CS7545, Spring 2024: Machine Learning Theory - Homework #3

Due: April 5th, 2024

1) **The Doubling Trick.** In class, we proved the regret of OGD. In the last step of the proof, we showed that the regret is upper bounded by (Last line of Step 2 in the proof):

$$R_T^{\text{POGD}} \leq \frac{\eta G^2 T}{2} + \frac{D^2}{2\eta}.$$

By optimally tuning η as $\eta = \frac{D}{G\sqrt{T}}$, we obtain a bound of the form $DG\sqrt{T}$. The problem with this approach is that it requires us to know T in advance. Is there a way around this issue?

Imagine constructing a modified algorithm $\mathcal{A}'$ that does the following iterative procedure. $\mathcal{A}'$ starts with an initial parameter η_1, implements the P-OGD algorithm using this step size for T_1 rounds, then adjusts the parameter to η_2, and runs P-OGD for T_2 rounds, and so forth. Let's say η gets updated k times, where $T_1 + T_2 + \ldots + T_k = T$.

Can you construct a schedule for the sequence of (η_i, T_i) that achieves the same *order* of bound as the optimally tuned bound (namely, $O(DG\sqrt{T})$), even though T is unknown in advance? In other words, you want to choose the sequence of $T_1, T_2, \ldots$, with the associated $\eta_1, \eta_2, \ldots$ so that whenever the online learning procedure truly ends, at a previously unknown T, the bound $R_T^{\mathcal{A}} = O(\sqrt{MT})$ will always hold. (Hint: this is referred to as a "doubling trick".)

2) **Dynamic Regret.** In Lecture 16, we learned how to get the *regret* guarantee of convex functions by P-OGD. Note that in regret, the compatetor $\mathbf{w}^*$ is fixed. In this problem, we consider a generalized notation of regret: dynamic regret, which is defined as

$$\text{Regret}_T := \sum_{t=1}^{T} f_t(\mathbf{w}_t) - \sum_{t=1}^{T} f_t(\mathbf{w}_t^*)$$

where $\mathbf{w}_t^* \in K$ and $\sum_{t=2}^{T} \|\mathbf{w}_t^* - \mathbf{w}_{t-1}^*\|_2 \leq P_T$. For convenience we assume $0 \in K$. Can you get the regret guarantee in terms of G, D, T, and P_T? What is the optimal η in this scenario?

Hint: 1) you can start from

$$\text{Regret}_T \leq \sum_{t=1}^{T} \frac{\eta}{2} G^2 + \sum_{t=1}^{T} \frac{\|\mathbf{w}_t - \mathbf{w}_t^*\|_2^2 - \|\mathbf{w}_{t+1} - \mathbf{w}_t^*\|_2^2}{2\eta},$$

and consider how to use the definitions of D and P_T to finish the telescoping sum. 2) When tuning η in the final step, consider the upper bound as a function of η, and then try to find the optimal configuration.

3) **Tuning Parameters.** We are going to imagine we have some algorithm $\mathcal{A}$ with a performance bound that depends on some input values (which can not be adjusted) and some tuning parameters (which can be optimized). We will use greek letters (α, η, ζ, etc.) for the tuning parameters and capital letters (T, D, N,

etc.) for inputs. We would like the bound to be the tightest possible, up to multiplicative constants. For each of the following, tune the parameters to obtain the optimal bound. Using *big-Oh* notation is fine to hide constants, but you *must not* ignore the dependence on the input parameters. For example, assuming $M, T > 0$, imagine we have a performance guarantee of the form:

$$\text{Performance}(\mathcal{A}; M, T, \epsilon) \leq M\epsilon + \frac{T}{\epsilon} \tag{1}$$

and we know $\epsilon > 0$. Then by optimizing the above expression with respect to the free parameter we can set $\epsilon = \sqrt{\frac{T}{M}}$. With this value we obtain Performance$(\mathcal{A}; M, T, \epsilon) = O(\sqrt{MT})$

NOTE: I didn't have to make up this problem, I actually pulled all the bounds below from different papers!

(a) Performance$(\mathcal{A}; T, \eta) \leq \max(T\eta, \eta^{-2})$

(b) Performance$(\mathcal{A}; T, \eta) \leq \frac{T}{\eta} + \exp(\eta)$. (Note: you needn't obtain the optimal choice of η here or the tightest possible bound, but try to tune in order to get a bound that is $o(T)$ – i.e. the bound should grow strictly slower than linear in T.)

(c) Performance$(\mathcal{A}; N, T, \eta, \epsilon) \leq \frac{T\epsilon}{\eta} + \frac{N}{\epsilon} + T\eta$

(d) Performance$(\mathcal{A}; N, T, \eta, \epsilon) \leq \frac{\log N}{\eta} + \frac{\eta T}{\epsilon^2} + \epsilon T$.

(e) Performance$(\mathcal{A}; T, N, \eta) \leq \frac{\log N + \eta T}{1 - \exp(-\eta)}$

4) **Online Non-Convex Optimization.** Sometimes our nice assumptions don't always hold. But maybe things will still work out just fine. For the rest of this problem assume that $X \subset \mathbb{R}^n$ is the learner's decision set, and the learner observes a sequence of functions $f_1, f_2, \ldots, f_T$ mapping $X \to \mathbb{R}$. The regret of an algorithm choosing a sequence of $\mathbf{x}_1, \mathbf{x}_2, \ldots$ is defined in the usual way:

$$\text{Regret}_T := \sum_{t=1}^{T} f_t(\mathbf{x}_t) - \min_{\mathbf{x} \in X} \sum_{t=1}^{T} f_t(\mathbf{x})$$

Wouldn't it ruin your lovely day if the functions f_t were not convex? Maybe the only two conditions you can guarantee is that the functions f_t are bounded (say in $[0, 1]$) and are 1-Lipschitz: they satisfy that $|f_t(\mathbf{x}) - f_t(\mathbf{x}')| \leq \|\mathbf{x} - \mathbf{x}'\|_2$. Prove that, assuming X is convex and bounded, there exists a randomized algorithm with a reasonable expected-regret bound. Something like $\mathbb{E}[\text{Regret}_T] \leq \sqrt{nT \log T}$ would be admirable. (Hint: Always good to ask the experts for ideas. And you needn't worry about efficiency.)