4/15/15

ENSEMBLE LEARNING

(1) Main Idea

→ Weak/Simple Learners (NB, LR, Linear SVMs, decision stumps)
  - → Underfit
  - → High bias
  - → Low Variance

→ Strong Learners (Kernel SVMs, Deep Neural Nets, Deep Decision Trees)
  - → Overfit
  - → Low Bias
  - → High Variance

Ensemble Methods: Can we get the best of both?

Bagging
→ Reducing the variance of (multiple) strong learners

Boosting
→ Reducing the bias of (multiple) weak learners

(2) Bagging ≡ Bootstrap Averaging

Train - Time	Test - Time
$D = \{(\vec{x}_1, y_1), \ldots (x_N, y_N)\}$	$\vec{x}$
$I = \{1, 2, 3, \ldots N\}$ [Set of indices]	$\hat{y}^{(1)} = h_1(\vec{x}) \quad \cdots \quad \hat{y}^{(M)} = h_M(\vec{x})$
Bootstrap Sampling [Sample with replacement N times]	Committee
$\xrightarrow{1} \quad \xrightarrow{2} \ldots \xrightarrow{M}$	
$I^{(1)} = \{1, 2, 3, \ldots\}_N \quad I^{(2)} \quad I^{(M)} = \{5, 5, 2, \ldots\}_N$	$\hat{y}^{com} = \frac{1}{M} \sum_{m=1}^{M} y^{(m)}$ [Regression]
$D^{(1)} \quad D^{(2)} \ldots D^{(M)}$	
$\downarrow$ Train $\quad \downarrow$ Train $\quad \downarrow$ Train	
$\hat{y}^{(1)} = h_1(\vec{x}) \quad \hat{y}^{(M)} = h_M(\vec{x})$	$= \underset{c}{\text{argmax}} \sum_{m=1}^{M} I(y^{(m)} = c)$ [Classification Majority Vote]

Why? Because committees make better predictions [under certain assumptions]

$$E_{P(X,Y)}\left[L(y, \hat{y}^{com}(x))\right]$$

$$= E\left[\left(y - \frac{1}{M}\sum_{m=1}^{M}\hat{y}^{(m)}\right)^2\right] \quad \text{[For Regression w/ } L_2\text{-Loss]}$$

$$= \frac{1}{M^2} E\left[\left(My - \sum_{m=1}^{M}\hat{y}^{(m)}\right)^2\right]$$

Let $e_m = y - \hat{y}^{(m)}$ Error of m^{th} predictor

$$E_{com} = \frac{1}{M^2} E\left[\left(\sum_{m=1}^{M} e_m\right)^2\right]$$

$$= \frac{1}{M^2} E\left[\sum_{m=1}^{M} e_m^2 + \sum_{m_1 \neq m_2} e_{m_1} e_{m_2}\right]$$

Now, Assume: ① Zero mean errors $E[e_m] = 0 \quad \forall m$

② Uncorrelated errors $E[e_{m_1} e_{m_2}] = E[e_{m_1}] \cdot E[e_{m_2}]$

$$E_{com} = \frac{1}{M^2}\left[\sum_{m=1}^{M} E[e_m^2] + \sum_{m_1 \neq m_2} \underbrace{E[e_{m_1}]}_{0} \underbrace{E[e_{m_2}]}_{0}\right]$$

$$= \frac{1}{M^2} \sum_{m=1}^{M} E[e_m^2] = \frac{1}{M^2} \sum_{m=1}^{M} E[(y - y^{(m)})^2]$$

$$\Rightarrow \boxed{E_{com} = \frac{1}{M} E_{av}}$$

$\Rightarrow$ Large (uncorrelated-error) committees cut error! $\frac{1}{M}$th of average error!

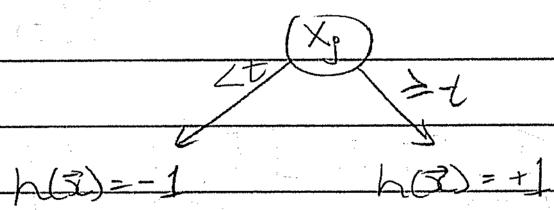
Assumption ① is okay. Why?

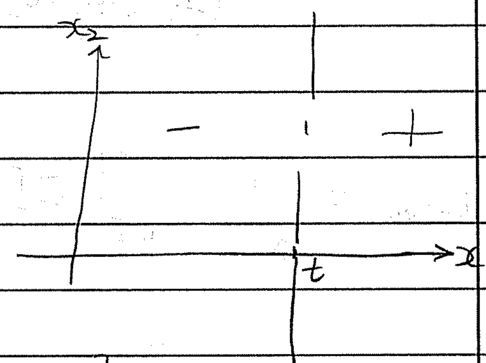
Assumption ② is difficult to satisfy in practice. True for bootstrap?

③ Boosting: Reducing Bias of Weak Learners
i.e. making weak learners strong.

Given: Hypothesis Class $\mathcal{H} = \{ h \mid h : X \to Y \}$
collection of weak learners

Example #1: Decision Stump [for binary classification]

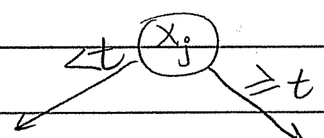




$$\mathcal{H} = \{ (j, t) \mid j \in \{1, 2, \ldots, d\}, t \in \mathbb{R} \}$$

Pick a feature & Pick a threshold
And you have a decision-stump classifier

Example #2: Decision Stump [for Regression]

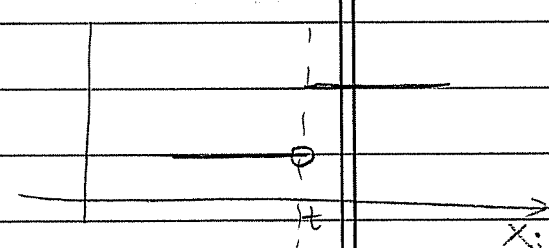


$L = \{$ set of training data-pts w/ $x_j < t \}$

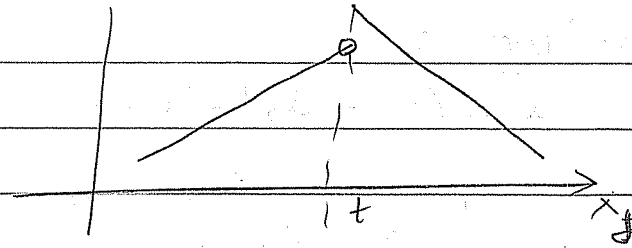
$R = \{$ set of training data-pts w/ $x_j \geq t \}$

$$h(\vec{x}) = \begin{cases} \frac{1}{|L|} \sum_{i \in L} y_i & x_j < t \\ \frac{1}{|R|} \sum_{i \in R} y_i & x_j \geq t \end{cases}$$

[Bucket Average]
0^{th}-order polynomial fit



$$h(\vec{x}) = \begin{cases} \vec{w}_L^T \vec{x} & x_j < t \\ \vec{w}_R^T \vec{x} & x_j \geq t \end{cases}$$



Assume: Access to a Black-Box Learning algorithm to pick a weak learner for a dataset, i.e.:

$$\boxed{h^* = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} L(y_i, h(\vec{x}_i))}$$

can be solved

Goal of Boosting: Learn a committee of weak-learners with low-loss

$$f(\vec{x}) = \sum \alpha_t h_t(\vec{x})$$

$$\underbrace{\min_{\alpha_t, h_t}} \frac{1}{N} \sum_{i=1}^{N} L(y_i, f(\vec{x}_i))$$

Problem: Joint optimization of all weak learners is difficult

Boosting involves greedy optimization.

At "time" t

$$\underbrace{\alpha_1 h_1(x) + \alpha_2 h_2(x) + \dots + \alpha_{t-1} h_{t-1}(x)}_{\text{fixed}} + \underbrace{\alpha_t h_t(x)}_{\text{Learnt / Optimized}}$$

$$f_t(x) = f_{t-1}(x) + \alpha_t h_t(x)$$

So

$$\alpha_t^*, h_t^* = \min_{\substack{\alpha_t \in \mathbb{R} \\ h_t \in \mathcal{H}}} \frac{1}{N}\sum_{i=1}^{N} L(y_i, f_t(x_i))$$

L_2-BOOST : Assume Regression & L_2-error

$$(\alpha_t^*, h_t^*) = \min_{\substack{\alpha_t \in \mathbb{R} \\ h_t \in \mathcal{H}}} \frac{1}{N}\sum_{i=1}^{N} \left[y_i - f_t(\vec{x}_i) \right]^2$$

$$= \frac{1}{N}\sum_{i=1}^{N} \left[\underbrace{y_i - f_{t-1}(\vec{x}_i)}_{\text{Current-Error}} - \alpha_t h_t(\vec{x}) \right]^2$$

$$= \frac{1}{N}\sum_{i=1}^{N} \left[e_i - \alpha_t h_t(\vec{x}_i) \right]^2$$

Usually α_t scalar can be learned inside h_t so we set $\alpha_t^* = 1$

$$\boxed{= \min_{h_t \in \mathcal{H}} \frac{1}{N}\sum_{i=1}^{N} \left[e_i - h_t(x_i) \right]^2}$$

Black-Box Call: Regress to Errors!

Train weak learner $h(\vec{x})$ with GT $y_i = e_i$