Lecture 6/25

- HW3 demo
- next: chapter 4
  - boosting / ensemble
  - Multilabel data, in particular ECOC
  - Feature selection
    - Activelearning
    - Feature aggregation / PCA
    - Features from similarity / TsNE
    - HW4
- chapter 5: SVM + kernels
- chapter 6: advanced NN
  - CNN - RecNN vs Transformers

"random forrests"

$H_t(x)$ = score produced by t^{th} weak learner (not probab, not class prediction)

Boosting (weak learner) = additive ensemble

datapoint $x = (x^1\ x^2\ \dots\ x^d)$ x_i

h_t $t = 1:T$

$$F(x) = \sum_{t=1}^{T} h_t(x)$$

final score

$h(x)$ weak learner

weak scores

acc $\simeq$ 65% better than random but poor

default: shallow Dectree

dec stump: 1-split DT

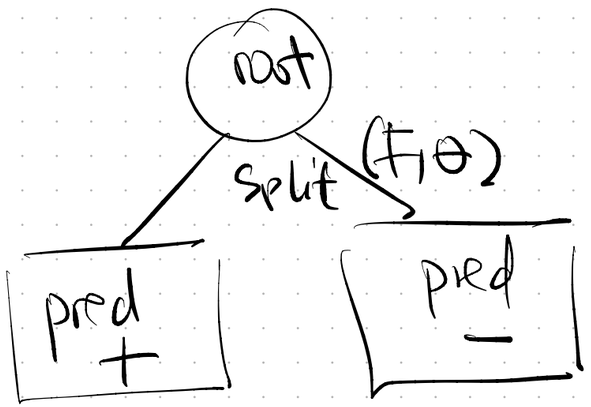
- Regression mode label y = quantity

want $(F(x) - y)^2 \simeq 0$

all i $\sum_{i=1}^{N} (F(x_i) - y_i)^2 \simeq 0$

- Classification mode $y \in \{+1, -1\}$ $y^* \in \{0, 1\}$ $y^* = \frac{1+y}{2}$

$F(x)$ score $\xrightarrow{\text{map}}$ $\begin{cases} pr(y=1|x) \\ p(y=-1|x) = 1 - p(y=1|x) \end{cases}$



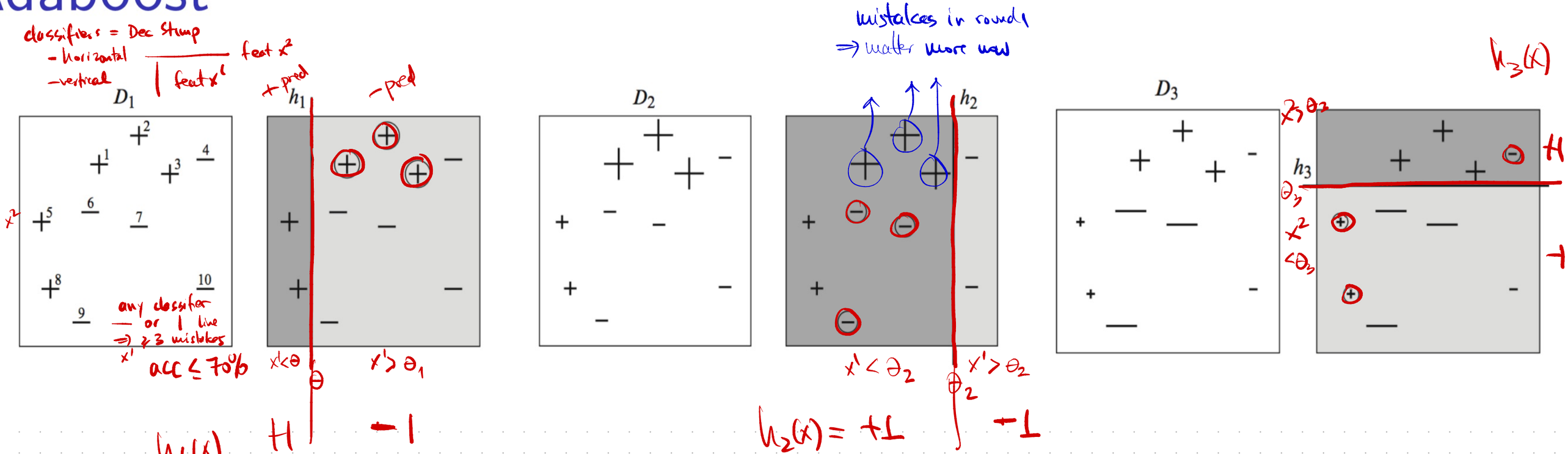
want max likelihood $\prod_{i=1}^{N} p(y^*|x)$ $p(y^*=1|x)^{y^*} \cdot p(y^*=0|x)^{1-y^*}$ filter which probab

intuition cca 1992: F(x) cannot be much better than h(x) weak learners.
VERY WRONG

Gradient Boosting = Gradient Descent + Boosting

Adaboost

classifiers = Dec Stump
- horizontal
- vertical



"complexity" increases linearly in T $\Theta(T)$ ⇒ not that much!

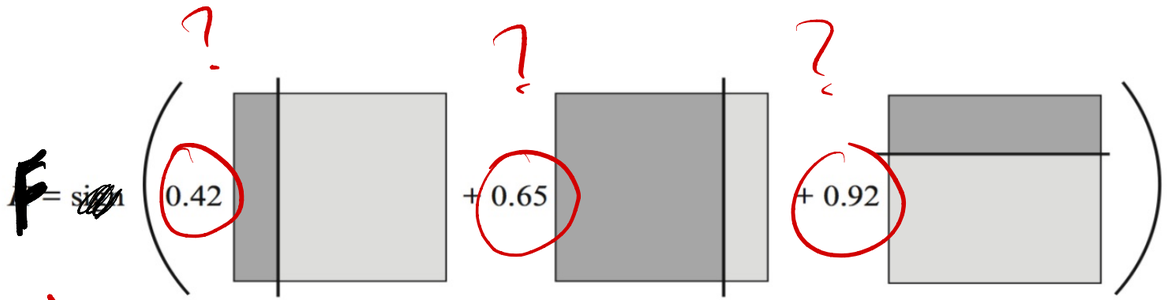
$F(x) = h_1(x) + h_2(x) + \dots + h_T(x)$ ensemble of weak learners $h(\cdot)$

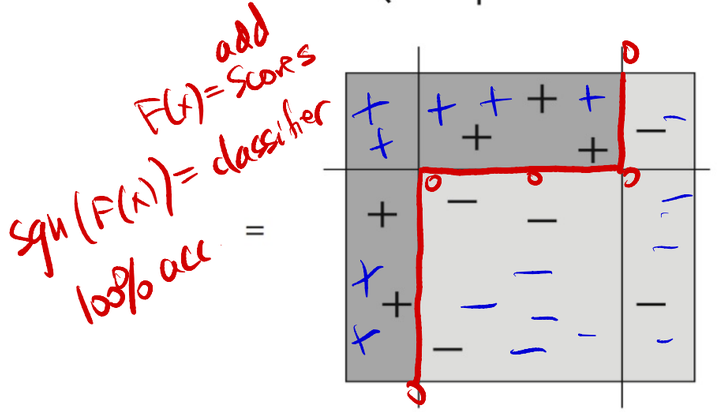
$h_t(x)$ = dec stump = 1-split decision tree.

Comparison: one big tree with T splits ⇒ complexity = $\Theta(2^T)$ ⇒ ⇒ overfit!

$$F(x) = \sum_t \rho_t h_t(x)$$

fixed coefficients ρ_t for $h_t(x)$





2 ways to implement boosting

1st (historically 1996): ADABOOST

- look at current w.learner $h_t(x)$
- reweight data $D_{t+1}(x_i)$ = distribution
  = importance of datapoint x_i next round
- train $h_{t+1}()$ = weak learner with error weighted by $D_{t+1}()$

$$e_{t+1} = \sum_{i=1}^{N} 1_{[h_{t+1}(x_i) \neq y_i]} \cdot D_{t+1}(x_i)$$

acc { 1 mistake / 0 correct

weight for round $t+1$

- add $h_{t+1}()$ to $F(x)$?

$$F(x) = F(x) + h_{t+1}(x)$$

$$F(x) = h_1(x) + h_2(x) \dots + h_{t+1}(x)$$

Key: After $t = 1 : T$ rounds of AdaBoost

weight on x $\quad D_{T+1}(x) \simeq$ # rounds t

where x mistake $h_t(x) \neq y$

$$\simeq \exp\left(\cdot \sum_{t=1}^{T} \alpha_t \left[h_t(x) \cdot -y \right] \right)$$

correct: $-1 : h_t(x) = y$

incorrect: $+1 : h_t(x) \neq y$

- $D_{T+1}(x)$ small $\Rightarrow$ most classif $h_t(x) = y$

$D_{T+1}(x)$ = large $\Rightarrow$ next classifier $h_{t+1}(x)$ focus on datapoint x

- $D_{T+1}(x)$ = large $\Rightarrow$ most classif $h_t(x) \neq y$

(Th) proper weights, $h()$ training etc (see Adaboost details) training can finish in one of these 2 ways.

- either train_error $\simeq 0$
  $F(x)$ ensemble

$$\sum_{i=1}^{N} 1_{[F(x_i) \neq y_i]} \longrightarrow 0$$

- or stuck: $D_{TH}()$ dist of point importance for next round is s.t. the best $h_{TH1}(x)$ classifier is random

$$e_{TH1} = \sum_{i=1}^{N} 1_{[h_{TH1}(x) \neq y_i]} \cdot D_{TH1}(x_i) \simeq 0.5$$

as long as $h_{TH1}(x)$ makes some progress (better than 0.5) training error ($F(x)$) has to go to zero.

Gradient Boosting.

2nd way much better

$F(x) = h_1(x) + h_2(x) + \cdots + h_T(x)$

want $F(x) \simeq y$ (regression mode) "house prices"

next weak learner $h_{T+1}(x) = F_{new}(x) - F_{current}(x)$

error $\frac{1}{2}(F(x) - y)^2 = J(x)$

diff w.r.t $F(x)$ $= F(x) - y = \frac{\partial J}{\partial F()}$

next $h_{T+1}(x)$ should watch residual gradient

Grad descent mode for $F(x)$ as variable ?? simple $\lambda = 1$

$F_{new}(x) = F(x) - \lambda \cdot \frac{\partial J}{\partial F()} = F(x) + y - F(x)$

trivial? want $F_{new}(x) = y$ (wrong thinking)

GD update : $F_{new}(x) = F(x) +$ $h_{T+1}(x) \simeq y - F(x)$ ← res grad

want the next weak learner

$$h_{T+1}(x) \simeq y - F(x)$$

train T+1 tree on (x, label $y - F(x)$) x=1:N

- if $h_{T+1}(x) \simeq y - F(x)$ well $\Rightarrow$

next w. lear

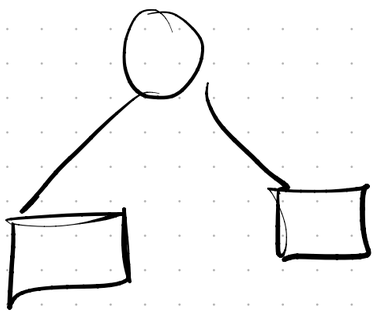
residual label (quantity)

$$F_{new}(x) = F(x) + h_{T+1}(x) \overset{\text{well}}{\simeq} y \quad \text{(good)}$$

- practice: train $h_{T+1}()$ to watch $h_{T+1}(x) \simeq y - F(x)$

new label

weak learner "dec stump"



- req updating label after every round

$y_{new} = y_{org} - F(x)$

- does not require weights like Adaboost, modif train procedure

Train $h_{T+1}(x)$: DecTreeAlg$(x, y - F(x))$

Classification $y \in \{+1, -1\}$ ⟺ $y^* = \frac{1+y}{2} \in \{1, 0\}$ — binary

$$F(x) = \sum_{t=1}^{T} h_t(x)$$

score — one F

$$p(x) = p(y^*=1|x) = \frac{e^{F(x)}}{e^{F(x)} + e^{-F(x)}}$$

$$p(\bar{x}) = p(y^*=0|x) = \frac{e^{-F(x)}}{e^{F(x)} + e^{-F(x)}} = 1 - p(x)$$

$$\text{loglikelihood} = \log\left(\prod_{i=1}^{N} p(x)^{y^*} \cdot (1-p(x))^{1-y^*} \right)$$

filter y^*: 0 / 1 — which probab.

want MAX

GD thinking ⟹ fit next weak learner $h_{T+1}(x) \simeq$ residual gradient ? $\frac{\partial \text{loglikelihood}}{\partial F(\cdot)}$

$F_{new} = F + h_{T+1}(x)$

L categories:

$y \in \{1, 2, \dots, L\}$

y-dist = 1/0/0/ - 0

pred dist $p_1(x)\ p_2(x) \cdots p_L(x)$

L $F_1(x)\ F_2(x) \cdots F_L(x)$

scores:

$$p(x) = p(y^*=1|x) = \frac{e^{F(x)}}{e^{F(x)}+e^{-F(x)}} \quad \Big\} \Rightarrow \quad F(x) = \frac{1}{2}\log\frac{p(x)}{1-p(x)} = F$$

$$1-p(x) = \frac{e^{-F(x)}}{e^{F(x)}+e^{-F(x)}}$$

(given x data)

x fixed

$$p(x)^{y^*}\cdot(1-p(x))^{1-y^*} = p(x)^{\frac{1+y}{2}}(1-p(x))^{\frac{1-y}{2}} = \frac{1}{1+e^{-2yF(x)}}$$

later

$F = F(x)$

$$\left(\frac{e^F}{e^F+e^{-F}}\right)^{\frac{1+y}{2}}\cdot\left(\frac{e^{-F}}{e^F+e^{-F}}\right)^{\frac{1-y}{2}} = \frac{(e^F)^{\frac{1+y}{2}+\frac{y-1}{2}}}{e^F+e^{-F}} = \frac{e^{Fy}}{e^F+e^{-F}}$$

$$= \frac{1}{\frac{e^F+e^{-F}}{e^{Fy}}} = \frac{1}{\frac{e^{Fy}+e^{-Fy}}{e^{Fy}}} = \frac{1}{1+\frac{e^{-Fy}}{e^{Fy}}} = \frac{1}{1+e^{-2Fy}}$$

$$e^F+e^{-F} = e^{Fy}+e^{-Fy}$$

because $y \in \{+1,-1\}$

$$\text{loglikelihood} = \sum_{i=1}^{N} \log\left[p(x_i)^{y_i^*} \left(1-p(x_i)\right)^{1-y_i^*} \right] = \sum_{i=1}^{N} \log\left(\frac{1}{1+e^{2y_i F(x_i)}} \right)$$

$$= \sum_{i=1}^{N} -\log\left(1+e^{-2y_i F(x_i)}\right) = \sum_{i=1}^{N} e^{-y_i F(x_i)}$$

approx $\to 1$

Funny? $\log(\,) \sim \exp(\,)$

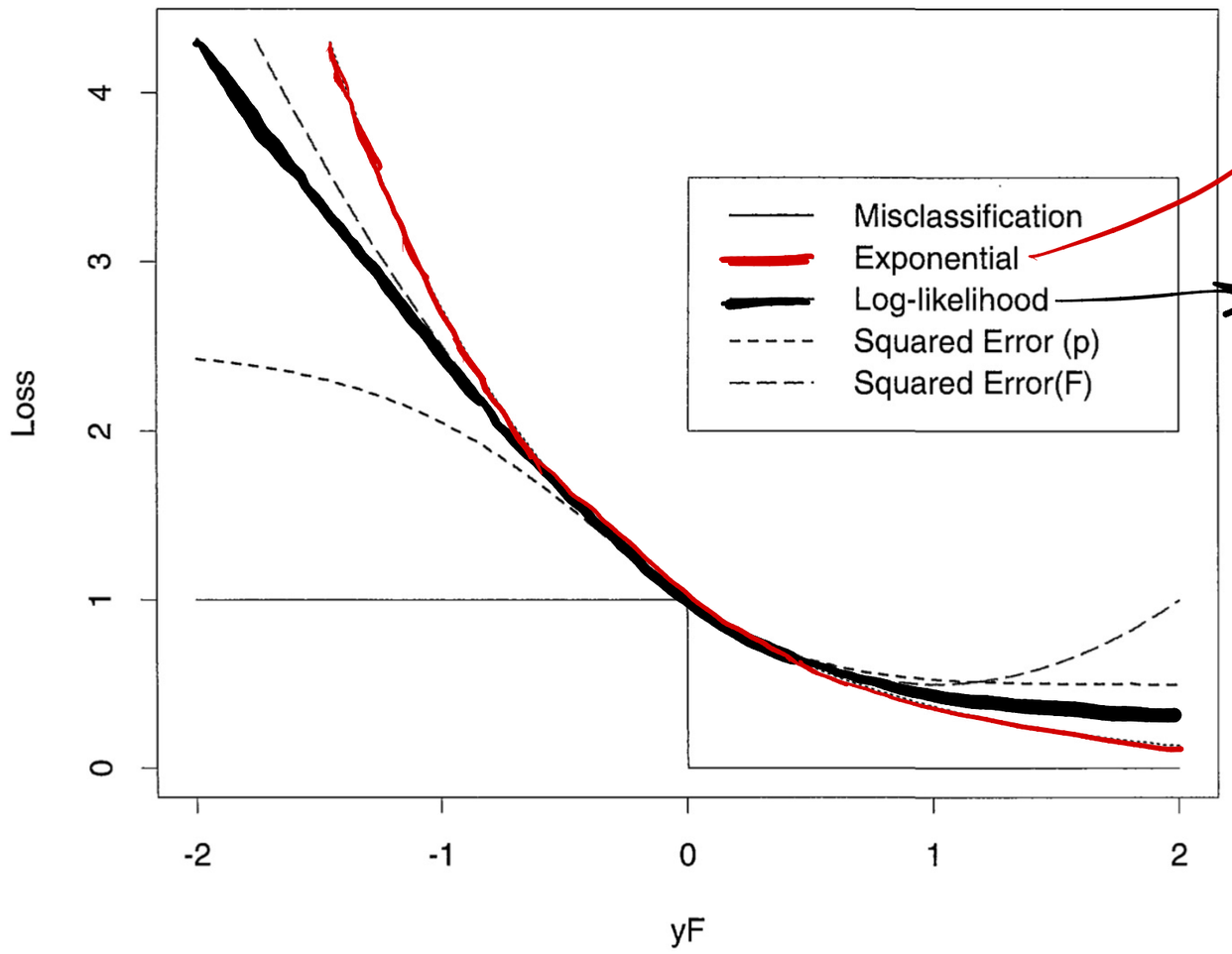
Friedman
Hastie
Tibshirani

ML point of view: important calculations are around 0

"Active learning"

$y_i F(x_i) \simeq 0$ → small

margin



$e^{-y_i F(x_i)}$

$\sum_{i=1}^{N} -\log\left(1+e^{-2y_i F(x_i)}\right)$

FIG. 2. *A variety of loss functions for estimating a function* $F(x)$ *for classification. The horizontal axis is* yF*, which is negative for errors and positive for correct classifications. All the loss functions are monotone in* yF*, and are centered and scaled to match* e^{-yF} *at* $F = 0$*. The curve labeled "Log-likelihood" is the binomial log-likelihood or cross-entropy* $y^* \log p + (1 - y^*)\log(1 - p)$*. The curve labeled "Squared Error*(p)*" is* $(y^* - p)^2$*. The curve labeled "Squared Error*(F)*" is* $(y - F)^2$ *and increases once* yF *exceeds* 1*, thereby increasingly penalizing classifications that are "too correct."*

$$\frac{\partial \text{loglikelihood}}{\partial F(x_i)} \propto \frac{\partial \sum_{i=1}^{N} e^{-y_i F(x_i)}}{\partial F(x_i)} = \frac{\partial\, e^{-y_i F(x_i)}}{\partial F(x_i)} = -y_i e^{-y_i F(x_i)}$$

- Fit next weak learner $h_{T+1}(x_i) \simeq y_i \cdot e^{-y_i F(x_i)}$ for all datapoints $i = 1:N$
- Update ensemble $F_{next}(x) = F(x) + h_{T+1}(x)$.

$$\log(\text{odds}) = \log \frac{p(y=+1|x)}{p(y=0|x)} = \log \frac{p(x)}{1-p(x)}$$

Alternative derivation for GB classification (easier, more practical)

$$F(x) = h_1(x) + h_2(x) \dots + h_T(x) \boxed{+ h_{T+1}(x)}$$

new tree $\sim$ fit $-\frac{\partial LL}{\partial F}$

BINARY

$$p(x) = p(y=+1|x) = \frac{e^{F(x)}}{e^{F(x)} + e^{-F(x)}}$$

$$1-p(x) = p(y=0|x) = \frac{e^{-F(x)}}{e^{F(x)} + e^{-F(x)}}$$

$$p(x) = \frac{1}{1+e^{-2F(x)}}$$

- making F half of the score $F \to \frac{1}{2}F \Rightarrow p(x) = \text{sigmoid}(F)$

$$p(x) = \frac{1}{1+e^{-F}} \qquad 1-p(x) = \frac{1}{1+e^{F}}$$

$$LL = \sum_{i=1}^{N} y_i \log p(x_i) + (1-y_i) \log(1-p(x_i))$$

$$\frac{\partial LL}{\partial F(x_i)} = \frac{\partial}{\partial F}\left[y_i \log\left(\frac{1}{1+e^{-F}}\right) + (1-y_i)\log\left(\frac{1}{1+e^{F}}\right)\right]$$

one datapoint x_i

$$= y_i \frac{e^{-F}}{1+e^{-F}} - (1-y_i)\left(1 + \frac{e^{-F}}{1+e^{-F}}\right)$$

$$= -(1-y_i) + \frac{e^{-F}}{1+e^{-F}}\left(y_i + 1 - y_i\right) \quad \rightarrow 1-p(x_i)$$

$$= y_i - 1 + 1 - p(x_i)$$

$$= y_i - p(x_i)$$

Fit next tree to gradient $h_{TH}(x_i) \simeq y_i - p(x_i)$

$\frac{\partial LL}{\partial F(x_i)}$. not really a gradient (formal math) NOT HW 4.

new tree $h_{T+1} = \underset{h=\text{tree}}{\operatorname{argmin}} \sum_{i=1}^{N} \text{Loss}\left(y_i, F_T(x_i) + h\right)$

$$L(y, F+h) \underset{\text{Taylor Approx}}{\simeq} L(y,F) + \frac{\partial}{\partial F} L(y,F)\, h + \frac{1}{2} \frac{\partial^2 L(y,F)}{\partial^2 F} h^2$$

solve for h ... $\dfrac{\sum y_i - p}{\sum p(1-p)}$ missing details

---

HWQ: multiple classes $y_i \in \{1, 2, \dots L\}$ one out of many $y_i = K$

Y_1	Y_2	y_K	y_L
0	0	1	0

20NG, 8NG (20L $\Rightarrow$ 8L)

- $F_1(x)$ $F_2(x)$ - - - $F_L(x)$ scoring functions ensembles of trees.
- probas per class via softmax

$$p_K(x) = \frac{e^{F_K(x)}}{\text{normalized}} = \frac{e^{F_K(x)}}{\sum_{j=1}^{L} e^{F_j(x)}}$$

- neg loglikelihood for x_i

$$LL_i = \sum_{K=1}^{L} \boxed{y_K} \log p_K(x_i) = -\log p_{[y_i = K]}(x_i)$$

filter 0/1

$$LLi = -\log\left(\frac{e^{F_k(x_i)}}{\sum_j e^{F_j(x_i)}}\right) = -F_k(x_i) + \log\left(\sum_j e^{F_j(x_i)}\right)$$

- gradient w.r.t $F_k(x_i)$ $k = 1, 2, 3 \ldots K$. 2 cases

$\partial / \partial F_k$

- $k = y_i$ = label $(Y_k = 1)$ $\dfrac{\partial LLi}{\partial F_k} = -1 + \dfrac{e^{F_k}}{\sum_j e^{F_j}} = -Y_k + P_k(x_i)$

-1

- $k \neq y_i$ = label $(Y_k = 0)$: $\dfrac{\partial LLi}{\partial F_k} = 0 + \dfrac{e^{F_k}}{\sum_j e^{F_j}} = P_k(x_i) = -Y_k + P_k$

0

$$\frac{\partial LLi}{\partial F_k} = P_k - Y_k$$

Fit next tree $h_{m+1} \simeq Y_k - P_k$