E1 245 - Online Prediction and Learning, Aug-Dec 2019
Homework #2

1. *Exp-concavity and common loss functions*

   (a) Show that if for a $y \in \mathscr{Y}$ and $\eta > 0$ the function $F(z) := e^{-\eta l(z,y)}$ is concave, then $l(z,y)$ is a convex function of z.

   (b) Show that the relative entropy loss $l(x,y) := y \log \frac{y}{x} + (1-y)\log\frac{1-y}{1-x}$, $x, y \in [0,1]$, is 1-exp-concave for all valid values[1] of y.

   (c) Show that the squared loss $l(x,y) := (x-y)^2$, $x, y \in [0,1]$, is $\frac{1}{2}$-exp-concave for all valid values of y.

   (d) Show that the absolute value loss $l(x,y) := |x-y|$, $x, y \in [0,1]$, *cannot* be η-exp-concave for any $\eta > 0$.

2. *Improved regret with exp-concave loss functions*
   Show that if the Exponential Weights algorithm is run in the prediction-with-expert-advice setting with a σ-exp-concave loss function $l : \mathscr{D} \times \mathscr{Y} \to [0,1]$ (over $\mathscr{D}$) and the learning rate $\eta = \sigma > 0$ over N experts, then the algorithm enjoys the regret bound
   $$\sum_{t=1}^{T} l(p_t, y_t) - \min_{i \in [N]} \sum_{t=1}^{T} l(f_{i,t}, y_t) \le \frac{\log N}{\sigma}.$$
   (note: regret does not grow with time T!) [Hint: Look at the place where Hoeffding's lemma was applied for the general convex loss function case.]

3. *Linear programming 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 that contains all these vectors. More precisely, we would like to find a vector $x \neq 0$ with $x^T a_j \ge 0 \, \forall j \in [m]$. 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 $x_*^T a_j \ge \varepsilon > 0$ 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 $\min_{1 \le j \le d} x_t^T a_j < 0$:

   (a) 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., $x_t^T a_{j_t} < 0$

   (b) run one iteration of Exponential-Weights(η), on the experts, with the loss vector as l_t, i.e., set $x_{t+1}(i) \propto x_t(i)\exp(-\eta l_t(i))$ $\forall 1 \le i \le d$, such that $\mathbf{1}^T x_{t+1} = 1$

   (c) increment t to $t+1$

   end while
   return x_t as a feasible solution

---
[1] By convention, we take $\frac{0}{0} := 0$ & $0 \cdot \log 0 := 0$.
[2] This is actually quite a general form of linear programming problem.
[3] $\mathbf{1}$ denotes the all-ones vector in $\mathbb{R}^d$.

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) 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 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) 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)?

4. *Sequential probability estimation*
Suppose you (the learner) are observing an arbitrary bit sequence $y_1, y_2, \ldots,\ y_i \in \{0,1\}$, generated from some source (think, e.g., a digital voice signal or someone typing on a keyboard). The following occurs at each round $t \geq 1$: You guess a probability distribution $\hat{p}_t \equiv (\hat{p}_t(0), \hat{p}_t(1)) \in \{(p, 1-p) : 0 \leq p \leq 1\}$ for the next bit y_t. Following your guess, y_t is revealed and you suffer a loss of $\log \frac{1}{\hat{p}_t(y_t)}$.

Consider competing in this guessing game with the class of all *constant* experts. A constant expert is a rule parameterized by $p \in [0,1]$ that always guesses the probability distribution $(p, 1-p)$ (the analog of a constantly rebalancing portfolio for 2 stocks in the sequential investment problem).

Write down the Exponential Weights prediction algorithm with uniform initial weights and learning rate $\eta = 1$. Can you express its prediction at each time t in the *simplest* possible form[4]? You may use the identity

$$\int_0^1 q^{n_1}(1-q)^{n_2}\, dq = \frac{1}{(n_1+n_2+1)\binom{n_1+n_2}{n_1}},$$

for any integers $n_1, n_2 \geq 0$, and where $\binom{a}{b}$ is the standard binomial coefficient $\frac{(a+b)!}{a!b!}$.

5. *Sequential probability estimation – continued*
With regard to the previous question, can you show that the (worst case) regret of the Exponential Weights algorithm you wrote down (with $\eta = 1$) in T rounds with respect to all the constant experts, for any sequence of bits $y_1, \ldots, y_T$, is no more than $\log(1+T)$?

[4] implementable using finitely many arithmetic operations