Name: ____________________

Question:	1	2	3	Total
Points:	10	12	18	40
Score:				

E1 245 – Online Prediction and Learning, Aug-Dec 2018 – Exam

Instructions

- Write your name on top of this question sheet, attach your solution sheets to it and return everything together.
- The total time for this exam is 2 hours. The exam has 3 questions, for a total of 40 points and 8 bonus points.
- Feel free to use any notes for this exam.
- Academic dishonesty will not be tolerated.

Useful results:

- Hoeffding's inequality: For independent random variables $X_1, \ldots, X_n$ taking values in $[a, b]$, and $v \geq 0$, $\mathbb{P}\left[\left|\sum_{t=1}^n X_t - \sum_{t=1}^n \mathbb{E}\left[X_t\right]\right| \geq v\right] \leq 2\exp\left(-\frac{2v^2}{n(b-a)^2}\right)$.

1. (Stochastic online learning)
Consider a stochastic online learning problem with 2 actions or arms $\{1, 2\}$ with Bernoulli reward distributions. It is known that their Bernoulli parameters (μ_1, μ_2) are either (μ_-, μ_+) or (μ_+, μ_-), where $\mu_- := \frac{1-\epsilon}{2}$ and $\mu_+ := \frac{1+\epsilon}{2}$, for some (potentially unknown) $\epsilon \in (0, \frac{1}{2})$.
At each round $1 \leq t \leq T$, a learner plays a single action $I_t \in \{1, 2\}$ and gets observations as described below. The total (pseudo) regret of the learner after T rounds is $\sum_{t=1}^T \left(\frac{1+\epsilon}{2}\right) - \mu_{I_t}$.

   (a) **(2 points)** Suppose that after each play, the learner only observes a reward sample from the action which it plays (independent of the past). Describe an algorithm for playing arms and a non-trivial (sub-linear in T) regret bound for it. (Just state without any proof.)

   (b) **(8 points)** Suppose now that after each play I_t, the learner observes rewards from both the actions' reward distributions, i.e., it observes $X_1(t) \sim \text{Ber}(\mu_1)$ and $X_2(t) \sim \text{Ber}(\mu_2)$, independent of each other and the past (note that the reward earned by the

learner is the same, μ_{I_t}, however the other arm is also observed). Design an algorithm with as small regret in T rounds as possible. (A concrete regret bound is expected, but without needing to be precise about constants.)
(Hint: Exploit the iid environment to do much better than before.)

(c) **(Bonus question: 8 points)** Can you argue a matching (up to constants) fundamental lower bound on the regret of any 'reasonable'[1] learning algorithm for this problem?

2. (Solving linear programs using Exponential-Weights)
Suppose we want to solve the following linear feasibility problem[2]: Given vectors $a_1, \ldots, a_m$ in $\mathbb{R}^d$, we want to find a linear half space, described by some vector x, that contains all these vectors. More precisely, we would like to find a vector $x \neq 0$ with $x^T a_j \geq 0 \; \forall j \in [m]$. Without loss of generality, we can also include the condition $\mathbf{1}^T x = 1$ in the specification[3] for x, so that our search is over all probability distributions on the dimensions $[d]$.

Suppose there really exists a vector x_* such that $x_*^T a_j \geq \epsilon > 0$ for all $j \in [m]$ (this is often called a large margin condition in machine learning). Consider the following procedure for the linear feasibility problem, based on the Exponential-Weights online algorithm.

initialize: experts $\{1, 2, \ldots, d\}$, x_1 as the uniform distribution over the experts, $t = 1$, $\rho = \max_j ||a_j||_\infty$, and $\eta > 0$

while $\min_{1 \leq j \leq d} x_t^T a_j < 0$:

1. set $l_t := -a_{j_t}/\rho$, where $j_t \in [d]$ is some constraint that is violated by the current distribution x_t, i.e., $x_t^T a_{j_t} < 0$
2. run one iteration of Exponential-Weights(η), on the experts, with the loss vector as l_t, i.e., set $x_{t+1}(i) \propto x_t(i) \exp(-\eta l_t(i))$ $\forall 1 \leq i \leq d$, such that $\mathbf{1}^T x_{t+1} = 1$
3. increment t to $t + 1$

end while
return x_t as a feasible solution

Intuitively, this procedure at each step feeds a 'hard' example (a point a_j that is on the wrong side of the current half space x_t, with large loss) to Exponential-Weights, i.e., it rewards constraint satisfaction and penalizes constraint violation to get Exponential-Weights to learn a good half space.

(a) **(6 points)** Note that by definition, each loss vector $l_t \in [-1, 1]^d$. It is a standard fact that Exponential-Weights enjoys the regret bound

$$\sum_{t=1}^{T} l_t^T x_t - \min_{x \in \Delta_d} \sum_{t=1}^{T} l_t^T x \leq \eta T + \frac{\log(d)}{\eta},$$

for any sequence of loss vectors $l_1, \ldots, l_T$ in $[-1, 1]^d$, where Δ_d denotes the set of all probability distribution vectors on $[d]$. Describe how you would use this to adjust

[1] You will have to identify a suitable notion for an algorithm to be a 'reasonable' learning algorithm.
[2] This is actually a rather general form of linear programming.
[3] $\mathbf{1}$ denotes the all-ones vector in $\mathbb{R}^d$.

the learning rate η in the procedure above, so that the number of rounds taken by it to terminate is bounded above by a suitable function of ρ, d and ϵ.

(b) **(6 points)** What if the linear feasibility problem admits a solution x_* but its margin ϵ is unknown? How would you modify the algorithm above that assumes knowledge of ϵ, to get an algorithm that still terminates, with a feasible solution, in the same number of rounds as above (upto constants)?

3. (Stochastic bandits)
Consider the iid[4] stochastic bandit problem with K Bernoulli-reward arms and total time T. Recall that if μ_i denotes the expected reward of the ith arm, then the regret of a bandit algorithm that plays an arm $I_t \in [N]$ at each time $1 \leq t \leq T$, and observes only the (random) reward from the chosen arm, is defined to be $R(T) := T \cdot \max_i \mu_i - \sum_{t=1}^{T} \mathbb{E}\left[\mu_{I_t}\right]$.

Explain briefly which of the following algorithms will/will not always achieve sublinear (pseudo-) regret with time horizon T (Recall: $R(T)$ is sublinear $\Leftrightarrow \lim_{T\to\infty} \frac{R(T)}{T} = 0$).

(a) **(3 points)** Play all arms exactly once. For each arm i, initialize s_i to be its observed reward and $n_i := 1$. At each time $t \leq T$, play $I_t := \arg\max_i s_i/n_i$ (break ties in any fixed manner), get (stochastic) reward R_t and update $s_{I_t} \leftarrow s_{I_t} + R_t$, $n_{I_t} \leftarrow n_{I_t} + 1$.

(b) **(3 points)** Play all arms exactly once. For each arm i, initialize s_i to be its observed reward and $n_i := 1$. At each time $t \leq T$, toss an independent coin with probability of heads $p := 1/\sqrt{T}$. Play $I_t := \arg\max_i s_i/n_i$ (break ties in any fixed manner) if the coin lands heads, else play a uniformly random arm, get (stochastic) reward R_t and update $s_{I_t} \leftarrow s_{I_t} + R_t$, $n_{I_t} \leftarrow n_{I_t} + 1$.

(c) **(3 points)** Same as the previous part but with $p := 1/T$.

(d) **(3 points)** Same as the previous part but with $p := 1/K$.

(e) **(3 points)** For each arm $i \in [N]$, initialize $u_i = 1, v_i = 1$. At each time $t \leq T$, sample independent random variables $\theta_i(t) \sim \text{Beta}(u_i, v_i)$, and play $I_t := \arg\max_i \theta_i(t)$ (break ties in any fixed manner). Get (stochastic) reward R_t and update $u_{I_t} \leftarrow u_{I_t} + R_t$, $v_{I_t} \leftarrow v_{I_t} + (1 - R_t)$.

(f) **(3 points)** Play all arms exactly once. For each arm i, initialize s_i to be its observed reward and $n_i := 1$. At each time $t \leq T$, let $A_t := \arg\max_i s_i/n_i$ and $B_t := \arg\max_{i \neq A_t} s_i/n_i$ denote the best and second-best arms in terms of sample mean, respectively. Play $I_t \in \{A_t, B_t\}$ chosen uniformly at random, get (stochastic) reward R_t and update $s_{I_t} \leftarrow s_{I_t} + R_t$, $n_{I_t} \leftarrow n_{I_t} + 1$.

[4] independent and identically distributed