CS7545, Spring 2023: Machine Learning Theory - Solutions #3

Jacob Abernethy, Zihao Hu, and Guanghui Wang Due: Tuesday, March 19 at 11:59 p.m.

1) **Parameter Tuning.**

(a) Write the bound as
$$\frac{\log N}{1-\exp(-\eta)}+\frac{\eta T}{1-\exp(-\eta)}.$$
The first term is a decreasing function of η, and the second term is an increasing function of η (To verify, take the derivative $\frac{e^n(e^n-n-1)}{(e^n-1)^2}>0, \forall n>0$). Since $T >> \log N$, the optimal value for η must be small, say less than 1.

Note that for $\eta \in (0,1)$,
$$\frac{\eta}{1-e^-\eta} \leq (\eta+1)$$
and
$$(1-\exp(-\eta))^{-1} \leq 2/\eta$$
Using the above inequalities, the bound becomes
$$\frac{\log N}{1-\exp(-\eta)}+\frac{\eta T}{1-\exp(-\eta)} \leq \frac{2\log N}{\eta}+(1+\eta)T=\left(\frac{2\log N}{\eta}+\eta T\right)+T$$
Take $\eta \leftarrow \sqrt{\frac{2\log N}{T}}$ and we have an upper bound $O(\sqrt{2T\log N}+T)$.

(b) The minimum occurs when $T\eta=\eta^{-2}$. So, $\eta=T^{-1/3}$. So, the upper bound is $O(T^{-2/3})$.

(c) Due to the exponential term, η must have the form $\log f(T)$ for some sub-linear function f of T. Furthermore, the first term $\frac{T}{\eta}$ requires that f is an increasing function of T. For example, we can choose $\eta=\log\sqrt{T}$, and the upper bound becomes $\frac{T}{\log\sqrt{T}}+\sqrt{T}$, which is sublinear.

(d) At the minimum, all three terms are equal. So, we want $\frac{T\epsilon}{\eta}=T\eta$, and $\frac{T\epsilon}{\eta}=\frac{N}{\epsilon}$

The first condition implies $\epsilon=\eta^2$. The second condition implies $T\epsilon^2=N\eta$, which then implies $\eta=\left(\frac{N}{T}\right)^{1/3}$ after the substitution $\epsilon=\eta^2$,

So, we get an upper bound $3T\eta=O(T\eta)=O(T^{\frac{2}{3}}N^{\frac{1}{3}})$.

2) **Doubling Trick.** Let $k=\lceil\log_2 T\rceil$. Let $T_i=2^{i-1}$ for $i=1,\ldots,k$, so that $T \leq \sum_{i=1}^k T_i$. So, the bound is (asymptotically)
$$\sum_{i=1}^k \sqrt{MT_i}=\sqrt{M\sum_{i=1}^k 2^{i-1}}=\sqrt{M}\frac{\sqrt{2}^k-1}{\sqrt{2}-1} \leq \sqrt{M}\frac{\sqrt{2T}-1}{\sqrt{2}-1}=O(\sqrt{MT})$$

3) **Dynamic Regret.** Following the proof of OGD in Lecture 16 but considering the dynamic regret, we have

$$\begin{aligned}
\text{Regret}_T &\le \sum_{t=1}^{T} \frac{\eta}{2} G^2 + \sum_{t=1}^{T} \frac{(\|\mathbf{w}_t - \mathbf{w}_t^*\|)^2 - (\|\mathbf{w}_{t+1} - \mathbf{w}_t^*\|)^2}{2\eta} \\
&\le \frac{TG^2\eta}{2} + \sum_{t=1}^{T} \frac{\left(\|\mathbf{w}_t\|^2 - 2\langle \mathbf{w}_t, \mathbf{w}_t^*\rangle + \|\mathbf{w}_t^*\|^2 - \|\mathbf{w}_{t+1}\|^2 + 2\langle \mathbf{w}_{t+1}, \mathbf{w}_t^*\rangle - \|\mathbf{w}_t^*\|^2\right)}{2\eta} \\
&\le \frac{TG^2\eta}{2} + \frac{1}{2\eta}(\|\mathbf{w}_1\|^2 - \|\mathbf{w}_{T+1}\|^2) + \frac{1}{\eta}\sum_{t=1}^{T} \langle \mathbf{w}_{t+1} - \mathbf{w}_t, \mathbf{w}_t^*\rangle \\
&\le \frac{TG^2\eta}{2} + \frac{D^2}{2\eta} + \frac{1}{\eta}(\langle \mathbf{w}_{T+1}, \mathbf{w}_T^*\rangle - \langle \mathbf{w}_1, \mathbf{w}_1^*\rangle) + \frac{1}{\eta}\sum_{t=2}^{T} \langle \mathbf{w}_{t-1}^* - \mathbf{w}_t^*, \mathbf{w}_t\rangle \\
&\le \frac{\eta G^2 T}{2} + \frac{7D^2}{4\eta} + \frac{D}{\eta}\sum_{t=2}^{T} \|\mathbf{w}_t^* - \mathbf{w}_{t-1}^*\| \\
&\le \frac{\eta G^2 T}{2} + \frac{7D^2 + 4DP_T}{4\eta}.
\end{aligned}$$

where we use the following relation

$$\begin{aligned}
\|\mathbf{w}_1\|^2 = \|\mathbf{w}_1 - \mathbf{0}\|^2 &\le D^2, \\
\mathbf{w}_{T+1}^\top \mathbf{w}_T^* \le \|\mathbf{w}_{T+1}\| \, \|\mathbf{w}_T^*\| &\le D^2, \\
-\mathbf{w}_1^\top \mathbf{w}_1^* \le \tfrac{1}{4}\|\mathbf{w}_1 - \mathbf{w}_1^*\|^2 &\le \tfrac{1}{4} D^2, \\
\langle \mathbf{w}_{t-1}^* - \mathbf{w}_t^*, \mathbf{w}_t\rangle \le \left\|\mathbf{w}_{t-1}^* - \mathbf{w}_t^*\right\| \|\mathbf{w}_t\| &\le D\left\|\mathbf{w}_{t-1}^* - \mathbf{w}_t^*\right\|.
\end{aligned}$$

4) **EWA with Prior.** The proof is essentially the same as the regular EWA. Using the same notations $W_t = \sum_i w_i^t$, $\Phi_t = -\log W_t$, and $L_T(\mathcal{A}) = \sum_{t=1}^T \ell^t(\hat{y}^t, y^t)$, we have

$$\Phi_{T+1} - \Phi_1 \ge (1 - \exp(-\eta))L_T(\mathcal{A}).$$

For any expert i,

$$\Phi_{T+1} - \Phi_1 = -\log W_{T+1} \le -\log w_i^{T+1} = -\log p_i \exp(-\eta L_T(\text{expert } i)) = -\log p_i + \eta L_T(\text{expert } i)$$

So, $(1 - \exp(-\eta))L_T(\mathcal{A}) \le \Phi_{T+1} - \Phi_1 \le -\log p_i + \eta L_T(\text{expert } i)$, which implies

$$L_T(\mathcal{A}) \le \frac{-\log p_i + \eta L_T(\text{expert } i)}{1 - \exp(-\eta)}$$

5) **Online Non-Convex Optimization.** We partition X into 2 -norm ϵ-balls. Each ϵ ball has size $O(\epsilon^n)$ and we need $N := O(1/\epsilon^n)$ of those to cover X. We treat each ball as an expert and run EWA.

EWA suffers $O(\sqrt{T\log N})$ regret with respect to the best expert. Now we need to analyze the best expert's regret with respect to the best fixed-point prediction. Let $x^* = \arg\min_{x\in X}\sum_t f_t(x)$. Then, one of the experts must satisfy $\|x - x^*\|_2 \le \epsilon$, which by Lipschitz assumption, implies $f_t(x) - f_t(x^*) \le \epsilon$ for all t. So, the best expert suffers at most $T\epsilon$ regret with respect to the best fixed-point prediction. The total regret of the algorithm is therefore upper-bounded by

$$O(\sqrt{T\log N} + T\epsilon) = O\left(\sqrt{Tn\log\frac{1}{\epsilon}} + T\epsilon\right).$$

Set $\epsilon = 1/T$, and the regret now becomes $O(\sqrt{nT\log T})$.

6) **Subsets as Experts.** Let L_t be the cumulative loss (of an expert or a hyper-expert) up to time t.

Note

$$\begin{aligned}
u_i^t &= \sum_{S \in S_k^N : i \in S} w_S^t \\
&= \sum_{S \in S_k^N : i \in S} \exp(-\eta L_t(S)) \\
&= \sum_{S \in S_k^N : i \in S} \exp(-\eta \sum_{j \in S} L_t(j)) \\
&= \sum_{S \in S_k^n : i \in S} \prod_{j \in S} \exp(-\eta L_t(j))
\end{aligned}$$

Define

$$\Phi_x^A := \sum_{S \in S_x^A} \prod_{j \in S} v_j$$

for all $x \in [k-1]$ and $A \in \{x, x+1, \ldots, n-1\}$. We will construct a lookup table for Φ. Our algorithm first computes the base cases: $\Phi_1^A = \sum_{i=1}^A v_i$ for all $A \in [n]$, and $\Phi_A^A = \prod_{j=1}^A v_j$ for all $A \in [k]$. T Then, we complete the table in $O(nk)$ time, using the following recursive formula:

$$\Phi_x^A \leftarrow v_A \Phi_{x-1}^{A-1} + \Phi_x^{A-1},$$

where the first term is the sumprod value over all elements in S_k^A that contain v_A, and the second term is the sumprod value over the rest.

Now we rewrite u_n^t as

$$u_n^t = v_n \Phi_{k-1}^{n-1} = v_n (\sum_{i=k-1}^{n-1} v_i \Phi_{k-2}^{i-1})$$