Very High-Dimensional Data: Greedy Boosting and Convex Lasso-Relaxation

Peter Bühlmann

ETH Zürich

1. High-dimensional data from gene expressions

central dogma from molecular biology

subregions of DNA (genes) $\xrightarrow{\text{transcription}}$ mRNA $\longrightarrow$ protein

building stones of cell

$\longrightarrow$ phenotype

can measure nowadays "whether and how strongly" genes are transcribed,

simultaneously for thousands of genes

technical terminology: gene is expressed (strongly expressed; or not expr.)

gene expression can be thought as a "continuous switch between on and off"

Affymetrix gene chip

tremendous breakthrough in molecular biology!

(Brown and Botstein labs at Stanford, mid 1990's)

structure of the data:

"unsupervised":

$X_1, \ldots, X_n,$ usually assumed to be i.i.d. , $\underbrace{X_i \in \mathbb{R}^p}$

p gene expressions from individual i

"supervised": additional information about the individuals

e.g. cancerous or non-cancerous (or survival time, etc.)

encoded as univariate response variables $Y_1, \ldots, Y_n$

$(X_1, Y_1), \ldots, (X_n, Y_n),$ usually assumed to be i.i.d.

in both cases:

typically: $n \approx 10 - 200, p \approx 5'000 - 20'000$

the $p \gg n$ problem!

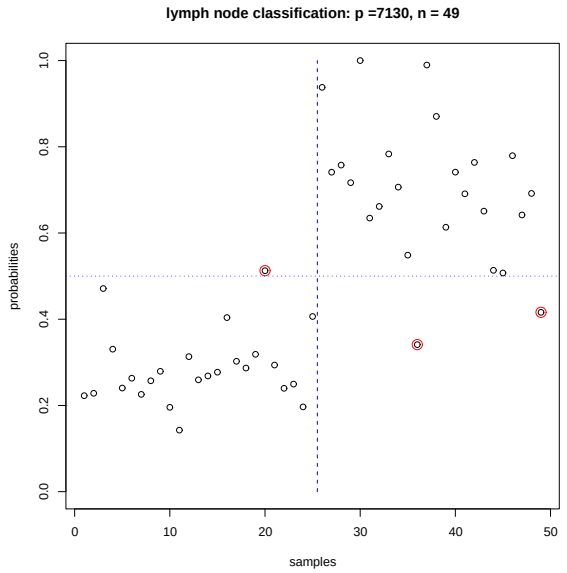
some goals for supervised problems:

- classification

  e.g. predicting in early stage of disease whether patient develops a tumor-subtype

  (prognosis in an early stage of a disease)

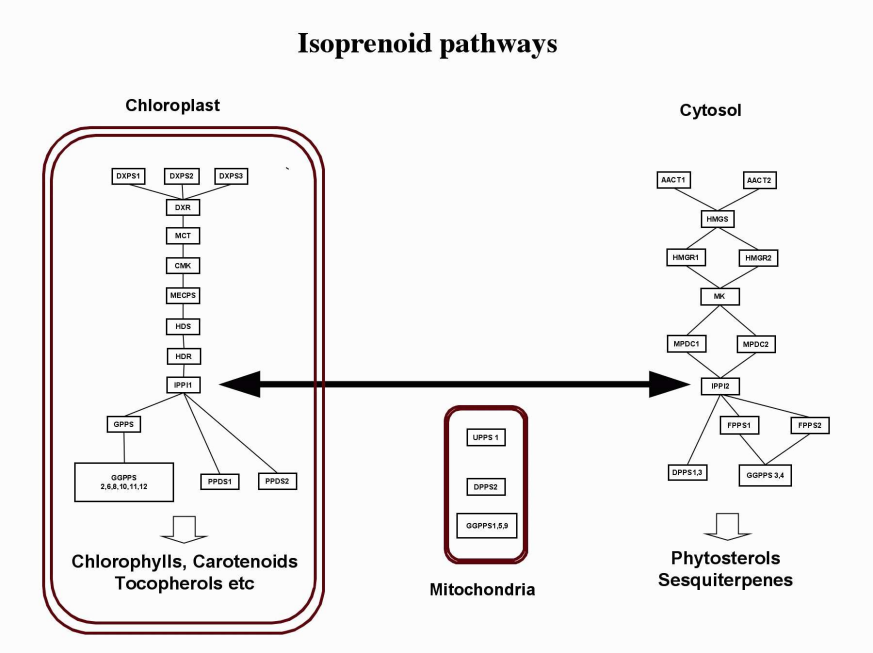
- estimating $\mathbb{P}[Y = 1 | X = x]$



- feature selection

  selection of genes which are “relevant” for e.g. certain tumor sup-type

an “unsupervised” problem: two isoprenoid pathways in Arabidopsis Thaliana



goal: associations not causality...

understand more about cross-talk on transcriptional (gene expression) level

data: $n = 118$ Affymetrix gene expression measurements; $p = 39$ genes

plus additional biological information

fairly high-dimensional

very high-dimensional when incorporating $\approx 1'000$ potential transcription factors

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

supervised data: $(X_1, Y_1), \ldots, (X_n, Y_n)$ (i.i.d. or stationary),

predictor variables $X_i \in \mathbb{R}^p$ (typically p very large)

uni- or multivariate response variables $Y_i \in \mathbb{R}$ (or $\mathbb{R}^q$) or $Y_i \in \{0, 1, \ldots, q-1\}$

aim: estimation of function $f(\cdot) : \mathbb{R}^p \to \mathbb{R}$ (or $\mathbb{R}^q$)

including feature selection e.g.

$f(x) = \mathbf{E}[Y|X = x]$ or $f(x) = \mathbb{P}[Y = y|X = x]$

$f(\cdot) =$ survival time function

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

base procedure:

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

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

generating multiple predictions:

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

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

$$\ldots \qquad\qquad \ldots$$

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

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 binary 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...)

assume univariate response Y in the sequel

Breiman (1998/99):

AdaBoost is functional gradient descent (FGD) procedure

Aim: find $f^*(\cdot) = \text{argmin}_{f(\cdot)} \mathbf{E}[\rho(Y, f(X))]$

e.g. for $\rho(y, f) = |y - f|^2 \rightsquigarrow f^*(x) = \mathbf{E}[Y|X = x]$

FGD solution: consider empirical risk $n^{-1} \sum_{i=1}^n \rho(Y_i, f(X_i))$ and

do iterative steepest descent in function space

Generic FGD algorithm

Step 1. $\hat{f}_0 \equiv 0$; set $m = 0$.

Step 2. Increase m by 1. Compute negative gradient $-\frac{\partial}{\partial f}\rho(Y, f)$
and evaluate at $f = \hat{f}_{m-1}(X_i) = U_i \ (i = 1, \ldots, n)$

Step 3. Fit negative gradient vector $U_1, \ldots, U_n$ by base procedure

$$(X_i, U_i)_{i=1}^n \xrightarrow{\text{algorithm A}} \hat{\theta}_m(\cdot)$$

e.g. $\hat{\theta}_m(\cdot)$ fitted by (weighted) least squares
i.e. $\hat{\theta}_m(\cdot)$ is an approximation of the negative gradient vector

Step 4. Up-date $\hat{f}_m = \hat{f}_{m-1}(\cdot) + \nu \cdot \hat{\theta}_m(\cdot)$ $(0 < \nu \leq 1$ a step-length)
i.e: proceed along an estimate of the negative gradient vector

Step 5. Iterate Step 2 until $m = m_{stop}$ for some stopping iteration m_{stop}

Why "functional gradient"?

Alternative formulation in function space:

empirical risk functional: $C(f) = n^{-1} \sum_{i=1}^{n} \rho(Y_i, f(X_i))$

inner product: $\langle f, g \rangle = n^{-1} \sum_{i=1}^{n} f(X_i) g(X_i)$

negative Gateaux derivative:

$$-dC(f)(x) = \frac{\partial}{\partial \alpha} C(f + \alpha 1_x)|_{\alpha=0}, \rightsquigarrow -dC(\hat{f}_{m-1})(X_i) = U_i$$

if $U_1, ..., U_n$ are fitted by least squares and base procedure is normed ($\|\hat{\theta}\| = 1$)

equivalent to maximize $\langle -dC(f_m), \theta \rangle$ w.r.t. $\theta(\cdot)$ (over all possible θ's from the base procedure)

i.e: $\hat{\theta}_m(\cdot)$ is the best approximation (most parallel)

to the negative gradient $-dC(f_m)$

By definition: FGD yields additive combination of base procedure fits

$$\nu \sum_{m=1}^{m_{stop}} \hat{\theta}_m(\cdot)$$

Breiman (1998):

FGD with $\rho(y, f) = \exp((2y-1) \cdot f)$ for binary classification yields the AdaBoost algorithm

(great result!)

Remark: FGD can not be represented as some explicit estimation function(al):

$$\hat{f}_m(\cdot) \neq \text{argmin}_{f \in \mathcal{F}} n^{-1} \sum_{i=1}^{n} \rho(Y_i, f(X_i)) \quad \text{for some function class } \mathcal{F}$$

$\rightsquigarrow$ FGD is mathematically more difficult to analyze but generically applicable (as an algorithm!) in very complex models

2.2. L_2Boosting

(see also Friedman, 2001)

loss function $\rho(y, f) = |y - f|^2$

population minimizer: $f^*(x) = \mathbf{E}[Y|X = x]$

FGD with base procedure $\hat{\theta}(\cdot)$: repeated fitting of residuals

$$m = 1: \ (X_i, Y_i)_{i=1}^n \rightsquigarrow \hat{\theta}_1(\cdot), \ f_1 = \hat{\theta}_1 \qquad \rightsquigarrow \text{ resid. } U_i = Y_i - f_1(X_i)$$

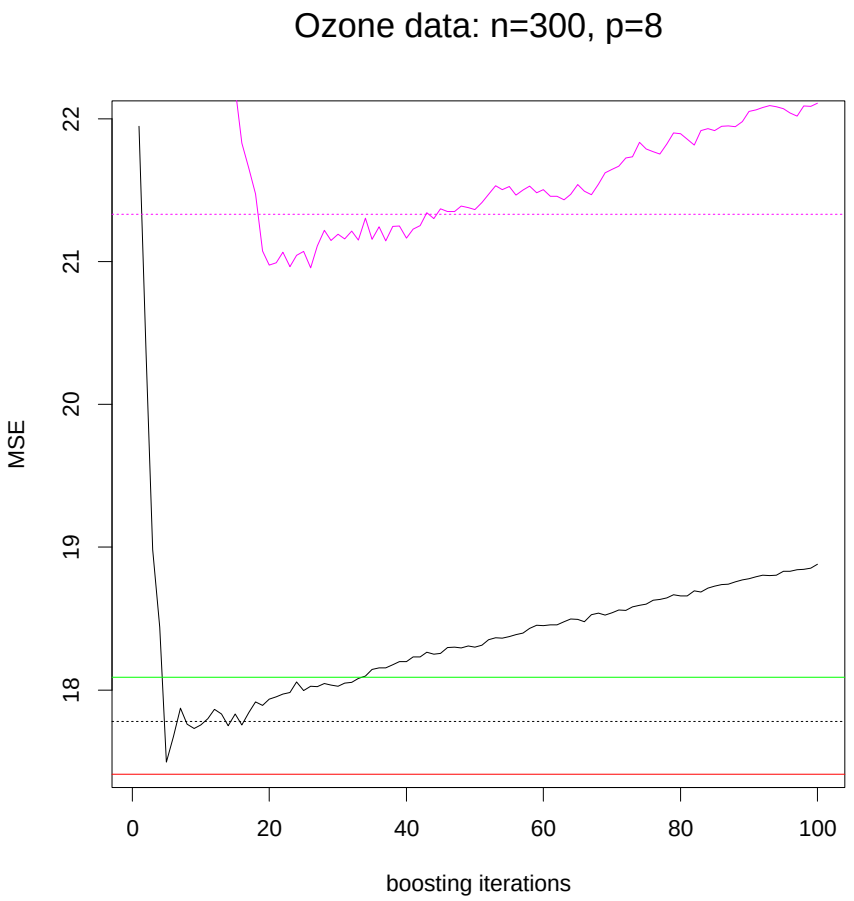
$$m = 2: \ (X_i, U_i)_{i=1}^n \rightsquigarrow \hat{\theta}_2(\cdot), \ f_2 = f_1 + \hat{\theta}_2 \qquad \rightsquigarrow \text{ resid. } U_i = Y_i - f_2(X_i)$$

$$\ldots \qquad\qquad \ldots$$

$f_{m_{stop}}(\cdot) = \nu \sum_{m=1}^{m_{stop}} \hat{\theta}_m(\cdot)$ (stagewise greedy fitting of residuals)

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

any gain over classical methods? (for additive modeling)



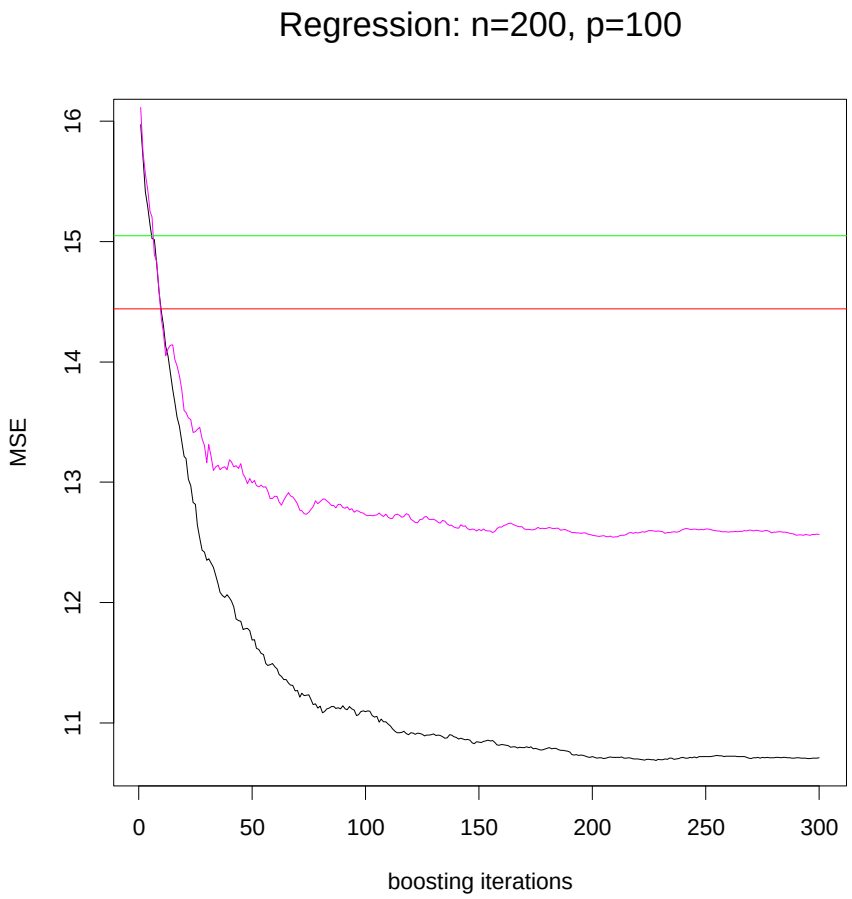
$n = 300,\ p = 8$

- 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:

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

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



- 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)

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

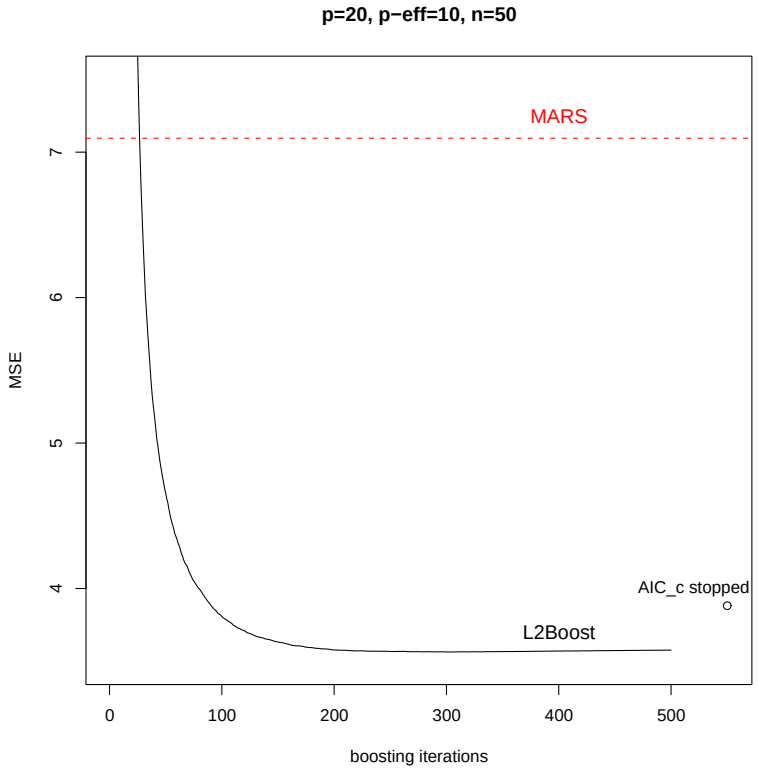
$\longrightarrow$ boosting yields an additive model fit

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

Example: degree 2 nonparametric interaction modelling

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 $p_{eff} = 5$

both methods have the same (high) degree of interpretability

3. 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 RSS most

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

corresponding $\hat{\beta}_{\hat{S}_1}$

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

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

etc.

this method does variable selection and

assigns variable amount of degrees of freedom for selected variables

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

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 $m = m_n$:

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

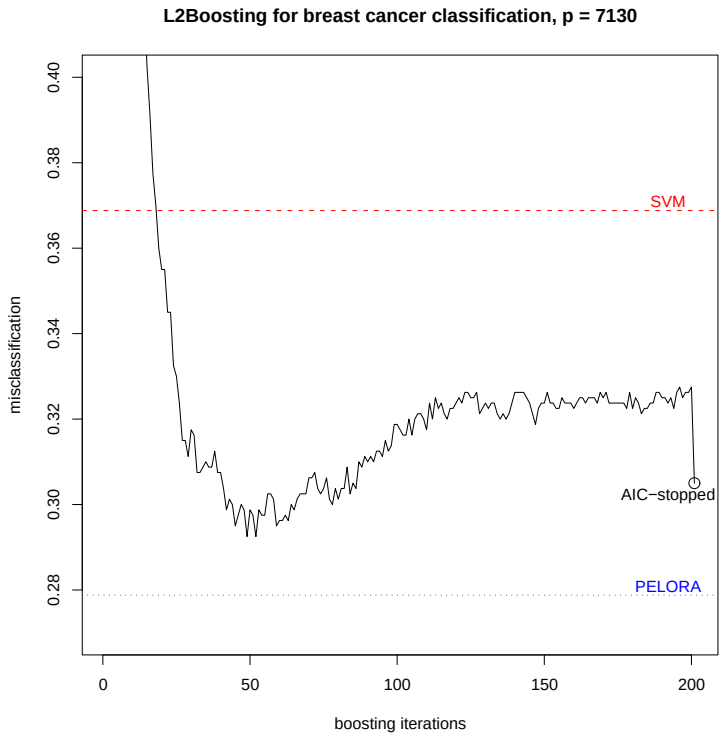
"no" assumptions about the predictor variables/design matrix

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: a high noise problem



42 out of $p = 7130$ genes are selected

$n = 49, p = 7130$ 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)

- competitive but clinically not very accurate prediction
- interesting gene selection

are these selected genes biologically meaningful?

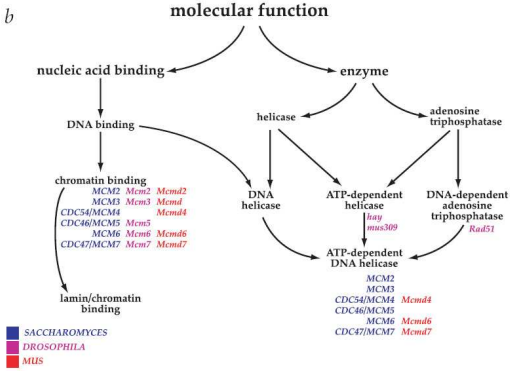
note: if $|\mathrm{Corr}(X_{sel}, X_k)| \approx 1$ for some $k \neq sel$

can "replace" X_{sel} by $X_k \rightsquigarrow$ a severe identifiability problem

a biologically useful "repair" via GO (**G**ene **O**ntology)

suppose we have selected $X_{\hat{S}_1}, \ldots, X_{\hat{S}_m}$

- build groups of genes $\mathcal{G}_{\hat{S}_j} = \{X_k;\ |\mathrm{Corr}(X_{\hat{S}_j}, X_k)| \text{ large}\}$
- assign significance of such gene groups in terms of functional GO categories



this yields classification in terms of functional gene categories

(instead of single genes)

Boosting/Gauss-Southwell idea is very generic

it can be used for possibly high-multivariate responses (Lutz and PB, 2005):

- high multi-category classification (e.g. gene annotation)

- high-dimensional linear time series

similar consistency theory:

for sparse multivariate linear regression and

for sparse linear time series models

with large dimensions $O(\exp(Cn^{1-\xi}))$ in the predictor and the response

(Lutz and PB, 2005)

4. Lasso-relaxation is good for $p \gg n$

consider again linear model (or highly overcomplete dictionary)

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

Lasso or ℓ^1-penalized regression (Tibshirani, 1996):

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

- does variable selection: some (many) β_j's exactly equal to 0
- does shrinkage
- involves a convex optimization only

this is convex relaxation:

replace the computationally hard/infeasible subset selection (ℓ^0-penalty)

by the convex ℓ^1-penalized problem

"similar" properties of convex realaxation (Lasso) and greedy algorithm (Boosting)

- variable selection
- shrinkage

and indeed: there are relations

Efron, Hastie, Johnstone, Tibshirani (2004): for special design matrices,

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

Zhao and Yu (2005): in general, when adding some backward step

the solutions from Lasso and Boosting coincide

greedy (plus backward steps) and convex relaxation are surprisingly similar

for $p \gg n$

both: Lasso/LARS and L_2Boosting are very useful

both are computationally attractive: $O(p)$ operation counts for $p \gg n$

and LARS is really fast

Results for high noise, binary lymph node classification

cross-validated misclassification rate:

Lasso (tuned by 5-fold CV): 27.3 %

L_2Boosting (tuned by AIC): 30.2 %

selected genes (on whole data set):

Lasso: 23 genes L_2Boosting: 42 genes

7 genes are selected by both methods

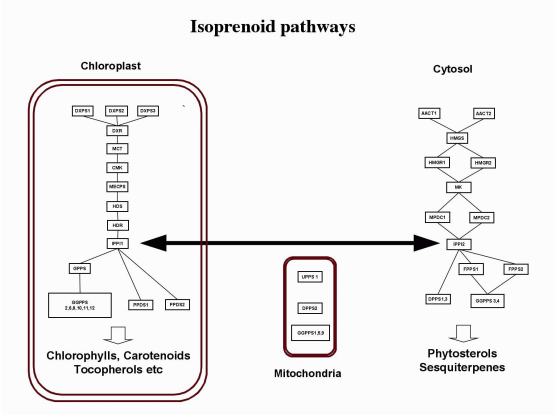
note the identifiability problem again

5. Variable selection and graphical modeling with the Lasso

goal: use the Lasso for variable selection in regression

determine presence/absence of associations between random variables

look-out: associations among expressions of 39 genes from the two biosynthesis pathways in Arabidopsis



5.1. Gaussian conditional independence graph

assume that $X = X_1, \ldots, X_p \sim \mathcal{N}_p(\mu, \Sigma)$

graph:

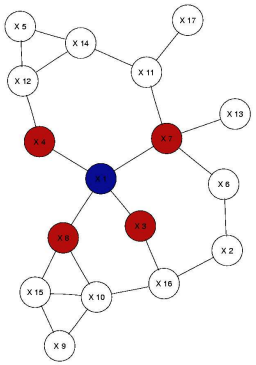
set of nodes $\Gamma = \{1, 2, \ldots, p\}$, corresponding to the p random variables

set of edges $E \subseteq \Gamma \times \Gamma$ defined as:

there is an undirected edge between node i and j

$\overset{\text{def}}{\Leftrightarrow}$ X_i conditionally dependent of X_j given all other $\{X_k;\ k \neq i, j\}$

$\Leftrightarrow$ $\Sigma^{-1}_{ij} \neq 0$



note: Σ^{-1}_{ij} corresponds to $\beta^{(i)}_j = \Sigma^{-1}_{ij}/\Sigma^{-1}_{ii}$, where

$$X_i = \beta^{(i)}_j X_j + \sum_{k \neq i,j} \beta^{(i)}_k X_k + \text{error}^{(i)}$$

$\rightsquigarrow$ we can infer the graph from variable selection in regression

$$\beta^{(i)}_j = 0 \Leftrightarrow \Sigma^{-1}_{ij} = 0$$

huge computational problem when using e.g. BIC: $p2^{p-1}$ least squares problems!

5.2. Just relax!

replace the computationally hard problem by a convex problem:

compute the Lasso estimates $\hat{\beta}_i^{(j)}$

Estimation of graph

estimate an edge between node i and j if

$$\hat{\beta}_j^{(i)} \neq 0 \text{ and } \hat{\beta}_i^{(j)} \neq 0$$

(for finite samples: it could happen that only one of the $\hat{\beta}_j^{(i)}, \hat{\beta}_i^{(j)}$ is $\neq 0$)

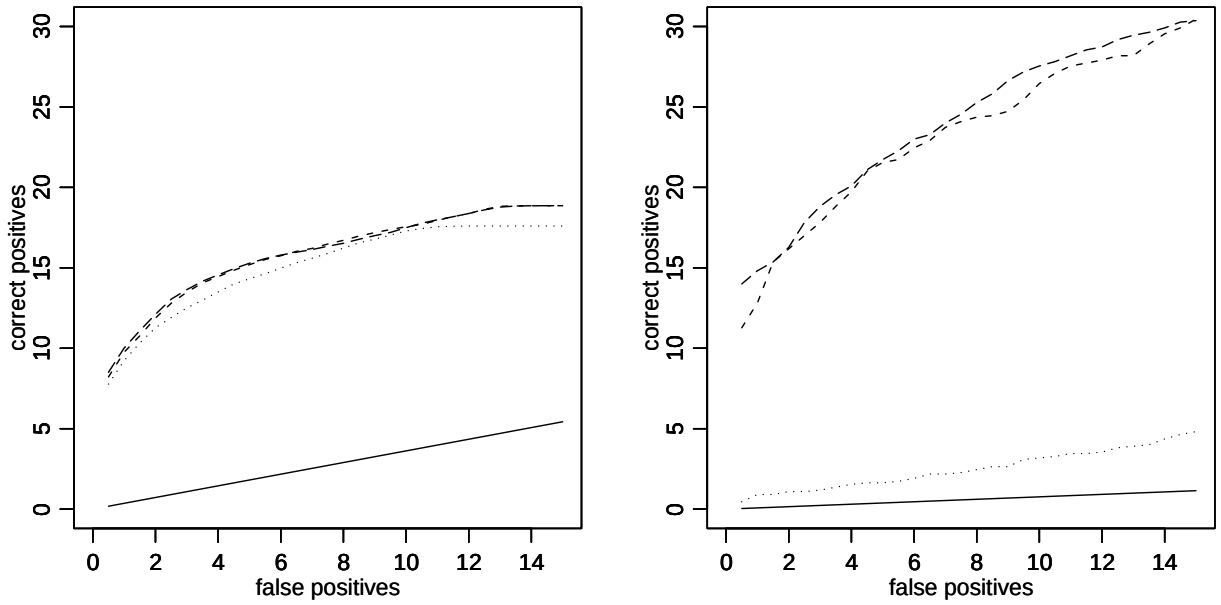
note: depends on the tuning parameter λ in Lasso

this involves only one convex optimization problem!

instead of checking exhaustively $2^{p-1}p$ least squares problems (e.g. using BIC)

Comparison of Lasso and classical stepwise selection

$p = 10$ $\qquad\qquad$ $p = 30$



dotted $\cdots$ stepwise selection

dashed $- - -$ Lasso

ROC-curves for estimated graphs with $p = 10, 30$ nodes and $n = 40$ obs.

true graphs are sparse, having at most 4 edges out of every node

5.3 Some theory for high dimensions

Theorem (Meinshausen & PB, 2004)

For $\lambda_n \sim Cn^{-1/2+\delta/2}$,

$$\mathbb{P}[\text{estimated graph}(\lambda_n) = \text{ true graph}] = 1 + O(\exp(-Cn^{\delta})) \quad (n \to \infty)$$

$$(0 < \delta < 1)$$

if

- Gaussian data
- $p = p_n = O(n^r)$ for any $r > 0$ (high-dimensional)
- maximal number of edges out of a node $= O(n^k)$ $(0 < k < 1)$ (sparseness)
- plus some other technical conditions

justification for relaxation with a computationally simple convex problem!

Choice of λ

Theorem doesn't say much about choosing λ...

first (not so good) idea: choose λ to optimize prediction

e.g. via some cross-validation scheme

but: for prediction oracle solution

$$\lambda^* = \arg\min_{\lambda} \mathbf{E}[(X_i - \sum_{j \neq i} \hat{\beta}_j^{(i)}(\lambda) X_j)^2]$$

$$\mathbb{P}[\text{estimated graph}(\lambda^*) = \text{true graph}] \to 0 \ (p_n \to \infty, n \to \infty)$$

asymptotically: the prediction optimal graph is too large

(Meinshausen & PB, 2004; related example by Meng et al., 2004)

A structural penalty parameter

goal: avoid connecting distinct connectivity components of the graph

Theorem (Meinshausen & PB, 2004)

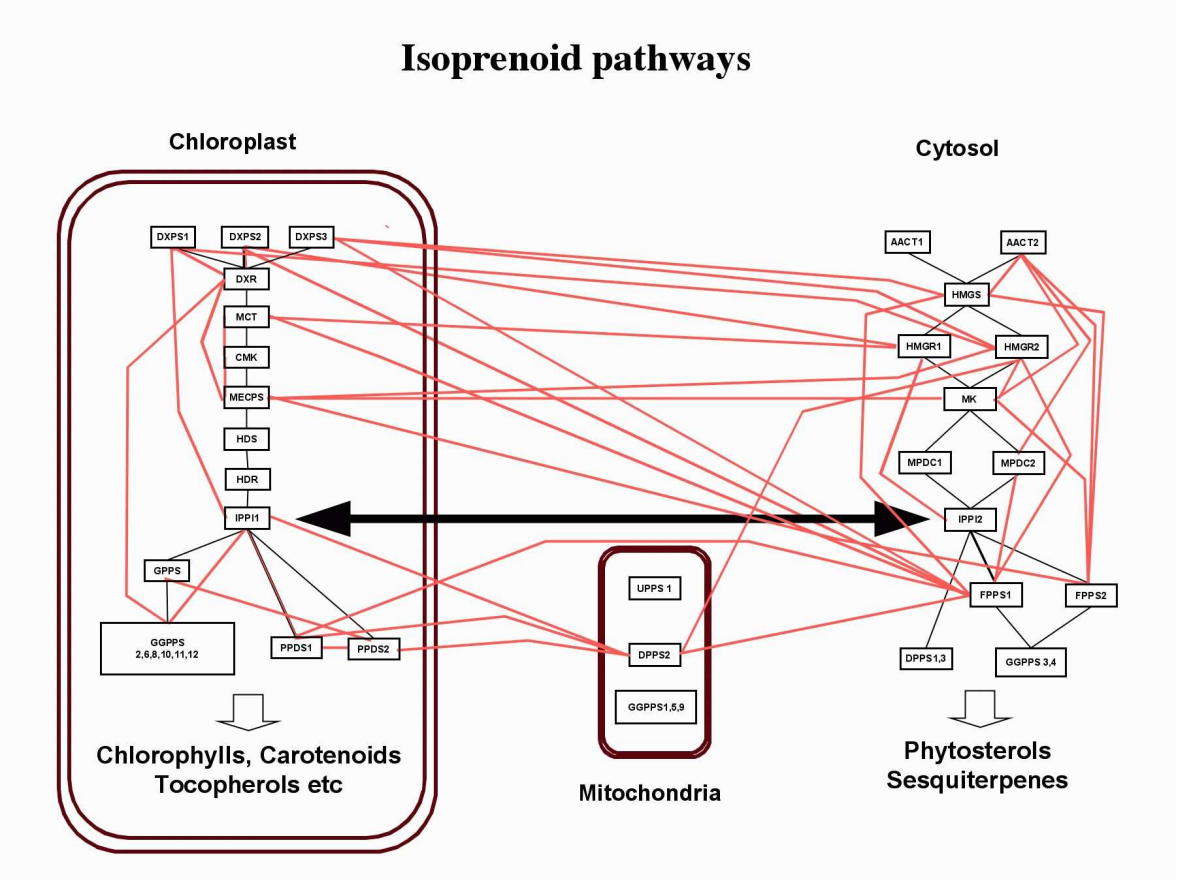
Finite sample control: when choosing the penalty

$$\lambda = \lambda_i = \frac{\hat{\sigma}_i}{\sqrt{n}} \tilde{\Phi}^{-1}(\frac{\alpha}{2p_n}),$$

$$\hat{\sigma}_i^2 = n^{-1} \sum_{r=1}^{n} X_{r,i}^2$$

the probability of falsely connecting distinct connectivity components is controlled at level α

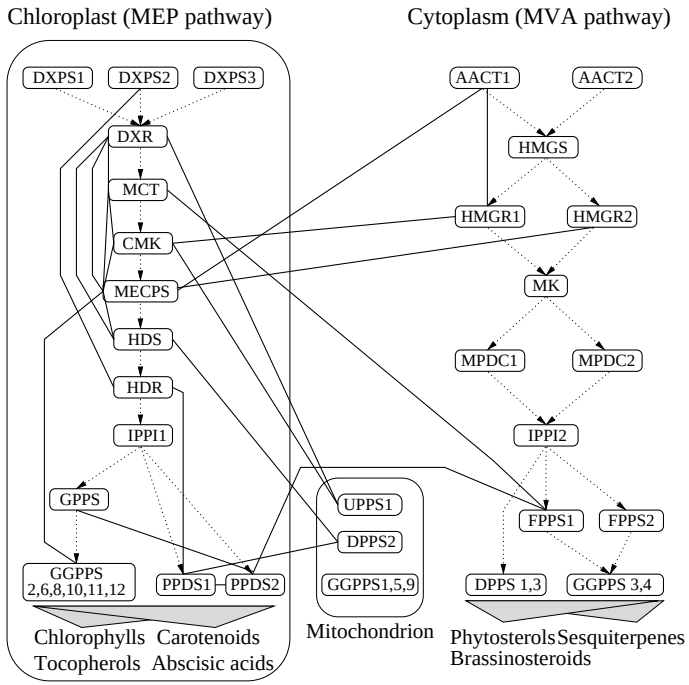
for the two biosynthesis pathways in Arabidopsis

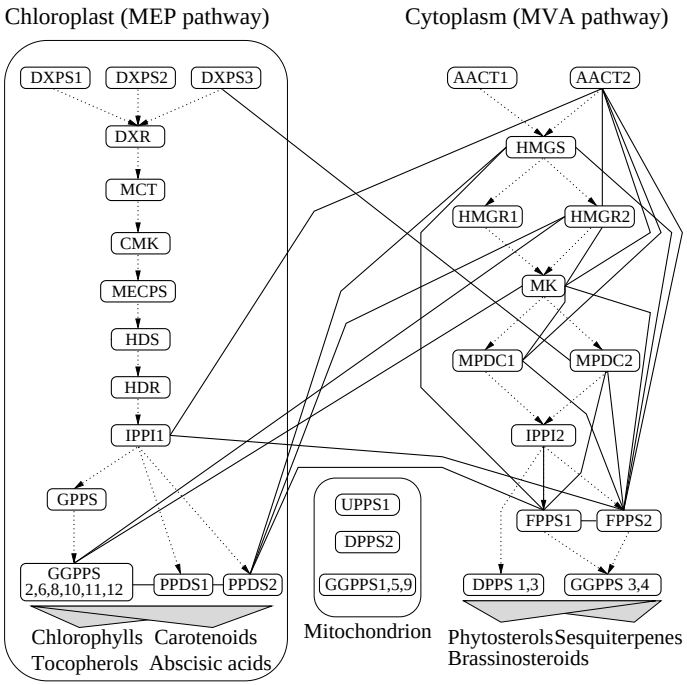


first observation: too many edges for biological interpretation

but it may serve as a first step in a further, biologically driven analysis...!

with further biological “constraints”





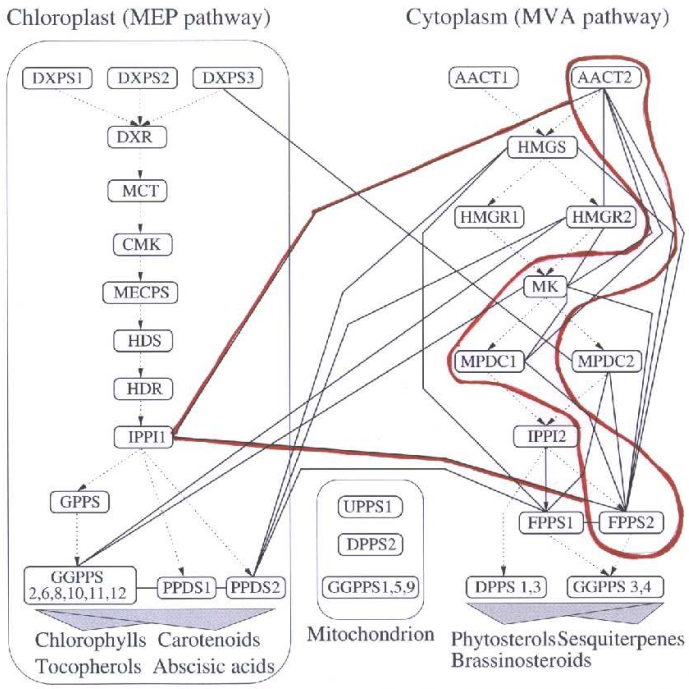
edges from MEP “module” to MVA

edges from MVA “module” to MEP

biologically most interesting novel connection: from IPPI1 to MVA “module”

identifiability problems: e.g. if variables are highly (partially) correlated

think in terms of “modules”



we are currently investigating whether potential common transcription factor “causes” the edge between IPPI1 and the MVA “module”

(Gruissem Lab, ETH Zürich)

6. Beyond Boosting and Lasso

consider regression $Y = X\beta + \varepsilon$

for orthonormal design: $\mathbf{X}^T\mathbf{X} = I$: Lasso/LARS and L_2Boosting yield the soft-threshold estimator:

$$\hat{\beta}^{(j)}_{soft} = \begin{cases} Z_j - \lambda, & \text{if } Z_j \geq \lambda, \\ 0, & \text{if } |Z_j| < \lambda, \\ Z_j + \lambda, & \text{if } Z_j \leq -\lambda. \end{cases} \qquad \text{where } Z_j = (\mathbf{X}^T\mathbf{Y})_j$$

6.1. Is soft-thresholding or Lasso a good thing?

- $\beta_1, \ldots \beta_p$ i.i.d. $\sim$ Double-Exponential,

  soft-thresholding and the Lasso yield the MAP (which often performs well)

- minimax results for soft-thresholding (Donoho & Johnstone, ...)

but: a different story in the very high-dimensional sparse case

assume:

- $p = p_n \sim C_1 \exp(C_2 n^{1-\xi})$ $(0 < \xi < 1)$
- effective number of variables is finite (finite ℓ^0-norm)

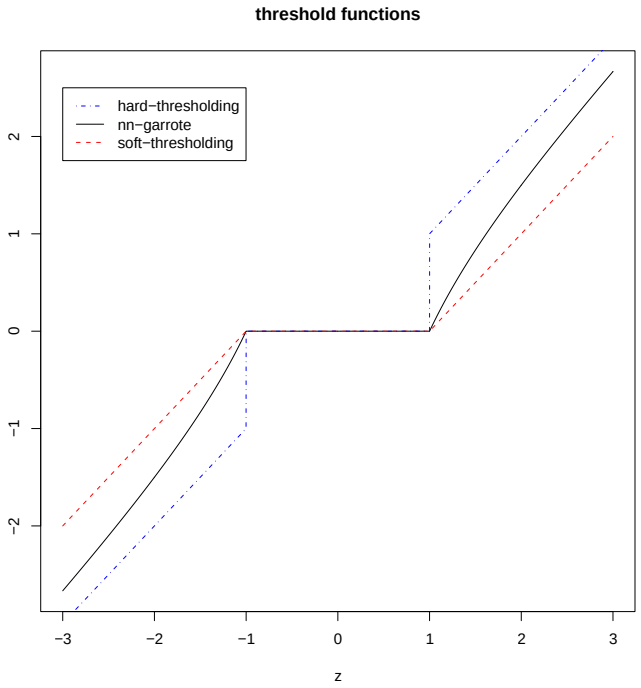
Theorem (Meinshausen, 2005)

$$\mathbb{P}[\inf_{\lambda} \underbrace{L(\lambda)}_{\text{risk of Lasso}} > cn^{-r}] \to 1 \ (n \to \infty) \text{ for } r > \xi$$

while optimal rate is n^{-1} (achieved e.g. by OLS with the true variables)

$\rightsquigarrow$ Lasso can have very poor convergence rate

reason: need large λ for variable selection $\rightsquigarrow$ strong bias of soft-thresholding



Better:

- SCAD (Fan and Li, 2001)
- Nonnegative Garrote (Breiman, 1995)
- Bridge estimation (Frank and Friedman, 1993)

they all work for general $\mathbf{X}$

for non-orthogonal $\mathbf{X}$:

- non-convex optimization for SCAD or Bridge estimation
- NN-Garrote only for $p \leq n$

6.2. The relaxed Lasso (Meinshausen, 2005)

for $\lambda \geq 0,\ 0 \leq \phi \leq 1$

$$\hat{\beta}_{\lambda,\phi} = \arg\min_{\beta} n^{-1} \sum_{i=1}^{n} (Y_i - \sum_{j \in \underbrace{\mathcal{M}_\lambda}_{\text{model from Lasso}(\lambda)}} \beta_j X_{i,j})^2 + \phi\lambda \|\beta\|_1$$

for $\phi = 0$: OLS on selected variables from Lasso(λ)

for $\phi = 1$: Lasso(λ)

amount of computation for finding all solutions over λ and ϕ:

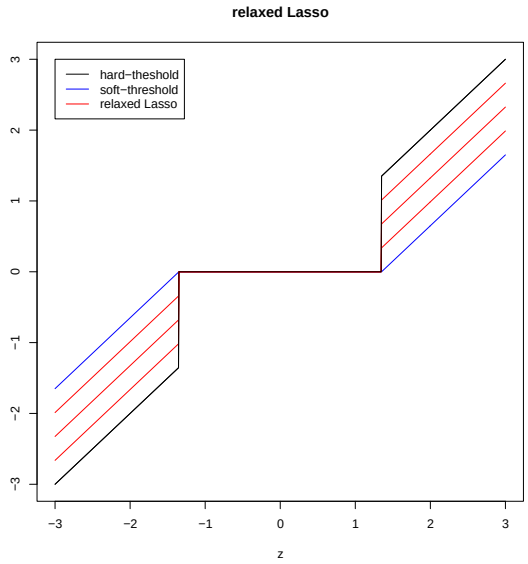
often, the same computational complexity as for Lasso/LARS (surprising):

$$O(np \min(n,p)) = O(n^2 p) \text{ if } p \gg n$$

worst case: $O(np \min(n,p)^2) = O(n^3 p)$ if $p \gg n$ still linear in p

for orthonormal case:

$$\mathbf{X}^T = \mathbf{X} = I$$



Theorem (Meinshausen, 2005)

with essentially the same assumptions as before

$$\inf_{\lambda,\phi} L(\lambda, \phi) = O_P(n^{-1})(n \to \infty)$$

also: use the relaxed Lasso for graphs/dependency networks

$\rightsquigarrow$ prediction optimal (or cross-validated) tuning parameters yield

consistent graph estimates

Results for high noise, binary lymph node classification

cross-validated misclassification rate:

relaxed Lasso (tuned by 5-fold CV): 24.4 %???

Lasso (tuned by 5-fold CV): 27.4 %

L_2Boosting (tuned by AIC): 30.3 %

selected genes (on whole data set):

relaxed Lasso: 2 genes (!) Lasso: 23 genes L_2Boosting: 42 genes

the 2 genes from relaxed Lasso are also selected by Lasso and L_2Boosting

note the identifiability problem again

6.3. L_2Boosting with FPE penalty

recap about L_2Boosting:

fit in iteration m the base procedure so that residual sum of squares is minimized

another idea:

fit in every iteration the base procedure so that the MSE (or out-sample squared error) is minimized

since the MSE is unknown: estimation by an FPE model selection criterion

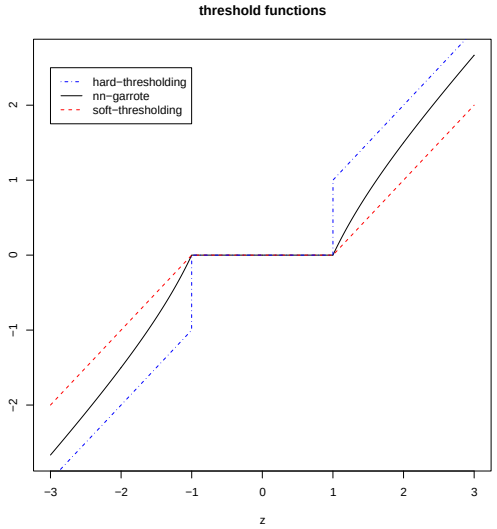
L_2Boosting with FPE penalty: fit in iteration m the base procedure $\hat{\theta}_m(\cdot)$ such that

$$\sum_{i=1}^{n}(U_i - \hat{\theta}_m(X_i))^2 + \gamma \cdot \underbrace{d.f.(\tilde{\mathcal{B}}_m)}_{\text{tr(FPE-boosting ``hat''-matrix)}} \quad \text{is minimal}$$

for linear regression with orthonormal design $\mathbf{X}^T\mathbf{X} = I$:

L_2Boosting with FPE penalty yields all solutions for Breiman's nonnegative garrote

(PB & Yu, 2005)



i.e. solutions which are

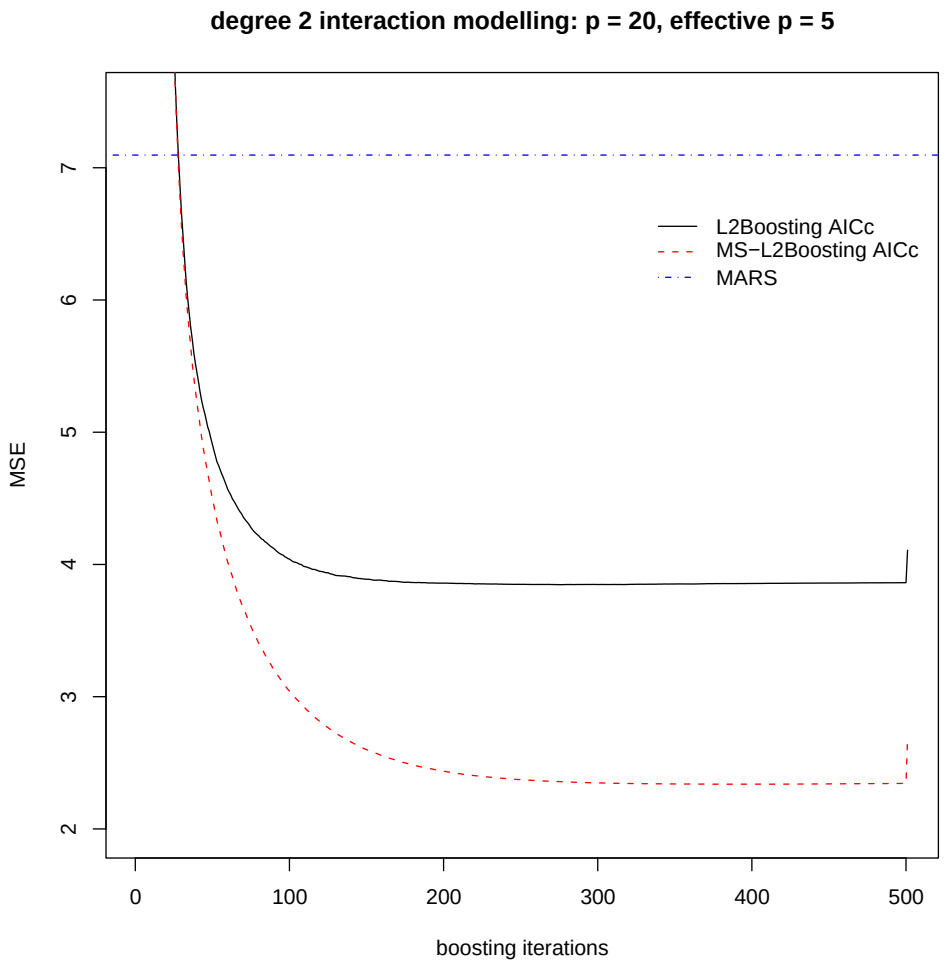
closer to hard-thresholding

less "exhaustive" than relaxed Lasso but

L_2Boosting with FPE penalty easily transfer to general base procedures

(nonparametrics)

Modelling with second-order interactions



Friedman #1 model:

$$Y = 10\sin(\pi X_1 X_2) + 20(X_3 - 0.5)^2 + 10X_4 + 5X_5 + \mathcal{N}(0,1)$$

$$X = (X_1, \ldots, X_{20}) \sim \text{Unif.}([0,1]^{20})$$

Sample size $n = 50$

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

7. Conclusions

Boosting: computationally greedy and very generic

Lasso / ℓ^1-penalty methods: convex optimization

- "surprisingly" similar and often very useful for $p \gg n$
- both explore a large space of solutions

relaxed Lasso (quasi-convex) and also Boosting with FPE penalty (quasi-greedy)

- provably/substantially better if signal is sparse w.r.t. ℓ^0-norm
- computationally very efficient for exploring an even much larger space of solutions
  (between ℓ^0- and ℓ^1-penalisation)

for biology/applications: improve by using

- biological constraints
  (e.g. conditioning on single potential transciption factors only)
- additional knowledge (e.g. GO categories)