CSCI699: Theory of Machine Learning	Fall 2021

Lecture 14: Boosting and Convex Learning Problems

Instructor: Vatsal Sharan *Scribe: Navid Hashemi*

Time slots for the presentation review have been posted on the project homepage:

1 Boosting

Definition 1. *Weak Learning: An algorithm A is a weak learner with edge/advantage γ for class $\mathcal{C}$ if: for any distribution $\mathcal{D}$ and any target $c \in \mathcal{C}$ given access to example (C, D) with probability $(1-\delta)$, A produces a hypothesis with* $\textbf{error}(h; c, D) \leq \frac{1}{2} - \gamma$.

If A runs in time $\textbf{poly}(d, \frac{1}{\delta})$ and $\gamma \geq \frac{1}{\textbf{poly}(d)}$, then $\mathcal{C}$ is efficiently weakly PAC-learnable.

Theorem 2. *If $\mathcal{C}$ is weakly PAC-learnable (efficiently) the $\mathcal{C}$ is PAC-learnable (efficiently).*

Proof. The proof relies on the AdaBoost algorithm due to Freund and Schapire. We first restate our setup.

There is a training set $\{(x_1, y_1), \cdots, (x_n, y_n)\}$, where $x_i \in \mathcal{X}, y_i \in \{-1, 1\}$.

By realizability,

$$\exists\ c \in \mathcal{C}\ s.t. \forall i\ y_i = c(x_i).$$

We assume there exists weak learning algorithm (WL), for $\mathcal{C}$.

AdaBoost (Freund and Schapire):

1. $\forall\ i \in [n]\ D_1(i) = \frac{1}{n}$
2. for $t = 1, 2, , \cdots, T$
3. use WL with dist D_t to get h_t
4. let $\epsilon_t = \mathbb{P}_{x \sim D_t}[h_t(x) = y]$. ($\epsilon_t \leq 1 - \gamma$ w.p $1 - \delta$)
5. choose $\alpha_t = \frac{1}{2}\textbf{log}\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$
6. $D_{t+1}(i) = \frac{D_t(i)}{z_{t+1}} = \begin{cases} e^{-\alpha_t} & if h_t(x_i) = y_i \\ e^{\alpha_t} & if h_t(x_i) \neq y_i \end{cases} = \frac{D_t(i)}{z_{t+1}} e^{-\alpha_t h_t(x_i) y_i}$, where z_{t+1} is normalizing constant.
7. Output $\text{sign}(H(x))$ where $H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$

Note:

1. We can assume $\epsilon_t \leq \frac{1}{2} - \gamma$ (union bound).
2. We can emulate the example oracle $EX(c, D_t)$, because D_t has finite support, Just reweight.

Aside: Ada Boost fits in the "Multiplicative Weight Update" framework. General framework with many algorithmic applications.

Lemma 3. *for* $T = \frac{1}{2\gamma^2}\mathbf{log}(2n)$*, training error is* 0.

Proof.

$$\text{Training error} = \mathbb{P}_{D_1}(\text{sign}(H(x) \neq y)) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(\text{sign}(H(x) \neq y_i)).$$

Note that

$$\mathbb{1}(\text{sign}(H(x) \neq y)) \leq e^{-yH(x)}.$$

Define $H_t(x) = \sum_{s=t}^{T} \alpha_s h_s(x)$. $H_t(x)$ has the following recursive form,

$$\begin{aligned} H_t(x) &= \alpha_t h_t(x) + H_{t+1}(x) \\ H_1(x) &= H(x) \end{aligned}$$

Therefore we can write the training error as,

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(\text{sign}(H(x_i) \neq y_i)) &\leq \sum_{i=1}^{n} D_1(i) e^{-y_i H_1(x_i)} \\ &= \sum_{i=1}^{n} D_1(i) e^{-d_1 y_i h_1(x_i) - y_i H_2(x_i)} \\ &= z_2 \sum_{i=1}^{n} D_2(i) e^{-y_i H_2(x_i)} \\ &= z_2 \sum_{i=1}^{n} D_2(i) e^{-\alpha_2 y_i h_2(x_i) - y_i H_3(x_i)} \\ &= z_2 z_3 \sum_{i=1}^{n} D_3(i) e^{-y_i H_3(x_i)} \\ &\vdots \\ &= \prod_{t=2}^{T+1} z_t \end{aligned}$$

We now need to bound z_{t+1}.

$$\begin{aligned}
z_{t+1} &= \sum_{i=1}^{n} D_t(i) e^{-\alpha_t y_i h_t(x_i)} \\
&= \sum_{i:y_i=h_t(x_i)} D_t(i) e^{-\alpha_t} + \sum_{i:y_i\neq h_t(x_i)} D_t(i) e^{\alpha_t} \\
&= (1-\epsilon_t)e^{-\alpha_t} + \epsilon_t e^{\alpha_t} \qquad \text{Recall: } \alpha_t = \frac{1}{2}\log(\frac{1-\epsilon_t}{\epsilon_t}) \\
z_{t+1} &= (1-\epsilon_t)\sqrt{\frac{\epsilon_t}{1-\epsilon_t}} + \epsilon_t\sqrt{\frac{1-\epsilon_t}{\epsilon_t}} - 2\sqrt{\epsilon_t(1-\epsilon_t)}
\end{aligned}$$

Define $\gamma_t = \frac{1}{2} - \epsilon_t$ Note that $\gamma_t \geq \gamma$:

$$\begin{aligned}
z_{t+1} &= 2\sqrt{(1/2-\gamma_t)(1/2+\gamma_t)} \\
&= \sqrt{1-4\gamma_t^2} \\
&\leq (e^{-4\gamma_t^2})^{\frac{1}{2}} = e^{2\gamma_t^2} \leq e^{-2\gamma^2} \\
\text{Therefore } \prod_{t=1}^{T+1} z_t &\leq e^{-2T\gamma^2} \leq \frac{1}{2n} \text{ if } T \geq \frac{1}{2\gamma^2}\log(2n)
\end{aligned}$$

$\square$

What about test error?

Suppose that WL(weak learning algo) always outputs a hypothesis from some class $\mathcal{H}$ whose VC-dim is d.

$$LC(\mathcal{H}, T) = \left\{ \text{sign}(\sum_{i=1}^{T} \alpha_i h_i(x)) : \; h_i \in \mathcal{H} \; d_i \in \mathbb{R} \right\}$$

Exercise: VC-dim$(LC(\mathcal{H}, T)) \leq c.T.d.\log(T)$ for some constant c. Hint: As in HW1, Compute Growth function and use Sauer's Lemma.

Due to VC-theorem,
If $n \geq \frac{c}{\epsilon}\left(\log(\frac{1}{\delta}) + Td\log(T)\log(1/\epsilon)\right)$ then the hypothesis produced by Ada Boost has **error** $\leq \epsilon$, with probability $(1-\delta)$.
Putting in the bound of T, if $n \geq \frac{c}{\epsilon}\left(\log(1/\delta) + \frac{1}{2\gamma^2}\log(2n) d \log\left(\frac{1}{2\gamma^2}\log(2n)\right)\log(1/\epsilon)\right)$, we will get **error** $\leq \epsilon$ w.p. $1-\delta$.
This is satisfied for $n \geq \frac{c}{\epsilon}\left(\frac{d}{\gamma^2}\textbf{poly}\left(\log(d, \frac{1}{\gamma^2}, \frac{1}{\epsilon})\right) + \log(1/\delta)\right)$ ($n \geq a\log(n)$ is satisfied for $n \geq O(a\log(a))$). $\square$

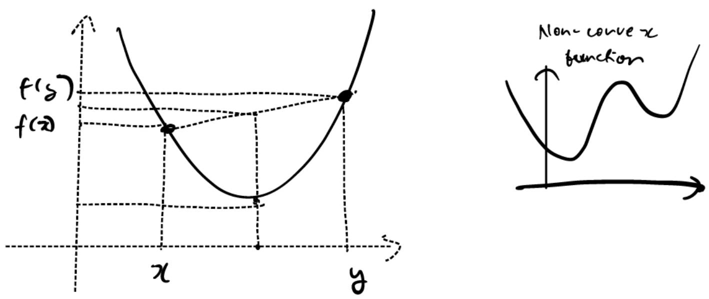
2 Convex Optimization

We start with some basic properties:

Definition 4. *(Convex set) A set $\mathcal{C} \subset \mathbb{R}^d$ is convex if $x, y \in \mathcal{C} \to tx + (1-t)y \in \mathcal{C},\ \forall 0 \le t \le 1$.*

Definition 5. *(Convex functions) A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if* **Domain**$(f) \subset \mathbb{R}^d$ *is convex and*

$$f(tx + (1-t)y) \le tf(x) + (1-t)f(y) \quad \forall\ t \in [0,1]\ \textbf{\textit{and}}\ (x,y) \in \textbf{Domain}(f).$$



Lemma 6. *If f is differentiable, then f is convex if and only if* **Domain(f)** *is convex and:*

$$f(y) - f(x) \ge \langle y - x, \nabla f(x) \rangle.$$
$$\textit{(or equivalently, } f(y) - f(x) \le \langle y - x, \nabla f(y) \rangle\textit{)}$$

Proof. Exercise: Use definition of convexity □

Corollary 7. *If f is convex and differentiable the $\nabla f(x) = 0$ implies x is a global minima of f (Local minima implies global minimum).*

Definition 8. *(Strong Convexity) A function f is λ-strongly convex if its domain is convex and,*

$$\forall\ x, y\ f(y) \ge f(x) + \langle y - x, \nabla f(x) \rangle + \frac{\lambda}{2} \|y - x\|^2.$$

Lemma 9. *(Some Properties of convex functions)*

1. *If f is twice differentiable, then f is convex iff domain(f) is convex and $\forall\ x \in domain(f), \nabla^2 f(x) \succeq 0$, (Recall that $A \succeq 0 \iff \forall\ x\ : x^\top Ax \ge 0$).*
2. *If $f_i(x)$ is convex $\forall i \in [n]$ then $y(x) = \sum_{i=1}^{n} w_i f_i(x)$ is convex, where $w_i \ge 0$.*
3. *If $f_i(x)$ is convex $\forall\ i \in [n]$ then $g(x) = \max_{i \in [n]} f_i(x)$ is convex.*

3 Convex learning Problems

An optimization problem

$$\min_{x\in A} f(x)$$

is called a convex optimization problem if (1) $f(x)$ is convex (2) A is convex.

Remark. *As we'll show next class, convex optimization problems can be solved efficiently.*

Recall the ERM problem of finding the ERM w.r.t some Hypotheis $\mathcal{H}$ in some training set $\mathcal{S} = (z_1, z_2, \cdots, z_n)$ where $z_i = (x_i, y_i)$.

$$\mathbf{ERM}_{\mathcal{H}}(\mathcal{S}) = \text{argmin}_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \ell(h, z_i)$$

Let $\mathcal{H} \subset \mathbb{R}^d$ parametrized by $w \in \mathcal{H}$.

$$\mathbf{ERM}_{\mathcal{H}}(\mathcal{S}) = \text{argmin}_{w\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} \ell(w, z_i).$$

Lemma 10. *If ℓ is a convex loss (in terms of w), and $\mathcal{H}$ is convex, then $\mathbf{ERM}_{\mathcal{H}}(\mathcal{S})$ is a convex optimization problem.*

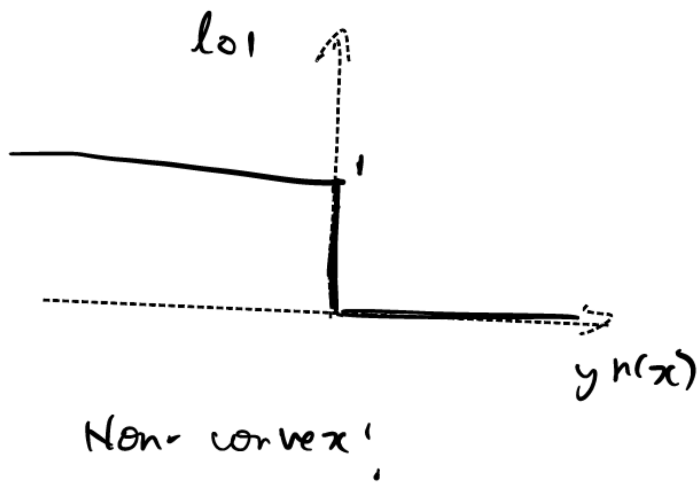
Proof. Average of convex functions is convex. $\square$

Example (Linear regression with squared loss):

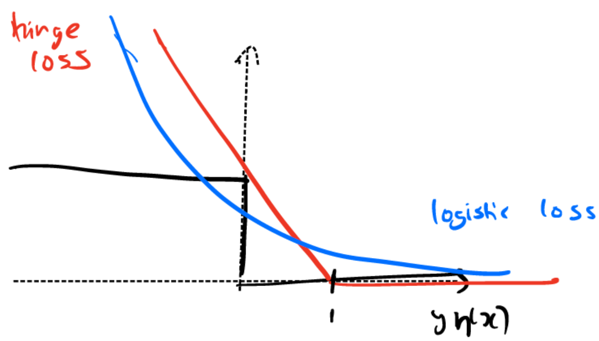
$$\begin{aligned}
&\text{Let } \mathcal{H} = \left\{ x \to \langle w, x\rangle, w \in \mathbb{R}^d \right\} \\
&\ell(h, (x, y)) = (h(x) - y)^2 \\
&\ell(w, (x, y)) = (\langle w, x\rangle - y)^2 \\
&\mathcal{H} = \mathbb{R}^d \text{ convex}
\end{aligned}$$

Exercise: $\ell(w, (x, y))$ is convex in terms of w.
$\mathbf{ERM}_{\mathcal{H}}(\mathcal{S})$ is a convex problem.
What about the 0/1 loss?

$$\ell_{01}(h, (x, y)) = \mathbb{1}(h(x) \neq y) = \mathbb{1}(yh(x) \leq 0)$$



A common technique to handle a non-convex loss is to instead consider a convex surrogate.



$$\text{hinge loss:}\quad \ell(h,(x,y)) = \max(1 - yh(x), 0)$$
$$\text{logistic loss}\quad \ell(h,(x,y)) = \log(1 + e^{-yh(x)})$$

Next class we will discuss more about convex surrogates.