Exercise Set V, Advanced Algorithms 2022

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students. Solve as many problems as you can and ask for help if you get stuck for too long. Problems marked * are more difficult but also more fun :).

These problems are taken from various sources at EPFL and on the Internet, too numerous to cite individually.

1 In this exercise we consider the Hedge algorithm and use the same notation as in the lecture notes. The average [external] regret of Hedge is defined as

$$\frac{\sum_{t \leq T} \vec{p}^{(t)} \cdot \vec{m}^{(t)} - \min_i \sum_{t \leq T} m_i^{(t)}}{T}$$

i.e., how much we "regret", on average over the days, compared to the best single strategy i.

1a If you knew the number of days T in advance, how would you set the parameter ϵ of Hedge to minimize the average external regret?

1b *(*)* Even if you do not know T in advance, describe a strategy that achieves roughly the same average external regret as in the case when T is known.

Hint: There is no need to redo the analysis from scratch. For example, you could consider restarting the algorithm each time you get to a day t of the form 4^i.

2 Suppose you are using the Hedge algorithm to invest your money (in a good way) into N different investments. Every day you see how well your investments go: for $i \in [N]$ you observe the change of each investment in percentages. For example, change(i) $= 20\%$ would mean that investment i increased in value by 20% and change(i) $= -10\%$ would mean that investment i decreased in value by 10%.

How would you implement the "adversary" at each day t so as to make sure that Hedge gives you (over time) almost as a good investment as the best one? In other words, how would you set the cost vector $\vec{m}^{(t)}$ each day?

3 Let $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$ and $c \in \mathbb{R}^n$. Consider the following linear program with n variables:

$$\begin{array}{ll} \textbf{maximize} & c^T x \\ \textbf{subject to} & Ax = b \\ & x \geq 0 \end{array}$$

Show that any extreme point x^* has at most m non-zero entries, i.e., $|\{i : x^*_i > 0\}| \leq m$.

Hint: what happens if the columns corresponding to non-zero entries in x^* are linearly dependent?

(If you are in a good mood you can prove the following stronger statement: x^* is an extreme point if and only if the columns of A corresponding to non-zero entries of x^* are linearly independent.)

4 Consider the following quadratic programming relaxation of the Max Cut problem on $G = (V, E)$:

$$\begin{array}{ll} \textbf{maximize} & \displaystyle\sum_{\{i,j\}\in E} (1 - x_i)x_j + x_i(1 - x_j) \\ \textbf{subject to} & x_i \in [0, 1] \quad \forall i \in V \end{array}$$

Show that the optimal value of the quadratic relaxation actually equals the value of an optimal cut. (Unfortunately, this does not give an exact algorithm for Max Cut as the above quadratic program is NP-hard to solve (so is Max Cut).)

Hint: analyze basic randomized rounding.