Sequential Learning - Home Assignment

Pierre Fernandez, Paul Jacob

March 2021

Part 1. Rock Paper Scissors

1. When playing "Rock Paper Scissors", the player has 3 possible actions, and the adversary has 3 possible actions too, hence $M = N = 3$. Now we can choose different matrices depending on the chosen order of the actions. We can choose for instance the matrix:

$$L = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}$$

Full information feedback

2. *Implementation of* `EWA`.
   (a) (Code)
   (b) (Code)
3. *Simulation against a fixed adversary.*
   (a) If the adversary chooses the action j_t, the loss incurred by the player if he chooses action i at time t is :
   $$l_t(i) = L(i, j_t)$$
   (b) The evolution of the weight vectors is shown in figure 1. It looks like the best strategy is to always choose the action that beats action 1, which is the most played by the adversary. Indeed, the expected loss is the lowest this way. In our setting, the action that beats action 1 is the action 3 (because $L(3, 1) = -1$), so the player always chooses action 3 at the end.

   (c) The evolution of the average loss is shown in figure 2. Once the algorithm has converged towards always selecting action 3, the average loss seems to stay negative. Indeed, the expected loss when choosing action 3 is: $\mathbb{E}_j[L(3, j)] = 0 \times 0.25 + 1 \times 0.25 - 1 \times 0.5 = -0.25$, so in the long term, the player should observe an average loss of -0.25 it he keeps this strategy. On average, the player is winning on the long term.

   (d) The evolution of the cumulative regret is shown in figure 3. Since the player chooses only action 3 after `EWA` converges, the cumulative regret stops increasing because action 3 is the optimal action.

   (e) The average loss obtained, in average, in maximum and in minimum over the n simulations is shown in figure 4. The algorithm seems to be stable, as both the maximum, average and minimum loss converge towards a similar value. As explained in question 3b, this value should be $\mathbb{E}_j[L(3, j)] = -0.25$ because the player only selects action 3 in the end. However, the max and min average loss seem to take a long time to converge, which is an undesired behaviour (we would like to have a fast convergence in all cases).

(f) The regret over time for different values of η, as well as the final regret as a function of η, are shown in figure 5 (figure 6 shows the same values but averaged over 100 experiments to better capture the expected behaviour). One can see that the larger η is, the lower the final regret seems to be, and the best value in practice between the ones that we have tested is $\eta = 1$. In theory, the best value for η in `EWA` is supposed to be: $\eta = \sqrt{\log M/T} = \sqrt{\log 3/100} \approx 0.1$. However, we are here in a particular context, where the adversary has a fixed strategy and does not adapt over time (oblivious adversary).

4. *Simulation against an adaptive adversary.*

(a) The average loss over time when playing against an adaptive adversary (playing `EWA` with $\eta = 0.05$) is shown in figure 7. One can see that the average loss converges towards 0, which seems reasonable since both players are playing a zero-sum game with a similar regret minimizing strategy.

(b) The evolution of $||\bar{p}_t - (1/3, 1/3, 1/3)||_2$ over time in log log scale is shown in figure 8. As expected, $\bar{p}_t$ converges towards the weight vector $(1/3, 1/3, 1/3)$. Indeed, as the subject says, $(\bar{p}_t, \bar{q}_t)$ converges towards a Nash equilibrium. Such an equilibrium cannot be reached if one player uses an unbalanced strategy: if some player plays one action more often than the other ones (for instance ”Rock”), the other player should change its strategy to play the action that beats it (for instance ”Paper”). Hence, the only Nash equilibrium is when both players use a uniform strategy.

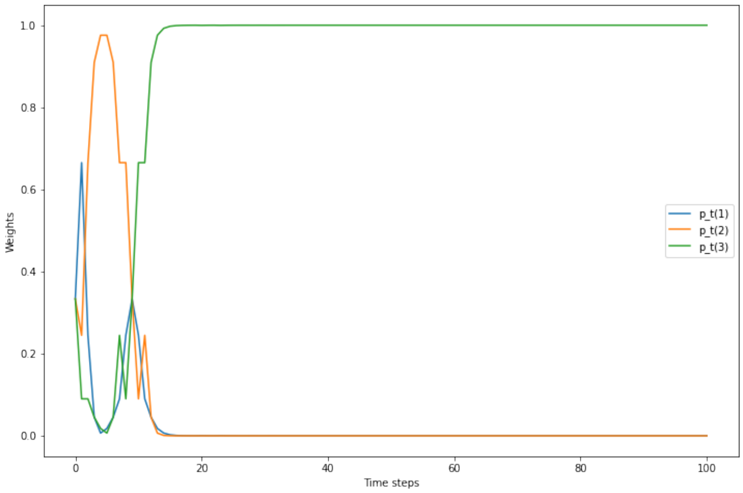

Figure 1: Evolution of the weight vectors (question 3b).

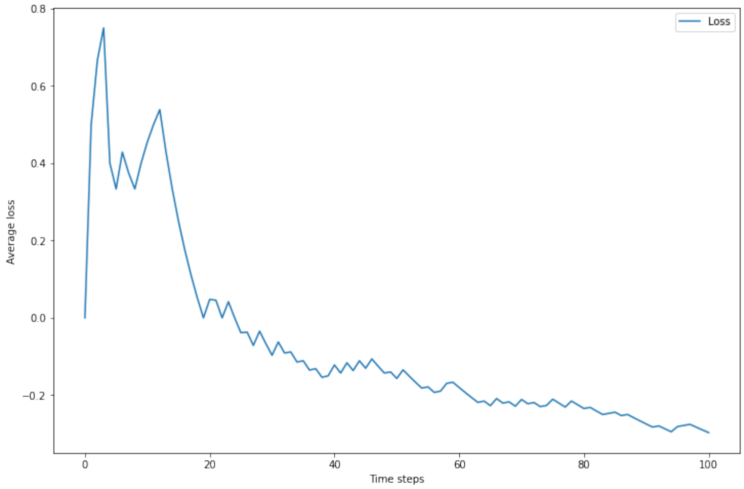

Figure 2: Evolution of the average loss (question 3c).

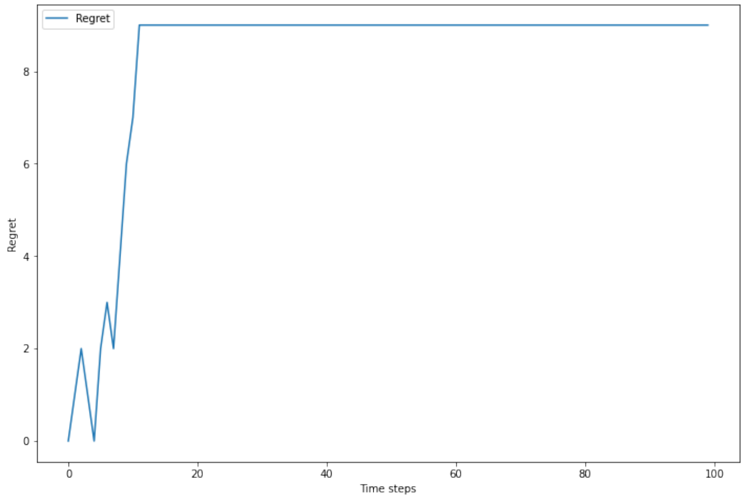

Figure 3: Evolution of the cumulative regret (question 3d).

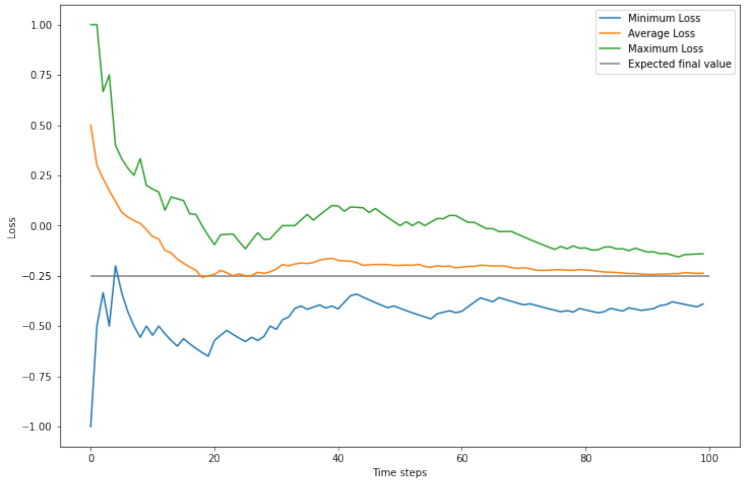

Figure 4: Stability of the algorithm (question 3e).

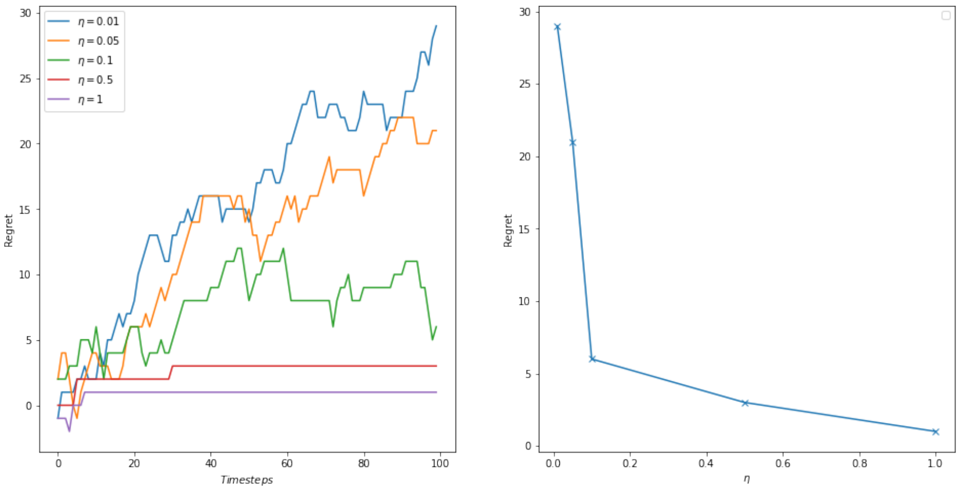

Figure 5: Regret over time, and final regret depending on η (question 3f).

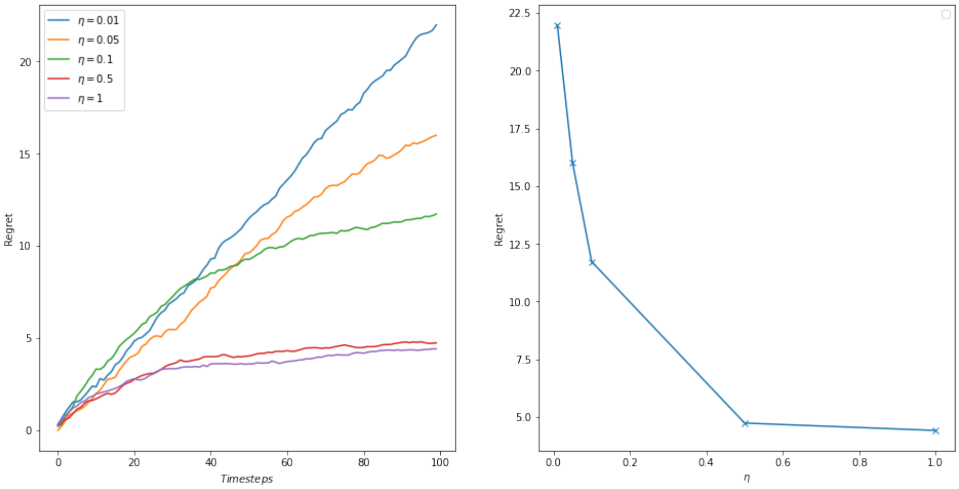

Figure 6: Regret depending on η, averaged over 100 experiments (question 3f).

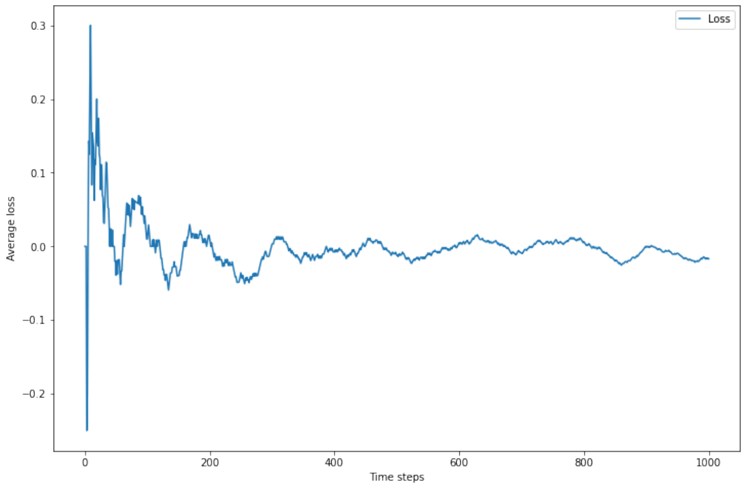

Figure 7: Average loss over time against an adaptive adversary (question 4a).

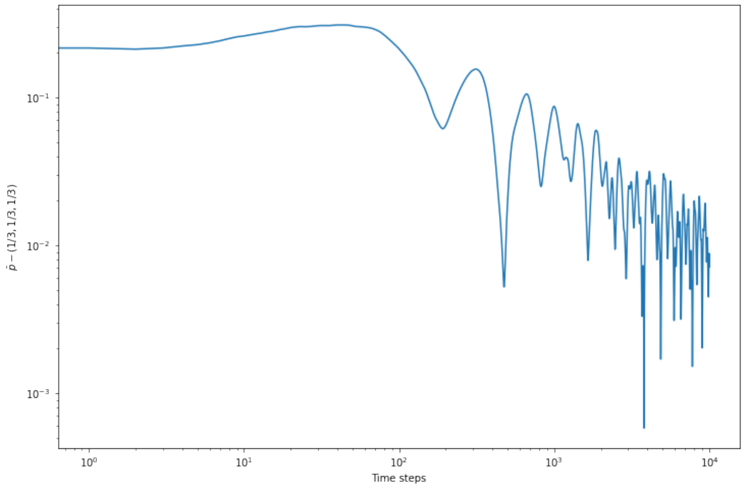

Figure 8: Convergence towards a Nash equilibrium (question 4b).

Bandit feedback

5. *Implementation of* `EXP3`*.*

   (a) (Code)

   (b) (Code)

6.

   (a) Here the answer is the same: if the adversary chooses the action j_t, the loss incurred by the player if he chooses action i at time t is :

   $$l_t(i) = L(i, j_t)$$

   The difference with the previous questions (full information feedback) is that the player does not observe the loss that he would have incurred with the other choices of actions, that is, $l_t(\hat{i}) = L(\hat{i}, j_t)$ for $\hat{i} \neq i$.

   (b) The evolution of the weight vectors is shown in figure 9. One can see that the `EXP3` algorithm can take a far longer time to converge than `EWA`, which is why the experiments were done at least over 1000 time steps. It can also stay stuck in a poor strategy for some time, for instance here it keeps playing the action 1 until step 300. In some cases also, the algorithm does not find the optimal strategy. Still, most of the time, `EXP3` finds the optimal strategy before the end of the 1000 steps.

   (c) The evolution of the average loss is shown in figure 10. Once again, and even if it takes more time than `EWA`, the average loss seems to stay negative once the `EXP3` algorithm has converged towards the optimal strategy. Note that sometimes, when the algorithm does not converge towards the optimal strategy, the average loss stays positive.

   (d) The evolution of the cumulative regret is shown in figure 11. Because the algorithm takes more time to converge, the final regret is larger than when using `EWA`. Also, since sometimes the algorithm does not converge towards the optimal strategy during the 1000 first steps, the regret can keep increasing.

   (e) The average loss obtained, in average, in maximum and in minimum over the n simulations is shown in figure 12. The `EXP3` algorithm seems to be less stable than `EWA`: even if the plot looks similar at first, some differences are crucial. First, recall that the experiments are led over 1000 experiments: that is, even after 10 times more steps, the extreme cases are far from the average behaviour. Also, notice that the maximum loss is still positive after 1000 steps, which indicates that in this extreme case, the algorithm has not converged towards the optimal strategy.

(f) The regret over time for different values of η, as well as the final regret as a function of η, are shown in figure 13 (figure 14 shows the same values but averaged over 100 experiments to better capture the expected behaviour). One can notice once again the unstability of the `EXP3` on the first plot: when performing different simulations, the behaviour can change drastically, and it is difficult to analyse the influence of the parameter η with only one simulation for each η. The second plot is more meaningful since it shows the behaviour of the algorithm in average: once again, the performance seems to increase with η.

7.

(a) The average loss over time when playing against an adaptive adversary (playing `EXP3` with $\eta = 0.05$) is shown in figure 15. The average loss converges towards 0, however, this time the strategy that the algorithm finds is different than what `EWA` did: in fact, both the player and adversary end up repeatedly playing the same action over and over again, leading to endless draws. Indeed, `EXP3` does not observe the losses for the other actions, and only observes the loss of 0 that they receive, whereas `EWA` would see that another action leads to a better reward and change its strategy.

(b) The evolution of $||\bar{p}_t - (1/3, 1/3, 1/3)||_2$ over time in log log scale is shown in figure 16. As said in question 7a, $\bar{p}_t$ does not converge towards the weight vector $(1/3, 1/3, 1/3)$. In particular, $\bar{p}_t$ does not converge towards a Nash equilibrium.

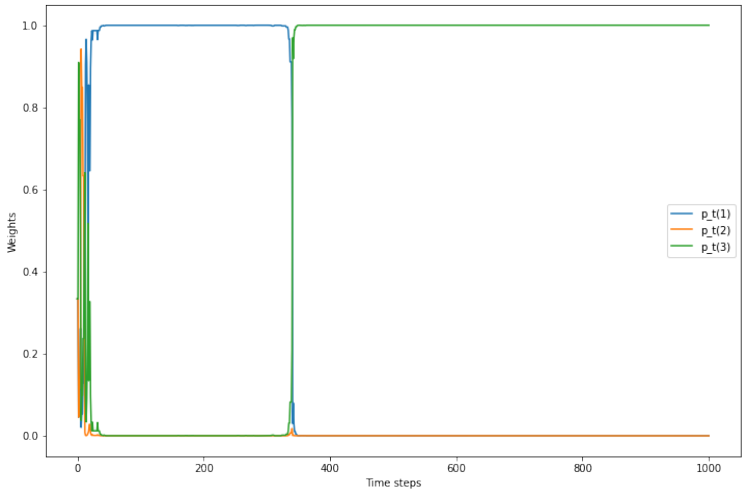

Figure 9: Evolution of the weight vectors with `EXP3` (question 6b).

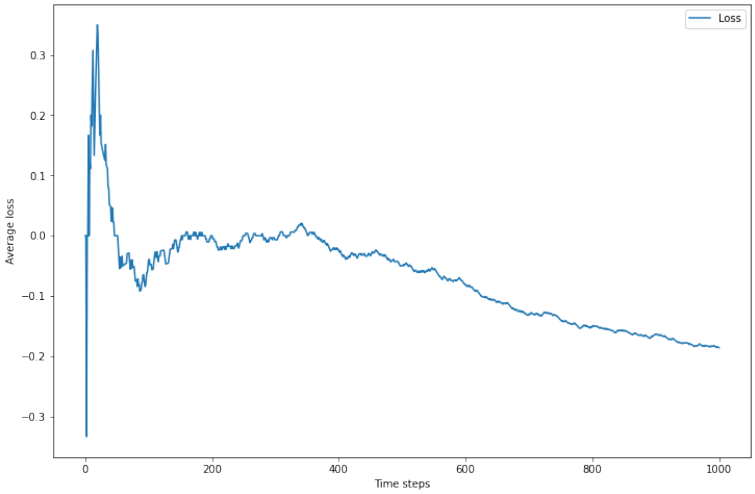

Figure 10: Evolution of the average loss with EXP3 (question 6c).

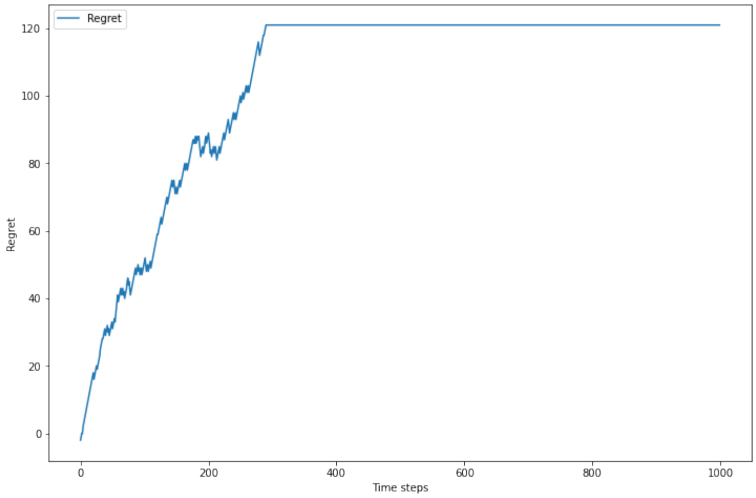

Figure 11: Evolution of the cumulative regret with EXP3 (question 6d).

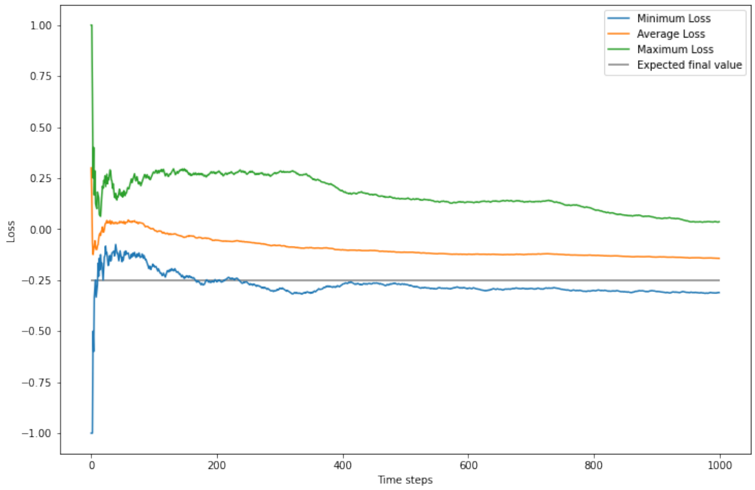

Figure 12: Stability of the algorithm EXP3 (question 6e).

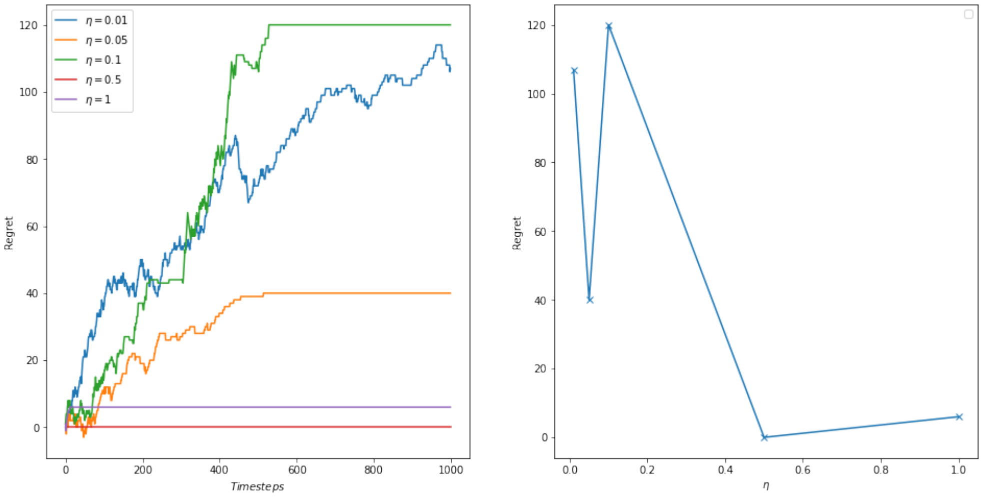

Figure 13: Regret over time, and final regret depending on η with `EXP3` (question 6f).

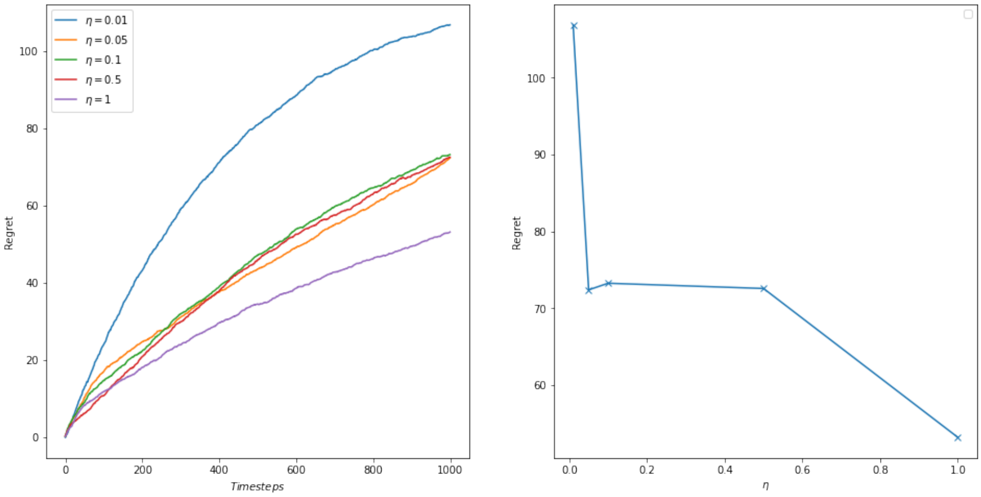

Figure 14: Regret depending on η, averaged over 100 experiments with `EXP3` (question 6f).

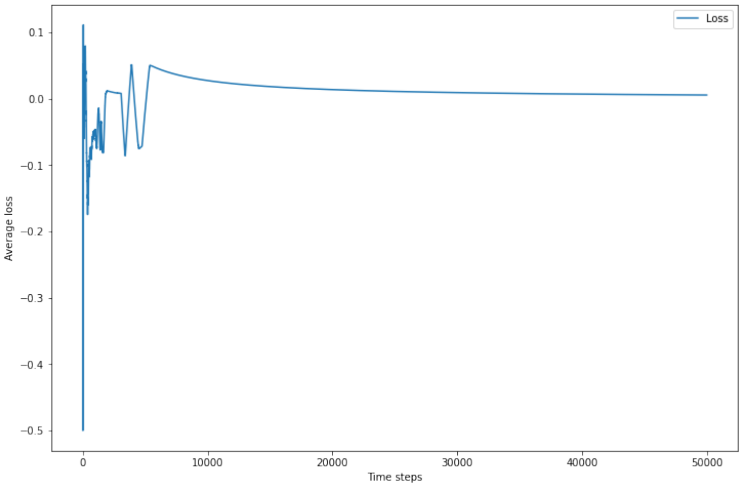

Figure 15: Average loss over time against an adaptive adversary with `EXP3` (question 7a).

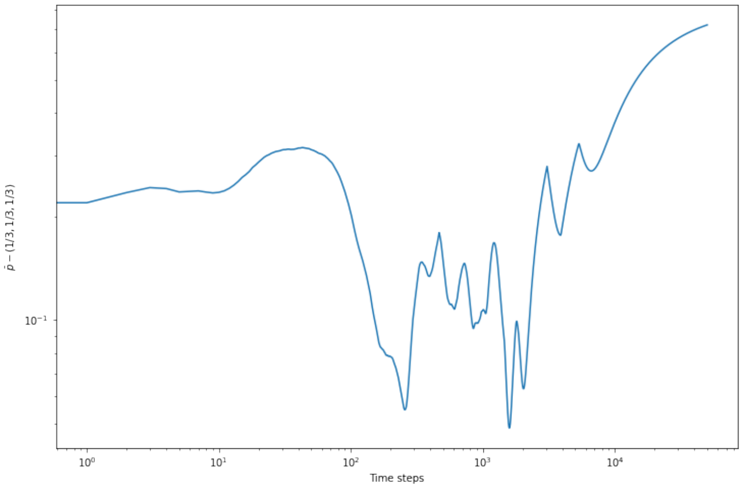

Figure 16: Evolution of $||\bar{p}_t - (1/3, 1/3, 1/3)||_2$ with `EXP3` (question 7b).

Optional extentions

8. The average loss of `EXP3` over time when playing against `UCB` is shown in figure 17. Before running the simulation, one can expect that `EXP3` performs better: indeed, `UCB` is not suited to adversarial contexts and cannot adapt its strategy against an opponent. On the contrary, `EXP3` is designed to perform well in adversarial contexts. As expected, when playing for 500 time steps, `EXP3` wins against `UCB`.

9. After implementing `EXP3-IX` and running 10 simulations, the average loss obtained on average, in maximum and in minimum is shown in figure 18. One can see that is it far more stable than `EXP3`, and that the maximum and minimum loss are far closer to the expected final value with `EXP3-IX`. Surprisingly, we had to choose a quite large value for the parameter γ to observe a real difference, that is, we chose $\gamma = \eta = 1$.

10. The behaviour of `EWA` on the prisoner's dilemma is shown in figure 19 (for the action weights) and 20 (for the average loss over time steps). The behaviour of `EXP3` is shown in figure 21 (for the action weights) and 22 (for the average loss over time steps). The convention is the following: action 1 corresponds to remain silent, and action 2 corresponds to betraying the other player. We only show the results for one player since the behaviour is the same for the second one.

In the first case (`EWA`) where both players see the consequences of their actions and the loss of other actions, both players end up always falling in the Nash equilibrium, that is, they always betray each other and choose action 2. In the second case (`EXP3`), the algorithm is less stable but most of the time, both players choose to betray each other. In both cases, the average loss seems to converge towards 2. This value is sub-optimal, since each player would always incur a loss of 1 if they did not betray each other. This illustrates the main point of the prisonner's dilemma: two completely rational individuals might not cooperate, even if it appears that it is in their best interests to do so.

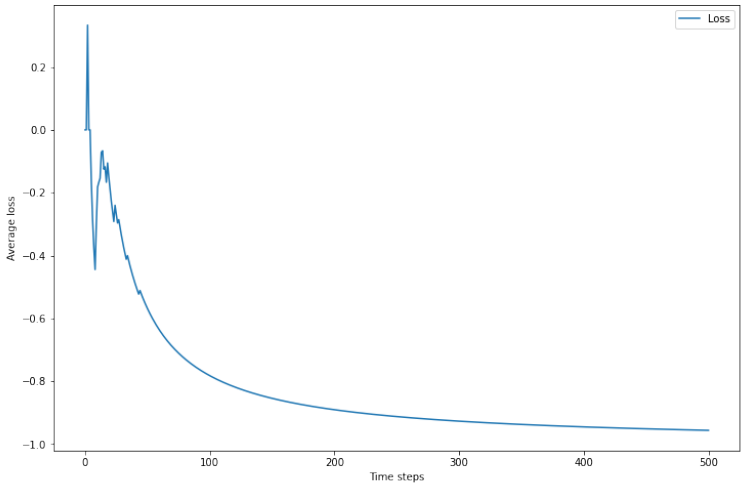

Figure 17: Average loss of `EXP3` over time when playing against `UCB` (question 8).

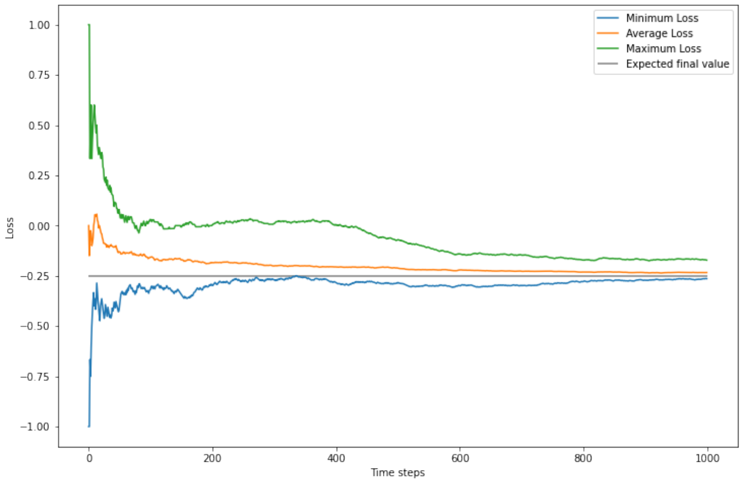

Figure 18: Stability of the algorithm `EXP3-IX` (question 9).

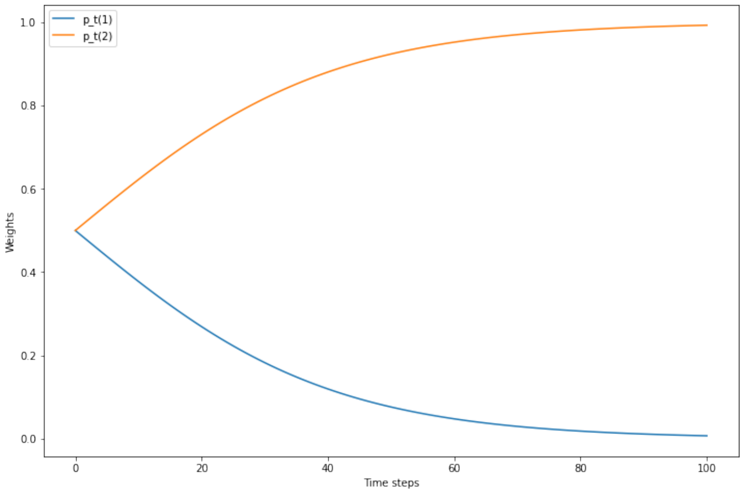

Figure 19: Action weights of `EWA` on the prisonner's dilemma (question 10).

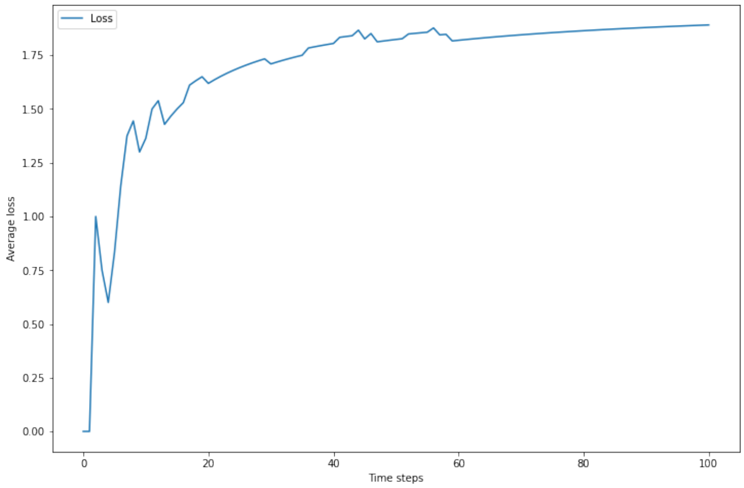

Figure 20: Average loss of `EWA` on the prisonner's dilemma (question 10).

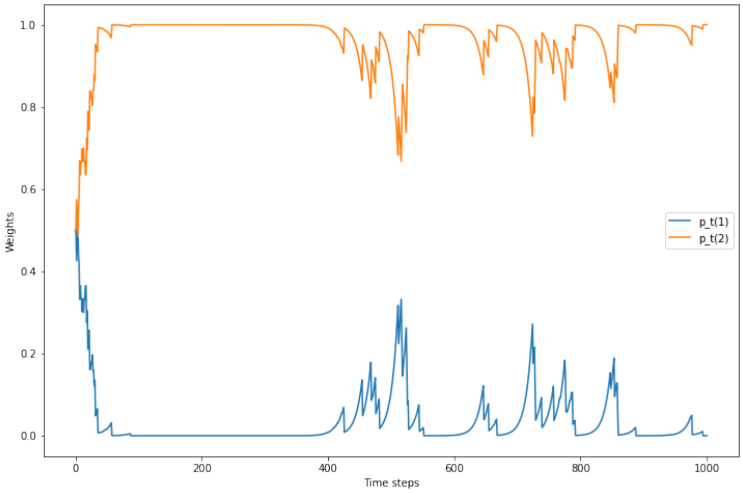

Figure 21: Action weights of `EXP3` on the prisonner's dilemma (question 10).

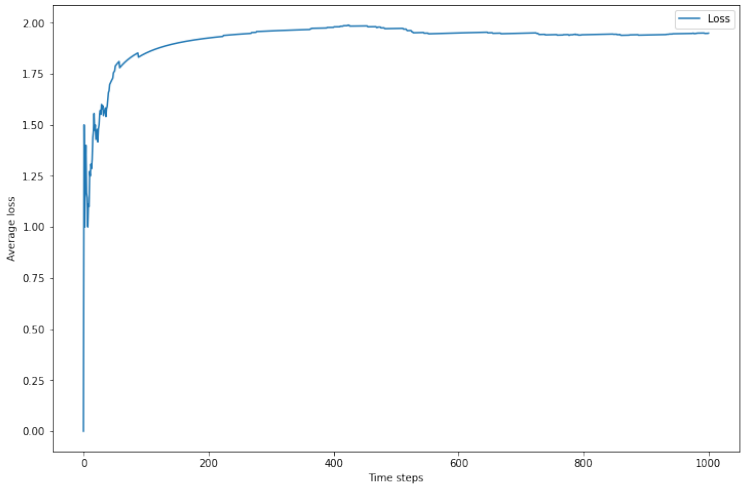

Figure 22: Average loss of `EXP3` on the prisonner's dilemma (question 10).

Part 2. Bernoulli Bandits

A bit of context: why Bernoulli bandits matter

1. *Follow the leader.*

   (a) We consider the case where there exists an arm $\tilde{k} \in [K]$ such that $p_{\tilde{k}} = \min_{k\in[K]} p_k < p^\star$ (otherwise, the pseudo-regret is always equal to 0). We also assume that $T > K$.

   Let Σ be the following event: in the first K steps (where we draw all the arms once) $X_t^{k_t} = \delta_{\tilde{k}}(k_t)$, where $k_t = t$, for $t \in [K]$. One has:

$$\mathbb{P}(\Sigma) = p_{\tilde{k}} \prod_{k\in[K], k\neq\tilde{k}} (1 - p_k) > 0$$

   Moreover, if the event Σ occurs, then all the empirical means are 0 except the one of $\tilde{k}$. Therefore, for all $t > K$, $k_t = \tilde{k}$ and:

$$\begin{aligned} R_T &= Tp^\star - \sum_{t=1}^{K} p_t - \sum_{t=K+1}^{T} p_{\tilde{k}} \\ &\geq Tp^\star - Kp^\star - (T-K)p_{\tilde{k}} \\ &\geq (T-K)(p^\star - p_{\tilde{k}}) \end{aligned}$$

   As a consequence:

$$\begin{aligned} \mathbb{E}R_T &= \mathbb{E}[R_T|\Sigma]\ \mathbb{P}(\Sigma) + \mathbb{E}[R_T|\bar{\Sigma}]\ \mathbb{P}(\bar{\Sigma}) \\ &\geq \mathbb{E}[R_T|\Sigma]\ \mathbb{P}(\Sigma) \\ &\geq \frac{T-K}{T}(p^\star - p_{\tilde{k}})\mathbb{P}(\Sigma)\ T \\ &\geq \frac{1}{K+1}(p^\star - p_{\tilde{k}})\mathbb{P}(\Sigma)\ T \end{aligned}$$

   To conclude with, for $\alpha = \dfrac{1}{K+1}(p^\star - p_{\tilde{k}})\mathbb{P}(\Sigma)$ the expected regret of the Follow-the-leader algorithm (FTL) verifies $\mathbb{E}R_T \geq \alpha T$.

   (b) (Code)

   (c) We can see that the regrets are split into 2 groups: 0 and 10, corresponding to the 2 arms. It means that the algorithm rapidly chooses one arm and keeps playing this one. If this arm is the first one, it is associated with regret 0 and the regret at time 100 is 0. Otherwise, the associated regret is 0.1 so the regret at time 100 is 100 × 0.1 = 10. Moreover, the two groups contain almost the same number of examples, so the algorithm is not good at choosing the best one (we expect a larger group at regret 0).

   (d) We can see that the mean regret is linear with regard to time. It is not optimal, since we can find a $O(\sqrt{T})$ solution. FTL is not a good algorithm for stochastic bandits.

2. *UCB.*

   (a) Let $\lambda \in \mathbb{R}$ and X a Bernoulli random variable with parameter p. One has:

$$\begin{aligned} \phi_X(\lambda) &= \log \mathbb{E}\left[e^{\lambda(X-\mathbb{E}(X))}\right] \\ &= \log \mathbb{E}\left[e^{\lambda X}\right] - \lambda p \\ &= \log\left(e^\lambda p + (1-p)\right) - \lambda p \qquad \text{(law of the unconscious statistician)} \end{aligned}$$

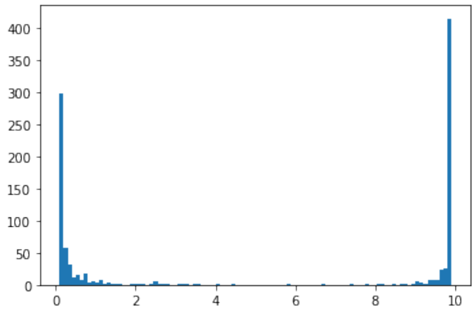

Figure 23: Histogram of the regret R_T of FTL over 1000 repetitions of the experiment, at time $T = 100$ (Question 1.(c))

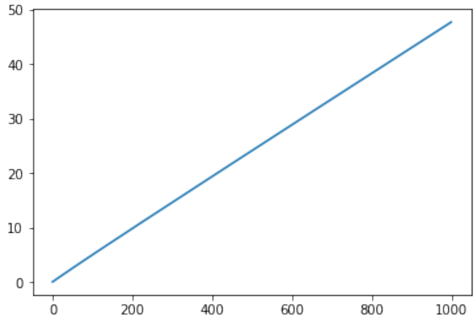

Figure 24: Mean regret of FTL over 1000 repetitions, as a function of $t \in \{1, ..., n\}$ (Question 1.(d))

(b) Let X a random variable such that $\phi''_X \leq \sigma^2$.

Let φ the function $\varphi := \lambda \mapsto \varphi(\lambda) = \phi_X(\lambda) - \frac{1}{2}\sigma^2\lambda^2$. φ is twice differentiable and $\varphi''(\lambda) = \phi''_X(\lambda) - \sigma^2 \leq 0$ for all $\lambda \in \mathbb{R}$, so φ is concave.

Recall that $\phi_X(\lambda) = \log \mathbb{E}\left[e^{\lambda(X-\mathbb{E}(X))}\right]$ so:

$$\begin{aligned} \phi'_X(\lambda) &= \frac{\mathbb{E}\left[(X-\mathbb{E}(X))e^{\lambda(X-\mathbb{E}(X))}\right]}{\mathbb{E}\left[e^{\lambda(X-\mathbb{E}(X))}\right]} \\ \implies \phi'_X(0) &= \frac{\mathbb{E}\left[(X-\mathbb{E}(X))\right]}{1} \\ &= 0 \end{aligned}$$

Therefore, φ admits a minimum in 0 so $\forall \lambda \in \mathbb{R}$, $\varphi(\lambda) \leq \varphi(0) = \log(1) - 0 = 0$.

We conclude that X is σ^2-sub-Gaussian from the fact that:

$$\varphi(\lambda) \leq 0 \implies \log\left(\frac{\mathbb{E}\left[e^{\lambda(X-\mathbb{E}(X))}\right]}{e^{\frac{1}{2}\sigma^2\lambda^2}}\right) \leq 0 \implies \mathbb{E}\left[e^{\lambda(X-\mathbb{E}(X))}\right] \leq e^{\frac{1}{2}\sigma^2\lambda^2}$$

(c) Let $\lambda \in \mathbb{R}$ and X a Bernoulli random variable with parameter p. According to question

2.(a), one has that $\phi_X(\lambda) = \log\big(e^\lambda p + (1-p)\big) - \lambda p$, so:

$$\begin{aligned}
\phi'_X(\lambda) &= -p + \frac{e^\lambda p}{e^\lambda p + (1-p)} \\
\phi''_X(\lambda) &= \frac{e^\lambda p(1-p)}{(e^\lambda p + (1-p))^2} \\
&= \frac{e^\lambda p(1-p)}{(e^\lambda p + (1-p))^2 - (e^\lambda p - (1-p))^2 + (e^\lambda p - (1-p))^2} \\
&= \frac{e^\lambda p(1-p)}{4e^\lambda p(1-p) + (e^\lambda p - (1-p))^2} \\
&\leq \frac{e^\lambda p(1-p)}{4e^\lambda p(1-p)} \\
&\leq \frac{p(1-p)}{4p(1-p)} = (1/2)^2
\end{aligned}$$

Therefore, according to question 2.(b), X is 1/4-sub-Gaussian.

(d) Let $x \in [0,1]$. We denote by g the function $\lambda \mapsto e^{\lambda x} - 1 + x - xe^\lambda$. g is differentiable and $g'(\lambda) = x(e^{\lambda x} - e^\lambda)$, so $g'(\lambda) \geq 0$ if $\lambda \leq 0$ and $g'(\lambda) \leq 0$ otherwise, because $x \in [0,1]$. Therefore, one has $\forall \lambda \in \mathbb{R}$, $g(\lambda) \leq g(0) = 0$. Hence:

$$\forall x \in [0,1], \quad \forall \lambda \in \mathbb{R}, \quad e^{\lambda x} \leq 1 - x + xe^\lambda$$

Now, let X a random variable supported on $[0,1]$ with mean p, Y a random variable with distribution $\mathcal{B}(p)$ and $\lambda \in \mathbb{R}$.

$$\begin{aligned}
\phi_X(\lambda) &= \log \mathbb{E}\left[e^{\lambda X}\right] - \lambda p \\
&\leq \log \mathbb{E}\left[1 - X + Xe^\lambda\right] - \lambda p \qquad \text{(according to the previous calculation)} \\
&\leq \log\Big(1 + (e^\lambda - 1)E\left[X\right]\Big) - \lambda p \\
&\leq \log\Big(1 + (e^\lambda - 1)p\Big) - \lambda p = \phi_Y(\lambda) \qquad \text{(according to (a))}
\end{aligned}$$

So $\forall \lambda \in \mathbb{R}$, $\phi_X(\lambda) \leq \phi_Y(\lambda)$.

(e) Let X a random variable supported on $[0,1]$ and denote p its mean.

The random variable Y with distribution $\mathcal{B}(p)$ is σ^2-sub-Gaussian according to (c) (with $\sigma = 1/2$), so $\forall \lambda \in \mathbb{R}$, $\mathbb{E}\left[e^{\lambda(Y - \mathbb{E}(Y))}\right] \leq e^{\frac{1}{2}\sigma^2\lambda^2} \implies \phi_Y(\lambda) \leq \frac{1}{2}\sigma^2\lambda^2$

Moreover, $\forall \lambda \in \mathbb{R}$, $\phi_X(\lambda) \leq \phi_Y(\lambda)$. Therefore $\phi_X(\lambda) \leq \frac{1}{2}\sigma^2\lambda^2$ which implies that:

$$\mathbb{E}\left[e^{\lambda(X - \mathbb{E}(X))}\right] \leq e^{\frac{1}{2}\sigma^2\lambda^2}$$

This proves that X is sub-Gaussian, and we can conclude that all random variables supported on $[0,1]$ are sub-Gaussian.

(f) (Code)

(g) As we can see, compared to 1.(d) the algorithm achieved a better mean regret, as it is no longer linear but in what seems to be $\log(T)$. It is expected since UCB is a better algorithm for stochastic bandits.

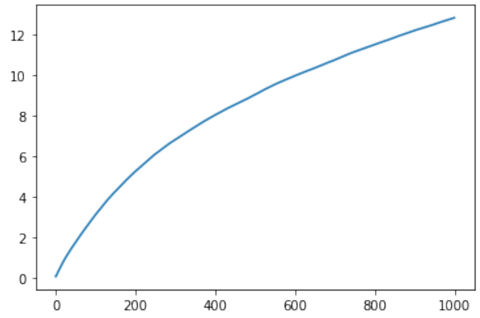

Figure 25: Mean regret of UCB over 1000 repetitions, as a function of $t \in \{1, ..., n\}$ (Question 2.(g))

(h) When plotting the mean regret at $T = 1000$ of UCB(σ^2) for different values of σ, we see that the optimal parameter changes according to p. For $p = (0.5, 0.6)$ the optimal σ^2 is around $0.25 \sim 1/4$ (theoretical parameter obtained before), while it is around 0.1 for $p = (0.95, 0.85)$. Therefore, the more we go away from $p \sim 0.5$, the less the value $1/4$ is optimal (see Question 3.).

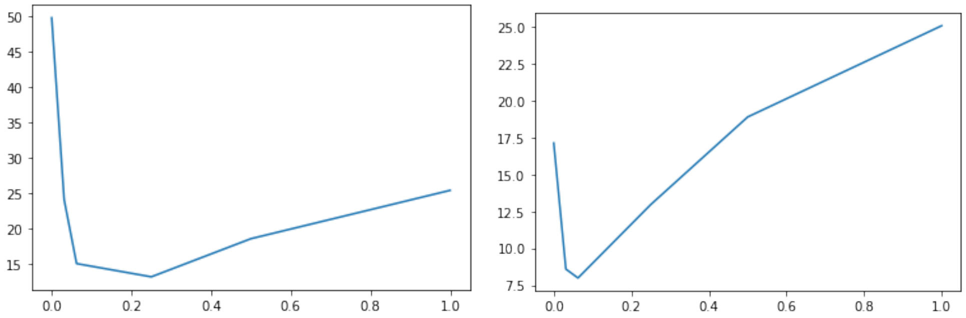

Figure 26: Mean regret of UCB over 1000 repetitions as a function of σ^2 at time $T = 1000$ for $p = (0.6, 0.5)$ (left) $p = (0.95, 0.85)$ (right) (Question 2.(h))

3. We can see that the variance of $\mathcal{B}(p)$ is bounded by the sub-gaussian constant. Moreover, we can observe, as it has been mentioned in 2.(h), that for p away from 0.5, the value $1/4$ is way less optimal than the new $\sigma^2(p)$.

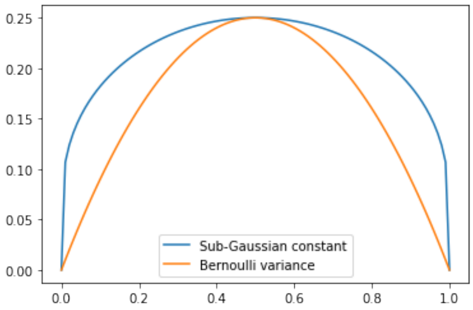

Figure 27: Variance of $\mathcal{B}(p)$ and sub-Gaussian constant $\sigma^2(p)$ as a function of $p \in [0, 1]$ (Question 3.)

4. Let X a σ^2-sub-Gaussian random variable. One has using a Taylor expansion for $\lambda \to 0$ and

the fact that $\mathbb{E}\left[e^{\lambda(X-\mathbb{E}(X))}\right] \leq e^{\frac{1}{2}\sigma^2\lambda^2}$:

$$\begin{aligned}
&\mathbb{E}\left[1+\lambda(X-\mathbb{E}[X])+\frac{\lambda^2}{2}(X-\mathbb{E}[X])^2+o(\lambda^2)\right] \leq 1+\frac{1}{2}\sigma^2\lambda^2+o(\lambda^2) \\
&\implies 1+\lambda(\mathbb{E}[X]-\mathbb{E}[X])+\frac{\lambda^2}{2}\mathbb{E}\left[(X-\mathbb{E}[X])^2\right]+o(\lambda^2) \leq 1+\frac{1}{2}\sigma^2\lambda^2+o(\lambda^2) \\
&\implies 1+\frac{\lambda^2}{2}\mathrm{Var}(X) \leq 1+\frac{1}{2}\sigma^2\lambda^2+o(\lambda^2) \\
&\implies \mathrm{Var}(X) \leq \sigma^2+o_{\lambda\to 0}(1)
\end{aligned}$$

The above equation gives for $\lambda \to 0$ that $\mathrm{Var}(X) \leq \sigma^2$, so a σ^2-sub-Gaussian random variable has variance bounded by σ^2.

Adaptation to the variance

5. *UCB-V.*

(a) Using the definition of $\hat{v}_t^k = \frac{1}{N_t^k}\sum_{s=1}^{t-1}\mathbb{I}\{k_s=k\}(X_s^{k_s}-\hat{\mu}_t^k)^2$, one has:

$$\begin{aligned}
N_t^k\hat{v}_t^k &= \sum_{s=1}^{t-1}\mathbb{I}\{k_s=k\}(X_s^{k_s}-\hat{\mu}_t^k)^2 \\
&= \sum_{s=1}^{t-1}\mathbb{I}\{k_s=k\}\left(X_s^{k_s}-\frac{1}{N_t^k}\sum_{s'=1}^{t-1}I\{k_{s'}=k\}X_{s'}^{k_{s'}}\right)^2 \\
&= \sum_{s=1}^{t-1}\mathbb{I}\{k_s=k\}(X_s^{k_s})^2+\frac{1}{(N_t^k)^2}\sum_{s=1}^{t-1}\mathbb{I}\{k_s=k\}\left(\sum_{s'=1}^{t-1}I\{k_{s'}=k\}X_{s'}^{k_{s'}}\right)^2 \\
&\qquad -\frac{2}{N_t^k}\sum_{s=1}^{t-1}\mathbb{I}\{k_s=k\}\left(X_s^{k_s}\sum_{s'=1}^{t-1}I\{k_{s'}=k\}X_{s'}^{k_{s'}}\right) \\
&= \sum_{s=1}^{t-1}\mathbb{I}\{k_s=k\}(X_s^{k_s})^2+\left(\sum_{s'=1}^{t-1}I\{k_{s'}=k\}X_{s'}^{k_{s'}}\right)^2\frac{1}{(N_t^k)^2}\sum_{s=1}^{t-1}\mathbb{I}\{k_s=k\} \\
&\qquad -\frac{2}{N_t^k}\sum_{s'=1}^{t-1}I\{k_{s'}=k\}X_{s'}^{k_{s'}}\sum_{s=1}^{t-1}\mathbb{I}\{k_s=k\}X_s^{k_s} \\
&= \sum_{s=1}^{t-1}\mathbb{I}\{k_s=k\}(X_s^{k_s})^2+\left(\sum_{s'=1}^{t-1}I\{k_{s'}=k\}X_{s'}^{k_{s'}}\right)^2\frac{1}{(N_t^k)^2}N_t^k \\
&\qquad -\frac{2}{N_t^k}\left(\sum_{s'=1}^{t-1}I\{k_{s'}=k\}X_{s'}^{k_{s'}}\right)^2 \\
&= \sum_{s=1}^{t-1}\mathbb{I}\{k_s=k\}(X_s^{k_s})^2-\frac{1}{N_t^k}\left(\sum_{s=1}^{t-1}I\{k_s=k\}X_s^{k_s}\right)^2
\end{aligned}$$

(b) Using previous question, we can write:

$$\begin{aligned}
N_{t+1}^{k_t}\hat{v}_{t+1}^{k_t}-N_t^{k_t}\hat{v}_t^{k_t} &= \sum_{s=1}^{t}\mathbb{I}\{k_s=k_t\}(X_s^{k_s})^2-\frac{1}{N_{t+1}^{k_t}}\left(\sum_{s=1}^{t}I\{k_s=k_t\}X_s^{k_s}\right)^2 \\
&\qquad -\sum_{s=1}^{t-1}\mathbb{I}\{k_s=k_t\}(X_s^{k_s})^2+\frac{1}{N_t^{k_t}}\left(\sum_{s=1}^{t-1}I\{k_s=k_t\}X_s^{k_s}\right)^2
\end{aligned}$$

Then, $\sum_{s=1}^{t} I\{k_s = k_t\}X_s^{k_s} = X_t^{k_t} + \sum_{s=1}^{t-1} I\{k_s = k_t\}X_s^{k_s}$, so $\hat{\mu}_{t+1}^{k_t} = \frac{N_t^{k_t}}{N_{t+1}^{k_t}}\hat{\mu}_t^{k_t} + \frac{1}{N_{t+1}^{k_t}}X_t^{k_t}$. Moreover, $N_{t+1}^{k_t} = 1 + N_t^{k_t}$. Using all of this:

$$\begin{aligned}
N_{t+1}^{k_t}\hat{v}_{t+1}^{k_t} - N_t^{k_t}\hat{v}_t^{k_t} &= (X_t^{k_t})^2 - \frac{1}{N_{t+1}^{k_t}}\left(\sum_{s=1}^{t} I\{k_s = k_t\}X_s^{k_s}\right)^2 + \frac{1}{N_t^{k_t}}\left(\sum_{s=1}^{t-1} I\{k_s = k_t\}X_s^{k_s}\right)^2 \\
&= (X_t^{k_t})^2 - \frac{1}{N_{t+1}^{k_t}}\left(N_t^{k_t}\hat{\mu}_t^{k_t} + X_t^{k_t}\right)^2 + \frac{1}{N_t^{k_t}}\left(N_t^{k_t}\hat{\mu}_t^{k_t}\right)^2 \\
&= (X_t^{k_t})^2 - \frac{(N_t^{k_t})^2}{N_{t+1}^{k_t}}(\hat{\mu}_t^{k_t})^2 - \frac{1}{N_{t+1}^{k_t}}(X_t^{k_t})^2 - 2\frac{N_t^{k_t}}{N_{t+1}^{k_t}}X_t^{k_t}\hat{\mu}_t^{k_t} + N_t^{k_t}(\hat{\mu}_t^{k_t})^2 \\
&= (X_t^{k_t})^2 - \frac{1}{N_{t+1}^{k_t}}(X_t^{k_t})^2 + \left(N_t^{k_t} - \frac{(N_t^{k_t})^2}{N_{t+1}^{k_t}}\right)(\hat{\mu}_t^{k_t})^2 - 2\frac{N_t^{k_t}}{N_{t+1}^{k_t}}X_t^{k_t}\hat{\mu}_t^{k_t} \\
&= (X_t^{k_t})^2 - \frac{1}{N_{t+1}^{k_t}}(X_t^{k_t})^2 + \frac{N_t^{k_t}(N_t^{k_t}+1) - (N_t^{k_t})^2}{N_{t+1}^{k_t}}(\hat{\mu}_t^{k_t})^2 - 2\frac{N_t^{k_t}}{N_{t+1}^{k_t}}X_t^{k_t}\hat{\mu}_t^{k_t} \\
&= (X_t^{k_t})^2 - \frac{1}{N_{t+1}^{k_t}}(X_t^{k_t})^2 + \frac{N_t^{k_t}}{N_{t+1}^{k_t}}(\hat{\mu}_t^{k_t})^2 + \frac{1 - N_t^{k_t} - N_{t+1}^{k_t}}{N_{t+1}^{k_t}}X_t^{k_t}\hat{\mu}_t^{k_t} \\
&= (X_t^{k_t})^2 - \frac{1}{N_{t+1}^{k_t}}(X_t^{k_t})^2 + \frac{N_t^{k_t}}{N_{t+1}^{k_t}}(\hat{\mu}_t^{k_t})^2 + \left(\frac{1}{N_{t+1}^{k_t}} - \frac{N_t^{k_t}}{N_{t+1}^{k_t}} - 1\right)X_t^{k_t}\hat{\mu}_t^{k_t} \\
&= (X_t^{k_t})^2 - \frac{N_t^{k_t}}{N_{t+1}^{k_t}}X_t^{k_t}\hat{\mu}_t^{k_t} - \frac{1}{N_{t+1}^{k_t}}(X_t^{k_t})^2 - X_t^{k_t}\hat{\mu}_t^{k_t} + \frac{N_t^{k_t}}{N_{t+1}^{k_t}}(\hat{\mu}_t^{k_t})^2 + \frac{1}{N_{t+1}^{k_t}}X_t^{k_t}\hat{\mu}_t^{k_t} \\
&= (X_t^{k_t})^2 - X_t^{k_t}\left(\frac{N_t^{k_t}}{N_{t+1}^{k_t}}\hat{\mu}_t^{k_t} + \frac{1}{N_{t+1}^{k_t}}X_t^{k_t} + \hat{\mu}_t^{k_t}\right) + \hat{\mu}_t^{k_t}\left(\frac{N_t^{k_t}}{N_{t+1}^{k_t}}\hat{\mu}_t^{k_t} + \frac{1}{N_{t+1}^{k_t}}X_t^{k_t}\right) \\
&= (X_t^{k_t})^2 - X_t^{k_t}(\hat{\mu}_{t+1}^{k_t} + \hat{\mu}_t^{k_t}) + \hat{\mu}_t^{k_t}\hat{\mu}_{t+1}^{k_t} \\
&= \left(X_t^{k_t} - \hat{\mu}_t^{k_t}\right)\left(X_t^{k_t} - \hat{\mu}_{t+1}^{k_t}\right)
\end{aligned}$$

Eventually, $N_{t+1}^{k_t}\hat{v}_{t+1}^{k_t} - N_t^{k_t}\hat{v}_t^{k_t} = \left(X_t^{k_t} - \hat{\mu}_t^{k_t}\right)\left(X_t^{k_t} - \hat{\mu}_{t+1}^{k_t}\right)$. The practical advantage of this formulation is that it allows to update the variance at time $t+1$ without having to actually compute a sum of t terms, since it uses the means that have to be computed either way.

(c) (Code)

(d) We observe that for this setup UCB is better than UCB-V.

(e) As p gets far away from 0.5 and $\sigma^2 = 1/4$ gets less optimal, UCB-V becomes better than UCB(1/4). The extreme case of $p = (0.0, 0.1)$ shows for instance than UCB-V gives better results when the actual variance of the Bernoulli arm is close to 0.

Algorithms for parametric distributions

6. Here we observe that KL-UCB is better than UCB(1/4) and UCB-V in all the cases, it gives results really close to UCB-V when p gets close to 0, and results close to UCB when p gets close to 0.5.

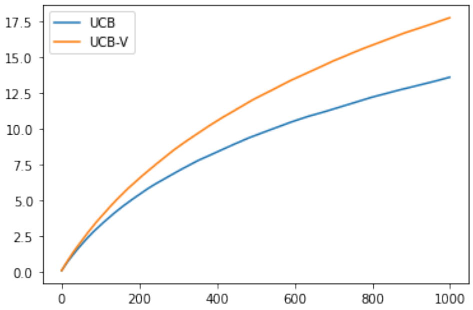

Figure 28: Mean regret of UCB Vs UCB-V over 1000 repetitions as a function of time t up to $T = 1000$ for $p = (0.6, 0.5)$ (Question 5.(d))

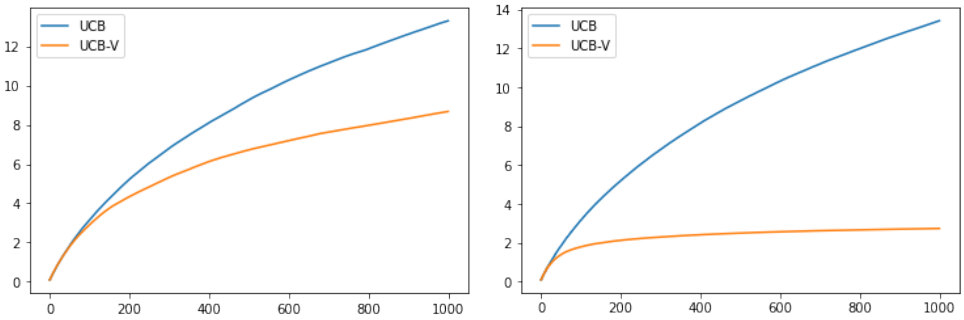

Figure 29: Mean regret of UCB Vs UCB-V over 1000 repetitions as a function of time t up to $T = 1000$ for $p = (0.1, 0.2)$ (left) and $p = (0.0, 0.1)$ (right) (Question 5.(e))

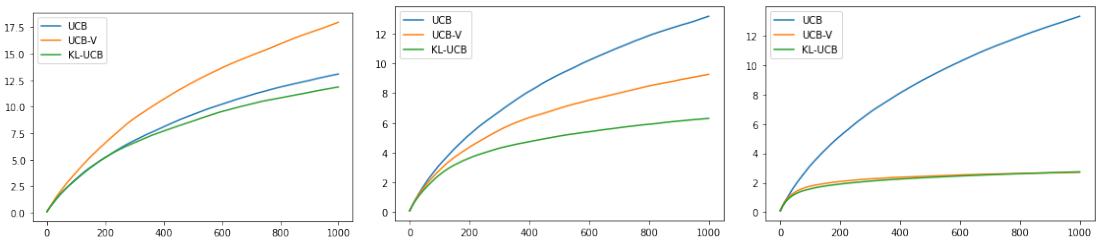

Figure 30: Mean regret of UCB Vs UCB-V Vs KL-UCB over 1000 repetitions as a function of time t up to $T = 1000$ for $p = (0.5, 0.6)$ (left), $p = (0.1, 0.2)$ (middle) and $p = (0.0, 0.1)$ (right) (Question 6.)