

Privacy in Statistics and Machine Learning
In-class Exercises for Lecture 17 (Multiplicative Weights)
March 28, 2023

Spring 2023

Adam Smith (based on materials developed with Jonathan Ullman)

Problems with marked with an asterisk () are more challenging or open-ended.*

1. **(Online Learning Requires Randomization)** Show that for every method \mathcal{D} that plays deterministic actions (where \mathbf{p}^t puts probability 1 on a single action) there is an adversary for which \mathcal{D} 's average regret is $\Omega(1)$.
2. **($\sqrt{\ln(k)/T}$ lower bound)** Show that for any method (randomized or not), if the adversary picks cost vectors uniformly at random in $\{0, 1\}^k$, the expected regret will be $\Theta(\sqrt{\ln(k)/T})$. (*Hint:* The expected cost paid by the algorithm is exactly $T/2$. Show that in hindsight, with probability at least $1/2$, one of the choices will have cost less than $\frac{T}{2} - \Omega(\sqrt{T \ln(k)})$. Start by showing $\frac{T}{2} - \Omega(\sqrt{T})$.)
This means that the guarantee obtained by multiplicative weights is tight, in general.
3. **(MW with a perfect action)** Suppose we know ahead of time that there is a perfect choice a^* that always has cost 0. Show that we can set η so that the algorithm achieves total cost at most $2 \ln(k)$. (Be careful: our proof required η to be at most $1/2$.)
4. **(*)** Show that if we know OPT ahead of time, we can set η to get an average cost of at most $\frac{OPT}{T} + \frac{2}{T} \sqrt{\ln(k) \cdot \max(OPT, \ln(k))}$.
5. Theorem 3.1 requires us to know the number of steps T ahead of time. Show that one can modify the algorithm to adapt automatically to the length of the process. Specifically, there is a standard trick known as “repeated doubling”: we start the algorithm assuming we will run for $T_0 = 4 \ln(k)$ steps. If the number of steps exceeds T_0 , we restart the algorithm assuming a length of $T_1 = 2T_0$. If the number of steps exceeds $T_0 + T_1$, we expand our time horizon to $T_2 = 2T_1$, and so on. Show that this variation achieves average regret $O(\sqrt{\ln(k)/T})$ of T (without knowing T).
6. How important is the fact that probabilities of higher-cost actions be selected with exponentially small probability? Consider an algorithm that, at each time t , selects action a with probability that scales polynomially in its cost so far $c_a^{<t}$. Specifically, suppose $p_a^t \propto \frac{1}{1+c_a^{<t}}$. Show a sequence of cost vectors on which the algorithm has expected average regret at least $\Omega(k/T)$ (one can prove a stronger bound; but even this bound highlights the bad dependency on k).