

Lecture 6/25

- HW3 demo

- Next: Chapter 4

- boosting / ensemble

- Multilabel data, in particular ECOC

- Feature selection

- Feature aggregation / PCA

- Features from similarity / TSNE

- HW4

- chapter 5: SVM + kernels

- chapter 6: advanced NN

- CNN
 - RecNN vs Transformers

"random Forests"

- Active learning.

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

Boosting (weak learner) = additive ensemble

datapoint
 $x = (x^1, x^2, \dots, x^D)$
 $t = 1:T$
 x_i

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

final score

$$h_t(x)$$

weak scores

$\frac{h(x)}{h(x)}$
 weak learner
 acc $\approx 65\%$
 better than random
 but poor
default: shallow
 decision tree
dec stump: 1-split DT

• Regression mode label y = quantity

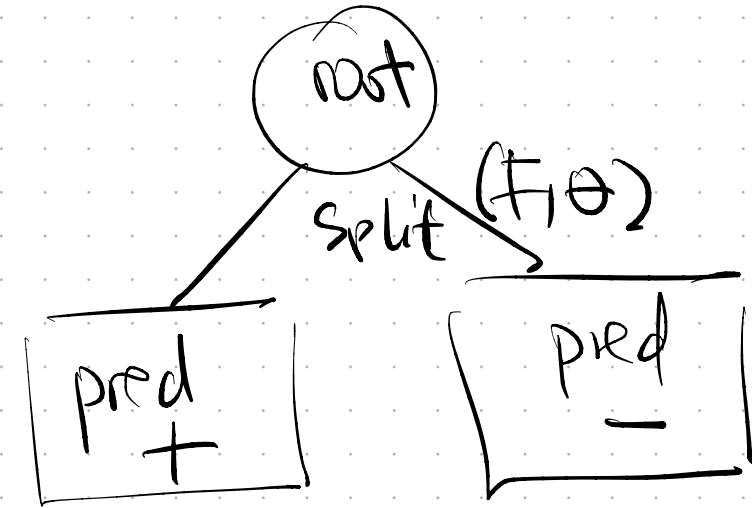
want $(F(x) - y)^2 \approx 0$
 all i $\sum_{i=1}^N (F(x_i) - y_i)^2 \approx 0$

• Classification mode $y \in \{-1, +1\}$ $y^* \in \{0, +1\}$

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

want max likelihood $\prod_{i=1}^N p(y^*_i|x) \cdot p(y^*_i=1|x)$

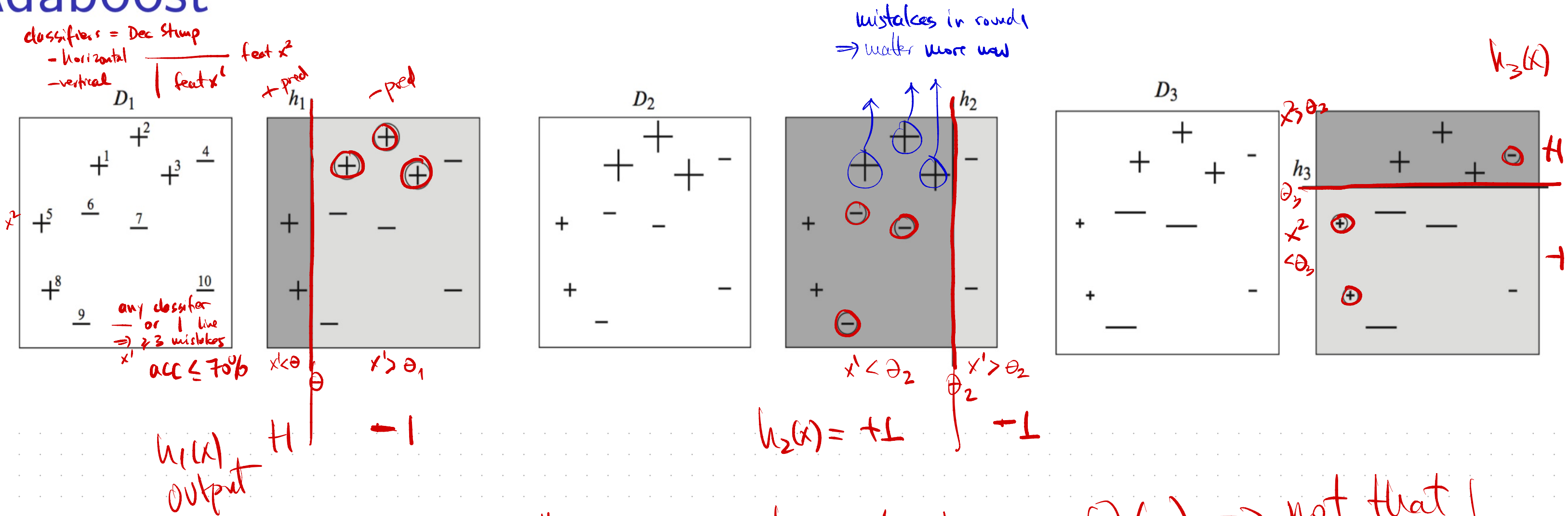
y^* $1-y^*$ filter which predicts



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

Gradient Boosting = Gradient Descent + Boosting

adaboost



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

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

$h_{\text{stump}}(x) = \text{dec stump} = 1\text{-split decision tree.}$

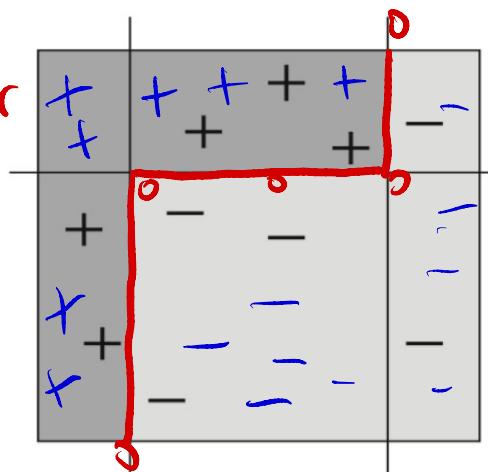
Comparison: one big tree with T splits $\Rightarrow$ complexity = $\Theta(2^T) \Rightarrow$ overfit!

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

(fixed coefficients ρ_t for $h_t(x)$)

$$F = \text{sign} \left(\begin{array}{c} ? \\ 0.42 \\ ? \end{array} + \begin{array}{c} ? \\ 0.65 \\ ? \end{array} + \begin{array}{c} ? \\ 0.92 \\ ? \end{array} \right)$$

$F(x) = \text{add scores}$
 $\text{sign}(F(x)) = \text{classifier}$
 100% acc =



2 ways to implement boosting

1st (historically 1996): ADABOOST

→ look at current weak learner $h_t(x)$

→ reweight data $D_{t+1}(x_i) =$

= importance of datapoint x_i next round

→ train $h_{t+1}() =$ weak learner

with error weighted by $D_{t+1}()$

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

acc $\begin{cases} 1 & \text{mistake} \\ 0 & \text{correct} \end{cases}$

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 $t = 1 \dots T$ rounds of Adaboost

weight on x $D_{t+1}(x) \approx \# \text{ rounds } t \text{ where } x \text{ mistake}$

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

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

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

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

• $D_{t+1}(x) = \text{large} \Rightarrow$ most classif $h_t(x) \neq y$

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

- either train error ≈ 0
 $F(x)$ ensemble $\sum_{i=1}^N \frac{1}{[F(x_i) \neq y_i]} \rightarrow 0$

• or stuck: $D_{T+1}()$ dist of point importance for next round is s.t. the best $h_{T+1}(x)$ classifier is random

$$E_{T+1} = \sum_{i=1}^N \frac{1}{[h_{T+1}(x_i) \neq y_i]} \cdot D_{T+1}(x_i) \approx 0.5$$

as long as $h_{T+1}(x)$ makes some progress (better than 0.5)
training error ($F(x)$) has to go to zero.

2nd way
much better

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

Gradient
Boosting.

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

next weak
learner

$$h_{T+1}(x) = F_{\text{new}}(x) - F_{\text{current}}(x)$$

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

diff
w.r.t $F(x)$

$$= F(x) - y = \frac{\partial J}{\partial F(x)}$$

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

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

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

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

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

want the next weak learner

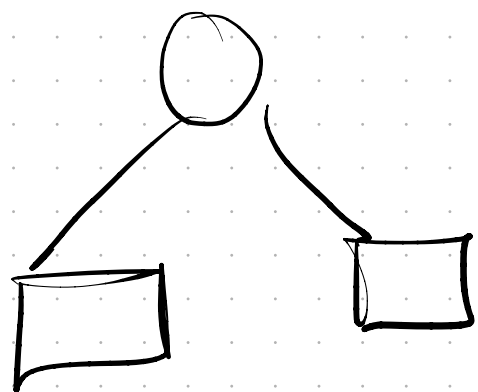
$$h_{T+1}(x) \approx y - F(x)$$

train the tree on
 $(x, \text{label } y - F(x))$ $x=1:N$

- if $h_{T+1}(x) \approx y - F(x)$ well $\Rightarrow$
residual label (quantity)

next w. learner $F_{\text{new}}(x) = F(x) + h_{T+1}(x) \approx \text{well } y$ (good)

- practice: train $h_{T+1}()$ to match $h_{T+1}(x) \approx y - F(x)$
weak learner
"dec Stump"
new label



- req updating label after every round
 $y_{\text{new}} = y_{\text{orig}} - F(x)$

- does not require weights like Adaboost, modify train procedure

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

Classification

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

binary

$$F(x) = \sum_{t=1}^T \text{h}_t(x)$$

Score
one F

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

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

want MAX

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

GD thinking $\Rightarrow$ fit next weak learner $h_{T+1}(x) \approx$

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

residual gradient?

$\frac{\partial \text{log likelihood}}{\partial F(x)}$

Categories:

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

$y\text{-dist} = 1/0/0/\dots$

pred dist $P_1(x) P_2(x) \dots P_L(x)$

Scores: $F_1(x) F_2(x) \dots F_L(x)$

filter $y^* \rightarrow 0$
 $\rightarrow 1$ which probab.

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

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

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

(Note: y^ is circled in red in the original image)*

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

(Note: $F=F(x)$ is written on the left)

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

(Note: $e^{Fy} + e^{-Fy}$ is boxed in red in the original image)

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

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

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

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

approx $\rightarrow 1$

Funny? $\log(\) \approx \exp(\)$

ML point of view: important
calculations are around 0

Friedman
Hastie
Tibshirani

$y_i F(x_i) \approx 0$
margin $\downarrow$ Small

"Active Learning"

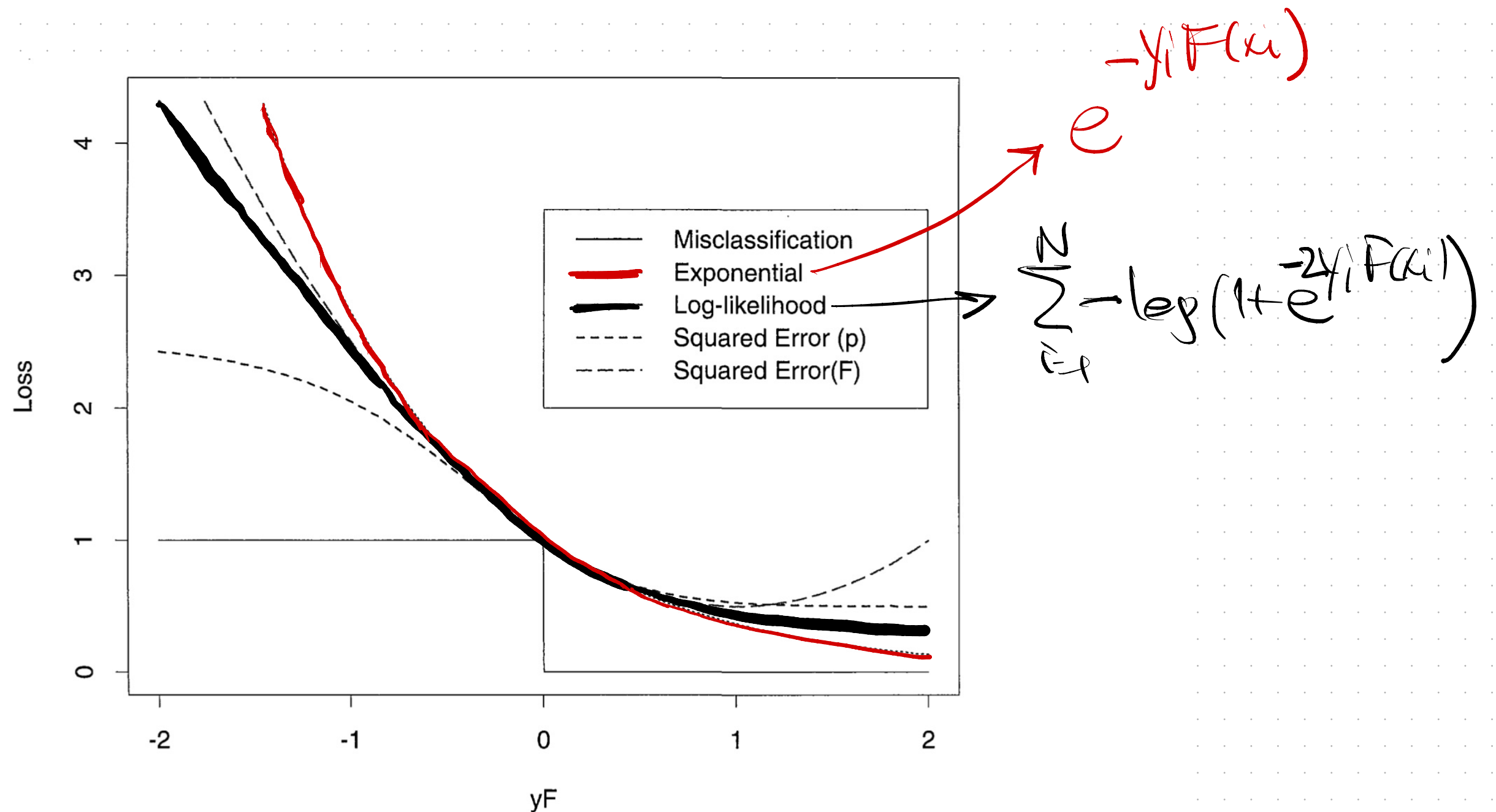


FIG. 2. A variety of loss functions for estimating a function $F(x)$ for classification. The horizontal axis is yF , which is negative for errors and positive for correct classifications. All the loss functions are monotone in yF , and are centered and scaled to match e^{-yF} at $F = 0$. The curve labeled “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.”

$$\frac{\partial \log \text{likelihood}}{\partial F(x_i)} \approx \frac{\partial \sum_{i=1}^N e^{-y_i F(x_i)}}{\partial F(x_i)} = \frac{\partial e^{-y_i F(x_i)}}{\partial F(x_i)} = -y_i e^{-y_i F(x_i)}$$

• Fit ^{next} weak learner $h_{TH}(x_i) \approx y_i \cdot e^{-y_i F(x_i)}$
for all datapoints $i = 1:N$

• Update $F_{\text{next}}(x) = F(x) + h_{TH}(x)$.
ensemble

$$\log(\text{odds}) = \log \frac{P(Y=1|X)}{P(Y=0|X)} = \log \frac{p(x)}{1-p(x)}$$

Alternative derivation for GB Classification (easier, more practical)

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

$$+ h_{TH}(x)$$

new tree

$$\approx - \frac{\partial LL}{\partial F}$$

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 $F \rightarrow \frac{1}{2}F \Rightarrow p(x) = \text{sigmoid}(F)$

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

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

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

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

one datapoint x_i

$$\begin{aligned}
&= y_i \frac{e^{-F}}{1+e^{-F}} - (1-y_i) \left(1 - \frac{e^{-F}}{1+e^{-F}} \right) \\
&= -(1-y_i) + \frac{e^F}{1+e^F} (y_i + 1-y_i) \xrightarrow{\quad} 1-p(x_i) \\
&= y_i - 1 + 1 - p(x_i) \\
&= y_i - p(x_i)
\end{aligned}$$

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

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

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

$$L(y, F+h) \approx L(y, F) + \frac{\partial}{\partial F} L(y, F) h + \frac{1}{2} \frac{\partial^2 L(y, F)}{\partial^2 F} h^2$$

Taylor
Approx

solve for h ...

$$\frac{\sum y_i - P}{\sum P(1-P)}$$

missing details

HW A:

multiple classes
one out of many

$y_i \in \{1, 2, \dots, L\}$
 $y_i = K$

y_1	y_2	y_K	y_L
0	0	1	0

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

- $F_1(x)$ $F_2(x)$... $F_L(x)$ scoring functions ensembles of trees
- probab per class via softmax $P_K(x) = \frac{e^{F_K(x)}}{\sum_{j=1}^L e^{F_j(x)}}$ normalized

neg loglikelihood for x_i
 $L_{hi} = -\sum_{K=1}^L \boxed{y_{iK}} \log P_K(x_i)$ filter %

$$= -\log P_{[y_i=K]}(x_i)$$

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

• gradient w.r.t $F_k(x_i)$ $k=1,2,\dots,h$, 2 cases

$$- k=y_i = \text{label} (Y_k=1) \quad \frac{\partial LL_i}{\partial F_k} = 1 + \frac{e^{F_k}}{\sum_j e^{F_j}} = \boxed{-Y_k + P_k(x_i)}$$

$$- k \neq y_i = \text{label} (Y_k=0) : \quad \frac{\partial LL_i}{\partial F_k} = 0 + \frac{e^{F_k}}{\sum_j e^{F_j}} = P_k(x_i) = \boxed{-Y_k + P_k}$$

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

Fit next tree $h_{T+1} \approx Y_k - P_k$