Boosting for regression

$x_1, x_2, \dots, x_n \in \mathbb{R}^p$

$y_1 \; y_2 \; \dots \; y_n \in \mathbb{R}$

Fix: B – number of trees

λ – shrinking λ-small – requires large B

d – number of splits in each tree ($d+1$ leaves)

1) Set $\hat{f}(x)=0$ and $r_i = y_i$, $i \in \{\text{training set}\}$

$\begin{cases} \hat{f}(x_i) + r_i = y_i \\ \text{prediction} \quad \text{error} \quad \text{actual value} \end{cases}$

2) For $b=1,2,\dots,B$:

(a) fit a tree $\hat{f}^b$ with d splits to the training data (X, r)

(b) update $\hat{f}$: $\text{new } \hat{f}(x) := \hat{f}(x) + \lambda \hat{f}^b(x)$

(c) update the residuals:

$\text{new } r_i := r_i - \lambda \hat{f}^b(x)$

3) $\rightsquigarrow$ model: $\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$

$\hat{f}^b(x) = \sum c_j \mathbb{1}_{R_j}(x)$

$\lambda=1$

$\lambda=0.1$

x:	135	141	156	132	131
y:	11	12	11	13	14
$\hat{f}$:	11	13	11	13	13
r:	0	−1	0	0	+1
r:	2.9	10.7	9.9	11.7	12.7

$= \hat{f}^1$

Ada Boost for binary classification

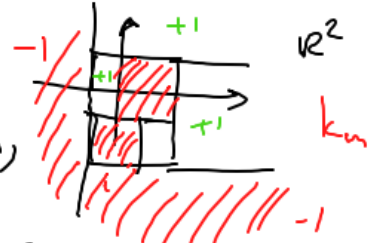
Dataset: (X, Y) $X=(x_1,\dots,x_N)$, $y=(y_1,\dots,y_N)$, $x_i \in \mathbb{R}^p$, $y_i \in \{-1,1\}$

{ we will construct a function $C: \mathbb{R}^p \to \mathbb{R}$
classification is done by $\operatorname{sgn}(C(x))$

- $C_0(x) = 0$
- $C_m(x) = C_{m-1}(x) + \alpha_m k_m(x)$

  $\alpha_m \geq 0$, k_m - classifier, $k_m: \mathbb{R}^p \to \{-1,1\}$

How to choose α_m, k_m? — To minimize the total error of C_m,



$$E = \sum_{i=1}^{N} e^{-y_i C_m(x_i)} = \sum_{i=1}^{N} e^{-y_i C_{m-1}(x_i)} e^{-y_i \alpha_m k_m(x_i)} = \sum_{i=1}^{N} w_i^{(m)} e^{-y_i \alpha_m k_m(x_i)}$$

$w_i^{(m)} := e^{-y_i C_{m-1}(x_i)}$

{ e.g. $y_i=-1$, $C_m(x_i) = -10 \rightsquigarrow e^{-y_i C_m(x_i)} = e^{-10}$ — very small
$C_m(x_i) = -0.1 \rightsquigarrow e^{-y_i C_m(x_i)} = e^{-0.1}$ — <1, but close to 1
$C_m(x_i) = 1 \rightsquigarrow e^{-y_i C_m(x_i)} = e^{1} = e$
$C_m(x_i) = 3 \rightsquigarrow = e^{3}$

$$E = \sum_{i=1}^{N} w_i^{(m)} e^{-y_i \alpha_m k_m(x_i)} = \sum_{i: y_i = k_m(x_i)} w_i^{(m)} e^{-\alpha_m} + \sum_{i: y_i \neq k_m(x_i)} w_i^{(m)} e^{\alpha_m} =$$

$$= \underbrace{\left(\sum_i w_i^{(m)}\right) \cdot e^{-\alpha_m}}_{\text{does not depend on } x} + \underbrace{\sum_{i: y_i \neq k_m(x_i)} w_i^{(m)} \left(e^{\alpha_m} - e^{-\alpha_m}\right)}$$

$$= \underbrace{\left(e^{\alpha_m} - e^{-\alpha_m}\right)}_{\text{does not depend on } k \text{ and } \neq 0} \underbrace{\sum_{i: y_i \neq k_m(x_i)} w_i^{(m)}}_{\text{does not depend on } \alpha_m, \text{ only on } k_m}$$

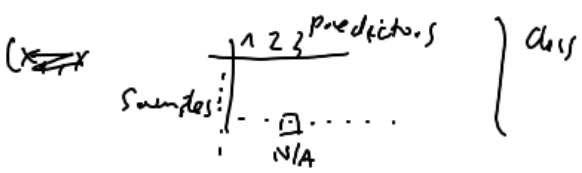
→ k_m should be chosen to minimise $\sum_{i: y_i \neq k_m(x_i)} w_i^{(m)}$, i.e., weighted error.

once k_m is chosen, we find α_m to minimise E

$$\rightsquigarrow \alpha_m = \frac{1}{2} \ln \frac{\sum_{i: y_i = k_m(x_i)} w_i^{(m)}}{\sum_{i \neq k_m(x_i)} w_i^{(m)}}$$

Drawback: excessive weight assigned to outliers

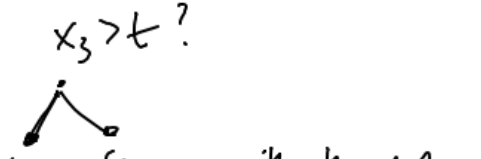
Handling missing data:



- ignore records with missing attribute values
  (or ignore an attribute)

  drawback:
  could result in bias
  smaller dataset

- 'guess' the missing attribute value
- treat missing values as a special category
- trees: split the sample (used in C4.5 by Ross Quinlan)
  C5.0



– with the data

Trees 'ask' question about just one attribute at a time, so missing data is less problematic than in the case of other methods. (But not in scipy)

Trees may be used to find important attributes.

LDA, QDA (Linear / Quadratic Discriminant Analysis)

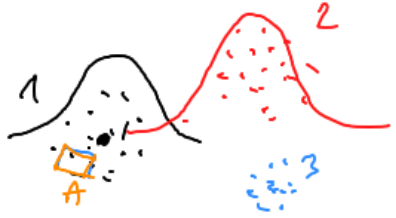
$X = (x_1, \dots, x_N)$, $x_i \in \mathbb{R}^p$ $\quad y_i \in \{1, 2, \dots, K\}$

Model distribution of X seperately in each of the responses classes (i.e., given Y)

and use Bayes Theorem

$$\Pr(Y = k \mid X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

these are predicted probabilities that given sample is in the class k $(k = 1, 2, \dots, K)$



$f_1 \sim N(\cdot, \cdot)$

$$\pi_k = \Pr(Y = k) = \underbrace{\frac{\#\{i : y_i = k\}}{N}}_{\text{estimate}}$$

How to estimate f_k?

we assume that f_k is of some special form and try to estimate the coefficients

$f_k(x) = \text{“}\Pr(X = x \mid Y = k)\text{”}$ density function

$$\Pr(X \in A \mid Y = k) = \int_A f_k(x)\, dx$$