① Term Project.

- A Experimental
- B Theoretical

5-pages

① summary

② advocate / critic

This time: Boosting (Adaboost) → $\boxed{\text{NUS Prof (Generalization)}}$

Recap: ℓ_1-regular. (Lasso / Compressed Sensing)

① Mathemetical

② Graph Proof

$$\min_{\theta} \frac{1}{2n}\sum_t (y_t - \theta^T x_t)^2 + \lambda\|\theta\|_1 .$$

$\Updownarrow$ Equivalent.

$$\min_{\theta} \frac{1}{2n}\|y - X\theta\|^2 \quad \text{s.t } \|\theta\|_1 \le t \quad \text{for some } t>0$$

(LASSO)

Figure:

$\tilde{\theta} = (X^T X)^{-1} X^T y$

Sparse Solution

this probability is very small!

(Lebesgue Measure)

Boosting / Adaboost (Feature selection)

Decision stump (Axis-aligned linear separators)

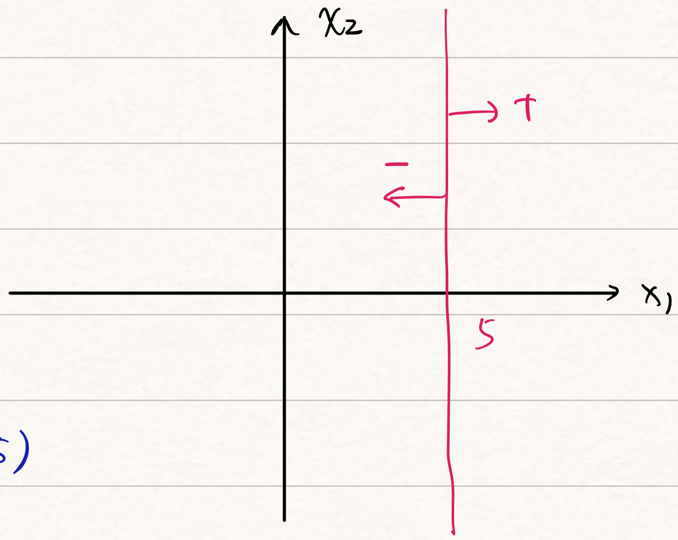
$$h(x; \underline{\theta}) = \text{sign}(s(x_k - \theta_0)), \quad \underline{x} \in \mathbb{R}^d \quad k \in \{1, \ldots, d\} \quad s \in \{\pm 1\}$$

$$\theta_0 \in \mathbb{R}$$

[Example]

$$\begin{cases} s = +1 \\ k = 1 \Rightarrow \text{Focus on } x_1 \\ \theta_0 = 5 \end{cases}$$

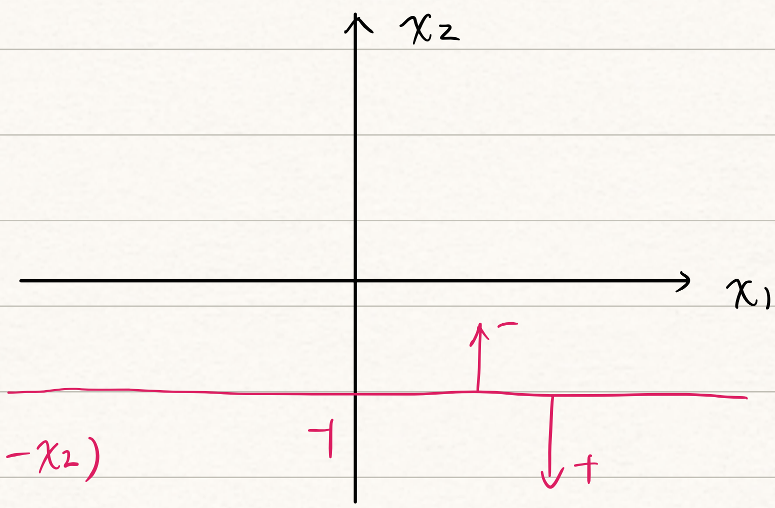
$h(\underline{x}, \underline{\theta}) = \text{sign}(x_1 - 5)$



[Example]

$$\begin{cases} s = - \\ k = 2 \\ \theta_0 = -1 \end{cases}$$

$h(x; \theta) = \text{sign}(-1 - x_2)$



Pretty Simple Classifier !

$\Downarrow$

Base Learner

Q: Can we "combine" several such $\boxed{\text{simple classifiers}}$ to form a "Strong" classifier ?

simple classifiers → "Weak"

simple classifiers → "Decision stumps"

~~Eg~~: $\phi(\underline{x}; \underline{\theta}) = [\, h(x_i; \underline{\theta_i}) : i=1, \dots, m \,]^T \in \{\pm 1\}^m \Rightarrow$ Consider $\underline{m}$ decision stumps

$\Downarrow$

$\underline{\theta_i} = \{ s_i, k_i, \theta_{0i} \} \Rightarrow$ Difficult to get (learn) parameters

Run a linear classifier based on $\phi(\underline{x}, \underline{\theta})$

Eg: Collect the output of all decision stumps into a SINGLE classifier.

$\Rightarrow \begin{cases} \text{Tractable} \\ \text{Generalization Well} \end{cases}$

$\Downarrow$

Ensemble $\rightarrow$ $h_m(\underline{x}) = \sum_{j=1}^{m} \alpha_j h(\underline{x}; \underline{\theta_j})$, $\underbrace{\alpha_j \geq 0 \ \& \ \sum \alpha_j = 1}_{\text{convex combin.}}$.

e.g. if $m=2$, and $h(\cdot; \underline{\theta_2})$ is more "reliable",

then we may choose $\alpha_2 = 0.8$, $\alpha_1 = 0.2$.

$\boxed{\text{Rmk}}$ Each stump has $\alpha_j \rightarrow$ "votes".

Ensemble $h_m(x)$ classifies a data $\underline{x}'$ according to the votes α_j of each stump $h(\cdot; \underline{\theta_j})$

Our Aim: Learn $\{(\underline{\theta_j}, \alpha_j)\}_{j=1}^{m}$ where $\underline{\theta_j} = \{s_j, k_j, \theta_{0j}\}$

Weak Learner $\xrightarrow{\text{Ensemble}}$ Strong Learner

↓

Stump

Exponential Loss → $\text{Loss}_{exp}(y, h(x)) = \exp(-y h(x)) = \exp(-z)$

label (y), prediction ($h(x)$)

Figure

$\text{Loss}_{0-1} \Rightarrow \begin{cases} \text{non-differentiable} \\ \text{non-convex} \end{cases} \Rightarrow \boxed{\text{BAD}}$

Rmk: $\text{Loss}_{0-1}(z) \leq \text{Loss}_{exp}(z)$

$\Downarrow$

The exp. loss is a SURROGATE to the 0-1 loss. $\Rightarrow \begin{cases} \text{convex} \\ \text{differentiable} \end{cases}$

Aim: Learn parameters θ_j and votes α_j **GREEDILY**.

$\Downarrow$ greedy alg.

Assume that we learn $(m-1)$-th weak classifier & their votes, i.e., $\{(\theta_j, \alpha_j)\}_{j=1}^{m-1}$. These are $\boxed{\text{fixed}}$ for learning the m-th term $\boxed{\alpha_m \cdot h(\cdot\,; \theta_m)}$

Consider the ensemble:

$$h_m(x) = \sum_{j=1}^{m} \alpha_j\, h(x; \theta_j)$$

$$= \left[\sum_{j=1}^{m-1} \alpha_j\, h(x; \theta_j) \right] + \alpha_m\, h(x; \theta_m)$$

$$= h_{m-1}(x) + \alpha_m\, h(x; \theta_m)$$

Compute the exp. loss on a given dataset $\mathcal{D}_n = \{(x_t, y_t)\}_{t=1}^{n}$

ensemble learner

$$\boxed{J(\underbrace{\alpha_m}_{\text{vote}}, \underbrace{\theta_m}_{\text{stump}})} = \sum_{t=1}^{n} \text{loss}_{\exp}(y_t, h_m(x_t))$$

ensemble at m-th iter

$$= \sum_{t=1}^{n} \exp(-y_t\, h_m(x_t))$$

Our aim is to minimize this!

ensemble at (m-1)-th iter

$$= \sum_{t=1}^{n} \exp(-y_t\, h_{m-1}(x_t) - y_t\, \alpha_m\, h(x_t; \theta_m))$$

Loss → t & m-1

$$= \sum_{t=1}^{n} W_{m-1}(t)\, \exp(-y_t\, \alpha_m\, h(x_t; \theta_m))$$

$$\Downarrow$$

$W_{m-1}(t) := \exp(-y_t\, h_{m-1}(x_t))$

"weights" Associated to data point (x_t, y_t) after (m-1)-th iteration

Note: $W_m(t) = W_{m-1}(t) * \exp(-y_t\, \alpha_m\, h(x_t; \theta_m))$

AdaBoost Alg.

$\Rightarrow$ Input: $\mathcal{D}_n = \{(x_t, y_t)\}_{t=1}^n$

Loss $= \exp(-yz)$

$\boxed{z = \sum \alpha_n g_m(x)}$

- Initialize weight: $\underline{W_0(t)} = \frac{1}{n}$ For all t.

we have $\underline{(\alpha_i, \hat{\theta}_i)}$ $i = 1, 2, \dots, m-1$ (stump)

- At Boosting stage m, find a base learner $h(\cdot\,; \hat{\theta}_m)$

that minimize:

$$\hat{\theta}_m = \arg\min_{\theta_m} - \sum_{t=1}^n \tilde{W}_{m-1}(t)\, y_t\, h(x_t; \theta_m)$$

weighted trg loss.

$\Downarrow$

Learn the Stump.

$$\tilde{W}_{m-1}(t) = \frac{W_{m-1}(t)}{\sum_t W_{m-1}(t)}$$

normalized weight

$$\boxed{\hat{\alpha}_m = \frac{1}{2}\ln\left(\frac{1-\hat{\varepsilon}_m}{\hat{\varepsilon}_m}\right)}$$

$\hat{\varepsilon}_m =$

- Choose vote $\hat{\alpha}_m \in \mathbb{R}$ using a formula (comes from Greedy)

- Update the weights $\tilde{W}_m(t) = \dfrac{\tilde{W}_{m-1}(t)}{Z_m} \exp(-y_t h(x_t, \hat{\theta}_m)\hat{\alpha}_m)$

$$= \bigcirc \begin{cases} \exp(\alpha_m), & \text{inequal} \\ \exp(-\alpha_m), & \text{equal} \end{cases}$$

Note: ①

$$\boxed{- \sum_{t=1}^n \tilde{W}_{m-1}(t)\, y_t\, h(x_t; \theta_m)}$$

$\Rightarrow$ Motivation

Claim: $-y_t h(x_t; \theta_m) = 2 \cdot \mathbb{1}\{y_t \neq h(x_t; \theta_m)\} - 1$

pf: ① $y_t = h(x_t; \theta_m) \Rightarrow$ LHS $= -1$ RHS $= -1$

② $y_t \neq h(x_t; \theta_m) \Rightarrow$ LHS $= +1$ RHS $= +1$

$$-\sum_{t=1}^{n} \tilde{W}_{m-1}(t)\, y_t\, h(x_t; \underline{\theta}_m) \begin{cases} y_t = h \Rightarrow 1 \\ y_t \neq h \Rightarrow -1 \end{cases}$$

$$= \sum_{t=1}^{n} \tilde{W}_{m-1}(t) \left[2 \cdot \mathbb{1}\{ y_t \neq h(x_t; \theta_m) \} - 1 \right]$$

$$= \sum_{t=1}^{n} 2\, \tilde{W}_{m-1}(t)\, \mathbb{1}\{ y_t \neq h(x_t; \theta_m) \} - 1 \quad \left(\text{since } \sum_{1}^{n} \tilde{W}_{m-1}(t) = 1\right)$$

$$:= 2\hat{\varepsilon}_m - 1.$$

$$\hat{\varepsilon}_m = \sum_{t=1}^{n} \tilde{W}_{m-1}(t)\, \mathbb{1}\{ y_t \neq h(x_t, \hat{\theta}_m) \}$$

weighted training loss $= \sum_{t:\text{ misclassified}} \tilde{W}_{m-1}(t)$

Rmk: Since $\tilde{W}_0(t) = \frac{1}{n}$, $t = 1, 2, \ldots, n$. $\varepsilon_1 =$ trg error $= \sum_{t: y_t \neq h(x_t, \theta_1)} \frac{1}{n}$

② Update Rule of Weights (when we attain $\hat{\theta}_m$)

$$\tilde{W}_m(t) = \frac{\tilde{W}_{m-1}(t)}{Z_m} \exp\big(\overbrace{-y_t h(x_t; \underline{\hat{\theta}}_m)}^{\in \{\pm 1\}}\, \hat{\alpha}_m\big)$$

$$= \frac{\tilde{W}_{m-1}(t)}{Z_m} \times \begin{cases} e^{-\hat{\alpha}_m}, & \text{if } y_t = h(x_t, \hat{\theta}_m) \\ e^{+\hat{\alpha}_m}, & \text{if } y_t \neq h(x_t, \hat{\theta}_m) \end{cases}$$

increase weight

$\Rightarrow$ So, if $\hat{\alpha}_m > 0$ (usual case), trg examples that are misclassified by $h(\cdot\,;\hat{\theta}_m)$ are given higher weights in the next boosting stage.

Final classifier $\rightarrow$ $h_m(x) = \sum_{j=1}^{m} \hat{\alpha}_j \, h(x\,;\hat{\theta}_m)$

Q: How to choose the votes $\hat{\alpha}_j$

$\Downarrow$

$$\hat{\alpha}_m = \frac{1}{2}\ln\left(\frac{1-\hat{\varepsilon}_m}{\hat{\varepsilon}_m}\right) \qquad \hat{\varepsilon}_m = \sum_{t=1}^{n} \tilde{W}_{m-1}(t)\, \mathbb{1}\{y_t \neq h(x_t\,;\hat{\theta}_m)\}$$

$\hat{\varepsilon}_m$: weighted training error when we consider the optimized stump $h(\cdot\,;\hat{\theta}_m)$ $\Rightarrow$ m-th iteration (update)

Rmk: ① $\hat{\varepsilon}_m = 0 \Rightarrow \hat{\alpha}_m = +\infty$ $\longleftrightarrow$ put all weights on classifier $h(\cdot\,;\hat{\theta}_m)$

$\updownarrow$ $\qquad\qquad \Updownarrow$

$h(\cdot\,;\hat{\theta}_m)$ perfectly classifies all training samples

② $\hat{\varepsilon}_m = \frac{1}{2} \Rightarrow \hat{\alpha}_m = 0$ $\longleftrightarrow$ No weights on classifier $h(\cdot\,;\hat{\theta}_m)$

$\updownarrow$

$h(\cdot\,;\hat{\theta}_m)$ is 'maximally confused'

How does AdaBoost Perform?

$\Downarrow$

Guarantee by a theorem:

[THM]: If each base classifier $h(\cdot\,; \hat{\theta}_j)$ is slightly better than random guessing, i.e., $\hat{\varepsilon}_j < \frac{1}{2}$

$\Downarrow$

$\sum \widetilde{W}_{j-1}(t)$

t: classified wrongly on $h(\cdot\,; \hat{\theta}_j)$

$\Downarrow$

the training error decrease to 0 exponentially,

i.e., $\frac{1}{n}\sum_{t=1}^{n} \mathbb{1}\{y_t h_m(x_t) \le 0\} \le \exp\left(-2\sum_{j=1}^{m}\left(\frac{1}{2}-\hat{\varepsilon}_j\right)^2\right)$

$\Downarrow$

In particular, if there exists γ s.t.

$\frac{1}{2} - \hat{\varepsilon}_j \ge \gamma$

then $\frac{1}{n}\sum \mathbb{1}\{y_t h_m(x_t) \le 0\} \le \exp(-2m\gamma^2)$

$\Downarrow$

training error (0-1)