Theoretical Machine Learning - COS 511

Homework Assignment 5

Due Date: 12 Apr 2017, till 22:00

(1) Solve 4 out of the following 6 problems.

(2) Consulting other students from this course is allowed. In this case - clearly state whom you consulted with for each problem separately.

(3) Searching the internet or literature for solutions, other than the course lecture notes, is NOT allowed.

(4) All problems are weighted equally at 10 points each. Indicate on your problem set which four problems you choose to solve. Feel free to write down solutions for the other two as well, but your homework grade will only depend upon the four you mark to be graded.

Ex. 1:

In this exercise we prove a tight lower bound on the regret of any algorithm for online optimization

- For any sequence of T fair coin tosses, let N_h be the number of heads and $N_t = T - N_h$. Give an asymptotically tight upper and lower bound for $\mathbb{E}(|N_h - N_t|)$.
- Consider an expert advice setting with two experts: At each round the experts disagree, and if one obtains loss 1, the other obtain loss 0. Use the fact above to design a setting where any expert algorithm incurs expected regret, asymptotically matching the upper bound for Hedge.

Ex. 2:

- Prove that f is α-expconcave over $\mathcal{K}$ iff

$$\nabla^2 f(\mathbf{x}) \succ \alpha \nabla f(\mathbf{x}) \nabla f(\mathbf{x})^\top$$

- Prove that a strongly convex function is also exp-concave (with appropriate parameters). Show that the convers is not necessarily true.

Ex. 3:
Derive an algorithm, based on online newton step, with sub–linear regret bound for the portfolio selection problem. (Do not assume that $\mathbf{r}_t$ is bounded for below, in other words– if you put all your money in you might lose everything!!!)

Ex. 4:

(1) Show that the dual norm induced by a give matrix $A \succ 0$ is given by A^{-1}.

(2) Show the generlized Cauchy Schwartz inequality, i.e.

$$\mathbf{x} \cdot \mathbf{y} \leq \|\mathbf{x}\| \cdot \|\mathbf{y}\|^*.$$

(3) Show that $\| \cdot \|^*{}_1 = \| \cdot \|_\infty^*$ and $\| \cdot \|_2^* = \| \cdot \|_2$.

Ex. 5:
Consider the setting of prediction from expert advice. We say that an algorithm **switches** if its decision at consecutive times is different.

In the **switching-cost** variant, the player (algorithm) incurs an extra loss of 1 everytime she switches actions (experts). Formally, she incurs cost $f_t(a_t) + \mathbf{1}\{a_t \neq a_{t-1}\}$ for playing experts a_{t-1} and then a_t in iterations $t-1$ and t, where f_t is the adversarially chosen loss. i.e. the switching–cost regret is given by

$$S\text{Regret}_T = \sum_{t=1}^{t} \mathbf{g}_t(a_t) + \mathbf{1}\{a_t \neq a_{t-1}\} - \min_{a^*} \sum_{t=1}^{T} \mathbf{g}_t(a^*)$$

We will make the distinction between an *oblivious* adversary and an adaptive adversary. An adaptive adversary can choose his decision based on previous rounds. An oblivious adversary chooses the set of losses at the beginning of the game (i.e. $\mathbf{g}_1, \ldots, \mathbf{g}_T$ are independent of the choice of the learner, but can still be arbitrary).

- Prove that the minimax regret against an adaptive adversary is $\theta(T)$ in the switching cost regret model.
- Construct an algorithm for the swtiching cost setting, against an oblivious adversary that attain sublinear regret

hint/spoiler:

Refine the horizon T into S blocks of fixed size: consider a strategy that chooses a fixed action within each block. Compare this to a regret game where the costs are bounded between $[0, S]$ to show that the regret is $O(S\sqrt{T})$. Choose S

Ex. 6(The doubling trick):

Some of the algorithms we've constructed depend on knowing the parameter T in advance, for example, when using a fixed step size of $\frac{1}{\sqrt{T}}$ in OGD. In this exercise we consider a simple trick to turn any such algorithm to an algorithm that achieves a similar regret without the knoweledge of T.

Let A be an algoirhtm that attains regret of $O(\alpha\sqrt{T})$. Consider the following algorithm

- **for** $m = 1, 2, \ldots,$:
- run A on rounds $t = 2^m, \ldots, 2^{m+1} - 1$

Show that the algorithm has regret at most $\frac{\sqrt{2}}{\sqrt{2}-1}\alpha\sqrt{T}$.