$\boxed{\text{GBDT}}$ Framework

→ $\boxed{\text{Idea}}$: $\underline{G_m(x) = G_{m-1}(x) + g_m(x)}$ Loss function: $L(G_m, y)$

① $L(G_m, y)$ on data point (x, y)

$= L(G_{m-1}(x) + g_m(x), y)$

$\boxed{\text{Aim:}}$ $g_m(x) = \underset{g}{\operatorname{argmin}}\ L(G_{m-1}(x) + g(x), y) \quad \forall x$

② $L(G_{m-1}(x) + g(x), y) \approx \left.\frac{\partial L(G, y)}{\partial G}\right|_{G = G_{m-1}(x)} \cdot g(x) + L(G_{m-1}(x), y)$

↓ $\boxed{\text{first-order}}$

recommend function $g_m(x)$ should fit to $-\left.\frac{\partial L(G, y)}{\partial G}\right|_{G = G_{m-1}(x)}$

$\boxed{\text{Alternative Interpretation}}$

this formulation give us the choice of LSE

$$L(G_{m-1} + g(x), y) \approx L(G_{m-1}(x), y) + \left.\frac{\partial L(G, y)}{\partial G}\right|_{G = G_{m-1}(x)} g(x) + \frac{1}{2} g(x)^2$$

$$= \text{const} + \frac{1}{2}\left(g(x) + \left.\frac{\partial L(G, y)}{\partial G}\right|_{G = G_{m-1}(x)}\right)^2$$

$$\propto \left[-g(x) - \left.\frac{\partial L(G, y)}{\partial G}\right|_{G = G_{m-1}(x)}\right]^2$$

⇒ as for GBDT, use LSE to achieve $g_m(x)$ → m-th base classifier

---

$\boxed{\text{XGBoost}}$ $\underline{G_m(x) = G_{m-1}(x) + g_m(x)}$ Loss function: $L(G_m, y)$

for simplicity, define $\begin{cases} g^m(x) = \left.\frac{\partial L(G, y)}{\partial G}\right|_{G = G_m(x)} \\ h^m(x) = \left.\frac{\partial^2 L(G, y)}{\partial G^2}\right|_{G = G_m(x)} \end{cases}$

↓

Similar Non-parametric Framework

① $g_m(x) = \underset{g(x)}{\operatorname{argmin}}\ L(G_{m-1}(x) + g(x), y)$

② $L(G_{m-1}(x) + g(x), y)$

$$\approx L(G_{m-1}(x), y) + g^m(x)\cdot g(x) + \frac{1}{2} h^m(x)\cdot g(x)^2$$

$$\propto \left(g^m(x)\, g(x) + \frac{1}{2} h^m(x)\, g(x)^2 \right)$$

$$\propto h^m(x) \left[\frac{g^m(x)}{h^m(x)} + g(x) \right]^2$$

$\boxed{\text{consider the } L(G(x), y) \text{ on total dataset}}$

$$L_{total} \approx \sum_n h^m(x_n) \left[-\frac{g^m(x_n)}{h^m(x_n)} - g(x_n) \right]^2$$

$$\propto \sum_n w^m(x_n) \left[-\frac{g^m(x_n)}{h^m(x_n)} - g(x_n) \right]^2$$

where $\boxed{w^m(x_n) = \frac{h^m(x_n)}{\sum_{n'} h^m(x_{n'})}}$

$\boxed{\text{weighted LSE}}$ compared with GBDT (LSE)

$\boxed{\text{AdaBoost}}$ → $\boxed{\text{for classification}}$

First. Conclusion AdaBoost ← { XGBoost with exponential loss to achieve $g_m(\cdot)$; use primal formulation without Taylor approximation on α

$$L_{exp}(G(x), y) = \exp(-y\, G(x))$$

① $$g^m(x) = \left. \frac{\partial L_{exp}(G, y)}{\partial G} \right|_{G = G_{m-1}(x)}$$

$$= -y \exp(-y\, G_{m-1}(x))$$

$$h^m(x) = \left. \frac{\partial^2 L_{exp}(G, y)}{\partial G^2} \right|_{G = G_{m-1}(x)}$$

$$= y^2 \exp(-y\, G_{m-1}(x))$$

$$= 1 \cdot \exp(-y\, G_{m-1}(x)) = \exp(-y\, G_{m-1}(x))$$

② suppose that additive model is $G_m(x) = G_{m-1}(x) + \alpha_m g_m(x)$

with XGBoost Framework, we have :

$$(\alpha_m, g_m) = \arg\min_{\alpha, g} \sum_{n=1}^{N} w^m(x_n) \left[- \frac{g^m(x_n)}{h^m(x_n)} - \alpha\, g(x_n) \right]^2$$

$$= \arg\min_{\alpha, g} \sum_{n=1}^{N} w^m(x_n) \left[y_n - \alpha\, g(x_n) \right]^2$$

$$= \arg\min_{\alpha, g} \sum_{n=1}^{N} w^m(x_n) \left[1 + \alpha^2 \overset{1}{g^2(x_n)} - 2\alpha\, y_n\, g(x_n) \right]$$

$$= \boxed{\arg\min_{\alpha, g} \sum_{n=1}^{N} w^m(x_n) \left[\alpha^2 - 2\alpha\, y_n\, g(x_n) \right]}$$

$$\text{where } w^m(x_n) = \frac{h^m(x_n)}{\sum_{n'} h^m(x_n)} = \frac{\exp(-y_n G_{m-1}(x_n))}{\sum_{n'} \exp(-y_{n'} G_{m-1}(x_{n'}))}$$

observe that, we can minimize on $g(\cdot)$ first for the separability

$$g_m = \arg\min_{g} \sum_{n=1}^{N} w^m(x_n) \cdot (- y_n\, g(x_n))$$

$$= \arg\min_{g} \sum_{n=1}^{N} w^m(x_n) \cdot \frac{1 - y_n\, g(x_n)}{2}$$

$$= \arg\min_{g} \sum_{n=1}^{N} w^m(x_n)\, \mathbb{1}\{ y_n \neq g(x_n) \}$$

What we do in Adaboost!

minimize weighted traing error to achieve $g_m(\cdot)$

then, we use the first objective (without second-order approximation) to optimize on α! (this is more trivial part)

$$\alpha_m = \arg\min_{\alpha} \sum_{n=1}^{N} \exp(-y_n G_{m-1}(x_n)) \cdot \exp(-y_n\, \alpha\, g_m(x_n))$$

$$\Downarrow$$

we first-order condition, we can achieve

that $\boxed{\alpha_m = \log\left(\frac{1-\varepsilon_m}{\varepsilon_m}\right)} \leftarrow \boxed{\varepsilon_m = \sum_{i=1}^{N} w^m(x_i)\, \mathbb{1}\{ y_i \neq g_m(x_i) \}}$

Rmk: this is what we do in GBDT framework with step α,

→ ① we generate the m-th optimal classifier by Taylor Approximation Loss function
② we generate the m-th optimal seep length α_m by Loss function without Taylor Approximation

☆

⇒ To conclude, Adaboost can be viewed as a Combination of { XGBoost, GBDT }

in the sense that
- { use 2-order Taylor Approximation on exp-loss to achieve the m-th base classifier → **XGBoost Part**
- { after achieving classifier, back to exp-loss without Taylor Approx. to directly solve for optimal seep length! → **GBDT Part**

Another View Point of AdaBoost → with no Taylor Approximation

1) Model: $G_m(x) = G_{m-1}(x) + \alpha_m g_m(x)$

2) Loss Function: $L_{exp}(y_i, G_m(x_i)) = \exp(-y_i G_m(x_i))$

$$R_{emp}(G_m) = \sum_{n=1}^{N} L_{exp}(y_n, G_m(x_n))$$

$$= \sum_{n=1}^{N} \exp(-y_n G_{m-1}(x_n)) \cdot \exp(-y_n \alpha_m g_m(x_n))$$

$$:= \sum_{n=1}^{N} w_n^{(m)} \exp(-y_n \alpha_m g_m(x_n))$$

$$= \sum_{n=1}^{N} w_n^{(m)} \exp(-\alpha_m) \mathbb{1}\{y_n = g_m(x_n)\}$$

$$+ \sum_{n=1}^{N} w_n^{(m)} \exp(\alpha_m) \mathbb{1}\{y_n \neq g_m(x_n)\}$$

$$= \exp(-\alpha_m) \sum_{n=1}^{N} w_n^{(m)} \mathbb{1}\{y_n = g_m(x_n)\}$$

$$+ \exp(\alpha_m) \sum_{n=1}^{N} w_n^{(m)} \mathbb{1}\{y_n \neq g_m(x_n)\}$$

$$= (\exp(\alpha_m) - \exp(-\alpha_m)) \cdot \sum_{n=1}^{N} w_n^{(m)} \mathbb{1}\{y_n \neq g_m(x_n)\}$$

$$+ \exp(-\alpha_m) \sum_{n=1}^{N} w_n^{(m)}$$

$$\longrightarrow \boxed{(\hat{\alpha}_m, \hat{g}_m) = \underset{\alpha_m, g_m}{\operatorname{argmin}} \; R_{emp}(G_m)}$$

$\Rightarrow$ ① $$\hat{g}_m = \underset{g_m}{\operatorname{argmin}} \sum_{n=1}^{N} w_n^{(m)} \mathbb{1}\{y_n \neq g_m(x_n)\}$$

② $$\hat{\alpha}_m = \log\left(\frac{1-e_m}{e_m}\right) \qquad e_m := \frac{\sum_{n=1}^{N} w_n^{(m)} \mathbb{1}\{y_n \neq g_m(x_n)\}}{\sum_{n=1}^{N} w_n^{(m)}}$$

③ update formula for $w_n^{(m)}$:

$$w_n^{(m+1)} = \begin{cases} w_n^{(m)} \cdot \exp(-\alpha_m) & \text{if } g_m(x_n) = y_n \\ w_n^{(m)} \exp(\alpha_m) & \text{if } g_m(x_n) \neq y_n \end{cases}$$