2. Greedy is good for $p \gg n$: Boosting

data: $(X_1, Y_1), \ldots, (X_n, Y_n)$ (i.i.d. or stationary),
predictor variables $X_i \in \mathbb{R}^p$
response variables $Y_i \in \mathbb{R}$ or $Y_i \in \{0, 1, \ldots, J-1\}$

aim: estimation of function $f(\cdot) : \mathbb{R}^p \to \mathbb{R}$ (including feature selection) e.g.
$f(x) = \mathbf{E}[Y|X = x]$ or $f(x) = \mathbb{P}[Y = 1|X = x]$ with $Y \in \{0, 1\}$
or distribution of survival time Y given X depends on some function $f(X)$ only

our setting: typically p is very large

historically: Boosting is an ensemble scheme (multiple predictions and averaging)

base procedure:

data $\xrightarrow{\text{algorithm A}}$ $\hat{\theta}(\cdot)$ (a function estimate)

e.g.: simple linear regression, tree, MARS, "classical" smoothing, neural nets, ...

generating multiple predictions:

weighted data 1 $\xrightarrow{\text{algorithm A}}$ $\hat{\theta}_1(\cdot)$

weighted data 2 $\xrightarrow{\text{algorithm A}}$ $\hat{\theta}_2(\cdot)$

$\ldots$ $\ldots$

weighted data M $\xrightarrow{\text{algorithm A}}$ $\hat{\theta}_M(\cdot)$

Aggregation: $\hat{f}_A(\cdot) = \sum_{m=1}^{M} a_m \hat{\theta}_m(\cdot)$

data weights? averaging weights a_m?

classification of 2 lymph nodal status in breast cancer using gene expressions from microarray data:

$n = 33$, $p = 7129$ (for CART: gene-preselection, reducing to $p = 50$)

method	test set error	gain over CART
CART	22.5%	–
LogitBoost with trees	16.3%	28%
LogitBoost with bagged trees	12.2%	46%

2.1. Boosting algorithms

AdaBoost proposed for classification by Freund & Schapire (1996)

data weights (rough original idea): large weights to previously heavily misclassified instances (sequential algorithm)

averaging weights a_m: large if in-sample performance in mth round was good

Why should this be good?

(actually: other weighting schemes are equally good or better...)

Breiman (1998/99):

AdaBoost is functional gradient descent (FGD) procedure

a mix of statistical estimation and numerical optimization...

2.2 L_2Boosting

(see also Friedman, 2001)

L_2Boosting with base procedure $\hat{\theta}(\cdot)$ is a "constrained minimization" of empirical risk $n^{-1}\sum_{i=1}^n (Y_i - f(X_i))^2$ w.r.t. $f(\cdot)$

$\rightsquigarrow$ useful for regression

$$m=1: \ (X_i, Y_i)_{i=1}^n \rightsquigarrow \hat{\theta}_1(\cdot), \ \hat{f}_1 = \nu\hat{\theta}_1 \quad \rightsquigarrow \text{ resid. } U_i = Y_i - \hat{f}_1(X_i)$$
$$m=2: \ (X_i, U_i)_{i=1}^n \rightsquigarrow \hat{\theta}_2(\cdot), \ \hat{f}_2 = \hat{f}_1 + \nu\hat{\theta}_2 \quad \rightsquigarrow \text{ resid. } U_i = Y_i - \hat{f}_2(X_i)$$
$$\ldots \qquad\qquad \ldots$$

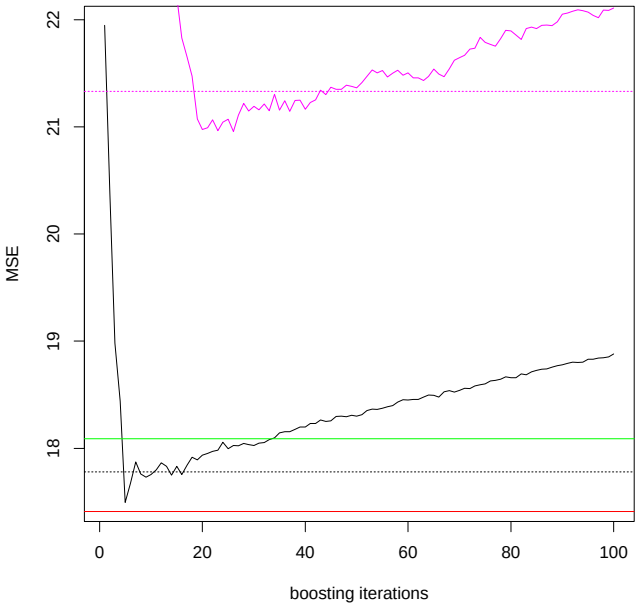
$f_{m_{stop}}(\cdot) = \nu \sum_{m=1}^{m_{stop}} \hat{\theta}_m(\cdot)$, m_{stop} a tuning parameter

repeated greedy fitting (with shrinkage ν) of residuals

Tukey (1977): twicing for $m_{stop} = 2$ and $\nu = 1$

Any gain over classical methods? (for additive modeling)

Ozone data: n=300, p=8



$n = 300,\ p = 8$

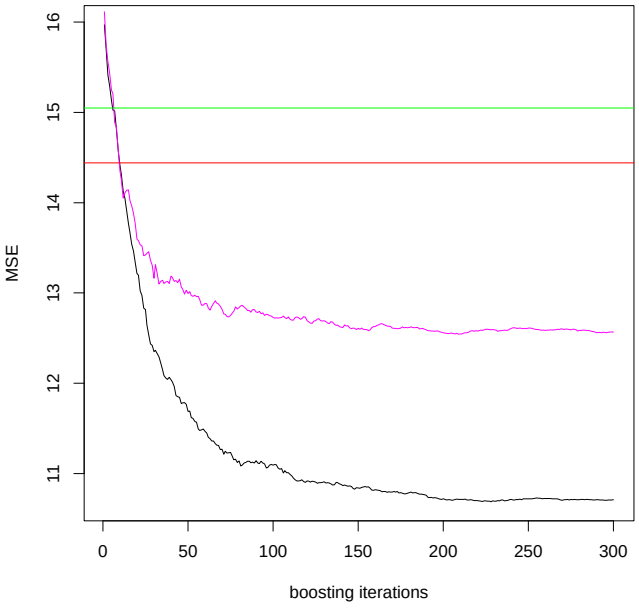
- magenta: L_2Boosting with stumps
  (horiz. line = cross-validated stopping)
- black: L_2Boosting with componentwise smoothing spline
  (horiz. line = cross-validated stopping)
  i.e: smoothing spline fi tting against the selected predictor which reduces RSS most
- green: MARS restricted to additive modeling
- red: additive model using backfi tting

L_2Boosting with stumps or comp. smoothing splines also yields additive model:

$$\sum_{m=0}^{M} \hat{\theta}_m(x^{(\hat{\mathcal{S}}_m)})) = \hat{g}_1(x^{(1)}) + \ldots + \hat{g}_p(x^{(p)})$$

Simulated data: non-additive regression function, $n = 200, p = 100$

Regression: n=200, p=100



- magenta: L_2Boosting with stumps
- black: L_2Boosting with componentwise
- green: MARS restricted to additive modeling
- red: additive model using backfi tting and fwd. var. selection

similar for classification

very often: boosting performs comparatively well in high-dimensions

(there is a lot of empirical evidence for this)

also SVM is often surprisingly accurate...

2.3. Choice of the base procedure

most popular in machine learning: tree algorithms (CART, C4.5)

they do variable/feature selection

have seen: for componentwise smoothing splines or stumps

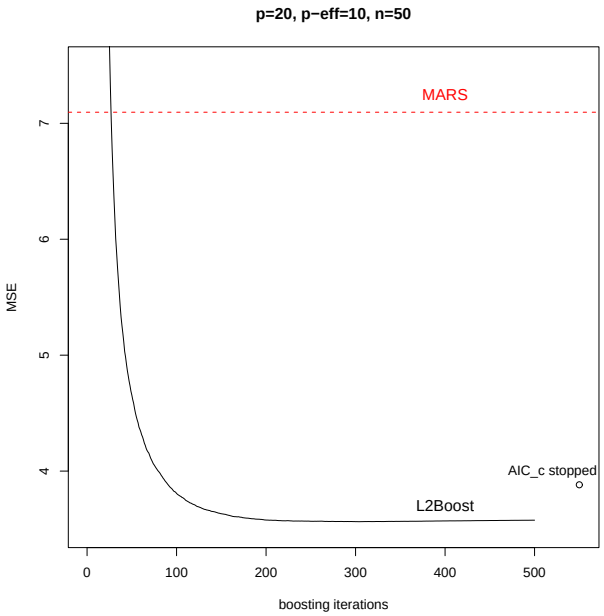
$\longrightarrow$ boosting yields an additive model fit

$\rightsquigarrow$ we can use boosting for fitting in "quite many" structural models

Example: degree 2 nonparametric interaction modeling

Friedman #1 model:

$$Y = 10\sin(\pi X_1 X_2) + 20(X_3 - 0.5)^2 + 10X_4 + 5X_5 + \mathcal{N}(0,1),\ X = (X_1, \ldots, X_{20}) \sim \text{Unif.}([0,1]^{20})$$



L_2Boosting with pairwise splines

sample size $n = 50$

$p = 20$, effective $p_{eff} = 5$

2.4. L_2Boosting for high-dimensional linear models

linear model

$$Y = f(X) + \varepsilon,$$

$$f(x) = \sum_{j=1}^{p} \beta_j x^{(j)}, \quad p \gg n$$

or: a highly over-complete dictionary $\{g_j(\cdot); j = 1, \ldots, p \gg n\}$

our approach: L_2Boosting with componentwise linear LS regression

This base procedure fits a univariate linear regression model against the one predictor variable which reduces residual sum of squares most

first round of estimation: selected predictor variable $X^{(\hat{S}_1)}$ (e.g. $= X^{(3)}$)

corresponding ordinary least squares $\hat{\beta}_{\hat{S}_1}$

use shrunken fit $\hat{f}_1 = \nu \hat{\beta}_{\hat{S}_1} X^{(\hat{S}_1)}$ (e.g. $\nu = 0.1$)

second round of estimation: selected predictor variable $X^{(\hat{S}_2)}$ (e.g.$= X^{(21)}$)

corresponding OLS $\hat{\beta}_{\hat{S}_2}$

use shrunken fit $\hat{f}_2 = \hat{f}_1 + \nu \hat{\beta}_{\hat{S}_2} X^{(\hat{S}_2)}$

etc.

very different from forward variable selection

this method does variable selection and

assigns variable amount of degrees of freedom for selected variables (shrinkage)

not full OLS on selected variables (even with $\nu = 1$)

For $\nu = 1$, this L_2Boosting is known as Matching Pursuit (Mallat and Zhang, 1993)

Gauss-Southwell algorithm

C.F. Gauss in 1803

"Princeps Mathematicorum"

R.V. Southwell in 1933

Professor in engineering

Oxford University

because of

variable selection and

assigning variable amount of degrees of freedom (shrinkage) for selected variables

reminds to Lasso (ℓ^1-penalized regression) (Tibshirani, 1996)

$$\hat{\beta}_{Lasso} = \text{argmin}_\beta \sum_{i=1}^{n}(Y_i - \sum_{j=1}^{p} \beta_j X_i^{(j)})^2 + \underbrace{\lambda}_{\geq 0} \sum_{j=1}^{p} |\beta_j|$$

and indeed: there is a relation (Efron, Hastie, Johnstone, Tibshirani, 2004)

but: the algorithms and estimates are not the same

Theorem for high dimensions (PB, 2004)

L_2Boosting with comp. linear LS regression is consistent (for suitable number of boosting iterations) if:

- $p_n = O(\exp(Cn^{1-\xi}))$ $(0 < \xi < 1)$
  essentially exponentially many variables relative to n
- $\sup_n \sum_{j=1}^{p_n} |\beta_{j,n}| < \infty$ ℓ_1-sparseness of true function

i.e. for suitable, slowly growing $m = m_n$:

$$\mathbf{E}_X |\hat{f}_{m_n,n}(X) - f_n(X)|^2 = o_P(1) \ (n \to \infty)$$

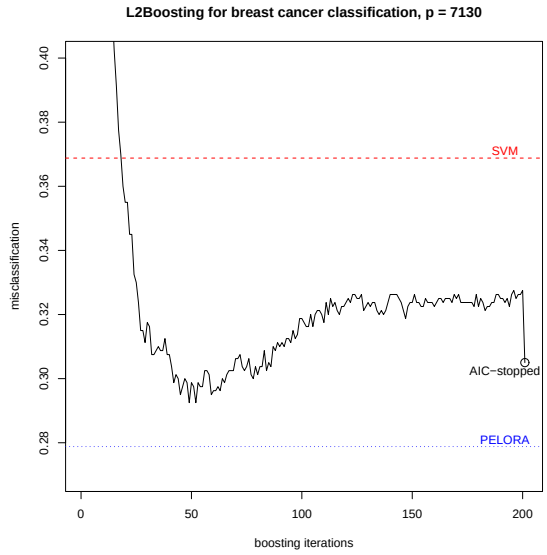
"no" assumptions about the predictor variables/design matrix

in other words:

consistency for de-noising sparse signal with highly over-complete dictionaries

similar result has been given for the Lasso by Greenshtein and Ritov (2004)

binary lymph node classification in breast cancer using gene expressions



$n = 49, p = 7130$ gene expressions

- black: L_2Boosting with componentwise linear LS regression
- red: SVM with radial basis kernel
- blue: Pelora: a "biologically inspired" gene grouping method (Dettling & PB, 2004)

42 out of $p = 7130$ genes are selected (some of them biologically meaningful)

good prediction and interesting gene selection

3. L_2Boosting, Lasso and LARS

Efron et al. (2004): intriguing relation between L_2Boosting and Lasso: $\hat{\beta}_{Lasso} = \text{argmin}_\beta \sum_{i=1}^n (Y_i - \sum_{j=1}^p \beta_j X_i^{(j)})^2 + \lambda \sum_{j=1}^p |\beta_j|$

for some special cases, roughly:

iterations of "L_2Boosting with "infinitesimally" small ν yield all Lasso solutions when varying λ"

$\rightsquigarrow$ computationally interesting to produce all Lasso solutions in one sweep of boosting

Least Angle Regression LARS (Efron et al., 2004) is computationally even more clever and efficient than L_2Boosting

for $p \gg n$

both: Lasso/LARS and L_2Boosting are very useful

and LARS is really fast