E1 245 - Online Prediction and Learning, Aug-Dec 2014
Homework #3

1. *Regret Minimization and Classical Optimization (8 points)*
Suppose you have a good algorithm for online convex optimization which, over the (convex) domain $\mathcal{K}$ and for a sequence of loss functions from the family $\mathcal{L}$ of convex functions, can achieve regret $R(T) = o(T)$ in T rounds, $T \in \{1, 2, \ldots\}$. Now, suppose you wish to minimize a given function $f \in \mathcal{L}$ over $\mathcal{K}$ to within accuracy $\epsilon > 0$, i.e., output $x \in \mathcal{K}$ satisfying $f(x) \leq \min_{y \in \mathcal{K}} f(y) + \epsilon$. Describe how you would accomplish this using the above.

2. *Exponential Weights as FTRL (10 points)*
Show that executing Follow The Regularized Leader (FTRL) on the simplex $\Delta_N := \left\{(x_1, \ldots, x_N) : \sum_{i=1}^N x_i = 1, \forall i \; x_i \geq 0\right\}$ with the entropic regularizer[1] $R(x) \equiv R_\eta(x) := \frac{1}{\eta} \sum_{i=1}^N x_i \log x_i$, and for linear loss functions $f_t(x) = \langle z_t, x \rangle$, $z_t \in \mathbb{R}^N$, is equivalent to running the Exponential Weights algorithm on N experts with losses $\{z_t\}$ and parameter η.
(Hint: It is possible to derive this directly from first principles and the definition of the FTRL step. However, an alternative way is by using (a) the equivalence between FTRL and (unconstrained minimization + Bregman projection) proven in class, and (b) observing that Bregman projection wrt the regularizer R onto Δ_N is equivalent to scaling by the $||\cdot||_1$ norm.)

3. *Bregman Divergences (2×3 points)*
Let $D_R(x, y)$ be the Bregman divergence corresponding to a strictly convex function $R : \mathbb{R}^d \to \mathbb{R}$. Show the following properties of the Bregman divergence:

   (a) ('Three point inequality') $\forall u, v, w \in \mathbb{R}^d : \; D_R(u, v) + D_R(v, w) = D_R(u, w) + \langle u - v, \nabla R(w) - \nabla R(v) \rangle$.

   (b) $\nabla_x D_R(x, y) = \nabla R(x) - \nabla R(y)$.

4. *Fenchel Duals (3×4 points)*
Compute the Fenchel dual functions for the following on $\mathbb{R}^d$.

   (a) $F(x) = e^{x_1} + \cdots + e^{x_d}$.

   (b) $F(x) = \log(e^{x_1} + \cdots + e^{x_d})$.

   (c) $F(x) = \frac{1}{2}||x||_p^2$, $p \in [1, \infty]$.

5. *Numerical Evaluation (40 points)*
Consider the stochastic multi-armed bandit with N arms and Bernoulli reward distributions for each arm. For $N \in \{10, 100, 1000\}$, setting the arms' mean rewards to be, say, equi-spaced in $(0, 1)$, simulate each of the following bandit algorithms (preferably in `MATLAB/C/C++`).

[1] $0 \log 0$ is to be interpreted as 0 in the definition.

(a) Upper Confidence Bound (UCB)

(b) Thompson Sampling with the `Beta(1,1)` prior per arm

(c) Thompson Sampling with the `Beta(0.5,0.5)` prior per arm

(d) EXP3 with (optimally tuned) horizon-dependent parameter $\eta := \sqrt{\frac{2\log N}{NT}}$

(e) EXP3.P with parameters $\delta := 0.05$, $\beta := \sqrt{\frac{\log(N/\delta)}{NT}}$, $\eta := 0.95\sqrt{\frac{\log N}{NT}}$, $\gamma := 1.05\sqrt{\frac{N\log N}{T}}$

For each algorithm,

(a) Plot the cumulative regret for various values of time horizon $T \in \{10, 50, 100, 500, 1000, 5000, \ldots, 10^6, 5 \times 10^6\}$, averaged across many, say 10^5, sample paths (the more the better). Along with the averaged regret, you should also display a measure of its variance in the form of, say, error bars of width 1 standard deviation around the mean values (the `errorbar` or `shadedErrorBar` function in `MATLAB` can help do this).

(b) Comment (broadly) on the nature of the average regret curves you have obtained. What is the observed scaling of regret with the time horizon? How does it fare compared to theoretical regret bounds for the algorithm?

Finally, compare the performance of all the algorithms together. Can you offer an explanation for why some perform better than others? Also, turn in your code by email.