CS-621 Theory Gems September 30, 2012

Problem Set 1

Lecturer: Aleksander Mądry *Due: October 14, 2012*

Problem 1. Recall the stock market framework (see Section 2 in notes from Lecture 1). Prove that, in the worst case, no *deterministic* prediction algorithm for this framework can make less than $2m^* + \log_2 n$ mistakes. Here, n is the size of the expert set and m^* is the number of mistakes made by the best expert.

Note: This means that it is essential to employ randomness to obtain the performance guarantee of the Randomized Weighted Majority algorithm (cf. Lemma 5 in the notes).

Problem 2. Consider the (general) learning-from-expert-advice framework (see Section 5 in notes from Lecture 1) with n experts. A popular measure of algorithms' performance in this framework is so-called *regret*. The regret is the difference $l_{ALG} - l^*$ between the total loss l_{ALG} of the algorithm and the loss $l^* = \min_i \sum_{t=1}^{T} l_i^t$ of the best expert in hindsight.

Assume that the game lasts for T rounds (and this value is known in advance) and the loss l_i^t in each round t is in the interval $[0, 1]$.

(a) Show that the regret suffered by the Multiplicative Weights Update algorithm (with appropriate parameters) is $O(\sqrt{T \log n})$.

(b) How important was it that the MWU algorithm was allowed to switch between experts in each round, while we compared its performance to the best *fixed* expert? To answer this, show that for any algorithm ALG, one can construct an example in which the total loss l_{ALG} of this algorithm will be at least $T(1-1/n)$, while the loss l^*_{CH} of the best "changing" expert is 0, i.e., $l^*_{CH} = \sum_{t=1}^{T} \min_i l_i^t = 0$.

Problem 3. Show that for any (non-empty) subset S of n experts, the Multiplicative Weights Update algorithm suffers a total loss l_{MWU} of at most

$$l_{MWU} \leq \max_{i \in S} \left(\sum_t l_i^t + \varepsilon \sum_t |l_i^t|\right) + \frac{\rho \ln \frac{n}{|S|}}{\varepsilon},$$

where $0 \leq \varepsilon \leq \frac{1}{2}$ and each $l_i^t \in [-\rho, \rho]$.

Note: This means that MWU algorithm has better performance when there are many good experts.

Problem 4. (Extra credit) Consider again the setting from Problem 2. That is, the (general) learning-from-expert-advice framework with n experts, the total number of rounds T known upfront, and $l_i^t \in [0, 1]$ for each i and t.

We now would like to be able to bound our regret not with respect to a one "fixed" expert, but to a "changing" expert that can switch *at most k times* during the execution of the game. (Of course, we do not know when the switches happen.) Design an algorithm for this setting whose regret is at most $O(\sqrt{kT \log Tn})$.[1]

[1] The optimal bound one can get here is $O(\sqrt{kT \log n})$, but you do not need to show that.