Advanced Topics in Artificial Intelligence: EE6180
Indian Institute of Technology Madras November 13, 2019
Instructor: Abhishek Sinha

Problem Set 5

- **Optional PSET**. You are *not* required to turn this in.
- Each problem carries 10 points.
- You may work on the problems in groups of size at most **two**. However, **each student must write their own solution**. If you collaborate on the problems, clearly mention the name of your collaborator.

1. ⌨**(Getting Started with Multi Armed Bandits)** This problem is designed to give you a step-by-step hands-on experience of working with Multi Armed Bandit (MAB) algorithms by understanding, modifying, and experimenting with an existing MAB code written in Python[1].

   (a) Download the Github repository:
   `https://github.com/johnmyleswhite/BanditsBook`
   The code is located in a directory named ∼`/BanditsBook/`.

   (b) Change your current directory to `/Banditsbook/`. Read the `README.md` file carefully and familiarize yourself with the structure of the codebase. This repository implements the following six standard bandit algorithms - ϵ`-Greedy, Softmax, UCB1, UCB2, Hedge,` and `Exp3`, which we have studied in the class.

   (c) Change your current directory to `/python/algorithms/` and check out the source code of each of the above algorithms. The codes differ in how the functions `select_arm()` and `update()` are implemented for each of the above algorithms. Make sure you fully understand the working of these two functions for each of the above algorithms.

   (d) The code implements three different models of bandits - `adversarial, Bernoulli,` and `Normal`. Check out the relevant codes at `/python/arms/`.

   (e) In this problem, we will compare the performance of ϵ`-Greedy` (for $\epsilon = 0.05$), `UCB1,` and `Exp3` (with random exploration probability $\gamma = 0.05$) policies for four Bernoulli bandits for a horizon of length $T = 10^4$ and averaging the result over $N = 100$ simulations. Set the expected reward values of the bandits to be $\boldsymbol{p} = [0.5, 0.95, 0.2, 0.8]$.

   (f) Modify the parameters in the file `/python/demo.py` to set up the required simulation environment.

[1] Refer to `https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-01sc-introduction-to-electrical-engineering-and-computer-science-i-spring-2011/python-tutorial/` for a quick Python tutorial.

(g) By suitably augmenting and modifying the function `test_algorithm()` (defined at `/python/testing_framework/tests.py`), investigate the following:

- For a bandit algorithm π, let $N_a^\pi(t)$ denote the average fraction of times (averaged over N runs) the arm a was selected by the algorithm π by the time t. Plot $N_a^\pi(t), a \in [0, 1, 2, 3]$ as a function of $t \in [0, T]$ for each of the above three algorithms. What do you observe from the nature of the plots? Can you guess what happens when $T \to \infty$?
- For a bandit algorithm π, let $R^\pi(t)$ denote the *pseudo-regret* of the algorithm π up to time t. In other words, if $\bar{r}^\pi(t)$ denotes the average-reward (over N runs) obtained by the algorithm π at time t, then the pseudo-regret is defined as $R^\pi(t) = t \max_i p_i - \sum_{\tau=1}^{t} \bar{r}^\pi(\tau)$. Plot the time-evolution of $R^\pi(t)$ for the above three algorithms in the same graph. What do you observe from the plots for different range of values of t? How sensitive is the plot with respect to the parameters ϵ and γ?

2. **(Bandits with Prediction)** Recall that in "bandits with predictions", after T rounds the algorithm outputs a prediction: a guess y_T for the best arm. We focus on the instantaneous regret $\Delta(y_T)$ for the prediction.

(a) Take any bandit algorithm with an instance-dependent regret bound $\mathbb{E}(R(T)) \leq f(T)$, and construct an algorithm for "bandits with predictions" such that $\mathbb{E}(\Delta(y_T)) \leq f(T)/T$.

(b) Consider Successive Elimination with $y_T = a_T$. Prove that this algorithm can achieve

$$\mathbb{E}(\Delta(y_T)) \leq T^{-\gamma}, \quad \text{if } \ T > T_{\mu,\gamma},$$

where $T_{\mu,\gamma}$ depends only on the mean rewards $\mu(a) : a \in \mathcal{A}$ and the γ. This holds for an arbitrarily large constant γ, with only a multiplicative factor increase in regret.

3. **(Foresight and Hindsight Regret for the IID cost model)** Consider the adversarial bandit setting as discussed in the class with full feedback and i.i.d. costs from the interval $c_t(a) \in [0, 1], \forall t, a$.

(a) Prove that

$$\min_a \mathbb{E}(\texttt{cost}(a)) \leq \mathbb{E}(\min_a \texttt{cost}(a)) + O(\sqrt{T \log(KT)}).$$

Take Away: All $\tilde{O}(\sqrt{T})$ regret bounds for algorithms for stochastic bandits (*e.g.,* `UCB`, `Successive Elimination`) carry over to "hindsight regret".

(b) (Lower bound for hindsight regret) Construct a problem instance with a deterministic adversary for which any algorithm suffers regret

$$\mathbb{E}[\texttt{cost(ALG)} - \min_a \texttt{cost}(a)] \geq \Omega(\sqrt{T \log K}).$$

Hint: Using Hoeffding's inequality, show that

$$\mathbb{E}[\min_a \texttt{cost}(a)] \leq \frac{T}{2} - \Omega(\sqrt{T \log K}).$$

(c) Prove that algorithms `UCB` and Successive Elimination achieve logarithmic regret bound even for hindsight regret, assuming that the best-in-foresight arm a^* is unique.

4. **(`Hedge` is an `FPL`)** Consider the following `FPL` strategy for the expert problem: at round t, play

$$i_t = \arg\min_i (L_{t-1}(i) - L_0(i)),$$

where $L_0(i), 1 \leq i \leq N$ are N i.i.d. variables with *Gumbel* distribution, i.e., $\mathbb{P}(L_0(i) \leq x) = \exp(-\exp(-\eta x))$ for some parameter $\eta, \forall i$.

(a) Prove that for any j, $\mathbb{P}(i_t = j) = \mathbb{P}[j = \arg\max_i \frac{\exp(-\eta L_{t-1}(i))}{\exp(-\eta L_0(i)}]$.

(b) Prove that the random variable $v(i) = \exp(-\eta L_0(i))$ follows the standard exponential distribution.

(c) For any positive numbers $a_i, 1 \leq i \leq N$, prove that $\mathbb{P}[j = \arg\max_i \frac{a(i)}{v(i)}] = \frac{a(j)}{\sum_{i=1}^N a(i)}$. Conclude that `FPL` with Gumbel noise is equivalent to sampling an expert using `Hedge`'s prediction.

5. **(Performance of the Deterministic Algorithms)** Prove that any deterministic algorithm for the online learning problem with K experts and $0-1$ costs can suffer total cost T for some deterministic-oblivious adversary, even if $\texttt{cost}^* \leq T/K$.

6. **(`EXP4` with Shifting Experts)** In the usual adversarial bandit setting as studied in the class, there are N experts, who, at each round recommends one of the K actions. As we proved in the class, the `EXP4` algorithm achieves a regret of $O(\sqrt{KT \log N})$, with respect to the best expert in the hindsight. In this problem, we use this result to get regret upper-bound with respect to certain "quasi-static" policies as defined below.

(a) Define an *S-shifting policy* to be a policy which makes at most S number of changes in its recommendation. More formally, an *S-shifting policy* is a sequence of arms $\pi = (a_t : t \in [T])$ with at most S "shifts": rounds t such that $a_t \neq a_{t+1}$. *S-shifting regret* is defined as the algorithm's total cost minus the total cost of the best *S-shifting policy*:

$$R_S(T) = \texttt{cost(ALG)} - \min_{S-\textit{shifting policies } \pi} \texttt{cost}(\pi).$$

Define the “Experts” in `EXP4` appropriately to show that

$$\mathbb{E}[R_S(T)] = O(\sqrt{KST\log(KT)}).$$

(b) **(Slowly Changing Costs):** Consider a randomized oblivious adversary such that the expected cost of each arm can change by at most ϵ in each round. Rather than compete with the best fixed arm, we compete with the far stronger benchmark of the best current arm: $c_t^* = \min_a c_t(a)$. More formally, we are interested in *dynamic regret*, defined as

$$R^*(T) = \min(\texttt{ALG}) - \sum_{t\in[T]} c_t^*.$$

Note that dynamic regret is the same as the S shifting regret as in part (a) with $S = T$. Use `EXP4` to obtain the following regret bound

$$\mathbb{E}[R^*(T)] \le O(T)(\epsilon K \log KT)^{1/3}.$$

Hint: Use part (a) with a suitably chosen value of S.

7. **(Regret bound against benchmarks with limited memory)** Consider an example of adversarial contextual bandits where the $x_1, x_2, \ldots, x_T$ is a reward sequence vectors chosen in advance by an adversary with $x_t \in [0,1]^k$. Furthermore, let $o_1, o_2, \ldots, o_T$ be a sequence of observations, also chosen in advance by an adversary with $o_t \in [O]$ for some fixed $O \in \mathbb{N}^+$. Then let $\mathcal{H}$ be the set of functions $\phi : [O]^m \to [k]$ where $m \in \mathbb{N}^+$. In each round the learner observes o_t and chooses an action A_t based on $o_1, A_1, X_1, \ldots, o_{t-1}, A_{t-1}, X_{t-1}, o_t$ and the regret is

$$R_T = \max_{\phi\in\mathcal{H}} \sum_{t=1}^{T} \left(x_{t\phi(o_t, o_{t-1}, \ldots, o_{t-m+1})} - x_{tA_t} \right),$$

where $o_t = 1$ for $t \le 0$. This means the learner is competing with the best predictor in hindsight that uses only the last m observations. Show that there exists an algorithm such that

$$\mathbb{E}[R_T] \le \sqrt{2TkO^m \log(k)}.$$