Problem Set 3

Note: We will discuss the first problem in the problem-solving session. However, you still need to write your own solution to every problem.

Problem 1: Learning to maximize two-dimensional functions . . . . . (10 points)
Consider the following maximization problem. In each round $i = 1, 2, \ldots, n$ you pick a pair of numbers $(x_1^i, x_2^i) \in [0,1]^2$. In round i, nature picks an arbitrary function $f^i : [0,1]^2 \longrightarrow [0,1]$, where f^i's gradient is guaranteed to be in $[-1,1]^2$. After the prediction is made, nature reveals only your payoff $f^i(x_1^i, x_2^i)$. You want to maximize your total payoff against the best-in-hindsight choice, i.e. minimizing the following regret:

$$R_n = \max_{(x_1,x_2)\in[0,1]^2} \frac{1}{n}\sum_{i=1}^{n} f^i(x_1, x_2) - \frac{1}{n}\sum_{i=1}^{n} f^i(x_1^i, x_2^i).$$

Design an online learning algorithm for this problem and provide a bound for R_n of your algorithm. Clearly describe your algorithm and analyze it (state and derive your regret bound). The regret bound should only depend on n and vanish as $n \to \infty$. Explain your answer.

Problem 2: Learning to predict . . . . . (10 points)
Consider the following prediction problem. On each day $t = 1, 2, \ldots, n$ you predict the probability of raining $p^t \in [0,1]$. After the prediction is made, nature reveals the state $\theta^t \in \{0,1\}$ (whether it really rains or not). You want to minimize your mean squared error on all days against the best-in-hindsight prediction, i.e. minimizing the following regret:

$$R_n = \frac{1}{n}\sum_{t=1}^{n}(p^t - \theta^t)^2 - \min_{p\in[0,1]} \frac{1}{n}\sum_{t=1}^{n}(p - \theta^t)^2.$$

Design an online learning algorithm for this problem and provide a bound for R_n of your algorithm. Clearly describe your algorithm and analyze it (state and derive your regret bound). The regret bound should only depend on n and vanish as $n \to \infty$. Explain your answer.

Hint: Minimizing the regret for squared error is equivalent to minimizing the regret for reward (1 − squared error). You might want to first discretize the prediction space to k levels, then use the EW algorithm and optimize k.