EC: Reinforcement Learning **TP3**

Bandit algorithm for Game Theory and RL

Lecturer: *Emilie Kaufmann* *emilie.kaufmann@inria.fr*

Deadline: December 20th, midnight

1 Adversarial MABs and Nash Equilibria

In a two players zero-sum game, players A and B choose simultaneously two actions $a \in \{1...N_A\}$ and $b \in \{1..N_B\}$. The reward of player A associated to the chosen pair of action (a, b) is denoted by $R_A(a, b)$. Then one has $R_B(a, b) = -R_A(a, b)$. Such a game is represented by a matrix of gains

$$G = (R_A(i, j))_{\substack{1 \leq i \leq N_A \\ 1 \leq j \leq N_B}}$$

The game described by the matrix below can model for example two persons choosing between two possible activities. The first person (player A) prefers consensus (with a preference for activity 1) whereas the other (player B) prefers being on his own (with a slight preference for activity 2).

G =

$A \backslash B$	1	2
1	2	-1
2	0	1

Consider the following game: at each time t, player A chooses an action x_A (based on the past observed rewards), player B also plays (simultaneously, without revealing his action x_B to A), player A receives the reward $R_A(x_A, x_B)$, and player B the reward $R_B(x_A, x_B) = -R_A(x_A, x_B)$. Each of the player is playing a bandit game against an adversary.

1. Assuming that both players are using an EXP3 strategy to select their action. Write a function

   `[ActionsA,ActionsB,Rew]= EXP3vEXP3(n,eta,beta,G)`

   that returns the sequences of actions chosen by each of the two players, and the rewards received by player A.

2. Illustrate that the quantities $p_{a,n} = \frac{1}{n} \sum_{t=1}^{n} 1_{(A_t=1)}$ and $p_{b,n}$ (similarly defined) almost surely converge to some values p_a^* and p_b^*.

3. A pair (p, q) is a Nash equilibrium if when player A uses the mixed strategy defined by p (that is he plays action A with probability p, action B otherwise) and player B uses the mixed strategy defined by q, none of the players has incentive to change its strategy if the other does not. Check that (p_a^*, p_b^*) is a Nash equilibrium.

4. Illustrate that the sequence $\frac{1}{n} \sum_{t=1}^{n} R_A(A_t, B_t)$ converges towards the value of the game.

Question 1: Illustrate the convergence towards the Nash equilibrium and the value of the game, for specified values of the parameters for EXP3.

2 MABs to solve a RL problem

A soda vending machine offers two kinds of soda: energy drink (E) or sugar free drink (N). E and N are sold at the same price of 1€ and at the beginning the machine contains the same amount of sodas (50 cans for each type). On a normal day, E has a slightly higher probability to be chosen, while on exam days E has a much higher probability to be chosen. Whenever one of the sodas is finished, the machine gets restock. Since refilling the machine is an expensive operation, the objective is to maximize the amount of money gained between two restocks. In order to achieve this objective, the machine can automatically decide to discount one of the two sodas in order to incentive the consumption of a soda.

A Markov Decision Process. Making the (realistic) assumption that the machine does not have knowledge of the type of day (exam, no exam), which is only a hidden state that governs the transition, the problem can be modeled as an MDP with:

- State space $\mathcal{X} = \{0, 50\}^2$: indicates the remaining quantity of each drink
- Action space $\mathcal{A} = [-1, 1]$: gives the level of discount that is applied. $a = 0$ means no discount, positive (resp. negative) values means that E (resp. N) is discounted by $|a|$.

The reward r_t is the price of the drink chosen by the user when it is presented with the discount choosen at round t, a_t.

Implementation of the Environment. The environment together with the preferences model and the evolution between normal and exam days is already provided in simulator.m.

Discount strategy. Given a discount strategy, the performance is evaluated as the mean reward collected before restock:

$$\mathbb{E}\left[\sum_{t=1}^{\tau} r_t\right],$$

where τ is the occurrence of the first restock. See test_simulator.m for an example. The expectation may be estimated based on the simulation of R restocks.

3 Using MABs to solve a RL problem

Assume the machine can only display three different values of price between two restocks. A natural discount policy is thus parameterized by three values of discount v_1, v_2, v_3 and three thresholds t_1, t_2, t_3 that belong to $\{0, \dots, 49\}$ and satisfy $t_1 \leq t_2 \leq t_3$. It is defined by

$$\begin{array}{rcll} \pi_{T,V}(n_E, n_N) & = & v_1 \mathbb{1}_{(|n_E - n_N| \in]t_1, t_2])} + v_2 \mathbb{1}_{(|n_E - n_N| \in]t_2, t_3])} + v_3 \mathbb{1}_{(|n_E - n_N| > t_3)} & \text{if } n_E > n_N \\ \pi_{T,V}(n_E, n_N) & = & -v_1 \mathbb{1}_{(|n_E - n_N| \in]t_1, t_2])} - v_2 \mathbb{1}_{(|n_E - n_N| \in]t_2, t_3])} - v_3 \mathbb{1}_{(|n_E - n_N| > t_3)} & \text{else} \end{array}$$

The goal of this part is to use a multi-armed bandit modeling and associated algorithms to progressively discover the best policy among a subset of policies of this type.

1. Write a function that returns the chosen discount (=action) under the policy parameterized by $T = (t_1, t_2, t_3)$ and $V = (v_1, v_2, v_3)$, when the current stock is (n_E, n_N):

$$[discount] = soda\_strategy\_param(n_E, n_N, T, V)$$

2. For a specific choice of $T = (t_1, t_2, t_3)$ and $V = (v_1, v_2, v_3)$, compare the performance of π_{t_1,t_2,t_3} to the two baselines soda_strategy_discount.m and soda_strategy_nodiscount.m that are given.

3. We consider a MAB model in which each set of parameters $V = (v_1, v_2, v_3), T = (t_1, t_2, t_3)$ corresponds to an arm, and drawing an arm (i.e., selecting a value for the parameters and running the corresponding discount policy until restock) yields a sample that is the result of applying policy $\pi_{V,T}$ until restock (i.e. the sum of rewards obtained over the round). Write a function

$$[reward] = DrawArm(T, V, s)$$

that produces a reward drawn from the arm parameterized by T, V (s is the simulator).

4. Fix a reasonable number of arms, and implement a bandit strategy such that each round $r = 1, \ldots, R$ a different value of the parameters for the parameterized policy is chosen and measure how the performance changes over rounds.

Question 2: Implement **TWO** out of these possible algorithms: UCB, KL-UCB, Thompson Sampling, EXP3. Notice that the normal implementation of bandit algorithms assumes that the samples are in $[0, 1]$. So, it is probably better to "normalize" the rewards. Illustrate the performance of the bandit algorithm compared to the previous baselines and the best policy in your class.