Binary Classification, Part 4

① AdaBoost (Freund-Schapire, 1997)

$\mathcal{G}$: base classifiers $g: \mathcal{X} \to \{-1, +1\}$ (weak learners)

$(X_1, Y_1), \dots, (X_n, Y_n)$ iid on $\mathcal{X} \times \{-1, +1\}$

Goal: return $\hat{f}_n \in \text{conv}(\mathcal{G})$ (use $\operatorname{sgn} \hat{f}_n$ to classify)

$$\hat{f}_n(x) = \frac{\sum_{k=1}^{K} \alpha_k g_k(x)}{\sum_{k=1}^{K} \alpha_k}$$

where K is fixed (# iterations) and $\alpha_k \geq 0$ and $g_k \in \mathcal{G}$ are determined iteratively from data.

Note: can use $\operatorname{sgn}\left(\sum_{k=1}^{K} \alpha_k g_k(x)\right)$ to classify

AdaBoost

- init: $w^{(1)} = (w_1^{(1)}, \dots, w_n^{(1)})$, $w_i^{(1)} = 1/n$ $\forall i$
- for $k = 1, \dots, K$ do:
  - $g_k := \operatorname{argmin}_{g \in \mathcal{G}} e_k(g)$ where $e_k(g) := \sum_{i=1}^{n} w_i^{(k)} 1_{\{Y_i \neq g(X_i)\}}$

    $e_k := e_k(g_k)$

    // standing assumption: $e_k \leq 1/2$
  - $w^{(k)} \to w^{(k+1)}$ update

    $\alpha_k := \frac{1}{2} \log \frac{1 - e_k}{e_k}$

    // $e_k \leq 1/2 \iff 1 - e_k \geq e_k$ $\quad \alpha_k \geq 0$

    $$w_i^{(k+1)} = \frac{w_i^{(k)} e^{-\alpha_k Y_i g_k(X_i)}}{Z_k}$$

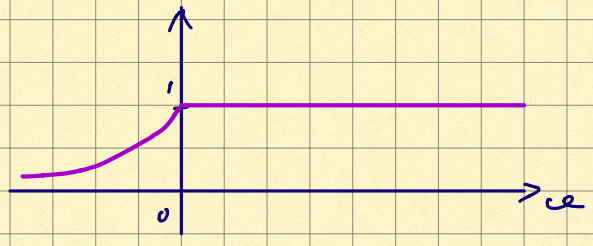
    Z_k : norm.

// $e^{-\alpha_k Y_i g_k(x_i)} = \begin{cases} e^{\alpha_k} & \text{if } Y_i \neq g_k(x_i) \\ e^{-\alpha_k} & \text{if } Y_i = g_k(x_i) \end{cases}$

- return $\bar{f}_n(x) := \dfrac{\sum_{k=1}^K \alpha_k g_k(x)}{\sum_{k=1}^K \alpha_k}$

Preview: • w.l.p., $L(\bar{f}_n) \leq \prod_{k=1}^K 2\sqrt{e_k(1-e_k)}$

• key analysis tool: $\varphi(u) = 1 \wedge e^u = \min\{1, e^u\}$



1-Lip., $0 \leq \varphi(\cdot) \leq 1$

$\gamma \geq \gamma' > 0 \Rightarrow$ $\varphi(\cdot/\gamma) \geq \varphi(\cdot/\gamma')$

• result due to Koltchinskii-Panchenko (2002)

Lemma $\frac{1}{n}\sum_{i=1}^n \exp\left(-Y_i \sum_{k=1}^K \alpha_k g_k(x_i)\right) = \prod_{k=1}^K 2\sqrt{e_k(1-e_k)}$

Proof $\exp\left(-Y_i \sum_{k=1}^K \alpha_k g_k(x_i)\right)$

$$= \prod_{k=1}^K \exp\left(-\alpha_k Y_i g_k(x_i)\right)$$

$$= \prod_{k=1}^K \frac{w_i^{(k+1)}}{w_i^{(k)}} Z_k$$

$$= \frac{\prod_{k=1}^K w_i^{(k+1)}}{w_i^{(1)}} \cdot \prod_{k=1}^K Z_k$$

$$= \frac{w_i^{(K+1)}}{w_i^{(1)}} \cdot \prod_{k=1}^K Z_k$$

$$= n w_i^{(K+1)} \cdot \prod_{k=1}^K Z_k$$

$$\frac{1}{n}\sum_{i=1}^{n} \exp\left(-Y_i \sum_{k=1}^{K} \alpha_k g_k(X_i)\right) = \underbrace{\sum_{i=1}^{n} w_i^{(K+1)}}_{=1} \cdot \prod_{k=1}^{K} Z_k$$

where $Z_k = 2\sqrt{e_k(1-e_k)}$ (can be shown from defn.) ∎

Per fcn $\varphi(u) = 1 \wedge e^u \le e^u$

By K.-P. adaptive margin bound, w.p. $\ge 1-\delta$,

$$L(\hat{f}_n) \le \inf_{\gamma \in (0,1]} \left\{ A_{\varphi(\cdot/\gamma),n}(\hat{f}_n) + \frac{C}{\gamma}\mathbb{E}R_n(\mathcal{G}(X^n)) + C\sqrt{\frac{\log\log(1/\gamma)}{n}} \right\} + \sqrt{\frac{\log(1/\delta)}{2n}}.$$

↓ take any $0 < \gamma \le 1$

Fix $\gamma \in (0,1]$

$$A_{\varphi(\cdot/\gamma),n}(\hat{f}_n) = \frac{1}{n}\sum_{i=1}^{n} \varphi\left(-\frac{1}{\gamma} Y_i \hat{f}_n(X_i)\right)$$

$$= \frac{1}{n}\sum_{i=1}^{n} \varphi\left(-\frac{1}{\gamma} Y_i \frac{\sum_{k=1}^{K} \alpha_k g_k(X_i)}{\sum_{k=1}^{K} \alpha_k}\right)$$

$$\le \frac{1}{n}\sum_{i=1}^{n} \varphi\left(-Y_i \sum_{k=1}^{K} \alpha_k g_k(X_i)\right) \quad \text{for a properly chosen } \gamma$$

(take $\gamma = 1 \wedge \frac{1}{\sum_{k=1}^{K} \alpha_k} \le \frac{1}{\sum_{k=1}^{K} \alpha_k}$)

$$\le \frac{1}{n}\sum_{i=1}^{n} \exp\left(-Y_i \sum_{k=1}^{K} \alpha_k g_k(X_i)\right) = \prod_{k=1}^{K} 2\sqrt{e_k(1-e_k)}$$

$$\therefore\ L(\hat{f}_n) \le \prod_{k=1}^{K} 2\sqrt{e_k(1-e_k)} + C\underbrace{\left(1 \vee \sum_{k=1}^{K} \alpha_k\right)}_{1/\gamma} \mathbb{E}R_n(G_S(X^n))$$

$$+ C\sqrt{\frac{\log\log\left(1 \vee \sum_{k=1}^{K} \alpha_k\right)}{n}} + \sqrt{\frac{\log(1/\delta)}{2n}} \qquad \text{w.p.} \ge 1-\delta$$

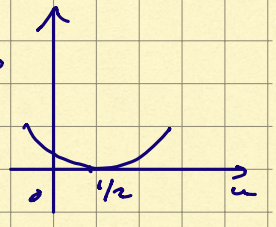
for $\gamma = 1 \wedge \dfrac{1}{\sum_{k=1}^{K} \alpha_k}$. ∎

Note: • $2\sqrt{e_k(1-e_k)} \le 1$

$e_k(1-e_k) \le \frac{1}{4}$

$e_k - e_k^2 \le \frac{1}{4}$ $\qquad e_k^2 - e_k + 1/4 \ge 0$

$(e_k - 1/2)^2 \ge 0$



• $\sum_{k=1}^{K} \alpha_k = \sum_{k=1}^{K} \frac{1}{2} \log \frac{1-e_k}{e_k} = \sum_{k=1}^{K} \log \sqrt{\frac{1-e_k}{e_k}}$

$$= \log \prod_{k=1}^{K} \left(\frac{1-e_k}{e_k}\right)^{1/2}.$$

• $e_k < 1/2 \ \ \forall k \quad \Rightarrow \quad \prod_{k=1}^{K} 2\sqrt{e_k(1-e_k)} \xrightarrow{K\to\infty} 0$

② Neural Nets

- linear classifier : $x \mapsto \operatorname{sgn}\langle w, x\rangle$ $\qquad w, x \in \mathbb{R}^d$

$\mathcal{F} := \{ x \mapsto \langle w, x\rangle : \ \|w\| \le B \}$

$\|w\| := \left(\sum_{j=1}^{d} |w_j|^2\right)^{1/2}$

w/o restriction on w, $R_n \le \sqrt{d/n}$

$$R_n(\mathcal{F}(X^n)) = \frac{1}{n}\mathbb{E}_\varepsilon\left[\sup_{\|w\|\le B}\left|\sum_{i=1}^n \varepsilon_i \langle w, X_i\rangle\right|\right]$$

$$= \frac{1}{n}\mathbb{E}_\varepsilon\left[\sup_{\|w\|\le B}\left|\left\langle w, \sum_{i=1}^n \varepsilon_i X_i\right\rangle\right|\right]$$

$$= \frac{B}{n}\mathbb{E}_\varepsilon\left\|\sum_{i=1}^n \varepsilon_i X_i\right\|$$

[Cauchy–Schwarz: $\|v\| = \sup_{\|w\|\le 1} |\langle w, v\rangle|$]

$$\mathbb{E}_\varepsilon\left\|\sum_{i=1}^n \varepsilon_i X_i\right\| = \mathbb{E}_\varepsilon\sqrt{\sum_{i=1}^n\sum_{j=1}^n \varepsilon_i\varepsilon_j\langle X_i, X_j\rangle}$$

$$\le \sqrt{\sum_{i=1}^n\sum_{j=1}^n \langle X_i, X_j\rangle \underbrace{\mathbb{E}[\varepsilon_i\varepsilon_j]}_{1_{\{i\neq j\}}}}$$

$$= \sqrt{\sum_{i=1}^n \|X_i\|^2}$$

$$\therefore \boxed{R_n(\mathcal{F}(X^n)) \le \frac{B}{n}\sqrt{\sum_{i=1}^n \|X_i\|^2}}$$

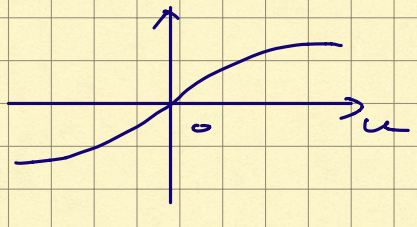
E.g. if $X_1, \dots, X_n$ are elements of a ball of rad. R centered at 0, then

$$R_n(\mathcal{F}(X^n)) \le \frac{BR}{\sqrt{n}} \leftarrow \text{dimension-free}$$

– add a nonlinearity

$\sigma: \mathbb{R} \to \mathbb{R}$ $\quad \sigma(0) = 0$, $\quad L$–Lipschitz

e.g. $\sigma(u) = \tanh u = \dfrac{e^u - e^{-u}}{e^u + e^{-u}}$



$$x \mapsto \operatorname{sgn}\sigma(\langle w, x\rangle)$$

$$\mathcal{F}_\sigma := \{ x \mapsto \sigma(\langle w, x\rangle) : \|w\| \le B \}$$

$$R_n(\mathcal{F}_\sigma(X^n)) = R_n(\sigma \circ \mathcal{F}(X^n)) \qquad \mathcal{F}: \text{linear}$$

$$\le 2L\, R_n(\mathcal{F}(X^n)) \qquad \text{(contraction principle)}$$

$$\le \frac{2LB}{n}\sqrt{\sum_{i=1}^n \|x_i\|^2}$$

– neural nets

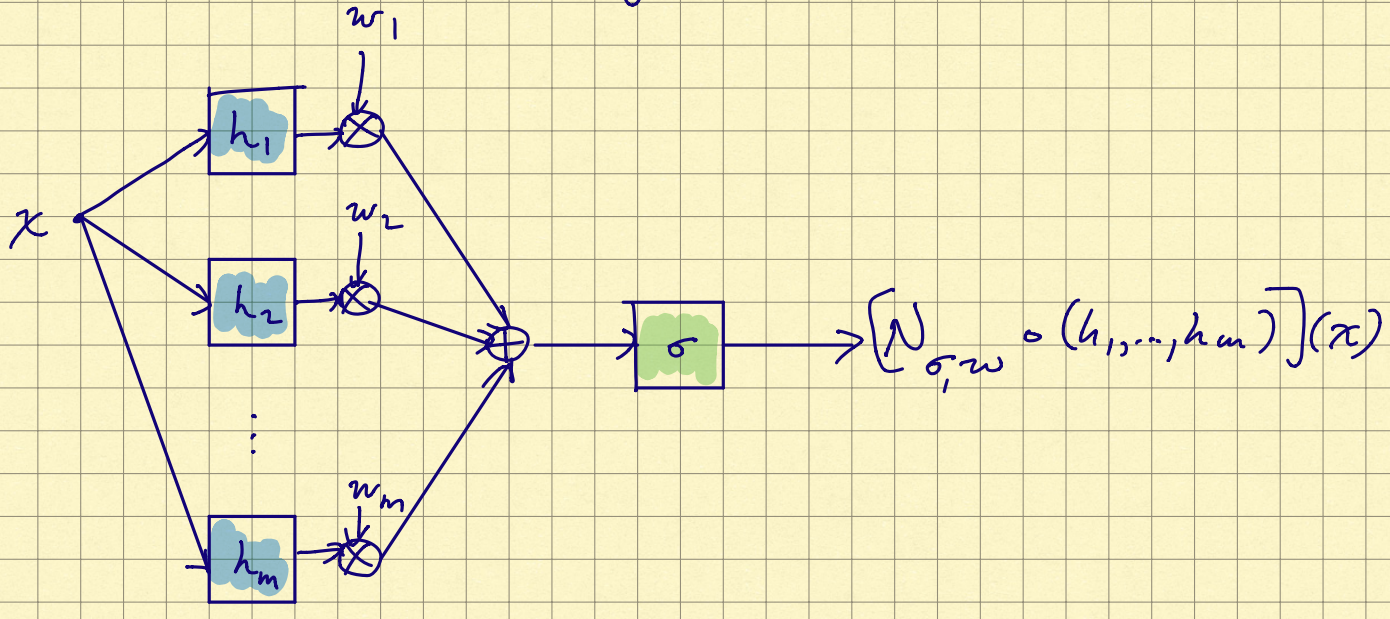
$\sigma : \mathbb{R} \to \mathbb{R}$, $w \in \mathbb{R}^m$ $(m \in \mathbb{N})$ $\Big\}$ $N_{\sigma,w} : \mathbb{R}^m \to \mathbb{R}$

$$N_{\sigma,w}(u) := \sigma\Big(\sum_{j=1}^m w_j u_j\Big)$$

$h_1, \dots, h_m : \mathbb{R}^d \to \mathbb{R}$

$$[N_{\sigma,w} \circ (h_1, \dots, h_m)](x) := N_{\sigma,w}(h_1(x), \dots, h_m(x))$$

$$= \sigma\Big(\sum_{j=1}^m w_j \cdot h_j(x)\Big)$$



deep neural nets – define recursively

- $\mathcal{G}_j$ – base classifiers $g: \mathcal{X} \to \mathbb{R}$, $\mathcal{X} \subseteq \mathbb{R}^d$

- fix $\ell \in \mathbb{N}$ (# layers)

$\sigma_1, \dots, \sigma_\ell : \mathbb{R} \to \mathbb{R}$ $\quad \sigma_k(0) = 0$, σ_k L_k-Lip.

$B_1, \dots, B_\ell > 0$

- $\mathcal{F}_0 := \mathcal{G}$
- given $\mathcal{F}_0, \dots, \mathcal{F}_{j-1}$ $\quad (j = 1, \dots, \ell)$

$$\mathcal{F}_j := \left\{ N_{\sigma_j, w} \circ (f_1, \dots, f_m) : \begin{array}{l} m \in \mathbb{N} \\ |w_1| + \dots + |w_m| \le B_j \\ f_1, \dots, f_m \in \mathcal{F}_{j-1} \end{array} \right\}$$

$\mathcal{F}_\ell$: class of all ℓ-layer nets w/ activations $\sigma_1, \dots, \sigma_\ell$, weight size constraints $B_1, \dots, B_\ell$

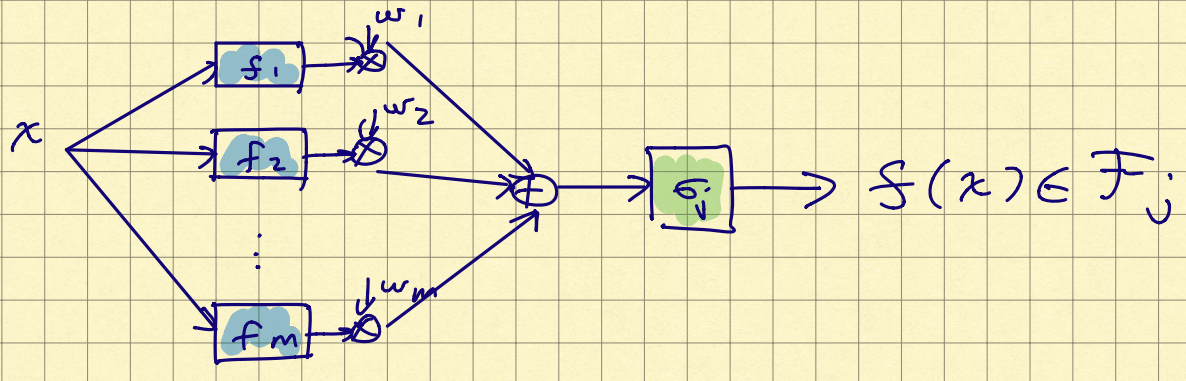
$$\boxed{R_n(\mathcal{F}_\ell) \le \, ?}$$

$\mathcal{F}_0 = \mathcal{G}$

$\mathcal{F}_1$: $\quad f(x) = \sigma_1\left(\sum_{k=1}^{m} w_k g_k(x) \right)$ $\quad \begin{array}{l} m \in \mathbb{N} \\ |w_1| + \dots + |w_m| \le B_1 \\ g_1, \dots, g_m \in \mathcal{G} \end{array}$

$\mathcal{F}_2$: $\quad f(x) = \sigma_2\left(\sum_{k=1}^{m} w_k f_k(x) \right), \quad f_1, \dots, f_m \in \mathcal{F}_1$

...



$\in \mathcal{F}_{j-1}$ $\quad \|w\|_1 \le B_j$

Naive bound: use contraction principle

$$\mathcal{F}_j = \sigma_j \circ \left(B_j \cdot \mathrm{absconv}(\mathcal{F}_{j-1}) \right)$$

$$R_n(\mathcal{F}_\ell) \le 2L_\ell B_\ell \cdot R_n(\mathcal{F}_{j-1})$$

$$\cdots$$

$$\le \prod_{j=1}^{\ell} (2L_j B_j) \cdot R_n(\mathcal{G})$$

– say, $L_1, \dots, L_\ell \le 1$

$$\prod_{j=1}^{\ell} (2B_j) = 2^\ell \prod_{j=1}^{\ell} B_j$$

needs to be $\le 2^{-\ell}$ for a bd that does not blow up w/ ℓ

Next lecture: $2^\ell \to \sqrt{\ell}$

(Golowich–Rakhlin–Shamir 2017)

- also see Bartlett–Foster–Telgarsky 2017