Machine Learning Theory (CS 6783)

Lecture 22: Relaxations & deriving algorithms

1 Recap

1. $\mathbf{Rel}_n : \bigcup_{t=0}^{n} \mathcal{X}^t \times \mathcal{Y}^t \mapsto \mathbb{R}$ is admissible if

$$\mathbf{Rel}_n\left(x_{1:n}, y_{1:n}\right) \geq -\inf_{f \in \mathcal{F}} \sum_{t=1}^{n} \ell(f(x_t), y_t)$$

and $$\sup_{x_t} \inf_{q_t} \sup_{y_t} \left\{\mathbb{E}_{\hat{y}_t \sim q_t}\left[\ell(\hat{y}_t, y_t)\right] + \mathbf{Rel}_n\left(x_{1:t}, y_{1:t}\right)\right\} \leq \mathbf{Rel}_n\left(x_{1:t-1}, y_{1:t-1}\right)$$

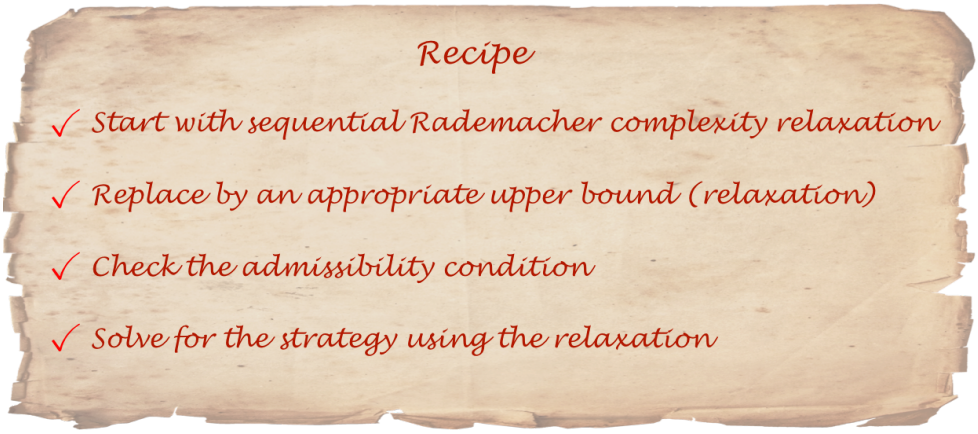
2. Algorithm: $q_t = \operatorname{argmin}_{q \in \Delta(\mathcal{Y})} \sup_{y_t} \left\{\mathbb{E}_{\hat{y}_t \sim q}\left[\ell(\hat{y}_t, y_t)\right] + \mathbf{Rel}_n\left(x_{1:t}, y_{1:t}\right)\right\}$ Bound: $\mathbb{E}\left[\mathrm{Reg}_n\right] \leq \frac{1}{n}\mathbf{Rel}_n\left(\cdot\right)$

3. For Binary classification, $q_t = 0.5 + \left(\mathbf{Rel}_n\left(x_{1:t}, (y_{1:t-1}, +1)\right) - \mathbf{Rel}_n\left(x_{1:t}, (y_{1:t-1}, -1)\right)\right)/2$ and for multi-class problems a waterfilling argument works.

4. Sequential Rademacher relaxation is admissible and gives the sequential Rademacher bound

$$\mathbf{Rad}_n\left(x_{1:t}, y_{1:t}\right) = \sup_{\mathbf{x},\mathbf{y}} \mathbb{E}_{\epsilon_{t+1:n}} \sup_{f \in \mathcal{F}} \left\{2 \sum_{s=t+1}^{n} \epsilon_s \ell(f(\mathbf{x}_{s-t}(\epsilon_{t+1:s-1})), \mathbf{y}_{s-t}(\epsilon_{t+1:s-1})) - \sum_{s=1}^{t} \ell(f(x_s), y_s)\right\}$$

2 The Recipe



1. Write down sequential Rademacher relaxation for the given problem (in a malleable form).

2. Move to upper bound $\mathbf{Rel}_n$ such that $\forall t \in [n]$ and for all $x_{1:t}$, $y_{1:t}$,

$$\mathbf{Rad}_n\left(x_{1:t}, y_{1:t}\right) \leq \mathbf{Rel}_n\left(x_{1:t}, y_{1:t}\right)$$

(notice this ensures that initial condition is satisfied, this is half the work).

3. Two equivalent ways of checking admissibility condition, $\forall x_t \in \mathcal{X}$:

$$\inf_{q_t} \sup_{y_t} \left\{\mathbb{E}_{\hat{y}_t \sim q_t}\left[\ell(\hat{y}_t, y_t)\right] + \mathbf{Rel}_n\left(x_{1:t}, y_{1:t}\right)\right\} \leq \mathbf{Rel}_n\left(x_{1:t-1}, y_{1:t-1}\right)$$

$$\sup_{p_t}\left\{\inf_{\hat{y}_t \in \mathcal{Y}} \mathbb{E}_{y_t \sim p_t}\left[\ell(\hat{y}_t, y_t)\right] + \mathbb{E}_{y_t \sim p_t}\left[\mathbf{Rel}_n\left(x_{1:t}, y_{1:t}\right)\right]\right\} \leq \mathbf{Rel}_n\left(x_{1:t-1}, y_{1:t-1}\right)$$

4. Algorithm: solve $\quad q_t(x_t) = \underset{q_t \in \Delta(\mathcal{Y})}{\operatorname{argmin}} \sup_{y_t}\left\{\mathbb{E}_{\hat{y}_t \sim q_t}\left[\ell(\hat{y}_t, y_t)\right] + \mathbf{Rel}_n\left(x_{1:t}, y_{1:t}\right)\right\}$

3 Finite Experts

$|\mathcal{F}| < \infty$ and ℓ bounded by say 1.

Step 1

$$\begin{aligned}
\mathbf{Rad}_n\left(x_{1:t}, y_{1:t}\right) &= \sup_{\mathbf{x},\mathbf{y}} \mathbb{E}_{\epsilon_{t+1},\ldots,\epsilon_n} \sup_{f \in \mathcal{F}}\left[2\sum_{s=t+1}^{n} \epsilon_s \ell(f(\mathbf{x}_{s-t}(\epsilon_{t+1:s-1})), \mathbf{y}_{s-t}(\epsilon_{t+1:s-1})) - \sum_{s=1}^{t} \ell(f(x_s), y_s)\right] \\
&= \sup_{\mathbf{x},\mathbf{y}} \mathbb{E}_{\epsilon_{t+1:n}} \inf_{\lambda > 0} \frac{1}{\lambda} \log\left(\sum_{f \in \mathcal{F}} \exp\left(2\lambda \sum_{s=t+1}^{n} \epsilon_s \ell(f(\mathbf{x}_{s-t}(\epsilon_{t+1:s-1})), \mathbf{y}_{s-t}(\epsilon_{t+1:s-1})) - \lambda \sum_{s=1}^{t} \ell(f(x_s), y_s)\right)\right)
\end{aligned}$$

That is we move to limit of soft-max.

Step 2

$$\begin{aligned}
&\mathbf{Rad}_n\left(x_{1:t}, y_{1:t}\right) = \sup_{\mathbf{x},\mathbf{y}} \mathbb{E}_{\epsilon_{t+1:n}} \inf_{\lambda > 0} \frac{1}{\lambda} \log\left(\sum_{f \in \mathcal{F}} \exp\left(2\lambda \sum_{s=t+1}^{n} \epsilon_s \ell(f(\mathbf{x}_{s-t}(\epsilon_{t+1:s-1})), \mathbf{y}_{s-t}(\epsilon_{t+1:s-1})) - \lambda \sum_{s=1}^{t} \ell(f(x_s), y_s)\right)\right) \\
&\leq \inf_{\lambda > 0} \sup_{\mathbf{x},\mathbf{y}} \mathbb{E}_{\epsilon_{t+1:n}} \frac{1}{\lambda} \log\left(\sum_{f \in \mathcal{F}} \exp\left(2\lambda \sum_{s=t+1}^{n} \epsilon_s \ell(f(\mathbf{x}_{s-t}(\epsilon_{t+1:s-1})), \mathbf{y}_{s-t}(\epsilon_{t+1:s-1})) - \lambda \sum_{s=1}^{t} \ell(f(x_s), y_s)\right)\right) \\
&\leq \inf_{\lambda > 0} \sup_{\mathbf{x},\mathbf{y}} \frac{1}{\lambda} \log\left(\sum_{f \in \mathcal{F}} \mathbb{E}_{\epsilon_{t+1:n}} \exp\left(2\lambda \sum_{s=t+1}^{n} \epsilon_s \ell(f(\mathbf{x}_{s-t}(\epsilon_{t+1:s-1})), \mathbf{y}_{s-t}(\epsilon_{t+1:s-1}))\right) \cdot \exp\left(-\lambda \sum_{s=1}^{t} \ell(f(x_s), y_s)\right)\right) \\
&= \inf_{\lambda > 0} \sup_{\mathbf{x},\mathbf{y}} \frac{1}{\lambda} \log\left(\sum_{f \in \mathcal{F}} \mathbb{E}_{\epsilon_{t+1:n}}\left[\prod_{s=t+1}^{n} \mathbb{E}_{\epsilon_t}\left[\exp\left(2\lambda \epsilon_s \ell(f(\mathbf{x}_{s-t}(\epsilon_{t+1:s-1})), \mathbf{y}_{s-t}(\epsilon_{t+1:s-1}))\right)\right]\right] \cdot \exp\left(-\lambda \sum_{s=1}^{t} \ell(f(x_s), y_s)\right)\right)
\end{aligned}$$

using $\mathbb{E}_\epsilon\left[e^{\epsilon x}\right] \le e^{x^2/2}$ exactly as in finite lemma proof, since loss is bounded,

$$
\begin{aligned}
&\le \inf_{\lambda>0} \sup_{\mathbf{x},\mathbf{y}} \frac{1}{\lambda} \log\left(\sum_{f\in\mathcal{F}} \prod_{s=t+1}^{n} \exp\left(2\lambda^2\right) \cdot \exp\left(-\lambda \sum_{s=1}^{t} \ell(f(x_s), y_s)\right)\right) \\
&= \inf_{\lambda>0} \left\{\frac{1}{\lambda} \log\left(\sum_{f\in\mathcal{F}} \exp\left(-\lambda \sum_{s=1}^{t} \ell(f(x_s), y_s)\right)\right) + 2\lambda(n-t)\right\} \\
&=: \mathbf{Rel}_n\left(x_{1:t}, y_{1:t}\right)
\end{aligned}
$$

Step 3. (admissibility)

$$
\begin{aligned}
&\sup_{p_t}\left\{\inf_{\hat{y}_t\in\mathcal{Y}} \mathbb{E}_{y_t\sim p_t}\left[\ell(\hat{y}_t, y_t)\right] + \mathbf{Rel}_n\left(x_{1:t}, y_{1:t}\right)\right\} \\
&\quad= \sup_{p_t}\left\{\inf_{\hat{y}_t\in\mathcal{Y}} \mathbb{E}_{y_t\sim p_t}\left[\ell(\hat{y}_t, y_t)\right] + \mathbb{E}_{y_t\sim p_t}\left[\inf_{\lambda>0}\left\{\frac{1}{\lambda}\log\left(\sum_{f\in\mathcal{F}}\exp\left(-\lambda\sum_{s=1}^{t}\ell(f(x_s),y_s)\right)\right) + 2\lambda(n-t)\right\}\right]\right\} \\
&\quad\le \inf_{\lambda>0}\sup_{p_t}\left\{\inf_{\hat{y}_t\in\mathcal{Y}} \mathbb{E}_{y_t\sim p_t}\left[\ell(\hat{y}_t, y_t)\right] + \mathbb{E}_{y_t\sim p_t}\left[\frac{1}{\lambda}\log\left(\sum_{f\in\mathcal{F}}\exp\left(-\lambda\sum_{s=1}^{t}\ell(f(x_s),y_s)\right)\right) + 2\lambda(n-t)\right]\right\} \\
&\quad= \inf_{\lambda>0}\sup_{p_t}\left\{\frac{1}{\lambda}\log(\exp(\lambda \inf_{\hat{y}_t\in\mathcal{Y}} \mathbb{E}_{y_t\sim p_t}\left[\ell(\hat{y}_t, y_t)\right])) + \mathbb{E}_{y_t\sim p_t}\left[\frac{1}{\lambda}\log\left(\sum_{f\in\mathcal{F}}\exp\left(-\lambda\sum_{s=1}^{t}\ell(f(x_s),y_s)\right)\right) + 2\lambda(n-t)\right]\right\} \\
&\quad= \inf_{\lambda>0}\sup_{p_t}\left\{\mathbb{E}_{y_t\sim p_t}\left[\frac{1}{\lambda}\log\left(\sum_{f\in\mathcal{F}}\exp\left(\lambda \inf_{\hat{y}_t\in\mathcal{Y}} \mathbb{E}_{y_t\sim p_t}\left[\ell(\hat{y}_t, y_t)\right] - \lambda\sum_{s=1}^{t}\ell(f(x_s),y_s)\right)\right) + 2\lambda(n-t)\right]\right\} \\
&\quad\le \inf_{\lambda>0}\sup_{p_t}\left\{\mathbb{E}_{y_t\sim p_t}\left[\frac{1}{\lambda}\log\left(\sum_{f\in\mathcal{F}}\exp\left(\lambda\left(\mathbb{E}_{y_t\sim p_t}\left[\ell(f(x_t),y_t)\right] - \ell(f(x_t),y_t)\right) - \lambda\sum_{s=1}^{t-1}\ell(f(x_s),y_s)\right)\right) + 2\lambda(n-t)\right]\right\} \\
&\quad\le \inf_{\lambda>0}\sup_{p_t}\left\{\frac{1}{\lambda}\log\left(\mathbb{E}_{y_t,y_t'\sim p_t}\left[\sum_{f\in\mathcal{F}}\exp\left(\lambda\left(\ell(f(x_t),y_t') - \ell(f(x_t),y_t)\right) - \lambda\sum_{s=1}^{t-1}\ell(f(x_s),y_s)\right)\right]\right) + 2\lambda(n-t)\right\} \\
&\quad= \inf_{\lambda>0}\sup_{p_t}\left\{\frac{1}{\lambda}\log\left(\mathbb{E}_{y_t,y_t'\sim p_t}\mathbb{E}_{\epsilon_t}\left[\sum_{f\in\mathcal{F}}\exp\left(\lambda\epsilon_t\left(\ell(f(x_t),y_t') - \ell(f(x_t),y_t)\right) - \lambda\sum_{s=1}^{t-1}\ell(f(x_s),y_s)\right)\right]\right) + 2\lambda(n-t)\right\}
\end{aligned}
$$

$(e^x + e^{-x})/2 \le e^{x^2/2}$ and loss is bounded by 1 and so,

$$
\begin{aligned}
&\le \inf_{\lambda>0}\left\{\frac{1}{\lambda}\log\left(\sum_{f\in\mathcal{F}}\exp\left(-\lambda\sum_{s=1}^{t-1}\ell(f(x_s),y_s)\right)\cdot e^{2\lambda^2}\right) + 2\lambda(n-t)\right\} \\
&= \inf_{\lambda>0}\left\{\frac{1}{\lambda}\log\left(\sum_{f\in\mathcal{F}}\exp\left(-\lambda\sum_{s=1}^{t-1}\ell(f(x_s),y_s)\right)\right) + 2\lambda(n-t+1)\right\} \\
&= \mathbf{Rel}_n\left(x_{1:t-1}, y_{1:t-1}\right)
\end{aligned}
$$

Step 4. (algorithm) We get a parameter free version of exponential weights algorithm,

$$\lambda_t^* = \operatorname*{argmin}_{\lambda>0} \left\{ \frac{1}{\lambda} \log \left(\sum_{f\in\mathcal{F}} \exp\left(-\lambda \sum_{s=1}^{t-1} \ell(f(x_s), y_s) \right) \right) + 2\lambda(n-t+1) \right\} \;\&\; q_t^* \propto \exp\left(-\lambda_t^* \sum_{s=1}^{t-1} \ell(f(x_s), y_s) \right)$$

Finally, we get the guarantee,

$$\mathbb{E}\left[\mathrm{Reg}_n\right] \le \tfrac{1}{n}\mathbf{Rel}_n\left(\cdot\right) = \frac{1}{n} \inf_{\lambda>0} \left\{ \frac{1}{\lambda} \log\left(|\mathcal{F}|\right) + 2\lambda n \right\} = \sqrt{\frac{8\log|\mathcal{F}|}{n}}$$

4 Online Linear Optimization: Euclidean space

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

Step 1

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

Step 2

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

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.

5 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

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

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

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

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

where $\lambda^* = \underset{\lambda>0}{\operatorname{argmin}} \left\{ \frac{R^2}{\lambda} + \frac{1}{\lambda}\mathbf{R}^* \left(-\lambda \sum_{s=1}^{t-1} \nabla_s \right) + 2\lambda(n-t+1) \right\}$. 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 :

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

5.1 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.