IE613: Online Machine Learning **Jan-Apr 2018**

Assignment 1: February 11

Instructions: *You are free to code in Python/Matlab/C/R. Discussion among the class participants is highly encouraged. But please make sure that you understand the algorithms and write your own code. If you share any code with any other student then you will be penalized and can be given 0 mark for that question.*

Submit the code and report by 11:59PM, 20th February on Moodle. Late submission will not be evaluated and given 0 mark.

Question 1 (Full information setting) *Consider the problem of prediction with expert advice with $d = 10$. Assume that the losses assigned to each expert are generated according to independent Bernoulii distributions. The adversary/environment generates loss for experts 1 to 8 according to $Ber(.5)$ in each round. For the 9th expert, loss is generated according to $Ber(.5 - \Delta)$ in each round. The losses for the 10th expert are generated according to different Bernoulii random variable in each round– for the first $T/2$ rounds, they are generated according to $Ber(0.5 + \Delta)$ and the remaining $T/2$ rounds they are generated according to Bernoulii random variable $Ber(0.5 - 2\Delta)$. $\Delta = 0.1$ and $T = 10^5$. Generate (pseudo) regret values for different learning rates (η) for each of the following algorithms. The averages should be taken over atleast 20 sample paths (more is better). Display 95% confidence intervals for each plot. Vary c in the interval [0.1 2.1] in steps of size 0.2 to get different learning rates. Implement Weighted Majoirty algorithm with $\eta = c\sqrt{2\log(d)/T}$.*

Question 2 (Bandit setting) *Consider the problem of multi-armed bandit with $K = 10$ arms. Assume that the losses are generated as in Question 1. For each of the following algorithms generate (pseudo) regret for different learning rates (η) for each of the following algorithms. The averages should be taken over atleast 50 sample paths (more is better). Display 95% confidence intervals for each plot. Vary c in the interval [0.1 2.1] in steps of size 0.2 to get different learning rates.*

- *EXP3. Set $\eta = c\sqrt{2\log(K)/KT}$.*
- *EXP3.P. Set $\eta = c\sqrt{2\log(K)/KT}$, $\beta = \eta$, $\gamma = K\eta$.*
- *EXP3-IX. Set $\eta = c\sqrt{2\log(K)/KT}$, $\gamma = \eta/2$.*

Question 3 *In Question 2, which one of EXP3, EXP3.P and EXP3-IX performs better and why?*

Question 4 *Consider the online learning setting over finite hypothesis class $\mathcal{H}$ under the realizability assumption. Assume that in round t, we get to observe $z_t = y_t + \nu_t$, where y_t is the true label generated according to the fixed (but unknown) hypothesis, ν_t is a Bernoulli noise (i.i.d.), and $+$ is the XOR operation. Suppose for some $\gamma \in [0\ 1/2)$, $\Pr\{y_t \neq z_t\} \leq \gamma$, give an online learning algorithm for this setting and its mistake bound.*

Question 5 *Consider an online algorithm that enjoys a regret of the form $\alpha\sqrt{T}$, but its parameters require the knowledge of T. The doubling trick, described next enables us to convert such an algorithm into an algorithm that does not need to know the time horizon. The idea is to divide the time into periods of increasing size and run the original algorithm in each period.*

Doubling Trick
Input: Algorithm A whose parameter depend on the time horizon $m = 0, 1, 2, \ldots$ Run A on the 2^m rounds $t = 2^m, \ldots, 2^{m+1} - 1$

Show that if regret of A on each period of 2^m rounds is at most $\alpha\sqrt{2^m}$, then the total regret is at most

$$\frac{\sqrt{2}}{\sqrt{2}-1}\alpha\sqrt{T}.$$

Submission Format and Evaluation: Your should submit a report along with your code. Please zip all your files and upload via moodle. The zipped folder should named as YourRegistrationNo.zip e.g. '154290002.zip'. The report should contain two figures: one figure should have two plots corresponding to each algorithm in Q.1 and the other should have 3 plots one corresponding to each algorithm in Q.2. For each figure, write a brief summary of your observations. We may also call you to a face-to-face session to explain your code.

Note: Please calculate (pseudo) regret for each algorithm in Q.2 for a given set of parameters as follows:

Let μ_t^i denote the mean of arm i in round t. Suppose an adversary generates sequence of loss vectors $\{l_t\}_{t=1}^T$ and an algorithm generates sequence of pulls $\{I_t\}_{t=1}^T$, the (pseudo) regret for this sample path is

$$\sum_{t=1}^{T} \mathbb{E}[l_t(I_t)] - \min_i \sum_{t=1}^{T} \mathbb{E}[l_t(i)] \tag{1.1}$$

$$= \sum_{t=1}^{T} \mu_t^{I_t} - \min_i \sum_{t=1}^{T} \mu_t^i \tag{1.2}$$

Note that in this calculation we only considered the mean values of losses, not the actual losses suffered. It is Okay if this value turns out to be negative. There is no expectation over random choices of I_ts here. Now generate 20 such sample paths and take their average.