Very high-dimensional data: greedy boosting and convex Lasso-optimization

Peter Bühlmann

ETH Zürich

1. High-dimensional data

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

$X_i \in \mathbb{R}^p$ predictor variable

Y_i univariate response variable, e.g. $Y_i \in \mathbb{R}$ or $Y_i \in \{0, 1\}$

high-dimensional: $p \gg n$

areas of application: astronomy, biology, imaging, marketing research, text classification,...

High-dimensional linear models

$$Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_i^{(j)} + \varepsilon_i,\ i = 1, \ldots, n$$

$$p \gg n$$

How should we fit this model?

approaches include:

Ridge regression (Tikhonov regularization); variable selection via AIC, BIC, gMDL (in a forward manner); Bayesian methods for regularization, ...

Boosting, Lasso, ...

our requirements:

- computationally feasible
- yields variable selection
- statistically accurate for prediction or selecting the correct variables

computational feasibility for high-dimensional problems

$$\rightsquigarrow$$

greedy methods

or

convex optimization

2. Greedy is good for $p \gg n$: L_2Boosting

(Friedman, 2001)

specify a base procedure ("weak learner"):

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

e.g.: simple linear regression, tree (CART), ...

L_2Boosting with base procedure $\hat{\theta}(\cdot)$: repeated fitting of residuals

$$m = 1: \ (X_i, Y_i)_{i=1}^n \rightsquigarrow \hat{\theta}_1(\cdot), \ f_1 = \underbrace{\nu}_{\text{e.g. } =0.1} \hat{\theta}_1 \rightsquigarrow \text{resid. } U_i = Y_i - f_1(X_i)$$

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

$$\ldots \qquad \ldots$$

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

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

Componentwise linear least squares base procedure:
linear OLS regression against the one predictor variable which reduces residual sum of squares most

$$\hat{\theta}(x) = \hat{\beta}_{\hat{\mathcal{S}}} x^{(\hat{\mathcal{S}})},$$

$$\hat{\beta}_j = \sum_{i=1}^{n} Y_i X_i^{(j)} / \sum_{i=1}^{n} (X_i^{(j)})^2, \;\; \hat{\mathcal{S}} = \arg\min_j \sum_{i=1}^{n} (Y_i - \hat{\beta}_j X_i^{(j)})^2$$

L_2Boosting with componentwise linear LS:

first round of estimation: selected predictor variable $X^{(\hat{\mathcal{S}}_1)}$ (e.g. $= X^{(3)}$)
corresponding $\hat{\beta}_{\hat{\mathcal{S}}_1}$
use shrunken fit $\hat{f}_1(x) = \nu \hat{\beta}_{\hat{\mathcal{S}}_1} x^{(\hat{\mathcal{S}}_1)}$ (e.g. $\nu = 0.1$)

second round of estimation: selected predictor variable $X^{(\hat{\mathcal{S}}_2)}$ (e.g.$= X^{(21)}$)
corresponding $\hat{\beta}_{\hat{\mathcal{S}}_2}$
use shrunken fit $\hat{f}_2(x) = \hat{f}_1(x) + \nu \hat{\beta}_{\hat{\mathcal{S}}_2} x^{(\hat{\mathcal{S}}_2)}$

etc.

for $\nu = 1$, this is known as

Matching Pursuit (Mallat and Zhang, 1993)

Weak greedy algorithm (deVore & Temlyakov, 1997)

a version of Boosting (Schapire, 1992; Freund & Schapire, 1996)

Gauss-Southwell algorithm

C.F. Gauss in 1803

"Princeps Mathematicorum"

R.V. Southwell in 1933

Professor in engineering, Oxford

Properties

variable selection

shrinkage towards zero for coefficients of selected variables

$\leadsto$ often much better performance than OLS on selected variables
("more stable" in Breiman's terminology)

"similar" to the Lasso (Efron, Hastie, Johnstone & Tibshirani, 2004)
but not the same

computational complexity:
$O(npm_{stop}) = O(p)$ if $p \gg n$, i.e. linear in dimension p

statistically consistent for very high-dimensional, sparse problems

Theorem (PB, 2004)

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

- $p_n = O(\exp(Cn^{1-\xi}))$ $(0 < \xi < 1)$ (high-dimensional)
  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

binary lymph node classification in breast cancer using gene expressions:

a high noise problem

$n = 49$ samples, $p = 7130$ gene expressions

	L_2Boosting	FPLR	Pelora	1-NN	DLDA	SVM
CV-misclassif.err.	24.8%	35.25%	27.8%	43.25%	36.12%	36.88%

L_2Boosting, Forward Penalized Logistic Regression (FPLR), Supervised Gene Grouping (Pelora)

no gene pre-selection $\rightsquigarrow$ all these methods do multivariate gene selection

Nearest Neighbor (1-NN), Diagonal Linear Discriminant Analysis (DLDA), SVM with radial basis kernel

with gene pre-selection: the best 200 genes from 2-sample Wilcoxon score

$\rightsquigarrow$ no additional gene selection anymore

L_2Boosting selected 42 out of $p = 7129$ genes

for this data-set: not good prediction with all the different methods

(although we will improve to 16.3%)

but L_2Boosting may be a good(?) multivariate gene selection method

Variable selection

do variable selection such that predictive performance is good
(not necessarily optimal)

"classical": subset selection using BIC, AIC, gMDL, etc.
computationally infeasible for high-dimensional problems

remedies:

- forward selection
  but often not competitive in terms of predictive performance
- L_2Boosting: seems quite interesting, but weak theoretical basis
- replace the computationally hard subset selection problem (2^p sub-models) by convex relaxation

3. 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 relaxation (Lasso) and greedy algorithm (Boosting):

variable selection; shrinkage;

consistency for prediction in high-dimensions (Greenshtein & Ritov (2004))

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

$O(np \min(n,p))$ essential operations to compute all Lasso solutions

$= O(p)$ if $p \gg n$

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

3.1. Variable selection and graphical modeling with the Lasso

goal: use the Lasso for determining presence/absence of associations between random variables (this includes regression)

Gaussian conditional independence graph

assume that $X = X^{(1)}, \ldots, X^{(p)} \sim \mathcal{N}_p(\mu, \Sigma)$

graph:

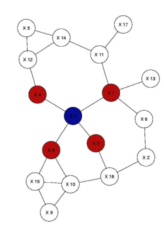
set of nodes $\Gamma = \{1, 2, \ldots, p\}$, corresponding to the p random variables

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!

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 $\neq 0$)

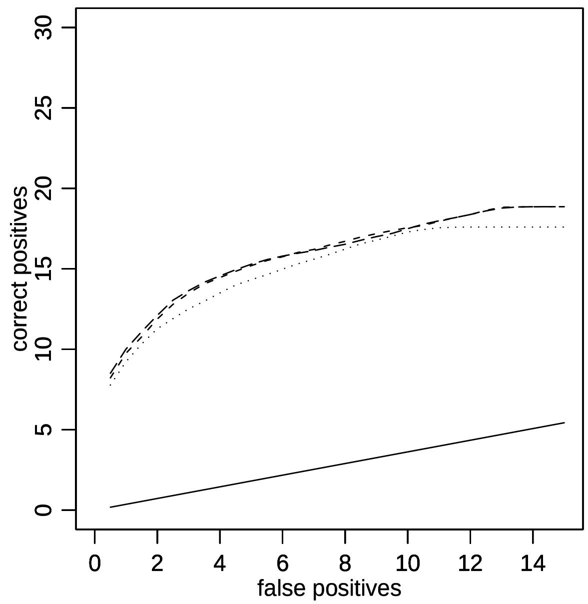
note: depends on the tuning parameter λ in Lasso

this involves only one convex optimization problem!

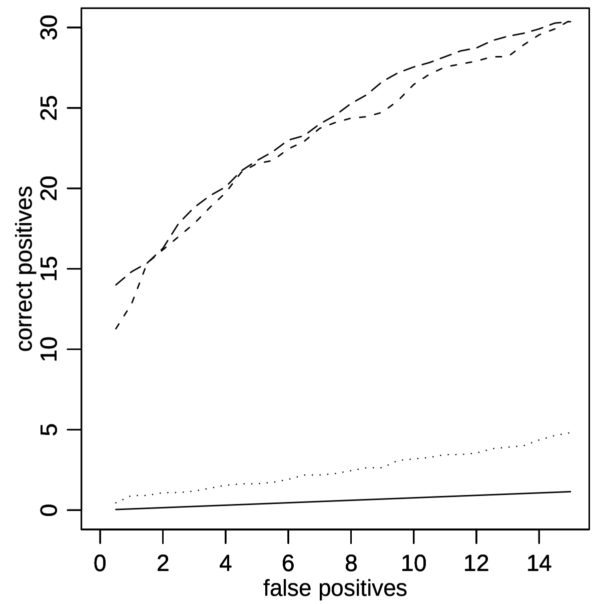
instead of checking exhaustively $2^{p-1}p$ least squares problems (e.g. using BIC)

Comparison of Lasso and classical stepwise selection

$p = 10$



$p = 30$



dotted $\cdot\cdot\cdot\cdot$	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

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) we have a simple proposal for choosing the penalty parameter

which avoids connecting distinct connectivity components

4. Beyond Lasso (and Boosting)

linear model $Y = X\beta + \varepsilon$

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

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

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)

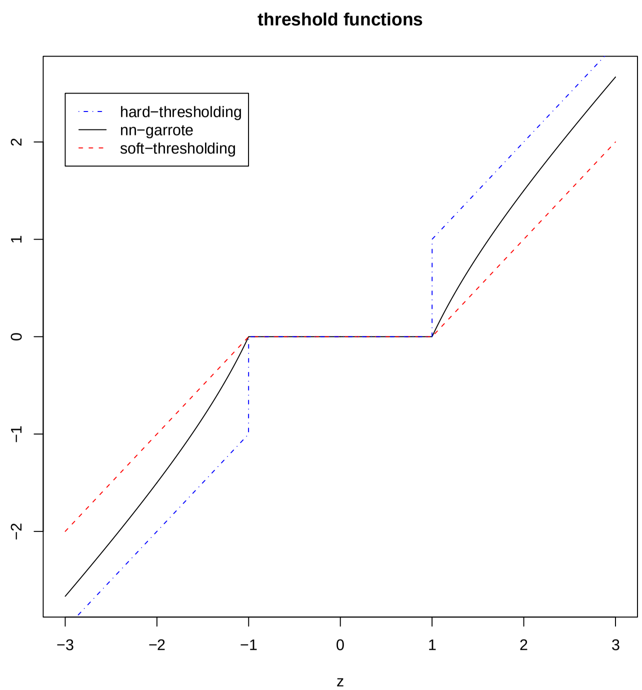
  non-effective variables are independent

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)

$\leadsto$ 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 $p \leq n$

4.1. 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:

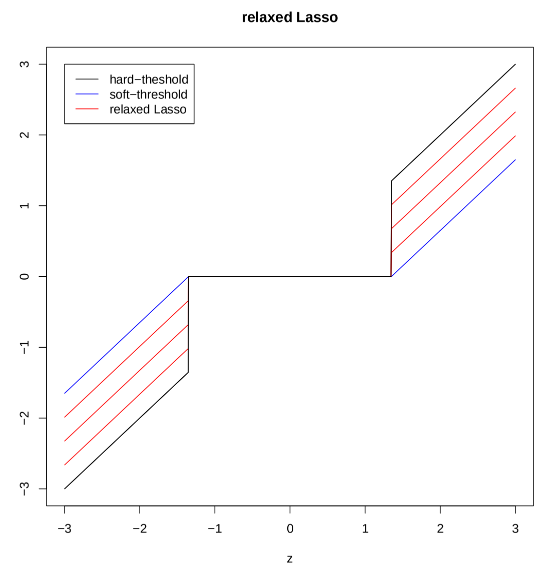
often, the same computational complexity as for Lasso/LARS (surprising):

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

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

this is "quasi-convex" optimization: two levels of a convex problem for orthonormal case:

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



Theorem (Meinshausen, 2005)

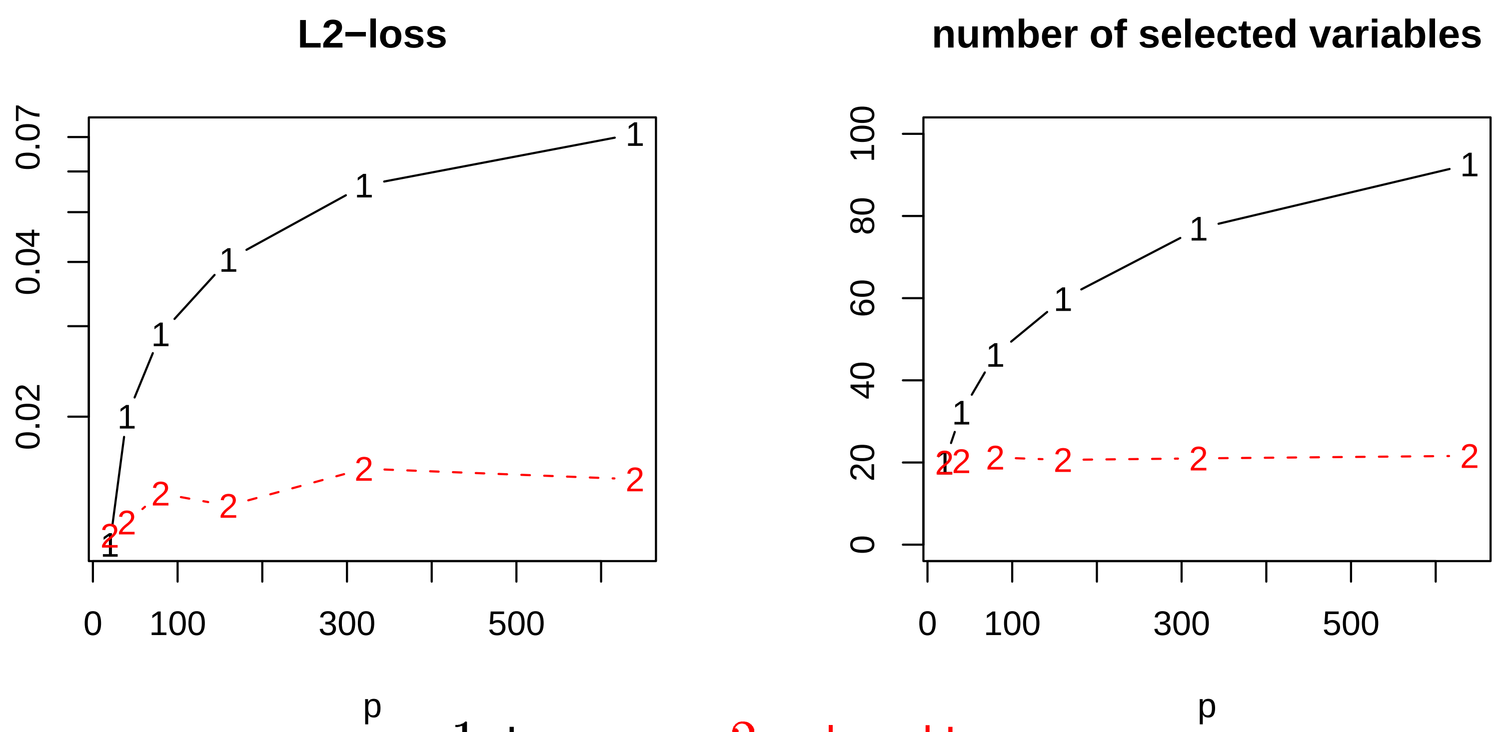
with essentially the same assumptions as before

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

also: use the relaxed Lasso for variable selection and graphs/dependency networks

$\leadsto$ prediction optimal (or cross-validated) tuning parameters yield (for some cases) consistent variable selection and graph estimates

$n = 300, p = 20, \ldots 650, p_{eff} = 20$

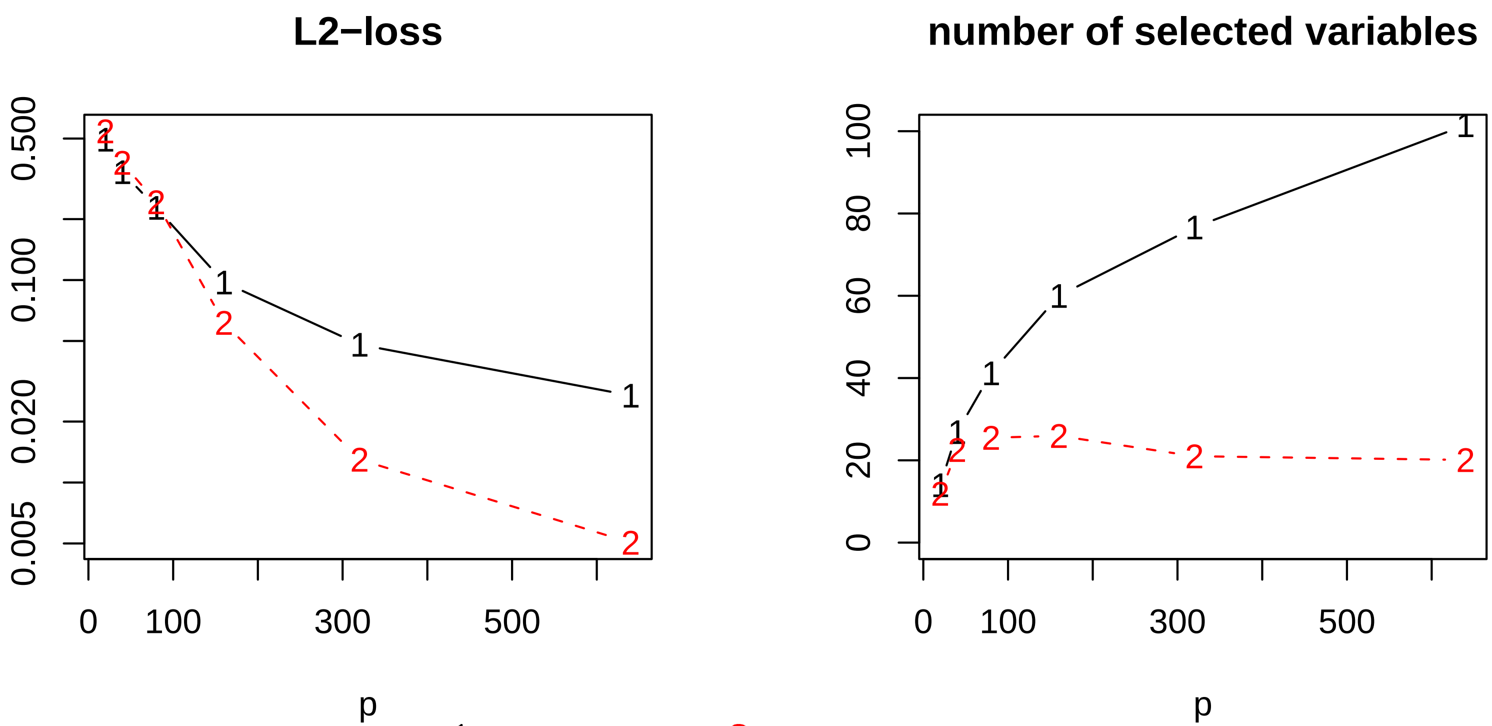


1: Lasso 2: relaxed Lasso

additional pure noise variables are much less damaging with the relaxed Lasso than for Lasso and Boosting

and they are very disturbing for Ridge-type regularization (e.g. SVM)

$n = p = 20, \ldots 650, p_{eff} = 20$



1: Lasso 2: relaxed Lasso

relaxed Lasso never substantially worse than the Lasso: the price for the flexibility of the relaxed Lasso is the larger search space $0 \leq \phi \leq 1$ (Lasso: $\phi = 1$)

Results for high noise, binary lymph node classification

cross-validated misclassification rate:

relaxed Lasso (tuned by 5-fold CV): 16.3%

Lasso (tuned by 5-fold CV): 21.0%

L_2Boosting (tuned by 5-fold CV): 24.8%

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

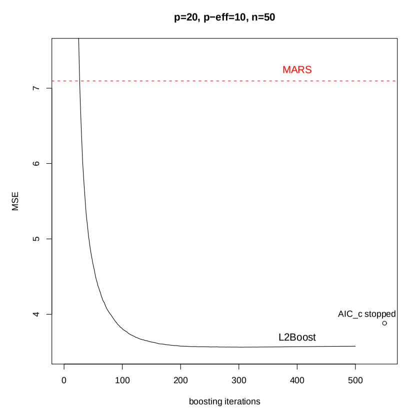
note the identifiability problem among highly correlated predictor variables

Conclusions

high-dimensional: blue greedy or convex?

the methods are similar and very useful

- Boosting is more generic: can be easily extended to e.g. the nonparametric setting



nonparametric interaction modeling

L_2Boosting with pairwise splines

sample size $n = 50$

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

- Lasso is more explicit (and hence better understood)

beyond Lasso (more sparse) is computationally feasible

via relaxed Lasso doing "quasi-convex" optimization