

# Boosting

$y$  = outcome variable

$\underline{x}$  = predictor variables (original)

define class of base (weak) learners

$$f(\underline{x}; \underline{b}) \in \mathcal{F}_B = \{f(\underline{x}, \underline{b})\}_{\underline{b} \in B}$$

↑ parameters  $(b_1, b_2, \dots, b_k)$

$B$  = set of all possible joint values of  $\underline{b}$

require  $\|f(\underline{z}, \underline{b})\|_2 = 1$  on training data

$$\frac{1}{N} \sum_{i=1}^N f^2(\underline{x}_i, \underline{b}) = 1$$

structural model for "boosting" base learner

$$F(\underline{x}) = \sum_{j=1}^J a_j f(\underline{x}; \underline{b}_j)$$

$F(\underline{x}; \{a_j\}_{j=1}^J) \in \mathcal{F}$  all linear combinations  
of  $f(\underline{x}; \underline{b}) \in \mathcal{F}_B$

usually  $J$  very large ( $J = \infty$ )

Regularized linear regression (GLS)

# GPS

## Definitions

$\nu \geq 0$ : path length

$\Delta\nu > 0$ : small increment

$$g_j(\nu) = - \left[ \frac{\partial \hat{R}(\mathbf{a})}{\partial a_j} \right]_{\mathbf{a}=\hat{\mathbf{a}}(\nu)} \quad (\text{loss} + \text{data})$$

$$p_j(\nu) = \left[ \frac{\partial P(\mathbf{a})}{\partial |a_j|} \right]_{\mathbf{a}=\hat{\mathbf{a}}(\nu)} \quad (\text{penalty})$$

$$\lambda_j(\nu) = g_j(\nu) / p_j(\nu) \quad (\text{combination})$$

$$P_j(v) = p(\hat{a}_j(v)) \quad \text{easy}$$

Compute  $g_j(v)$ :

$$\hat{R}(\underline{a}) = \frac{1}{N} \sum_{i=1}^N L(y_i, F_i)$$

$$F_i = \sum_{j=1}^J a_j f(x_i, \underline{b}_j)$$

$$\begin{aligned} g_j &= - \frac{\partial \hat{R}(\underline{a})}{\partial a_j} = \sum_{i=1}^N \frac{\partial L(y_i, F_i)}{\partial F_i} \frac{\partial F_i}{\partial a_j} \\ &= \frac{1}{N} \sum_{i=1}^N l_i \cdot f(x_i; \underline{b}_j) \end{aligned}$$

$$\text{where } l_i = - \frac{\partial L(y_i, F_i)}{\partial F_i}$$

= "pseudo" response

$$g_j(v) = \frac{1}{N} \sum_{i=1}^N l_i(v) f(x_i; \underline{b}_j)$$

$$l_i(v) = - \frac{\partial L(y_i, F_i(v))}{\partial F_i(v)}$$

$$F_i(v) = \sum_{j=1}^J \hat{a}_j(v) f(x_i; \underline{b}_j)$$

## Examples

$$L(y, F) = (y - F)^2 / 2 \quad (\text{least-squares})$$

$$l_i(v) = (y_i - F_i(v)) = r_i(v) \quad \text{residuals at } v$$

$$J_{ls}(v) = \frac{1}{N} \sum_{i=1}^N r_i(v) f(x_i; \underline{b}_{ls})$$

$$L(y, F) = |y - F| \quad (\text{least absolute deviation})$$

$$l_i(v) = \text{sign}(y_i - F_i(v)) = \text{sign}(r_i(v))$$

$$J_{lad}(v) = \frac{1}{N} \sum_{i=1}^N \text{sign}(r_i(v)) f(x_i; \underline{b}_{lad})$$

$$L(y, F) = \log(1 + e^{-yF}); \quad y \in \{-1, 1\}$$

(logistic regression)

$$p_i(v) = 1 / (1 + e^{-F_i(v)}) \quad \text{probability } y = 1 \text{ at } v$$

$$l_i(v) = \frac{y_i + 1}{2} - p_i(v) = r_i^{(p)}(v)$$

residual on probability scale

$$J_{lr}(v) = \frac{1}{N} \sum_{i=1}^N r_i^{(p)}(v) f(x_i; \underline{b}_{lr})$$

$$\underline{\text{Any } L(y, F)} : l_i(v) = -2 L(y_i, F_i(v)) / \partial F_i(v)$$

prediction for  $\underline{x}_i$  at path point  $v$

Note: 
$$F_i(v) = \sum_{j=1}^J \hat{a}_j(v) \underbrace{f(\underline{x}_i; \underline{b}_j)}_{j\text{th base learner}}$$

coefficient of  $j$ th base learner at  $v$

$$F_i(v) = \sum_{j \in A(v)} \hat{a}_j(v) F(\underline{x}_i; \underline{b}_j)$$

$$A(v) = \{j \mid \hat{a}_j(v) \neq 0\}$$

active set of parameters at path point  $v$

$$H(v) = \{f(\underline{x}; \underline{b}_j)\}_{j \in A(v)}$$

base learners in model at  $v$

Only need  $\{\hat{a}_j(v)\}_{j \in A}$  &  $\{f_j(\underline{x}; \underline{b}_j)\}_{f \in H}$  to predict  $F(\underline{x})$  at  $v$

## Boosting (GPS) algorithm

$v = 0$ ;  $A(v) = \text{empty}$ ;  $H(v) = \text{empty}$

loop {

compute  $\{ \lambda_j(v) = g_j(v) / p(\hat{a}_j(v)) \}_{j \in A(v)}$

$S = \{ j \mid \lambda_j(v) \cdot \hat{a}_j(v) < 0 \}_{j \in A(v)}$

if  $S \neq \text{empty}$  {  $j^* = \underset{j \in S}{\operatorname{argmax}} |\lambda_j(v)|$

$\hat{a}_{j^*}(v + \Delta v) = \hat{a}_{j^*}(v) + \Delta v \cdot \operatorname{sign}(\lambda_{j^*}(v))$

$v \leftarrow v + \Delta v$ ; next;

}

$j^* = \underset{j \in A(v)}{\operatorname{argmax}} |\lambda_j(v)|$

compute  $\{ \lambda_k(v) = g_k(v) / p(0) \}_{k \notin A(v)}$

$k^* = \underset{k \notin A(v)}{\operatorname{argmax}} |\lambda_k(v)|$

if  $\lambda_{j^*}(v) > \lambda_{k^*}(v)$  {

$\hat{a}_{j^*}(v + \Delta v) = \hat{a}_{j^*}(v) + \Delta v \cdot \operatorname{sign}(\lambda_{j^*}(v))$

$v \leftarrow v + \Delta v$ ; next;

}

$A(v + \Delta v) = A(v) \cup \{k^*\}$

$H(v + \Delta v) = H(v) \cup \{f(x; \underline{b}_{k^*})\}$

$\hat{a}_{k^*}(v + \Delta v) = \Delta v \cdot \operatorname{sign}(\lambda_{k^*}(v))$

$v \leftarrow v + \Delta v$

} until  $\{g_j(v) = 0\}_1^J \Rightarrow v = v_{\max}$

## Path

$$F(x; v) = \sum_{j \in A(v)} \hat{a}_j(v) f(x; b_j)$$

$$v \in [0, v_{\max}]$$

## Problem

compute  $\{g_k(v)\}_{k \notin A(v)}$

$$k^* = \operatorname{argmax}_{k \notin A(v)} |g_k(v)|$$

too many  $k \notin A(v)$

## Trick

$$g(\underline{b}; v) = \frac{1}{N} \sum_{i=1}^N l_i(v) f(x_i; \underline{b})$$

$$l_i(v) = - \frac{\partial L(y_i, F_i(v))}{\partial F_i(v)}$$

$$F_i(v) = \sum_{j \in A(v)} \hat{a}_j(v) f(x_i; \underline{b}_j)$$

$$\underline{b}^*(v) = \operatorname{argmax}_{\underline{b}} |g(\underline{b}; v)|$$

$$(\underline{b}^*(v), \rho^*(v)) = \operatorname{argmin}_{\underline{b}, \rho} \sum_{i=1}^N (l_i(v) - \rho \cdot f(x_i; \underline{b}))^2$$

$$\operatorname{sign}(g(\underline{b}^*; v)) = \operatorname{sign}(\rho^*(v))$$

$\{l_i(v)\}_1^N =$  pseudo responses at  $v$

Nonlinear least-squares optimization

replace  $\kappa^* = \operatorname{argmax}_{\kappa \notin A(v)} |g_{\kappa}(v)|$

with  $\underline{b}_{\kappa^*} = \underline{b}^*(v)$  at each step.

$\Rightarrow$  generalized gradient boosting  
with any penalty  $P(\underline{g})$

## (Lasso) Gradient boosting

(1)  $P(\underline{a}) = \sum_{j=1}^J |a_j|$  lasso penalty

$$P_j'(a_j) = \frac{\partial P(\underline{a})}{\partial |a_j|} = 1$$

(2) assume  $S^j = \text{empty}$  always

(3) step size

$$\Delta v(v) = \varepsilon \cdot \underset{\eta}{\operatorname{argmin}} \sum_{i=1}^N L(y_i, F_i(v) + \eta \cdot f(x_i; \underline{b}^k(v)))$$

↑ "shrinkage" factor ( $\varepsilon \sim 0.1$ )

"learning rate"

# Gradient Boosting

Initialize :  $k = 0$  ;  $\{F_i = 0\}_1^N$

loop {

$$\{l_i = -\partial L(y_i, F_i) / \partial F_i\}_1^N$$

$$(b^*, g^*) = \underset{\underline{b}, g}{\operatorname{argmin}} \sum_{i=1}^N (l_i - g \cdot f(x_i; \underline{b}))^2$$

$$\Delta v = \epsilon \cdot \underset{\eta}{\operatorname{argmin}} \sum_{i=1}^N L(y_i, F_i + \eta f(x_i, \underline{b}^*))$$

$$k \leftarrow k + 1 ; \underline{b}_k = \underline{b}^* ; \hat{a}_k = \Delta v$$

$$\{F_i \leftarrow F_i + \Delta v \cdot f(x_i; \underline{b}_k)\}_1^N$$

until ( $k = K$ )

Path indexed by  $k \in [1, K]$

$$F_k(x) = \sum_{j=1}^k \hat{a}_j f(x; \underline{b}_j)$$

So far, arbitrary base learners:

$$f(x; \underline{b}) \in \mathcal{F}_B$$

Now, base learners = all  $M$ -terminal node decision trees

$$f(x; \{c_m, R_m\}_1^M) = \sum_{m=1}^M c_m I(x \in R_m)$$

$\underline{b} = \{c_m\}_1^M$  & split variables or subsets defining regions  $\{R_m\}_1^M$

At each boosting step ( $k$ ):

$$(\{c_m^*, R_m^*\}_1^M) = \operatorname{argmin}_{\{c_m, R_m\}_1^M} \sum_{i=1}^N \left( l_i - \sum_{m=1}^M c_m I(x_i \in R_m) \right)^2$$

least-squares regression tree problem

Approximate with CART greedy top-down search procedure.

$$(\{c_m^*, R_m^*\}_1^M) \approx \text{CART}_{\text{reg}}(\{l_i, x_i\}_1^N)$$

# Gradient tree boosting

Initialize :  $k = 1$  ;  $\{F_i = 0\}_1^N$

loop {

$$\{l_i = -\partial L(y_i, F_i) / \partial F_i\}_1^N$$

$$T_k(x) = \text{CART}_{\text{reg}}(\{l_i, x_i\}_1^N)$$

$$\Delta v = \epsilon \cdot \underset{\eta}{\text{argmin}} \sum_{i=1}^N L(y_i, F_i + \eta T_k(x_i))$$

$$T_k(x) \leftarrow \Delta v \cdot T_k(x)$$

$$\{F_i \leftarrow F_i + T_k(x_i)\}_1^N$$

$$k = k + 1$$

} until ( $k = K$ )

Path indexed by  $k \in [1, K]$

$$F_k(x) = \sum_{j=1}^k T_j(x)$$

For trees at  $n$ th iteration

$$T_n(\underline{x}) \leftarrow \varepsilon \cdot \eta_{n2}^* T_n(\underline{x})$$

$$\eta_{n2}^* = \operatorname{argmin}_{\eta} \sum_{i=1}^N L(y_i, F_i + \sum_{m=1}^M \eta \cdot C_{n2m} \mathbb{I}(\underline{x}_i \in R_{n2m}))$$

Optimize within each region

$$\{\hat{a}_{n2m}\}_1^M = \operatorname{argmin}_{\{a_{nm}\}_1^M} \sum_{i=1}^N L(y_i, F_i + \sum_{m=1}^M a_{nm} \mathbb{I}(\underline{x}_i \in R_{n2m}))$$

$$T_n(\underline{x}) = \sum_{m=1}^M \varepsilon \cdot \hat{a}_{n2m} \mathbb{I}(\underline{x} \in R_{n2m})$$

Since  $\{R_{n2m}\}_{m=1}^M = \text{disjoint}$

reduces to separate min. within each  $R_{n2m}$

$$\hat{a}_{n2m} = \operatorname{argmin}_a \sum_{\underline{x}_i \in R_{n2m}} L(y_i, F_i + a)$$

simple min. wRT constant

Examples:

$$L(y, F) = (y - F)^2$$

$$\hat{a}_{n2m} = \operatorname{mean}_{\underline{x}_i \in R_{n2m}} \{y_i - F_i\}$$

$$L(y, F) = |y - F|$$

$$\hat{a}_{n2m} = \operatorname{median}_{\underline{x}_i \in R_{n2m}} \{y_i - F_i\}$$

## Multiple additive Regression trees (MART)

Initialize:  $k=1$ ;  $\{F_i=0\}_1^N$

loop {

$$\{l_i = -\partial L(y_i, F_i) / \partial F_i\}_1^N$$

$$\{R_{k,m}\}_{m=1}^M = \text{CARTreg}(\{l_i, x_i\}_1^N)$$

$$\{\hat{a}_{k,m} = \underset{a}{\operatorname{argmin}} \sum_{x_i \in R_{k,m}} L(y_i, F_i + a)\}_{m=1}^M$$

$$T_k(x) = \sum_{m=1}^M \hat{a}_{k,m} I(x \in R_{k,m})$$

$$\{F_i = F_i + T_k(x_i)\}_1^N$$

$$k = k + 1$$

} until ( $k = K$ )

Path indexed by  $k \in [1, K]$

$$F_k(x) = \sum_{j=1}^k T_j(x)$$

$L(y, F) = |y - F|$  much more robust  
than  $(y - F)^2$

$$F^*(x) = \text{median}(y|x)$$

immune to outliers in  $y$

Trees immune to outliers in  $x$

$\Rightarrow$  MART total immune to outliers

$$l_i = \tilde{y}_i = \text{sign}(y_i - F_{m-1}(x_i)) \text{ splitting}$$

$$\hat{a}_{\text{rem}} = \text{median}_{x_i \in \text{Rem}} \{y_i - F_{m-1}(x_i)\} \text{ terminal nodes}$$

Very important for data mining

But, less efficient than  $L(y, F) = (y - F)^2$   
when noise well behaved

$$y = F^*(x) + \varepsilon ; \varepsilon \sim N(0, \sigma^2)$$

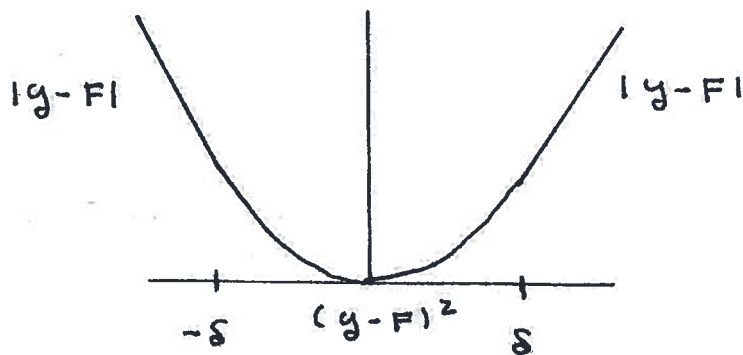
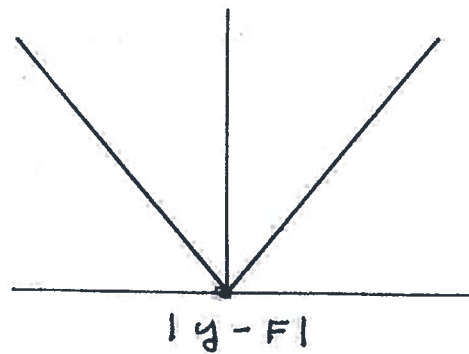
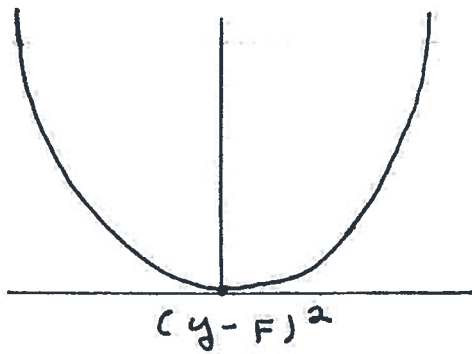
$|y - F|$  sacrifices a little accuracy  
in good situations for insurance  
against bad situations

## Huber M - regression

(best of both)

$$L(y, F) = M(y, F) = \begin{cases} 1/2 (y-F)^2 & |y-F| \leq \delta \\ \delta (|y-F| - \delta/2) & |y-F| > \delta \end{cases}$$

$\delta$  = "trimming" factor; defines "outlier"



$$\delta \rightarrow 0 \Rightarrow M(y, F) \rightarrow |y - F|$$

$$\delta \rightarrow \infty \Rightarrow M(y, F) \rightarrow (y - F)^2$$

$$\delta = \underset{\alpha}{\text{quantile}} |y - F|$$

$$\Rightarrow \text{breakdown} = 1 - \alpha$$

usually  $\delta \geq 0.8$  or  $0.9$

$M(y, F)$  properties:

$\varepsilon \sim N(0, \sigma^2)$  almost as good as  $(y - F)^2$

$\varepsilon \sim$  very bad  $(1 - \alpha)$  " " " "  $|y - F|$

$\varepsilon \sim$  in between  $\Rightarrow$  better than both

Boosting  $M(y, F)$ :  $\lambda_m(x_i) = y_i - F_{m-1}(x_i)$

$$\tilde{y}_i = \lambda_m(x_i) \mathbb{I}(|\lambda_m(x_i)| \leq S_m)$$

$$+ S_m \text{sign}(\lambda_m(x_i)) \mathbb{I}(|\lambda_m(x_i)| > S_m)$$

$$S_m = \underset{\alpha}{\text{quantile}} \{ |y_i - F_{m-1}(x_i)| \}_1^N$$

current residuals at  $(m-1)$ th step

$$\hat{a}_{em} = \underset{a}{\text{argmin}} \sum_{x_i \in R_{em}} M(y_i, F_{m-1}(x_i) + a)$$

$\hat{a}_{em} \approx$  winsorized mean ( $S_m$ )

## Classification

$y \in \{c_1, c_2, \dots, c_K\}$  unordered categorical values

$d_{ke} = I(y = c_{ke})$  class indicator dummy

$$p_{ke}(x) = \Pr(y = c_{ke} | x) = E[d_{ke} | x]$$

obtain estimates  $\{\hat{p}_{ke}(x)\}_1^K$

$$\hat{y}(x) = \underset{1 \leq k \leq K}{\operatorname{argmin}} \sum_{e=1}^K L_{e|k} \hat{p}_{ke}(x); \hat{y} = c_{k^*}^1(x)$$

$$\text{let } \underline{d} = \{d_{ke}\}_1^K; \underline{\hat{p}}(x) = \{\hat{p}_{ke}(x)\}_1^K$$

$$\text{Need } L(\underline{d}, \underline{\hat{p}}) \Rightarrow E_{\underline{d}} = \underline{\hat{p}} = \underset{\underline{\hat{p}}}{\operatorname{argmin}} L(\underline{d}, \underline{\hat{p}})$$

One possibility:

$$\begin{aligned} L(\underline{d}, \underline{\hat{p}}) &= \|\underline{d} - \underline{\hat{p}}\|^2 \\ &= \sum_{k=1}^K (d_{ke} - \hat{p}_{ke}(x))^2 \quad \text{Gini criterion} \end{aligned}$$

used by CART  $\rightarrow$  rapid computation  
updating formulae for means  
and variances.

With gradient boosting can use any (differentiable) loss criterion with least-squares CART

$d_{jk} \in \{0, 1\}$  multinomial random variable  
 $\Rightarrow$  negative multinomial log-likelihood

$$L(\underline{d}, \hat{p}(\underline{x})) = - \sum_{k=1}^K d_{jk} \log \hat{p}_{jk}(\underline{x})$$

Problems:

(1) must have  $0 \leq \hat{p}_{jk}(\underline{x}) < 1$   
for all  $k$  &  $\underline{x}$

(2) would like  $\sum_{k=1}^K \hat{p}_{jk}(\underline{x}) = 1$  for all  $\underline{x}$

Solution: symmetric logistic transform

$$\hat{p}_{jk}(\underline{x}) = \exp(F_{jk}(\underline{x})) / \sum_{e=1}^K \exp(F_e(\underline{x}))$$

$$F_{jk}(\underline{x}) = \log \hat{p}_{jk}(\underline{x}) - \frac{1}{K} \sum_{e=1}^K \log \hat{p}_e(\underline{x})$$

$$L(\underline{d}, F(\underline{x})) = - \sum_{k=1}^K d_{jk} F_{jk}(\underline{x}) + \log \sum_{k=1}^K e^{F_k(\underline{x})}$$

K-class  
logistic  
regression

## Gradient boosting

$$\begin{aligned}
 \ell_{ik} &= \tilde{y}_{ik} = - \left[ \frac{\partial L(\alpha_i, F(x_i))}{\partial F_k(x_i)} \right]_{F(x) = F_{m-1}(x)} \quad \text{pseudo-responses} \\
 &\quad \begin{array}{c} \nearrow \text{obs} \\ \nwarrow \text{class} \end{array} \\
 &= \alpha_{ik} - \hat{p}_{ik}(x) \quad \text{ordinary residual} \\
 &\quad \text{on } \underline{\text{probability scale}}
 \end{aligned}$$

$\{\tilde{y}_{ik}\}_1^K \Rightarrow K$ -pseudo responses  
 $\Rightarrow K$  trees, one for each class  $F_{ik}(x)$

$\{R_{k, \ell m}\}_{\ell=1}^J = J$ -term node tree  $(\{\tilde{y}_{ik}, x_i\}_1^N)$   
 $\begin{array}{c} \text{term. node} \\ \downarrow \\ \{R_{k, \ell m}\}_{\ell=1}^J \\ \uparrow \\ \text{class iteration} \end{array}$

Terminal node updates:

$$\begin{aligned}
 \{\hat{\alpha}_{k, \ell m}\} &= \underset{\{\alpha_{k, \ell m}\}}{\text{argmin}} \\
 &= \sum_{i=1}^N \left\{ \sum_{k=1}^K \alpha_{ik} \left[ F_{k, m-1}(x_i) + \sum_{\ell=1}^J \alpha_{k, \ell m} I(x_i \in R_{k, \ell m}) \right] \right. \\
 &\quad \left. + \log \sum_{k=1}^K \exp \left[ F_{k, m-1}(x_i) + \sum_{\ell=1}^J \alpha_{k, \ell m} I(x_i \in R_{k, \ell m}) \right] \right\}
 \end{aligned}$$

- (1) not independent because regions associated with different classes overlap
- (2) no closed form solution

What to do?

Numerical optimization: Newton-Raphson

$$\text{Let } \underline{a} = \{a_{k\ell m}\}_{k=1, \ell=1}^{K, J}$$

$$\hat{\underline{a}} = \underset{\underline{a}}{\text{argmin}} \Phi(\underline{a}); \Phi(\underline{a}) = \text{above}$$

Let:

$$\underline{g}(\underline{a}^*) = \text{gradient vector}$$

$$g_j(\underline{a}^*) = \frac{\partial \Phi(\underline{a})}{\partial a_j} \Big|_{\underline{a}=\underline{a}^*}$$

$$\underline{H}(\underline{a}^*) = \text{Hessian matrix}$$

$$H_{ij}(\underline{a}^*) = \frac{\partial^2 \Phi(\underline{a})}{\partial a_i \partial a_j} \Big|_{\underline{a}=\underline{a}^*}$$

Newton-Raphson search:

$$\underline{a}^* = \underline{0}$$

$$\text{loop } \{ \underline{a}^* \leftarrow \underline{a}^* - \underline{H}(\underline{a}^*) \underline{g}(\underline{a}^*) \}$$

$$\text{until } \|\underline{g}(\underline{a}^*)\| = 0$$

$$\hat{\underline{a}} = \underline{a}^*$$

## Approximations:

$$(1) \underline{H}(\underline{a}^*) \approx \text{diag } \underline{H}(\underline{a}^*) \\ = \text{diag} \left( \frac{\partial^2 \Phi(\underline{a})}{\partial a_j^2} \Big|_{\underline{a}=\underline{a}^*} \right)$$

$\Rightarrow$  solution partitions into separate calculation in each region  $R_{k \text{ rem}}$

(2) one step starting at  $\underline{a} = \{a_{k \text{ rem}}\} = 0$

$$\hat{a}_{k \text{ rem}} = \frac{K-1}{K} \sum_{x_i \in R_{k \text{ rem}}} \tilde{y}_{ik} / \sum_{x_i \in R_{k \text{ rem}}} | \tilde{y}_{ik} | (1 - | \tilde{y}_{ik} |)$$

$$\tilde{y}_{ik} = d_{ik} - \hat{p}_{ik}(x_i) = \text{pseudo response}$$

## Algorithm K-MART

$$\{F_{1k}(x) = 0\}_{k=1}^K$$

For  $m = 1$  to  $M$  {

$$\hat{p}_{1k}(x) = e^{F_{1k}(x)} / \sum_{k=1}^K e^{F_{1k}(x)}$$

For  $k = 1$  to  $K$  {

$$\{\tilde{y}_{ik} = d_{ik} - \hat{p}_{1k}(x_i)\}_{i=1}^N$$

$$\{R_{krem}\} = \text{J-term tree}(\{\tilde{y}_{ik}, x_i\}_{i=1}^N)$$

$$\{\hat{a}_{krem} = \frac{k-1}{K} \sum_{x_i \in R_{krem}} \tilde{y}_{ik} / \sum_{x_i \in R_{krem}} |\tilde{y}_{ik}| (1 - |\tilde{y}_{ik}|)\}_{r=1}^J$$

$$F_{1k}(x) \leftarrow F_{1k}(x) + \sum_{r=1}^J \hat{a}_{krem} I(x \in R_{krem})$$

end For

end For

## Classification:

$$\hat{k}(x) = \operatorname{argmin}_k \sum_{l=1}^K L_{lk} \hat{p}_l(x)$$

$$\hat{c}(x) = c_{\hat{k}(x)}$$

MART (with shrinkage)  $\sim$  SVM

$$\hat{F}(\mathbf{x}) = \sum a_m b_m(\mathbf{x}), \quad \lambda \cdot P(\{a_m\})$$

	SVM	MART
Basis fun's	polyn's, RBF, NN	any (fast LS) Trees
Loss	class: $(1 - yF)_+$ reg: $( y - F  - \epsilon)_+$	differentiable class: -log like reg: Huber M
Penalty	$\sum a_m^2$	$\sum  a_m $
Trick	kernel	stage/shrink

“Right-sized” trees for boosting

“Boosting”:

build and prune at each step - NOT GOOD

expensive + inaccurate

“MART”: fixed tree size  $J$  at each step

Friedman 1999:

$J =$  small, but not too small

$\simeq 4 - 8$  (6) (insensitive)

"Right-sized" trees ( $L$ )

ANOVA expansion

$$F^*(\mathbf{x}) = \sum_j f_j(x_j) + \sum_{j,k} f_{jk}(x_j, x_k) + \sum_{j,k,l} f_{jkl}(x_j, x_k, x_l) + \dots$$

*main (additive) effects*      *two-var interactions*      *three-var. interactions*

Interaction order of MART  $\hat{F}(\mathbf{x})$

$$\leq \min(L-1, n) \quad \Rightarrow \quad \begin{array}{l} \text{individual trees} \\ \Sigma \text{ trees} \end{array}$$

"stumps"  $L = 2 \Rightarrow$  additive (main effects) model

Choose  $L$  to match int.-order of  $F^*(\mathbf{x})$  *usually approx. by low order interactions*

Usually unknown  $\Rightarrow$  "meta"-parameter for

model selection – but usually small

empirical:  $L \leq 6$ .

Try several values: indep test set or  $x$ -validation to judge results

WHY?

$J$  controls interaction order of  $\hat{F}(\mathbf{x})$

$J = 2$  ("stumps")  $\Rightarrow$  additive (main effects)

$J > 2$  permits interactions (to order  $J - 1$ )

Empirical:  $4 \leq J \leq 8$  (6)

## DATA MINING

Breiman:

“Boosted trees best off-the-shelf classifiers”

MART → regression.

Inherited tree advantages:

- (1) naturally handle mixed variable types  
and missing values
- (2) invariant to monotone transformations of  $x_j$
- (3) immune to bad  $x_j$ -distributions
- (4) internal variable subset selection
- (5) robust to irrelevant inputs.

Tree disadvantage - inaccuracy:

(1) coarse piecewise-constant approx.

(2) instability  $\Rightarrow$  high variance (large trees)  
 *$\sim$  search strategy*

(3) high interaction order approx.

(4) Fragmentation

MART (Boosting)

(1) piecewise-constant approx. much finer

(2) small trees + averaging  $\Rightarrow$  stable

(3) interaction order controlled by  $L$ .

In addition (any loss criterion)

(4) LAD\_MART immune, and

(5) M\_MART resistant to

$y$ -outliers

Boosting  $\Rightarrow$  loose interpretability

## INTERPRETATION

Relative importance of input variables

Trees (Breiman *et al* 1983):

$$\hat{J}_j^2(T) = \sum_{t=1}^{L-1} \hat{I}_t^2 \cdot 1(v_t = j)$$

$\uparrow$  non-terminal nodes

MART: average over trees

$$\hat{J}_j^2 = \frac{1}{M} \sum_{m=1}^M \hat{J}_j^2(T_m) ; \hat{J}_j \leftarrow \hat{J}_j \cdot 100 / \max_k \hat{J}_k$$

works better than for single tree

Justification: Friedman 1999

Classification: average over trees  
over all classes (K)

For classification, can also do separately for each class:

$$F_{le}(x) = \sum_{m=1}^M a_{lem} T_{lem}(x); \quad p_{le}(x) = \frac{e^{F_{le}(x)}}{\sum_{l=1}^K e^{F_{le}(x)}} \\ 1 \leq l \leq K$$

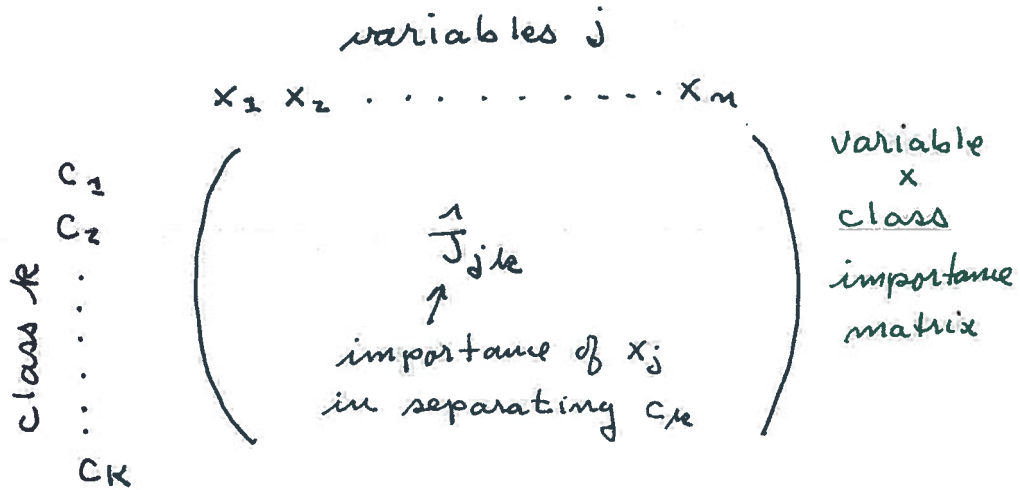
$$\hat{J}_{j|le}^2 = \sum_{m=1}^M \hat{J}_j(T_{lem})$$

- A. Which variables are most important for separating a particular class or subsets of classes
- B. Which classes do particular variables or subsets of variables contribute most towards separation

K-class classification:

$$\hat{J}_j \rightarrow \hat{J}_{j|k} \quad k=1 \text{ to } K$$

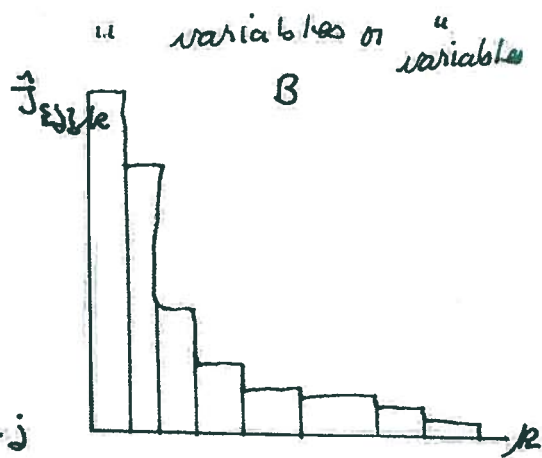
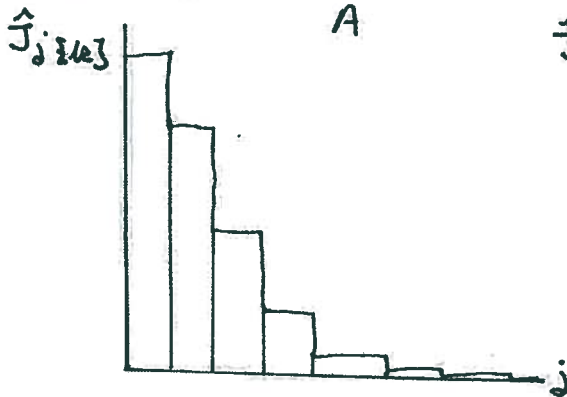
↑ averaged over trees of class  $k$  only



Can examine and average:

(1) row-wise (A) condition on class or sets of classes

(2) column-wise (B) " variables or " variables



## MORE INTERPRETATION

Joint dependence of  $\hat{F}(\mathbf{x})$  on relevant  $\{x_j\}_1^J$

Visualization most powerful tool

*plot  $\hat{F}(\underline{x}_R)$*

*vs  $\underline{x}_R \subseteq \underline{x}$*

BUT, limited to  $J \lesssim 3$

*↑ relevant  
variable  
subset*

Plot partial (average) dependence of  $\hat{F}(\mathbf{x})$

on selected small subsets of  $\{x_j\}_1^J$

Incomplete - but useful if subsets chosen carefully

Any black-box model (SVM, NN, etc)

## PARTIAL DEPENDENCE FUNCTIONS

↓ focus on

$$z_l = \{z_1, \dots, z_l\} \subset \mathbf{x}, \quad z_{\setminus l} \cup z_l = \mathbf{x}$$

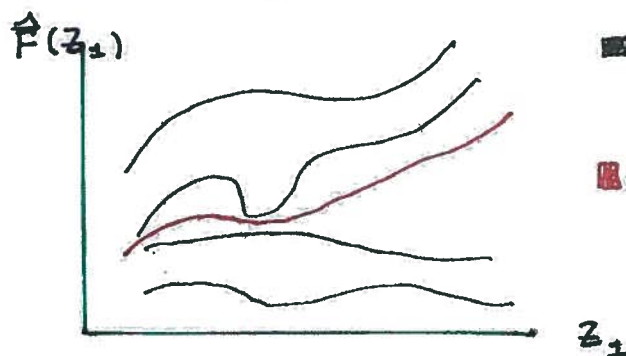
$$F(\mathbf{x}) = F(z_l, z_{\setminus l}) = F(z_l | z_{\setminus l}) = F_{z_{\setminus l}}(z_l)$$

= function of  $z_l$  with "parameters"  $z_{\setminus l}$

"Partial" (average) dependence of  $F(\mathbf{x})$  on  $z_l$ :

$$\bar{F}_l(z_l) = E_{z_{\setminus l}}[F(\mathbf{x})] = \int F(\mathbf{x}) p_{\setminus l}(z_{\setminus l}) dz_{\setminus l}$$

$$\hat{\bar{F}}_l(z_l) = \frac{1}{N} \sum_{i=1}^N F(z_l, z_{i \setminus l})$$



- different values of  $z_{\setminus l}$
- pointwise average = partial dep.

Describes *complete* dependence if

$$F(\mathbf{x}) = F_l(z_l) + F_{\setminus l}(z_{\setminus l})$$

$$F(\mathbf{x}) = F_l(z_l) \cdot F_{\setminus l}(z_{\setminus l}) \quad (\text{diagnostics})$$

$\bar{F}_l(z_l)$  = contrib. of  $z_l$  to  $F(\mathbf{x})$  after accounting  
NOT equivalent to for (average) effects of  $z_{\setminus l}$

$$\bar{F}_l(z_l) = E_{\mathbf{x}}[F(\mathbf{x}) | z_l] = \int F(\mathbf{x}) p(\mathbf{x} | z_l) dx$$

contribution of  $z_l$  ignoring  $z_{\setminus l}$

Trees:  $\bar{F}_l(z_l) \sim$  weighted tree traversal

MART: average over trees.

Classification: average over trees  
separately for each class  $\Rightarrow$   
partial dep. of  $F(\mathbf{x})$  on  $z_l$   
for class  $k$ .