E1 245 - Online Prediction and Learning, Aug-Dec 2019
Homework #1

1. *Exponential inequalities*
   Prove the following inequalities (useful in showing mistake bounds for the Weighted-Majority algorithm, for instance):

   (a) $\forall x \in \mathbb{R} \quad e^x \geq 1 + x$

   (b) $\forall 0 \leq x \leq \frac{1}{2} \quad -\log(1-x) \leq x + x^2$

2. *Generalizing HALVING*
   We showed that the HALVING algorithm for binary prediction makes at most $\log_2 N$ mistakes using the advice of N experts whenever some expert is always predicting correctly. Show that a "straightforward" modification of MAJORITY makes at most $O((m+1)\log_2 N)$ mistakes[1] whenever the best expert makes $m \geq 0$ mistakes. (Hint: When there is no best expert, think about how HALVING can "crash", and a very simple way to make it carry on.)

3. *Optimality of HALVING*
   For the problem of 1-bit prediction with expert advice, prove the following statement that establishes that the $\log_2 N$ mistake bound for HALVING, with a perfect expert, cannot be improved.

   Given *any* deterministic 1-bit prediction algorithm (i.e., an algorithm that issues a non-random prediction at each round) with $N = 2^n$ experts, there exist 1-bit outcomes $y_1, \ldots, y_n$, and 1-bit expert advice $x_{ti} \in \{0,1\}$, $1 \leq i \leq N$, $1 \leq t \leq n$, on which (a) some expert makes no mistakes, and (b) the given algorithm makes n mistakes.

4. *Smarter Exponential Weights when the best expert's loss is known beforehand*
   Consider prediction with expert advice with a convex loss (in the first argument) bounded in $[0,1]$. Suppose you know in advance what the best expert's total loss is going to be at time T (this could be much less than $O(T)$, e.g., a constant). The aim of this exercise is to see if this information a priori can be used to learn faster and reduce regret. (Recall that we already showed an analogous performance bound in the mistake count or 0-1 loss setting for the Weighted Majority algorithm, in terms of the mistake performance of a good expert.)

   (a) First, prove that $\log \mathbb{E}[e^{sX}] \leq (e^s - 1)\mathbb{E}[X]$ for any random variable $X \in [0,1]$.

   (b) Let the experts be indexed by $\{1, 2, \ldots, K\}$. Use the previous result instead of (the weaker) Hoeffding's inequality to show that $L_T(\text{ExpWts}) \leq (\eta L_T^* + \log K)/(1 - e^{-\eta})$ for ExpWeights run with parameter $\eta > 0$. Here, $L_T(\text{ExpWts})$ is the cumulative loss of the algorithm and $L_T^* := \min_{i=1,\ldots,K} L_{i,T}$ is the cumulative loss of the best expert, over T rounds.

   (c) Use the elementary inequality $\eta \leq (e^{\eta} - e^{-\eta})/2$ in the above bound to obtain a further bound. Then, assuming that the value of L_T^* is known beforehand, show that setting the ExpWeight learning rate to $\eta := \log(1 + \sqrt{(2\log K)/L_T^*})$ gives regret at most $\sqrt{2L_T^* \log K} + \log K$, which can be *significantly small when the best expert's cumulative loss is small.*

[1] Big-Oh notation: We say that $f(m) = O(g(m))$ if there exist constants α, m_0 such that $f(m) \leq \alpha g(m)\ \forall m \geq m_0$.

(d) What if the best expert's loss L_t^* is not known beforehand, but available only sequentially, i.e., at time t for each t? Using a doubling trick idea, can you design an algorithm that does not require advance knowledge of the cumulative loss of the best expert, and show that its regret bound is only worse by a constant factor compared to the one in part (c) above?

5. *The* doubling trick *for obtaining "anytime" learning algorithms*
Suppose an online learning algorithm with a parameter $\eta > 0$ enjoys a regret bound of $\frac{\beta}{\eta} + \gamma\eta T$ for a total of T rounds, where β and γ are some positive constants (think of the Exponential Weights forecaster for instance). If the time horizon T is known in advance, then setting $\eta := \sqrt{\frac{\beta}{\gamma T}}$ minimizes the bound. Consider the following tweak to obtain an algorithm (and bound) that does NOT require knowing the horizon T beforehand (i.e., an "anytime" algorithm). Time is divided into periods: the m-th period is formed by rounds $2^m, 2^m+1, \ldots, 2^{m+1}-1$, where $m = 0, 1, 2, \ldots$ In every m-th period, starting at round 2^m, the original algorithm is re-initialized and run with a parameter $\eta_m := \sqrt{\frac{\beta}{\gamma 2^m}}$. Prove that for *any* T, this modified algorithm enjoys a regret bound which is at most $\frac{\sqrt{2}}{\sqrt{2}-1}$ times the original optimal regret bound.