Machine Learning Theory (CS 6783)

Lecture 23: Relaxations & deriving algorithms

1 Recap

1. We saw how we can derive relaxations and algorithms for finite experts problem and for bit prediction problem
2. In general relaxations give a generic way of deriving algorithms for online learning problems with guaranteed bounds on regret.

2 Online Linear Optimization: Euclidean space

$\mathcal{F} = \{\mathbf{f} : \|\mathbf{f}\|_2 \leq 1\}$, $\mathcal{D} = \{\nabla : \|\nabla\|_2 \leq 1\}$

Step 1

$$\begin{aligned}
\mathbf{Rad}_n(x_{1:t}, y_{1:t}) &= \sup_{\boldsymbol{\nabla}} \mathbb{E}_{\epsilon_{t+1},\ldots,\epsilon_n} \sup_{\mathbf{f}\in\mathcal{F}} \left[\left\langle \mathbf{f}, 2\sum_{s=t+1}^{n} \epsilon_s \boldsymbol{\nabla}_{s-t}(\epsilon_{t+1:s-1})) - \sum_{s=1}^{t} \nabla_s \right\rangle \right] \\
&= \sup_{\boldsymbol{\nabla}} \mathbb{E}_{\epsilon_{t+1},\ldots,\epsilon_n} \left\| 2\sum_{s=t+1}^{n} \epsilon_s \boldsymbol{\nabla}_{s-t}(\epsilon_{t+1:s-1})) - \sum_{s=1}^{t} \nabla_s \right\|_2 \\
&= \sup_{\boldsymbol{\nabla}} \mathbb{E}_{\epsilon_{t+1},\ldots,\epsilon_n} \sqrt{\left\| 2\sum_{s=t+1}^{n} \epsilon_s \boldsymbol{\nabla}_{s-t}(\epsilon_{t+1:s-1})) - \sum_{s=1}^{t} \nabla_s \right\|_2^2}
\end{aligned}$$

Step 2

$$\begin{aligned}
\mathbf{Rad}_n(\nabla_{1:t}) &\leq \sup_{\boldsymbol{\nabla}} \sqrt{\mathbb{E}_{\epsilon_{t+1},\ldots,\epsilon_n} \left\| 2\sum_{s=t+1}^{n} \epsilon_s \boldsymbol{\nabla}_{s-t}(\epsilon_{t+1:s-1})) - \sum_{s=1}^{t} \nabla_s \right\|_2^2} \\
&= \sup_{\boldsymbol{\nabla}} \sqrt{\left\| \sum_{s=1}^{t} \nabla_s \right\|_2^2 + \mathbb{E}_{\epsilon_{t+1},\ldots,\epsilon_n} [\text{Cross terms}] + 4\mathbb{E}_{\epsilon_{t+1},\ldots,\epsilon_n} \left[\sum_{s=t+1}^{n} \|\boldsymbol{\nabla}_{s-t}(\epsilon_{t+1:s-1}))\|_2^2 \right]} \\
&= \sup_{\boldsymbol{\nabla}} \sqrt{\left\| \sum_{s=1}^{t} \nabla_s \right\|_2^2 + 4\mathbb{E}_{\epsilon_{t+1},\ldots,\epsilon_n} \left[\sum_{s=t+1}^{n} \|\boldsymbol{\nabla}_{s-t}(\epsilon_{t+1:s-1}))\|_2^2 \right]} \\
&\leq \sqrt{\left\| \sum_{s=1}^{t} \nabla_s \right\|_2^2 + 4(n-t)} =: \mathbf{Rel}_n(\nabla_{1:t})
\end{aligned}$$

Step 3 & 4

$$\inf_{\mathbf{f}_t}\sup_{\nabla_t}\left\{\langle \mathbf{f}_t,\nabla_t\rangle + \mathbf{Rel}_n\left(\nabla_{1:t}\right)\right\} = \inf_{\mathbf{f}_t}\sup_{\nabla_t}\left\{\langle \mathbf{f}_t,\nabla_t\rangle + \sqrt{\left\|\sum_{s=1}^{t}\nabla_s\right\|_2^2 + 4(n-t)}\right\}$$

$$= \inf_{\mathbf{f}_t}\sup_{\nabla_t}\left\{\langle \mathbf{f}_t,\nabla_t\rangle + \sqrt{\left\|\sum_{s=1}^{t-1}\nabla_s\right\|_2^2 + 2\left\langle\sum_{s=1}^{t-1}\nabla_s,\nabla_t\right\rangle + \|\nabla_t\|_2^2 + 4(n-t)}\right\}$$

$$\leq \inf_{\mathbf{f}_t}\sup_{\nabla_t}\left\{\langle \mathbf{f}_t,\nabla_t\rangle + \sqrt{\left\|\sum_{s=1}^{t-1}\nabla_s\right\|_2^2 + 2\left\langle\sum_{s=1}^{t-1}\nabla_s,\nabla_t\right\rangle + 4(n-t+1)}\right\}$$

Now in the above note that the second term depends on ∇_t only through $\left\langle\sum_{s=1}^{t-1}\nabla_s,\nabla_t\right\rangle$. This means that if $\mathbf{f}_t$ has any component orthogonal to $\sum_{s=1}^{t-1}\nabla_s$ then ∇_t can gain on the first term without loosing on the second term (as the component of ∇_t that increases first term is perpendicular to the second term). Hence $\mathbf{f}_t$ has to be of form $\mathbf{f}_t = -\alpha\sum_{s=1}^{t-1}\nabla_s$ for some positive α. Hence

$$\inf_{\mathbf{f}_t}\sup_{\nabla_t}\left\{\langle \mathbf{f}_t,\nabla_t\rangle + \mathbf{Rel}_n\left(\nabla_{1:t}\right)\right\}$$

$$= \inf_{\alpha>0}\sup_{\nabla_t}\left\{-\alpha\underbrace{\left\langle\sum_{s=1}^{t-1}\nabla_s,\nabla_t\right\rangle}_{\beta} + \sqrt{\left\|\sum_{s=1}^{t-1}\nabla_s\right\|_2^2 + 2\left\langle\sum_{s=1}^{t-1}\nabla_s,\nabla_t\right\rangle + 4(n-t+1)}\right\}$$

$$\leq \inf_{\alpha>0}\sup_{\beta\in\mathbb{R}}\left\{-\alpha\beta + \sqrt{\left\|\sum_{s=1}^{t-1}\nabla_s\right\|_2^2 + 2\beta + 4(n-t+1)}\right\}$$

Taking derivative to optimize over β for a given α we see that β is optimized when,

$$-\alpha + \frac{1}{\sqrt{\left\|\sum_{s=1}^{t-1}\nabla_s\right\|_2^2 + 2\beta + 4(n-t+1)}} = 0$$

Hence if we use

$$\alpha = \frac{1}{\sqrt{\left\|\sum_{s=1}^{t-1}\nabla_s\right\|_2^2 + 4(n-t+1)}}$$

then clearly the corresponding β that maximizes is at $\beta = 0$. Hence,

$$\inf_{\mathbf{f}_t}\sup_{\nabla_t}\left\{\langle \mathbf{f}_t,\nabla_t\rangle + \mathbf{Rel}_n\left(\nabla_{1:t}\right)\right\} \leq \sup_{\beta\in\mathbb{R}}\left\{-\frac{\beta}{\sqrt{\left\|\sum_{s=1}^{t-1}\nabla_s\right\|_2^2 + 4(n-t+1)}} + \sqrt{\left\|\sum_{s=1}^{t-1}\nabla_s\right\|_2^2 + 2\beta + 4(n-t+1)}\right\}$$

$$\leq \sqrt{\left\|\sum_{s=1}^{t-1}\nabla_s\right\|_2^2 + 4(n-t+1)} = \mathbf{Rel}_n\left(\nabla_{1:t-1}\right)$$

Algorithm is given by

$$\mathbf{f}_t = -\alpha \sum_{s=1}^{t-1} \nabla_s = -\frac{\sum_{s=1}^{t-1} \nabla_s}{\sqrt{\left\|\sum_{s=1}^{t-1} \nabla_s\right\|_2^2 + 4(n-t+1)}}$$

Notice that we don't need any projection, the solutions automatically have norm at most 1. The final guarantee we get is

$$\mathbb{E}\left[\mathrm{Reg}_n\right] \leq \frac{1}{n}\mathbf{Rel}_n\left(\cdot\right) = \frac{1}{n}\sqrt{4n} = \frac{2}{\sqrt{n}}$$

This gives an alternative for gradient descent and can be used for online convex optimization. For other norms, as long as the dual norm squared is a strongly-smooth function (or equivalently the norm squared is a strongly convex function) the same technique can be used where the equality due to Pythagorus theorem in the proof is replaced by inequality due to strong smoothness of norm squared. This can also be viewed as a modified, projection free form of gradient descent with automatically tuned step-sizes. The key thing to note is that the step size depends on past gradients and so if sequence is nicer, we take stronger steps.

3 Other Algorithms

3.1 Follow the Regularized Leader

We have arbitrary convex set $\mathcal{F}$, and set of gradients $\mathcal{D}$ such that $\sup_{\nabla\in\mathcal{D}} \|\nabla\|_* \leq 1$ where $\|\cdot\|_*$ is the dual of some norm $\|\cdot\|$. Assume that $\mathbf{R}$ is function that is 1-strongly convex w.r.t. norm $\|\cdot\|$. Let $R = \sqrt{\sup_{\mathbf{f}\in\mathcal{F}} \mathbf{R}(f)}$.
Fact : If $\mathbf{R}$ is strongly convex, its Fenchel conjugate $\mathbf{R}^*$ is strongly smooth w.r.t. dual norm, ie,

$$\mathbf{R}^*(x) \leq \mathbf{R}^*(x') + \left\langle \nabla\mathbf{R}^*(x'), x - x' \right\rangle + \frac{1}{2}\left\|x - x'\right\|^2$$

Steps 1 & 2

$$\begin{aligned}
\mathbf{Rad}_n\left(x_{1:t}, y_{1:t}\right) &= \inf_{\lambda>0}\sup_{\boldsymbol{\nabla}} \mathbb{E}_{\epsilon_{t+1},\ldots,\epsilon_n} \sup_{\mathbf{f}\in\mathcal{F}} \left[\left\langle \mathbf{f}, 2\sum_{s=t+1}^{n} \epsilon_s \boldsymbol{\nabla}_{s-t}(\epsilon_{t+1:s-1})) - \sum_{s=1}^{t}\nabla_s \right\rangle\right] \\
&\leq \inf_{\lambda>0}\sup_{\boldsymbol{\nabla}} \left\{ \frac{1}{\lambda}\sup_{f\in\mathcal{F}} R(f) + \frac{1}{\lambda}\mathbb{E}_{\epsilon_{t+1},\ldots,\epsilon_n}\mathbf{R}^*\left(2\lambda\sum_{s=t+1}^{n}\epsilon_s\boldsymbol{\nabla}_{s-t}(\epsilon_{t+1:s-1}) - \lambda\sum_{s=1}^{t}\nabla_s\right)\right\} \\
* \;&\leq \inf_{\lambda>0}\sup_{\boldsymbol{\nabla}} \left\{ \frac{1}{\lambda}\sup_{f\in\mathcal{F}} \mathbf{R}(f) + \frac{1}{\lambda}\mathbb{E}_{\epsilon_{t+1},\ldots,\epsilon_n}\left[R^*\left(2\lambda\sum_{s=t+1}^{n-1}\epsilon_s\boldsymbol{\nabla}_{s-t}(\epsilon_{t+1:s-1}) - \lambda\sum_{s=1}^{t}\nabla_s\right)\right.\right. \\
&\quad \left.\left. + \left\langle \nabla\mathbf{R}^*\left(2\lambda\sum_{s=t+1}^{n-1}\epsilon_s\boldsymbol{\nabla}_{s-t}(\epsilon_{t+1:s-1}) - \lambda\sum_{s=1}^{t}\nabla_s\right), 2\lambda\epsilon_n\boldsymbol{\nabla}_{n-t}(\epsilon_{t+1:n-1})\right\rangle + 2\lambda^2\left\|\boldsymbol{\nabla}_{n-t}(\epsilon_{t+1:n-1})\right\|_*^2\right]\right\} \\
** \;&\leq \inf_{\lambda>0}\sup_{\boldsymbol{\nabla}} \left\{ \frac{R^2}{\lambda} + \frac{1}{\lambda}\mathbb{E}_{\epsilon_{t+1},\ldots,\epsilon_{n-1}}\left[\mathbf{R}^*\left(2\lambda\sum_{s=t+1}^{n-1}\epsilon_s\boldsymbol{\nabla}_{s-t}(\epsilon_{t+1:s-1}) - \lambda\sum_{s=1}^{t}\nabla_s\right) + 2\lambda^2\right]\right\}
\end{aligned}$$

$$\leq \ldots \leq \inf_{\lambda>0}\left\{\frac{R^2}{\lambda}+\frac{1}{\lambda}\mathbf{R}^*\left(-\lambda\sum_{s=1}^{t}\nabla_s\right)+2\lambda(n-t)\right\}$$
$$=\inf_{\lambda>0}\left\{\frac{R^2}{\lambda}-\inf_{f\in\mathcal{F}}\left\{\sum_{s=1}^{t}\langle f,\nabla_s\rangle+\frac{1}{\lambda}\mathbf{R}(f)\right\}+2\lambda(n-t)\right\}=:\mathbf{Rel}_n\left(\nabla_{1:t}\right)$$

Step 3 One step of Symmetrization + $*$ and $**$ (for time index t)
Step 4 Algorithm :

$$\hat{\mathbf{y}}_t=\operatorname*{argmin}_{\mathbf{f}\in\mathcal{F}}\sum_{i=1}^{t-1}\langle\mathbf{f},\nabla_i\rangle+\frac{1}{\lambda_t^*}\mathbf{R}(\mathbf{f})$$

where $\lambda^*=\operatorname*{argmin}_{\lambda>0}\left\{\frac{R^2}{\lambda}+\frac{1}{\lambda}\mathbf{R}^*\left(-\lambda\sum_{s=1}^{t-1}\nabla_s\right)+2\lambda(n-t+1)\right\}$. This algorithm is also known as Follow The Regularized Leader algorithm where $\mathbf{R}$ is the regularizer. Of course in the one above the regularization parameter $\frac{1}{\lambda_t^*}$ is auto tuned for each round and is dependent on past sequence which we were able to derive from the relaxations.
Bound :

$$\mathrm{Reg}_n\leq\frac{1}{n}\inf_{\lambda}\left\{\frac{R^2}{\lambda}+2\lambda n\right\}\leq\sqrt{\frac{8R^2}{n}}$$

3.2 Mirror Descent, Online Newton's methods etc.

Mirror descent algorithm can be derived in a manner very similar to the above FTRL approach. Only $\mathbf{R}$ is replaces by Bregman divergence, $\Delta_{\mathbf{R}}(\mathbf{f}|\nabla\mathbf{R}(\hat{\mathbf{y}}_t)-\eta\nabla_t)$. The relaxation is

$$\mathbf{Rel}_n\left(\nabla_{1:t}\right)=\inf_{\eta>0}\left\{\sup_{\mathbf{f}\in\mathcal{F}}\left\{\sum_{i=1}^{t}\langle\mathbf{f},-\nabla_i\rangle+\frac{1}{\eta}\Delta_{\mathbf{R}}(\mathbf{f}|\nabla\mathbf{R}(\hat{\mathbf{y}}_t)-\eta\nabla_t)\right\}+2\eta(n-t)\right\}$$

(at the n^{th} step we put an arbitrary $\hat{\mathbf{y}}_n$ then we see that the admissibility step essentially tells us how to update). Admissibility is basically MD proof and algorithm is MD with adaptive step size. Mirror descent for strongly convex objective Online newton's step for exp-concave losses have very similar derivations. Online Newton's step we start with the dampened notion of sequential Rademacher process from the assignment 3.

4 Online Binary Classification

Step 1

$$\mathbf{Rad}_n\left(x_{1:t},y_{1:t}\right)=\sup_{\mathbf{x}}\mathbb{E}_\epsilon\sup_{f\in\mathcal{F}}\left\{\sum_{i=1}^{n-t}\epsilon_i f(\mathbf{x}_i(\epsilon))-\sum_{i=1}^{t}\mathbb{1}_{\{f(x_i)\neq y_i\}}\right\}$$
$$=\sup_{\mathbf{x}}\mathbb{E}_\epsilon\max_{(\sigma,\omega):(\sigma,\omega)\in\mathcal{F}|_{(x_{1:t},\mathbf{x}(\epsilon))}}\left\{\sum_{i=1}^{n-t}\epsilon_i\omega_i-\sum_{i=1}^{t}\mathbb{1}_{\{\sigma_i\neq y_i\}}\right\}$$

Step 2

$$\mathbf{Rad}_n\left(x_{1:t}, y_{1:t}\right) \leq \sup_{\mathbf{x}} \mathbb{E}_\epsilon \max_{\sigma \in \mathcal{F}|_{x_{1:t}}} \max_{\mathbf{v} \in V(\mathcal{F}_t(\sigma), \mathbf{x})} \left\{ \sum_{i=1}^{n-t} \epsilon_i \mathbf{v}_i(\epsilon) - \sum_{i=1}^{t} \mathbb{1}_{\{\sigma_i \neq y_i\}} \right\}$$

where $\mathcal{F}|_{(x_{1:t}, \mathbf{x}(\epsilon))}$ is the projection of $\mathcal{F}$ onto $(x_{1:t}, \mathbf{x}(\epsilon))$, $\mathcal{F}_t(\sigma) = \{f \in \mathcal{F} : f(x_{1:t}) = \sigma\}$, and $V(\mathcal{F}_t(\sigma), \mathbf{x})$ is the smallest zero-cover of the set $\mathcal{F}_t(\sigma)$ on the tree $\mathbf{x}$. Using soft-max:

$$\leq \inf_{\lambda} \frac{1}{\lambda} \log \left(\sup_{\mathbf{x}} \mathbb{E}_\epsilon \sum_{\sigma : \sigma \in \mathcal{F}|_{x_{1:t}}} \sum_{\mathbf{v} \in V(\mathcal{F}_t(\sigma), \mathbf{x})} \exp \left\{ \lambda \sum_{i=1}^{n-t} \epsilon_i \mathbf{v}_i(\epsilon) - \lambda L_t(\sigma) \right\} \right)$$

where $L_t(\sigma) = \sum_{i=1}^{t} \mathbb{1}_{\{\sigma_i \neq y_i\}}$

$$\leq \frac{1}{\lambda} \log \left(\sup_{\mathbf{x}} \sum_{\sigma \in \mathcal{F}|_{x_{1:t}}} \exp\left\{-\lambda L_t(\sigma)\right\} \mathbb{E}_\epsilon \sum_{\mathbf{v} \in V(\mathcal{F}_t(\sigma), \mathbf{x})} \exp \left\{ \lambda \sum_{i=1}^{n-t} \epsilon_i \mathbf{v}_i(\epsilon) \right\} \right)$$

$$\leq \frac{1}{\lambda} \log \left(\sup_{\mathbf{x}} \sum_{\sigma \in \mathcal{F}|_{x_{1:t}}} \exp\left\{-\lambda L_t(\sigma)\right\} \ \left|V(\mathcal{F}_t(\sigma), \mathbf{x})\right| \ \exp\left\{\lambda^2 (n-t)/2\right\} \right)$$

Remember the Littleton dimension, we have, $|V(\mathcal{F}_t(\sigma), \mathbf{x})| \leq g(\mathrm{Ldim}(\mathcal{F}_t(\sigma)), n-t) = \sum_{i=1}^{\mathrm{Ldim}(\mathcal{F}_t(\sigma))} \binom{n-t}{i}$

$$\leq \frac{1}{\lambda} \log \left(\sum_{\sigma \in \mathcal{F}|_{x_{1:t}}} g(\mathrm{Ldim}(\mathcal{F}_t(\sigma)), n-t) \exp\left\{-\lambda L_t(\sigma)\right\} \right) + \lambda (n-t)/2$$
$$=: \mathbf{Rel}_n\left(x_{1:t}, y_{1:t}\right)$$

Steps. 3 & 4 Let us do step 4 first. If the relaxation was admissible then the algorithm for binary classification would be,

$$q_t^*(x_t) = \frac{1}{2} + \frac{1}{2} \left(\mathbf{Rel}_n\left(x_{1:t}, y_{1:t-1}, +1\right) - \mathbf{Rel}_n\left(x_{1:t}, y_{1:t-1}, -1\right) \right)$$

However if the relaxation were admissible then plugging in the above solution, the admissibility condition becomes,

$$\begin{aligned}
\mathbf{Rel}_n\left(x_{1:t-1}, y_{1:t-1}\right) &\geq \sup_{x_t} \max_{y_t \in \{\pm 1\}} \left\{ q_t^*(x_t) \, \mathbb{1}_{\{y_t \neq 1\}} + (1 - q_t^*(x_t)) \, \mathbb{1}_{\{y_t \neq -1\}} + \mathbf{Rel}_n\left(x_{1:t}, y_{1:t-1}, y_t\right) \right\} \\
&= \sup_{x_t} \max \left\{ q_t^*(x_t) + \mathbf{Rel}_n\left(x_{1:t}, y_{1:t-1}, -1\right), (1 - q_t^*(x_t)) + \mathbf{Rel}_n\left(x_{1:t}, y_{1:t-1}, +1\right) \right\} \\
&= \sup_{x_t} \frac{\mathbf{Rel}_n\left(x_{1:t}, y_{1:t-1}, +1\right) + \mathbf{Rel}_n\left(x_{1:t}, y_{1:t-1}, -1\right)}{2} \\
&= \sup_{x_t} \mathbb{E}_{\epsilon_t} \left[\mathbf{Rel}_n\left(x_{1:t}, y_{1:t-1}, \epsilon_t\right) \right]
\end{aligned}$$

So we need to show the above inequality. Now note that

$$
\begin{aligned}
&\sup_{x_t}\mathbb{E}_{\epsilon_t}\left[\mathbf{Rel}_n\left(x_{1:t},y_{1:t-1},\epsilon_t\right)\right]\\
&\quad\le\frac{1}{\lambda}\sup_{x_t}\mathbb{E}_{\epsilon_t}\left[\log\left(\sum_{\sigma\in\mathcal{F}|_{x_{1:t-1}}}\sum_{\sigma_t\in\{\pm1\}}g(\mathrm{Ldim}(\mathcal{F}_t(\sigma,\sigma_t)),n-t)\exp\left(-\lambda L_{t-1}(\sigma)-\lambda\,\mathbb{1}_{\{\sigma_t\neq\epsilon_t\}}\right)\right)\right]+\lambda(n-t)/2\\
&\quad\le\frac{1}{\lambda}\sup_{x_t}\log\left(\sum_{\sigma\in\mathcal{F}|_{x_{1:t-1}}}\sum_{\sigma_t\in\{\pm1\}}g(\mathrm{Ldim}(\mathcal{F}_t(\sigma,\sigma_t)),n-t)\mathbb{E}_{\epsilon_t}\left[\exp\left(-\lambda L_{t-1}(\sigma)-\lambda\,\mathbb{1}_{\{\sigma_t\neq\epsilon_t\}}\right)\right]\right)+\lambda(n-t)/2\\
&\quad=\frac{1}{\lambda}\sup_{x_t}\log\left(\sum_{\sigma\in\mathcal{F}|_{x_{1:t-1}}}\sum_{\sigma_t\in\{\pm1\}}g(\mathrm{Ldim}(\mathcal{F}_t(\sigma,\sigma_t)),n-t)\exp\left(-\lambda L_{t-1}(\sigma)\right)\mathbb{E}_{\epsilon_t}\left[e^{-\lambda\,\mathbb{1}_{\{\sigma_t\neq\epsilon_t\}}}\right]\right)+\lambda(n-t)/2\\
&\quad\le\frac{1}{\lambda}\sup_{x_t}\log\left(\sum_{\sigma\in\mathcal{F}|_{x_{1:t-1}}}\sum_{\sigma_t\in\{\pm1\}}g(\mathrm{Ldim}(\mathcal{F}_t(\sigma,\sigma_t)),n-t)\exp\left(-\lambda L_{t-1}(\sigma)\right)\right)+\lambda(n-t+1)/2\\
&\quad\le\frac{1}{\lambda}\log\left(\sum_{\sigma\in\mathcal{F}|_{x_{1:t-1}}}\left(\sup_{x_t}\sum_{\sigma_t\in\{\pm1\}}g(\mathrm{Ldim}(\mathcal{F}_t(\sigma,\sigma_t)),n-t)\right)\exp\left(-\lambda L_{t-1}(\sigma)\right)\right)+\lambda(n-t+1)/2
\end{aligned}
$$

Now note that $\mathcal{F}_{t-1}(\sigma)=\mathcal{F}_t(\sigma,+1)\bigcup\mathcal{F}_{t-1}(\sigma,-1)$ and note that at most one of $\mathcal{F}_t(\sigma,+1)$ or $\mathcal{F}_t(\sigma,-1)$ can have Littleton dimension of $\mathrm{Ldim}(\mathcal{F}_{t-1}(\sigma))$. Because if not, then naming x_t as the root of the new tree, a tree of depth $\mathrm{Ldim}(\mathcal{F}_{t-1}(\sigma))+1$ can be made that is shattered by $\mathcal{F}_{t-1}(\sigma)$ which is a contradiction. Hence,

$$
\sup_{x_t}\sum_{\sigma_t\in\{\pm1\}}g(\mathrm{Ldim}(\mathcal{F}_t(\sigma,\sigma_t)),n-t)\le g(\mathrm{Ldim}(\mathcal{F}_{t-1}),n-t)+g(\mathrm{Ldim}(\mathcal{F}_{t-1})-1,n-t)\le g(\mathrm{Ldim}(\mathcal{F}_{t-1}),n-t+1)
$$

Hence we conclude that for all $x_t\in\mathcal{X}$,

$$
\begin{aligned}
\mathbb{E}_{\epsilon_t}\left[\mathbf{Rel}_n\left(x_{1:t},y_{1:t-1},\epsilon_t\right)\right]&\le\frac{1}{\lambda}\log\left(\sum_{\sigma\in\mathcal{F}|_{x_{1:t-1}}}g(\mathrm{Ldim}(\mathcal{F}_{t-1}),n-t+1)\exp\left(-\lambda L_{t-1}(\sigma)\right)\right)+\lambda(n-t+1)/2\\
&=\mathbf{Rel}_n\left(x_{1:t-1},y_{1:t-1}\right)
\end{aligned}
$$

This concludes admissibility.

Bound :

$$
\mathbb{E}\left[\mathrm{Reg}_n\right]\le\frac{1}{n}\inf_{\lambda}\left\{\frac{1}{\lambda}\log\left(g(\mathrm{Ldim}(\mathcal{F}),n)\right)+\tfrac{\lambda}{2}n\right\}=\sqrt{\frac{2\log\left(g(\mathrm{Ldim}(\mathcal{F}),n)\right)}{n}}\le\sqrt{\frac{2\mathrm{Ldim}(\mathcal{F})\log\left(n\right)}{n}}
$$

5 Remarks

1. Just like for online binary classification, there is a generic for for q_t based only on relaxation as long as it is admissible, similar result can also be obtained for multi-class version only q_t is not closed form but rather got by solving a simple convex program based on the relaxations.
2. Checking for admissibility in the binary classification case again was simplified into checking for inequality involving only the relaxations, similar result can be extended to multi-class case.