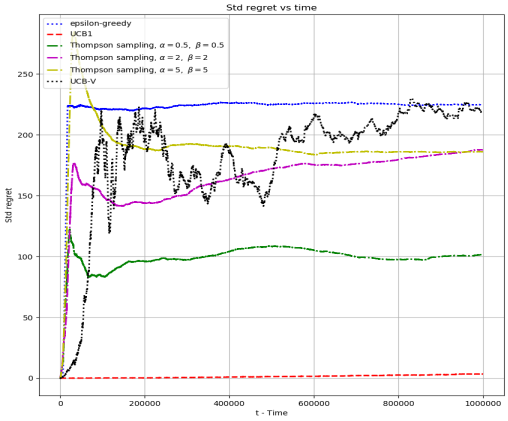
(b) $K = 2, \delta = 0.01$



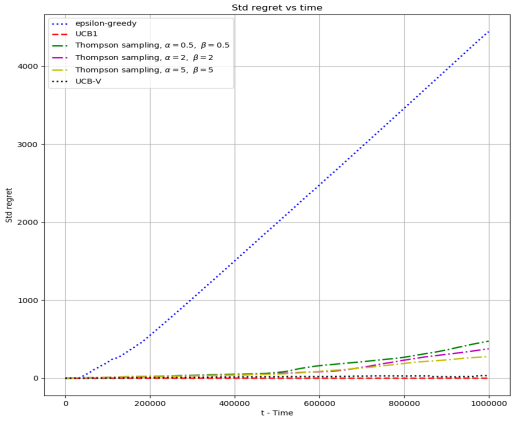
(c) $K = 10, \delta = 0.1$



(d) $K = 10, \delta = 0.01$



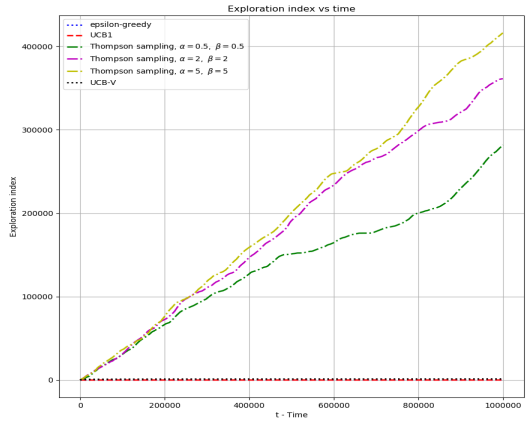
(e) $K = 100, \delta = 0.1$



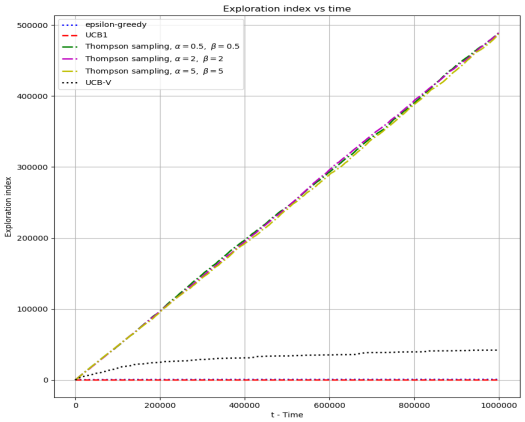
(f) $K = 100, \delta = 0.01$

Figure 3: Value of the std of the regret versus time for all the cases of $K = \{2, 10, 100\}, \delta = \{0.1, 0.01\}$

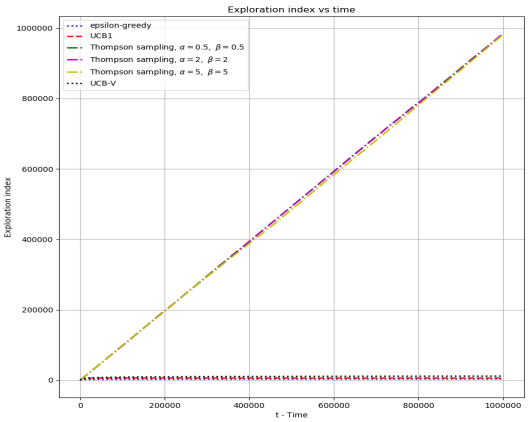
example, we see that while UCB1 and epsilon-greedy gave the best results and UCB-V gave the worst results, all these algorithms chose the greedy choice. Therefore choosing the greedy choice may help one algorithm while ruin for other ones. Moreover if we look sub-figure (f) we see that epsilon-greedy kept choosing the greedy-choice which what gave the best results in this case like we mentioned above. We also see that Thompson sampling method kept choosing the non-greedy choice which for the most cases gave better results. Therefore we conclude that there is no best method and every method perform better in different choice of number of arms to explore and exploit and different values of δ.
In figure [5] we show a comparison of UCB1 and UCB-V algorithms, where we take $K = 2$ Armed Bandits. In sub-figure (a) we constructed the following setup: The optimal arm's reward is Bernoulli RV with parameter $p = 0.51$, and the sub-optimal arms has deterministic reward of 0.48. We run the simulation for $T = 10^5$ and average over 100 repetitions. By taking the sub-optimal arms reward to be deterministic we get that the variance for the sub-optimal arms is zero, and therefore the upper Bound suppose to be smaller than the optimal arm's Bound. This setup is suppose to help the UCB-V algorithm in finding the optimal arm faster. We can see that indeed UCB-V outperform UCB1 and has almost no std, while UCB1 may have some simulations where it converge to the optimal arm. at average it gives poor results. In sub-figure (b) we constructed the following setup: The optimal arm has deterministic reward 0.51. while the sub-optimal arms reward is Bernoulli RV with parameter $p = 0.5$. We run the simulation for $T = 10^5$ and average over 100 repetitions as well. Oppose to the previous setup. this time the variance of the optimal arm is suppose to be zero, and therefore the upper Bound suppose to be smaller than the sub-optimal arms Bound. This setup is suppose to discourage UCB-V in finding the optimal arm, and indeed we see that UCB1 is outperform UCB-V. Although the std of UCB1 is much higher than of UCB-V we see that UCB-V is struggling with finding the optimal arm, much more than UCB1. We can conclude that UCB1 is not a stable algorithm and for some iteration it can find the optimal arm pretty quick. while at most of the time it can be stuck at sub-optimal arm.



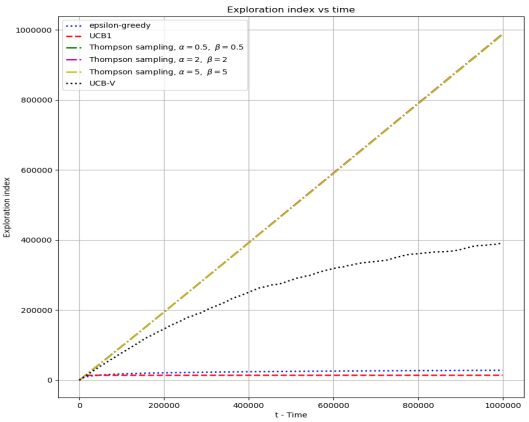
(a) $K = 2, \delta = 0.1$



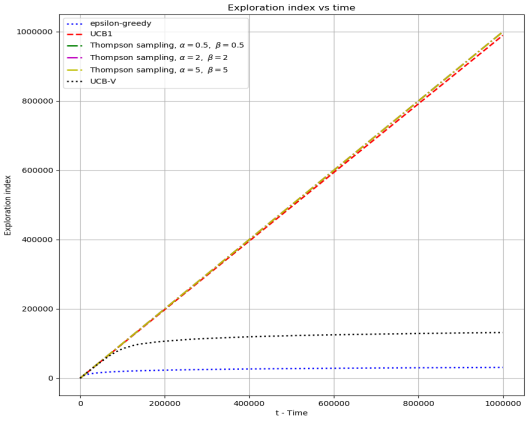
(b) $K = 2, \delta = 0.01$



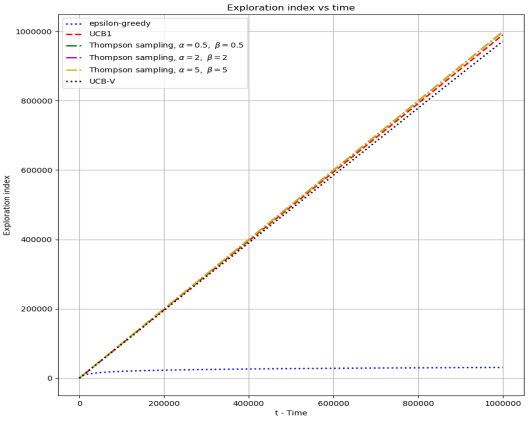
(c) $K = 10, \delta = 0.1$



(d) $K = 10, \delta = 0.01$

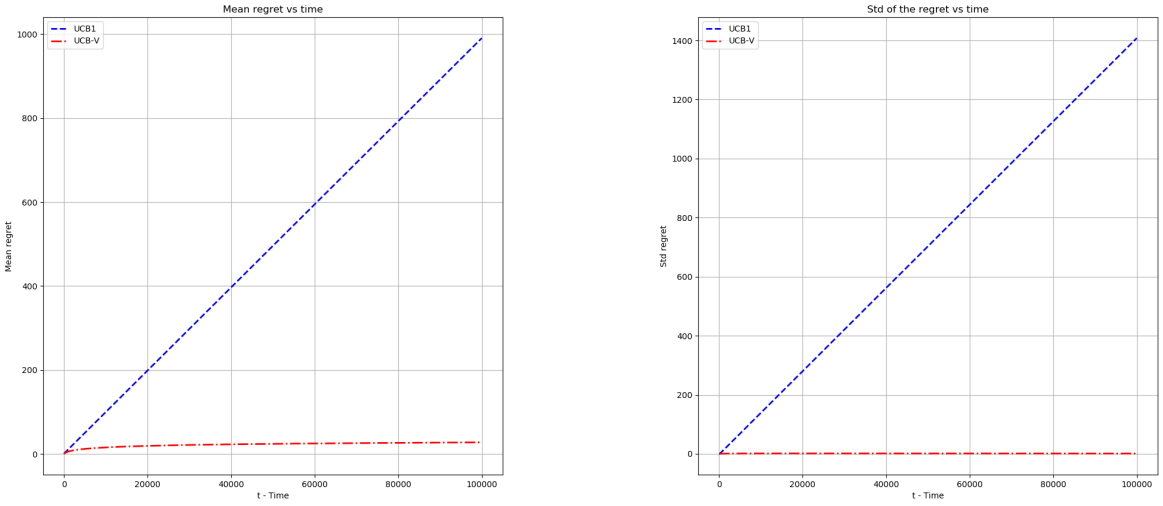


(e) $K = 100, \delta = 0.1$

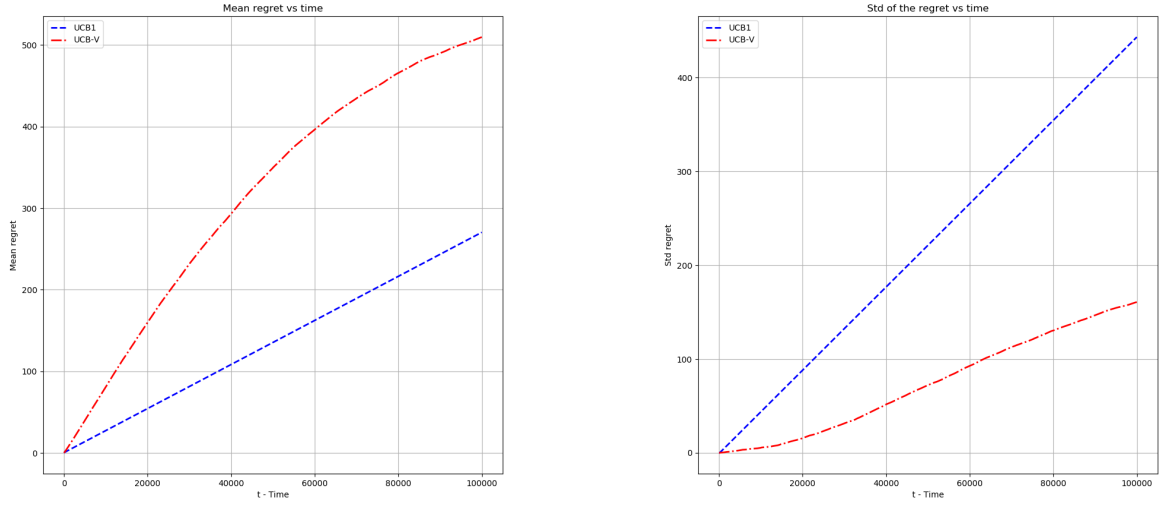


(f) $K = 100, \delta = 0.01$

Figure 4: Mean value of the exploration index versus time for all the cases of $K = \{2, 10, 100\}, \delta = \{0.1, 0.01\}$



(a) The optimal arm's reward is Bernoulli RV with p = 0.51, while the sub-optimal arms has deterministic reward of 0.48. On the left side we have the mean regret of both algorithms versus time, and on the right we have the std of the regret value versus time.



(b) The optimal arm has deterministic reward of 0.51, while the sub-optimal arms reward is Bernoulli RV with p = 0.5. On the left side we have the mean regret of both algorithms versus time, and on the right we have the std of the regret value versus time.

Figure 5: Comparison between UCB1 and UCB-V algorithms for 2 Armed Bandits. Both cases were computed over T = 10^5 and repeated 100 time.