15-451/651 Algorithms, Spring 2019

Homework #7 **Due: Tuesday–Friday, April 23–26, 2019**

This is an oral presentation assignment. Same rules as HW2/4. Prove your answers correct. There is no programming assignment for this HW. *The problems in this HW are simple but have a few parts. Also, they cover new concepts, so please start and finish early!*

(100/3 pts) 1. **(Can you Hear me Now?)** There are two providers in town, called 1 and 2. You start off with a phone from 1. The rules are simple: every day you receive exactly one call. If its from someone with the same provider as your current provider, it's free, else it costs \$1. You can change providers on any day for the (low low) cost of \$$X$.

Consider the algorithm: maintain a counter K (initially zero), and whenever you get a call from someone with the other provider, increment K. When the counter reaches X, reset K to zero and then switch providers.

Show this algorithm is $O(1)$-competitive. For full credit, make sure this constant is at most 5. (Doing better than 5 is possible, but not required.)

Hint: think of a good potential function. Just like in the MTF analysis from lecture, consider the various actions you and OPT can take, and how each of them changes the potential.

(100/3 pts) 2. **(Fickle Experts.)** In class you saw the randomized weighted majority theorem, in which we were given n experts. Then over any sequence of T rounds, and any expert i, we had

$$\mathbf{E}[\text{number of mistakes by RWM}] \leq (1+\varepsilon)m_i + \frac{\ln n}{\varepsilon}.$$

Here m_i is the number of mistakes made by expert i until time T . In Section 4 of the notes, we observed that $m_i \leq T$, so dividing the above by T and choosing $\varepsilon := \sqrt{\frac{\ln n}{T}}$ we get that for any i

$$\mathbf{E}[\text{rate of mistakes by RWM}] \leq \frac{m_i}{T} + 2\sqrt{\frac{\ln n}{T}}.$$

I.e., for any expert i (which includes the best expert at time T) the *average regret (versus that expert)*, which is our mistake rate minus that of the expert's mistake rate, goes to zero as $T \to \infty$.

(a) Above, we assumed we knew the time horizon T and hence could set $\varepsilon = \sqrt{\frac{\ln n}{T}}$. What if we don't know T? Here's one algorithm: for $s = 1, 2, \ldots$, play 2^s rounds of RWM (starting from scratch) with $\varepsilon = \sqrt{\frac{\ln n}{2^s}}$.

Show that the average regret of this algorithm after time T is $O\left(\sqrt{\frac{\ln n}{T}}\right)$.

(b) Here's a different extension. Now you don't just want to compare yourself to the best you could have done by choosing a single expert and sticking with them. Call an deterministic algorithm K-*fickle* if over the time horizon T, it follows the advice of some expert i_1 for the first t_1 steps, then i_2 for the next t_2 steps, etc, and then i_K for the last t_K steps, where each $i_j \in [n]$, $t_j \geq 0$ and $\sum_{j=1}^{K} t_j = T$. (Assume you know T, else you can use the "guess-and-double" idea from part (a).) Give an algorithm such that for any K-fickle (deterministic) algorithm A,

$$\mathbf{E}[\#\text{ mistakes by your algo }] \leq (\#\text{ mistakes by } A)(1+\varepsilon) + \frac{O(K\log(nT))}{\varepsilon}.$$

Your algorithm is allowed to run in time $(nT)^{O(K)}$.

(100/3 pts) 3. **(Let's Eliminate Gauss!)** Given an $n \times n$ *symmetric* matrix A and an $n \times 1$ vector b, our goal is to solve the equation $Ax = b$ to high accuracy. We will use gradient descent to solve this problem quickly given some assumptions about A; see the lecture notes for background on gradient descent. *(The analysis here is independent of the one from lecture, but you should be comfortable with the ideas there.)*

Recall from linear algebra that every symmetric $n \times n$ matrix A can be written as $V\Lambda V^T$, where V is an $n \times n$ matrix whose columns are the eigenvectors of A, and Λ is a diagonal matrix whose entries are the eigenvalues of A. Recall that for $x \in \mathbb{R}^n$, $\|x\|^2 = \sum_{i=1}^{n} x_i^2$.

(a) Consider the function $f(x) = \frac{1}{2}\|Ax - b\|^2$. Prove that f is convex and the gradient $\nabla f(x) = A(Ax - b)$. *(Hint: if $g(y)$ is a convex function, what about $f(x) = g(Ax - b)$?)*

(b) Suppose $x^* = \operatorname{argmin}_x \frac{1}{2}\|Ax - b\|^2$. State why $A^2 x^* = Ab$.

(c) Suppose we set $x^{(0)} = 0^n$, and

$$x^{(t+1)} \leftarrow x^{(t)} - \nabla f(x^{(t)}).$$

Argue for any $i \geq 0$, $A(x^{(i+1)} - x^*) = (I - A^2)(A(x^{(i)} - x^*))$.

(d) Argue that $\|Ax^{(t)} - b\|^2 = \|A(x^{(t)} - x^*)\|^2 + \|Ax^* - b\|^2$. Hint: for $x, y \in \mathbb{R}^n$, if $\langle x, y\rangle = 0$, then $\|x + y\|^2 = \|x\|^2 + \|y\|^2$. You may also find part (b) useful.

The above parts should all be proven for any symmetric matrix A, regardless of whether it is invertible or not.

For the next parts, you may find the following statements helpful: (1) for a symmetric matrix B and a vector y, $\|By\| \leq \max(|\lambda_{max}|, |\lambda_{min}|) \cdot \|y\|$ where λ_{max} is the maximum eigenvalue of B and λ_{min} is the minimum eigenvalue of B, and (2) for a symmetric matrix B, the eigenvalues of $I - B$ are in the range $[1 - \lambda_{max}, 1 - \lambda_{min}]$. Please try to prove these facts about eigenvalues yourself for practice, though you will not need to prove these to us in the oral presentation.

For the following parts, assume that all eigenvalues of A are in the range $[.9, 1.1]$; such an A is called *well-conditioned*. (Although you need not use this fact: is such a matrix invertible, i.e., does A^{-1} exist?)

(e) Show that $\|A(x^{(i+1)} - x^*)\| \leq \frac{1}{2}\|A(x^{(i)} - x^*)\|$.

(f) Prove that there exists a constant c such that for any $\epsilon \in (0,1)$, if $t \geq c\log(1/\epsilon))$, then

$$\|A(x^{(t)} - x^*)\|^2 \leq \epsilon\|b\|^2.$$

(g) Assuming that A has m non-zero entries, what is the overall running time of the algorithm for outputting an $x^{(t)}$ for which $\|Ax^{(t)} - b\|^2 \leq \|Ax^* - b\|^2 + \epsilon\|b\|^2$? Give an answer in terms of m, n, ϵ. Assume the non-zero entries of A are represented in such a way so that for any vector z, $A \cdot z$ can be computed in $O(m)$ time.