E1 245 - Online Prediction and Learning, Aug-Dec 2015
Homework #3

1. *Strong convexity of the entropy function (Pinsker's inequality)*

   (a) *(5 points)* Show that the negative entropy function $R(x) = \sum_i x_i \log x_i$ over the 2 dimensional simplex $\Delta_2 := \{(x, 1-x) : 0 \leq x \leq 1\}$ is 1-strongly convex with respect to the $||\cdot||_1$ norm.

   (b) *(10 points)* Prove the same in any number of dimensions $d \geq 2$.
   Hint: One way is to find a reduction to the $d = 2$ case. Let x and y be two vectors in Δ_d. Let $A := \{i : x_i \geq y_i\}$ be the coordinates where x dominates y. Can you find two new vectors x_A and y_A in Δ_2 so that $||x-y||_1 = ||x_A - y_A||_1$ and carry on?

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

3. *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]
   $R_\eta(x) := \frac{1}{\eta} \sum_{i=1}^N x_i \log x_i$, and linear loss functions $f_t(x) = \langle z_t, x \rangle$, is equivalent to running the Exponential Weights algorithm on N experts with loss vectors $\{z_t\}_{t \geq 1}$ and parameter η.
   Hint: You can derive this directly from first principles and the definition of the FTRL rule. 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.

4. *Programming exercise – multi-armed bandits, synthetic data (40 points)*
   Consider a stochastic multi-armed bandit problem with N arms and Bernoulli reward distributions. For each $N \in \{10, 100, 1000\}$, setting the arms' mean rewards to be, say, equispaced in $(0,1)$, simulate each of the following bandit algorithms in your favourite language/scientific package (`MATLAB/Python/C/C++/...`).

   (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 the (optimally tuned) horizon-dependent learning rate $\eta := \sqrt{\frac{2 \log N}{NT}}$

   (e) EXP3-IX (EXP3-Implicit Exploration, Neu 2015) with parameters $\delta := 0.05$, and time-dependent η_t, γ_t prescribed as per Thm. 1 of Neu's paper[2].

   For each of these algorithms,

---

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

[2] `http://cs.bme.hu/~gergo/files/N15b.pdf` , link also on course homepage

(a) Plot the cumulative regret for various values of total time horizon $T \in \{10, 50, 100, 500, 1000, 5000, \ldots, 10^6\}$, averaged across many (say about $10^3 - 10^5$) sample paths (the more the better). Along with the averaged regret, you should also display a measure of the regret variability in the form of, say, error bars of width 1-standard deviation around the mean (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? (Please turn in your code by email.)

5. *Programming exercise – multi-armed bandits, real-world data (40 points)*
Benchmark the bandit algorithms in the previous problem to adaptively find the fastest network server across time with respect to the `university-latencies` dataset, available as a zipped file download on the course page. The time horizon in this case corresponds to the total number of data records (1361), and each arm is a university server defined on the first line of the data file. Make sure you read the `license.txt` file in the dataset to understand the dataset; also note that the range of the measured values is not $[0, 1]$ (you may want to normalize). You must plot the regret as outlined before (mean as well as spread).