CS6046: Multi-Armed Bandits
Assignment 1
Course Instructor : Arun Rajkumar.
Release Date : February-21, 2020
Submission Date: On or before 5 PM on March-5,2020

SCORING: There are 3 question in this assignment which contributes to 15 points towards your final grade. Each question carries 5 points.

WHAT SHOULD YOU SUBMIT? You should submit a zip file titled 'Solutions_ roll-number.zip' Your assignment will NOT be graded if it does not contain all of the following:

- A PDF file which includes explanations regarding each of the solution as required in the question. Title this file as 'Report.pdf'
- Source code for all the programs that you write for the assignment clearly named.

CODE LIBRARY: You are expected to code all algorithms from scratch. You cannot use standard inbuilt libraries for the algorithms. You are free to use inbuilt libraries for plots. You can code using either Python or Matlab or C.

GUIDELINES: Keep the below points in mind before submission.

- Plagiarism of any kind is unacceptable. These include copying text or code from any online sources. These will lead to disciplinary actions according to institute guidelines.
- Any graph that you plot is unacceptable for grading unless it labels the x-axis and y-axis clearly.
- Don't be vague in your explanations. The clearer your answer is, the more chance it will be scored higher.

LATE SUBMISSION POLICY You are expected to submit your assignment on or before the deadline to avoid any penalty. Late submission incurs a penalty equal to the number of days your submission is late by.

Q1: Perceptron and OGD
Consider an online classification scenario where the learner receives $\mathbf{x}_t \in \mathbb{R}^d$ and predicts $\hat{y}_t = \{0, 1\}$ as the sign of $\mathbf{w}_t^T \mathbf{x}_t$. Further assume that the adversary has a true $\mathbf{w}^*$ based on which it decides if the learner makes a mistake or not. Instead of revealing the true answer y_t after every round, the adversary instead chooses a convex function $f_t : \mathbb{R}^d \to \mathbb{R}$ at every round and the loss is computed w.r.t to the same. Assume that the learner runs an Online Gradient Descent algorithm. Show that there are convex functions the adversary can choose at each round so that the OGD recovers the Perceptron algorithm. Use the regret analysis for OGD to obtain a mistake bound for the Perceptron algorithm. What can you say about choosing the learning rate η for OGD in this scenario?

Q2: Binary World
You are given a dataset *Dataset1_X.txt* which contains 10000 rows and 1000 columns where each entry is either $0, 1$. Assume an online classification scenario where expert $i \in [1000]$ at time instant $t \in [10000]$ predicts the bit in the tth row and i-th column of this dataset. Assume that once the algorithm predicts $\hat{y}_t \in \{0, 1\}$, the adversary reveals the true answer y_t which is the bit in the t-th row of the file *Dataset1_Y.txt*.

Implement the following algorithms for this problem: Perceptron, Winnow, Weighted Majority, Randomized Weighted Majority. Plot the mistakes as a function of time. Based on what you observe, do you think the adversary is *realizable*? If so, describe what could be the nature of the adversary. If not, explain why you believe so.

Q3: Randomizing the Leader: Consider the problem of online learning on the simplex Δ_d where d $= 1000$; At round t, you predict p_t and receive a vector z_t and suffer a loss of $p_t^T z_t$. Assume the adversary picks the vector z_t as the t-th row in the dataset *Dataset2_Z*. Implement FTRL with quadratic and entropic regularization for this problem and plot the regret over time.
Now, consider the following algorithm which first picks and fixes a random 1000 dimensional vector R sampled uniformly from $[0, 1/\eta]^d$ and uses the following rule for prediction

$$p_{t+1} = \arg\min_{p \in \Delta_d} \sum_{i=1}^{t} \left(p^T (z_i + R)\right)$$

How would you choose η for this problem? For the value chosen, plot the regret bound for this algorithm as well. How does the regret bound compare with the previous two algorithms for this problem?