Lasso and Group Lasso

Peter Bühlmann
ETH Zürich

September 2009

High-dimensional data

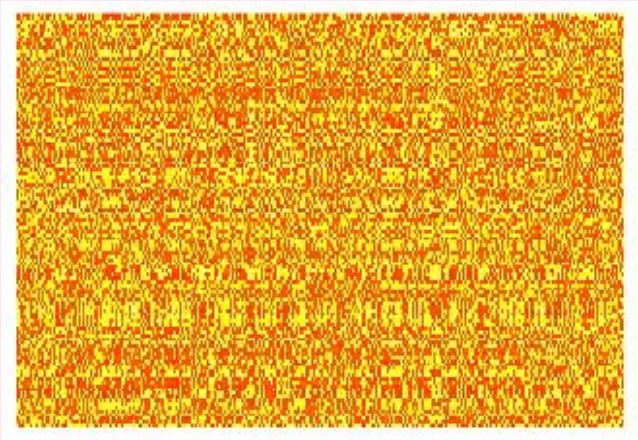
Riboflavin production with Bacillus Subtilis

(in collaboration with DSM (Switzerland))

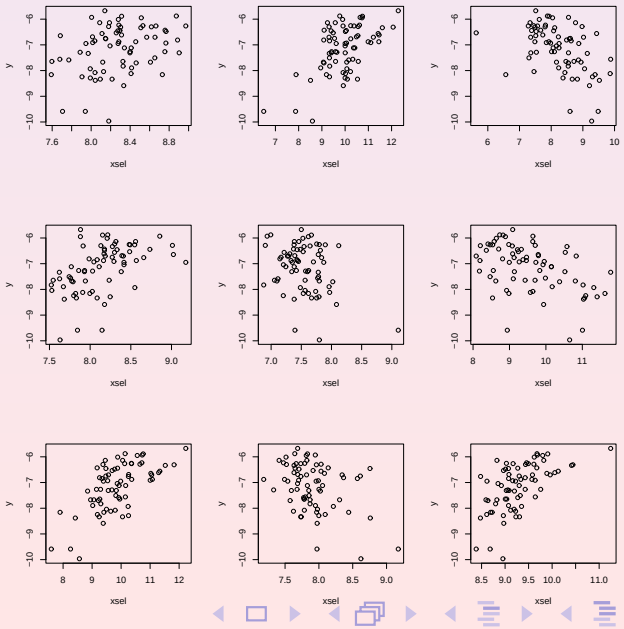
goal: improve riboflavin production rate of Bacillus Subtilis
using clever genetic engineering

response variables $Y \in \mathbb{R}$: riboflavin (log-) production rate
covariates $X \in \mathbb{R}^p$: expressions from $p = 4088$ genes
sample size $n = 115$, $p \gg n$

gene expression data



Y versus 9 “reasonable” genes



general framework:

$$Z_1, \ldots, Z_n \text{ i.i.d. or stationary}$$
$$\dim(Z_i) \gg n$$

for example:
$Z_i = (X_i, Y_i),\ X_i \in \mathbb{R}^p, Y_i \in \mathbb{R}$: regression with $p \gg n$
$Z_i = (X_i, Y_i),\ X_i \in \mathbb{R}^p, Y_i \in \{0, 1\}$: classification with $p \gg n$

numerous applications:
biology, imaging, economy, environmental sciences, ...

High-dimensional linear models

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

$p \gg n$

in short: $\mathbf{Y} = \mathbf{X}\beta + \epsilon$

goals:

- prediction, e.g. w.r.t. squared prediction error
- variable selection
  i.e. estimating the effective variables
  (having corresponding coefficient $\neq 0$)

Prediction

binary lymph node classification using gene expressions:
a high noise problem
$n = 49$ samples, $p = 7130$ gene expressions

despite that it is classification: $\mathbb{P}[Y = 1|X = x] = \mathbb{E}[Y|X = x]$
$\rightsquigarrow$ $\hat{p}(x)$ via linear model; can then do classification

cross-validated misclassification error (2/3 training; 1/3 test)

Lasso	L_2Boosting	FPLR	Pelora	1-NN	DLDA	SVM
21.1%	17.7%	35.25%	27.8%	43.25%	36.12%	36.88%

with variable selection

best 200 genes (Wilcoxon test)
no additional variable selection

from a practical perspective:
if you trust in cross-validation: can validate how good we are
i.e. prediction may be a black box, but we can evaluate it!

Prediction

binary lymph node classification using gene expressions:
a high noise problem
$n = 49$ samples, $p = 7130$ gene expressions

despite that it is classification: $\mathbb{P}[Y = 1|X = x] = \mathbb{E}[Y|X = x]$
$\rightsquigarrow$ $\hat{p}(x)$ via linear model; can then do classification

cross-validated misclassification error (2/3 training; 1/3 test)

Lasso	L_2Boosting	FPLR	Pelora	1-NN	DLDA	SVM
21.1%	17.7%	35.25%	27.8%	43.25%	36.12%	36.88%

with variable selection

best 200 genes (Wilcoxon test)
no additional variable selection

from a practical perspective:
if you trust in cross-validation: can validate how good we are
i.e. prediction may be a black box, but we can evaluate it!

Variable selection: Motif regression

for finding HIF1α transcription factor binding sites in DNA seq.
Müller, Meier, PB & Ricci

$Y_i \in \mathbb{R}$: univariate response measuring binding intensity of HIF1α on coarse DNA segment i (from CHIP-chip experiments)
$X_i = (X_i^{(1)}, \ldots, X_i^{(p)}) \in \mathbb{R}^p$:
$X_i^{(j)}$ = abundance score of candidate motif j in DNA segment i (using sequence data and computational biology algorithms, e.g. MDSCAN)

question: relation between the binding intensity Y and the abundance of short candidate motifs?

$\rightsquigarrow$ linear model is often reasonable
"motif regression" (Conlon, X.S. Liu, Lieb & J.S. Liu, 2003)

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon, \; n = 287, \; p = 195$$

goal: variable selection
$\rightsquigarrow$ find the relevant motifs among the $p = 195$ candidates

from a practical perspective:
not easy to evaluate how good we are!
$\rightsquigarrow$ it is highly desirable to
assess uncertainty, assign relevance or significance

question: relation between the binding intensity Y and the abundance of short candidate motifs?

$\rightsquigarrow$ linear model is often reasonable
"motif regression" (Conlon, X.S. Liu, Lieb & J.S. Liu, 2003)

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon, \; n = 287, \; p = 195$$

goal: variable selection
$\rightsquigarrow$ find the relevant motifs among the $p = 195$ candidates

from a practical perspective:
not easy to evaluate how good we are!
$\rightsquigarrow$ it is highly desirable to
assess uncertainty, assign relevance or significance

High-dimensional linear model

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon, \quad p \text{ large; or } p \gg n$$

we need to regularize...

and there are many proposals

- Bayesian methods for regularization
- greedy algorithms: aka forward selection or boosting
- preliminary dimension reduction
- ...

e.g. 2'650'000 entries on Google Scholar for
"high dimensional linear model" ...

High-dimensional linear model

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon, \quad p \text{ large; or } p \gg n$$

we need to regularize...
and there are many proposals

- Bayesian methods for regularization
- greedy algorithms: aka forward selection or boosting
- preliminary dimension reduction
- ...

e.g. 2'650'000 entries on Google Scholar for
"high dimensional linear model" ...

High-dimensional linear model

$$\mathbf{Y} = \mathbf{X}\beta + \epsilon, \quad p \text{ large; or } p \gg n$$

we need to regularize...
and there are many proposals

- Bayesian methods for regularization
- greedy algorithms: aka forward selection or boosting
- preliminary dimension reduction
- ...

e.g. 2'650'000 entries on Google Scholar for
"high dimensional linear model" ...

Penalty-based methods

if true $\beta_{\text{true}} = \beta^0$ is sparse w.r.t.

- $\|\beta^0\|_0 =$ number of non-zero coefficients
  $\rightsquigarrow$ penalize with the $\|\cdot\|_0$-norm:
  $\text{argmin}_\beta(n^{-1}\|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda\|\beta\|_0)$, e.g. AIC, BIC
  $\rightsquigarrow$ computationally infeasible if p is large (2^p sub-models)

- $\|\beta^0\|_1 = \sum_{j=1}^p |\beta_j^0|$
  $\rightsquigarrow$ penalize with the $\|\cdot\|_1$-norm, i.e. Lasso:
  $\text{argmin}_\beta(n^{-1}\|\mathbf{Y} - \mathbf{X}\beta\|^2 + \lambda\|\beta\|_1)$
  $\rightsquigarrow$ convex optimization:
  computationally feasible and very fast for large p

The Lasso (Tibshirani, 1996)

Lasso for linear models

$$\hat{\beta}(\lambda) = \text{argmin}_\beta(n^{-1}\|\mathbf{Y} - \mathbf{X}\beta\|^2 + \underbrace{\lambda}_{\geq 0} \underbrace{\|\beta\|_1}_{\sum_{j=1}^p |\beta_j|})$$

$\rightsquigarrow$ convex optimization problem

- Lasso does variable selection
  some of the $\hat{\beta}_j(\lambda) = 0$
  (because of "ℓ_1-geometry")
- $\hat{\beta}(\lambda)$ is a shrunken LS-estimate

more about "ℓ_1-geometry"

equivalence to primal problem

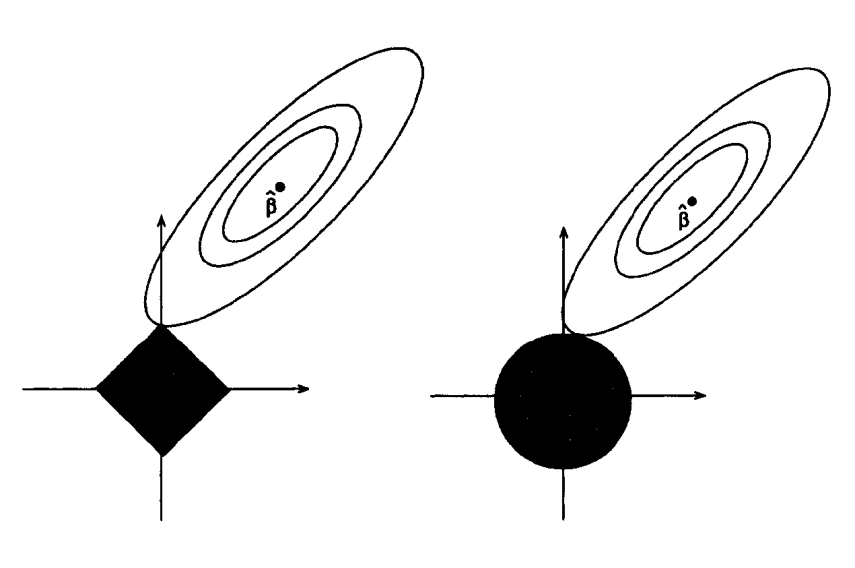
$$\hat{\beta}_{\text{primal}}(R) = \text{argmin}_{\beta;\|\beta\|_1 \le R} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2/n,$$

with a one-to-one correspondence between λ and R which depends on the data $(X_1, Y_1), \ldots, (X_n, Y_n)$
[such an equivalence holds since

- $\|\mathbf{Y} - \mathbf{X}\beta\|_2^2/n$ is convex in β
- convex constraint $\|\beta\|_1 \le R$

see e.g. Bertsekas (1995)]

p=2



left: ℓ_1-"world"
residual sum of squares reaches a minimal value (for certain constellations of the data) if its contour lines hit the ℓ_1-ball in its corner
$\rightsquigarrow \ \hat{\beta}_1 = 0$

ℓ_2-"world" is different

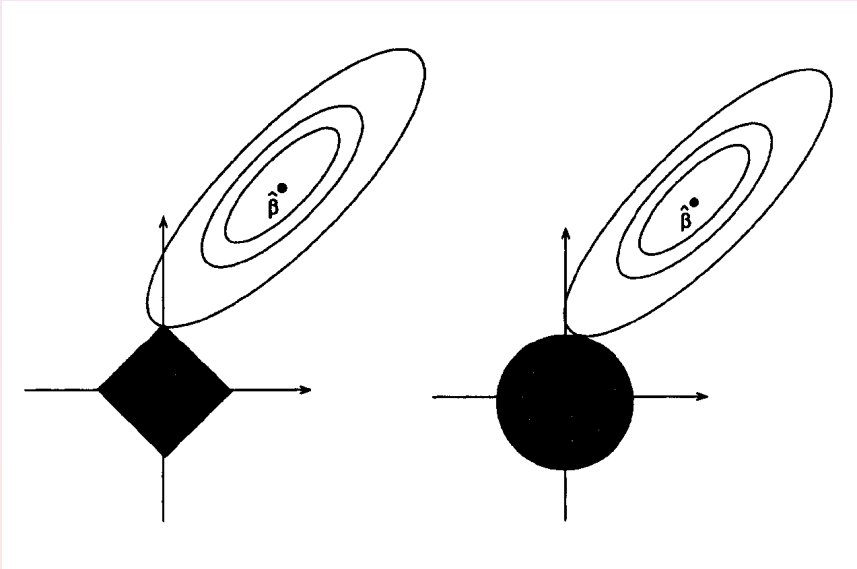
Ridge regression,

$$\hat{\beta}_{\text{Ridge}}(\lambda) = \text{argmin}_\beta \left(\|\mathbf{Y} - \mathbf{X}\beta\|_2^2/n + \lambda\|\beta\|_2^2 \right),$$

equivalent primal equivalent solution

$$\hat{\beta}_{\text{Ridge;primal}}(R) = \text{argmin}_{\beta;\|\beta\|_2 \le R} \|\mathbf{Y} - \mathbf{X}\beta\|_2^2/n,$$

with a one-to-one correspondence between λ and R



ℓ_q-penalized estimator:

$$\hat{\beta}_{\ell_q}(\lambda) = \text{argmin}_\beta \left(\|\mathbf{Y} - \mathbf{X}\beta\|_2^2/n + \lambda \|\beta\|_q^q \right),$$

$$\begin{aligned} \text{convex optimization} &\quad \Leftrightarrow \quad 1 \leq q \leq \infty \\ \text{variable selection; "sparse"} &\quad \Leftrightarrow \quad 0 \leq q \leq 1 \end{aligned}$$

$\rightsquigarrow$ Lasso ($q = 1$) is the "only" computationally feasible method doing variable selection

Orthonormal design

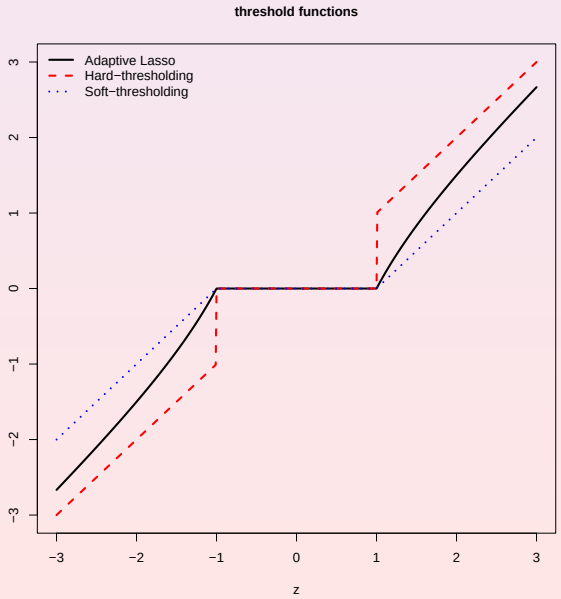
$$\mathbf{Y} = \mathbf{X}\beta + \epsilon, \quad \mathbf{X}^T\mathbf{X} = I$$

Lasso = soft-thresholding estimator

$$\hat{\beta}_j(\lambda) = \text{sign}(Z_j)(|Z_j| - \lambda/2)_+, \quad \underbrace{Z_j}_{=\text{OLS}} = (\mathbf{X}^T\mathbf{Y})_j,$$

$$\hat{\beta}_j(\lambda) = g_{\text{soft}}(Z_j),$$

[follows from more general characterization, see later]



Lasso for prediction: $x_{new}\hat{\beta}(\lambda)$

Lasso for variable selection:

$$\hat{S}(\lambda) = \{j;\ \hat{\beta}_j(\lambda) \neq 0\}$$

for $$S_0 = \{j; \beta_j^0 \neq 0\}$$

no significance testing involved

it's convex optimization only!

(and that can be a problem... see later)

Lasso for prediction: $x_{new}\hat{\beta}(\lambda)$

Lasso for variable selection:

$$\hat{S}(\lambda) = \{j;\ \hat{\beta}_j(\lambda) \neq 0\}$$

for $$S_0 = \{j; \beta_j^0 \neq 0\}$$

no significance testing involved

it's convex optimization only!

(and that can be a problem... see later)

Prediction (with the Lasso)

choose λ via cross-validation (e.g. 10-fold)

from a practical perspective:
if you trust in cross-validation: can validate how good we are
(need double cross-validation if $\lambda = \hat{\lambda}_{CV}$)
i.e. prediction may be a black box, but we can evaluate it!

binary lymph node classification using gene expressions:
a high noise problem
$n = 49$ samples, $p = 7130$ gene expressions

cross-validated misclassification error (2/3 training; 1/3 test)

Lasso	L_2Boosting	FPLR	Pelora	1-NN	DLDA	SVM
21.1%	17.7%	35.25%	27.8%	43.25%	36.12%	36.88%

with variable selection

best 200 genes (Wilcoxon test)
no additional variable selection

consistency and optimality (oracle inequality) for prediction
(see later)

Variable selection (with the Lasso): Motif regression

for finding HIF1α transcription factor binding sites in DNA seq.

$Y_i \in \mathbb{R}$: univariate response measuring binding intensity on coarse DNA segment i (from CHIP-chip experiments)
$X_i^{(j)}$ = abundance score of candidate motif j in DNA segment i

variable selection in linear model $Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_i^{(j)} + \varepsilon_i$,

$i = 1, \ldots, n = 287,\ p = 195$

$\rightsquigarrow$ Lasso selects 26 covariates and $R^2 \approx 50\%$
i.e. 26 interesting candidate motifs

and hence report these findings to the biologists...
really? do we trust our selection algorithm?
how stable are the findings?

Variable selection (with the Lasso): Motif regression

for finding HIF1α transcription factor binding sites in DNA seq.

$Y_i \in \mathbb{R}$: univariate response measuring binding intensity on coarse DNA segment i (from CHIP-chip experiments)
$X_i^{(j)}$ = abundance score of candidate motif j in DNA segment i

variable selection in linear model $Y_i = \beta_0 + \sum_{j=1}^{p} \beta_j X_i^{(j)} + \varepsilon_i$,

$i = 1, \ldots, n = 287,\ p = 195$

$\rightsquigarrow$ Lasso selects 26 covariates and $R^2 \approx 50\%$
i.e. 26 interesting candidate motifs

and hence report these findings to the biologists...
really? do we trust our selection algorithm?
how stable are the findings?

Some theory for variable selection with Lasso

an older formulation:

Theorem (Meinshausen & PB, 2004 (publ: 2006))

- sufficient and necessary neighborhood stability condition on the design X; see also Zhao & Yu (2006)
- $p = p_n$ is growing with n
  - $p_n = O(n^\alpha)$ for some $0 < \alpha < \infty$ (high-dimensionality)
  - $|S_{true,n}| = |S_{0,n}| = O(n^\kappa)$ for some $0 < \kappa < 1$ (sparsity)
  - the non-zero β_j's are outside the $n^{-1/2}$-range
  - Y, $X^{(j)}$'s Gaussian (not crucial)

Then: if $\lambda = \lambda_n \sim const.n^{-1/2-\delta/2}$ $(0 < \delta < 1/2)$,

$$\begin{aligned} \mathbb{P}[\hat{S}(\lambda) = S_0] &= 1 - O(\exp(-Cn^{1-\delta})) \ \ (n \to \infty) \\ &\approx 1 \text{ even for relatively small } n \end{aligned}$$

Problem 1:

Neighborhood stability condition is restrictive

sufficient and necessary for consistent model selection with Lasso

it fails to hold if design matrix exhibits
"strong linear dependence" (in terms of sub-matrices)

if it fails and because of necessity of the condition
$\Rightarrow$ Lasso is not consistent for selecting the relevant variables

neighborhood stability condition $\Leftrightarrow$ irrepresentable condition

(Zhao & Yu, 2006)

$$n^{-1} X^T X \to \Sigma$$

active set $S_0 = \{j;\ \beta_j \neq 0\} = \{1, \ldots, s_0\}$ consists of the first s_0 variables; partition

$$\Sigma = \begin{pmatrix} \Sigma_{S_0,S_0} & \Sigma_{S_0,S_0^c} \\ \Sigma_{S_0^c,S_0} & \Sigma_{S_0^c,S_0^c} \end{pmatrix}$$

irrep. condition : $|\Sigma_{S_0^c,S_0} \Sigma_{S_0,S_0}^{-1} \operatorname{sign}(\beta_1, \ldots, \beta_{s_0})| < 1$

not easy to get insights when it holds...

Problem 2: Choice of λ

for prediction oracle solution

$$\lambda_{\text{opt}} = \text{argmin}_\lambda \mathbb{E}[(Y - \sum_{j=1}^{p} \hat{\beta}_j(\lambda) X^{(j)})^2]$$

$\mathbb{P}[\hat{S}(\lambda_{\text{opt}}) = S_0] < 1 \ (n \to \infty)$ (or $= 0$ if $p_n \to \infty \ (n \to \infty)$)

asymptotically: prediction optimality yields too large models
(Meinshausen & PB, 2004; related example by Leng et al., 2006)

“Problem 3”: small non-zero regression coefficients (i.e. high noise level)

we cannot reliably detect variables with small non-zero coefficients

but (under some conditions)
we can still detect the variables with large regression effects

If neighborhood stability condition fails to hold (problem 1)

under compatibility conditions on the design $\mathbf{X}$
"typically" much weaker ass. than neighborhood stability

for suitable $\lambda = \lambda_n$ and with large probability

$$\|\hat{\beta} - \beta\|_1 = \sum_{j=1}^{p} |\hat{\beta}_j - \beta_j| \leq \underbrace{C}_{\text{depending on } \mathbf{X}, \sigma^2} \sqrt{\log(p) s_0 / n}$$

hence: $\max_j |\hat{\beta}_j - \beta_j| \leq \|\hat{\beta} - \beta\|_1 \leq C\sqrt{\log(p) s_0 / n}$

and if $\min_j \{|\beta_j|;\ \beta_j \neq 0\} > C\sqrt{\log(p) s_0 / n}$

then $\hat{\beta}_j \neq 0$ for all $j \in S_0$, i.e. $\hat{S} \supseteq S_0$

with large probability

$$\hat{S} \supseteq S_0$$

$$|\hat{S}| \leq O(min(n, p)) \underbrace{=}_{\text{if } p \gg n} O(n)$$

i.e. a huge dimensionality reduction in the original covariates!

furthermore: "typically", for prediction-optimal λ_{opt}

$$\hat{S}(\lambda_{\text{opt}}) \supseteq S_0$$

$\rightsquigarrow$ Lasso as an
excellent screening procedure

i.e. true active set is contained in estimated active set from Lasso

with large probability

$$\hat{S} \supseteq S_0$$

$$|\hat{S}| \leq O(min(n,p)) \underbrace{=}_{\text{if } p \gg n} O(n)$$

i.e. a huge dimensionality reduction in the original covariates!

furthermore: "typically", for prediction-optimal λ_{opt}

$$\hat{S}(\lambda_{\text{opt}}) \supseteq S_0$$

$\rightsquigarrow$ Lasso as an
excellent screening procedure

i.e. true active set is contained in estimated active set from Lasso

with large probability

$$\hat{S} \supseteq S_0$$

$$|\hat{S}| \leq O(min(n,p)) \underbrace{=}_{\text{if } p \gg n} O(n)$$

i.e. a huge dimensionality reduction in the original covariates!

furthermore: "typically", for prediction-optimal λ_{opt}

$$\hat{S}(\lambda_{\text{opt}}) \supseteq S_0$$

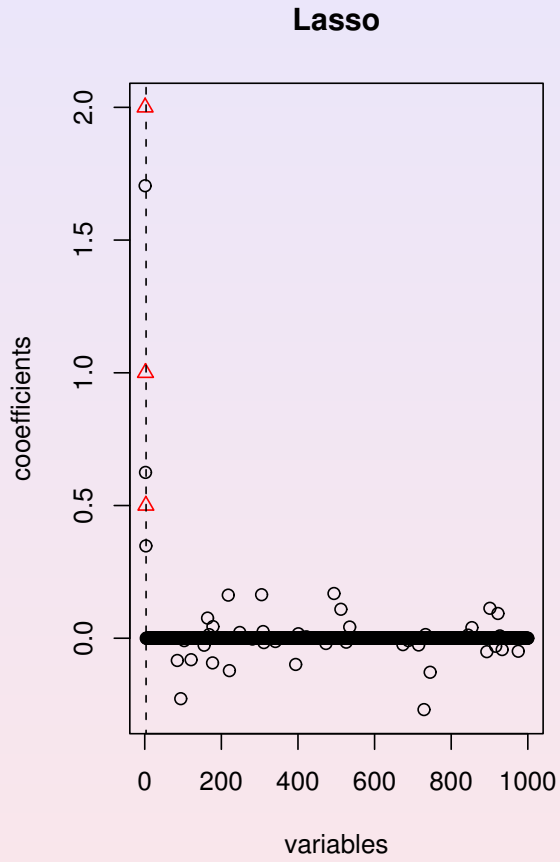
$\rightsquigarrow$ Lasso as an
excellent screening procedure

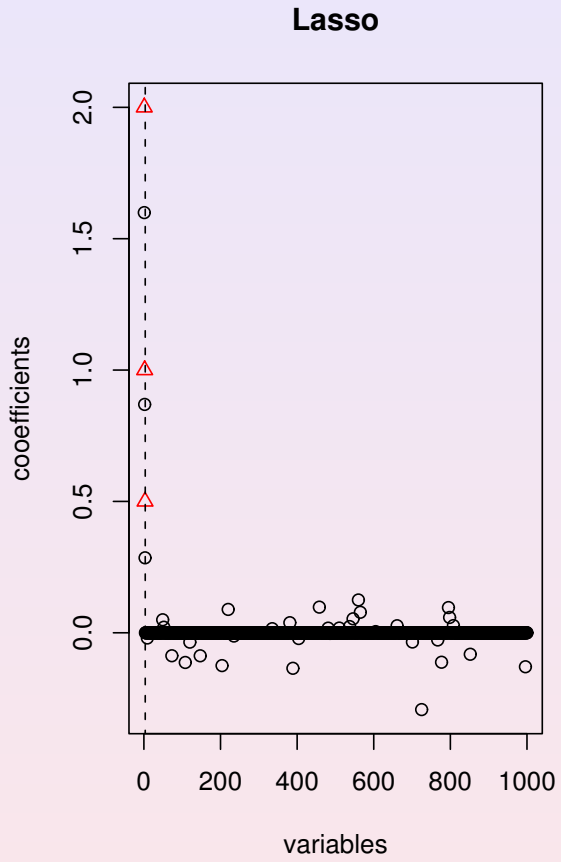
i.e. true active set is contained in estimated active set from Lasso

Lasso screening is easy to use, prediction optimal tuning computationally efficient, and statistically accurate

$O(np\min(n,p))$

$s_0 = 3,\ p = 1'000,\ n = 50$; 2 independent realizations



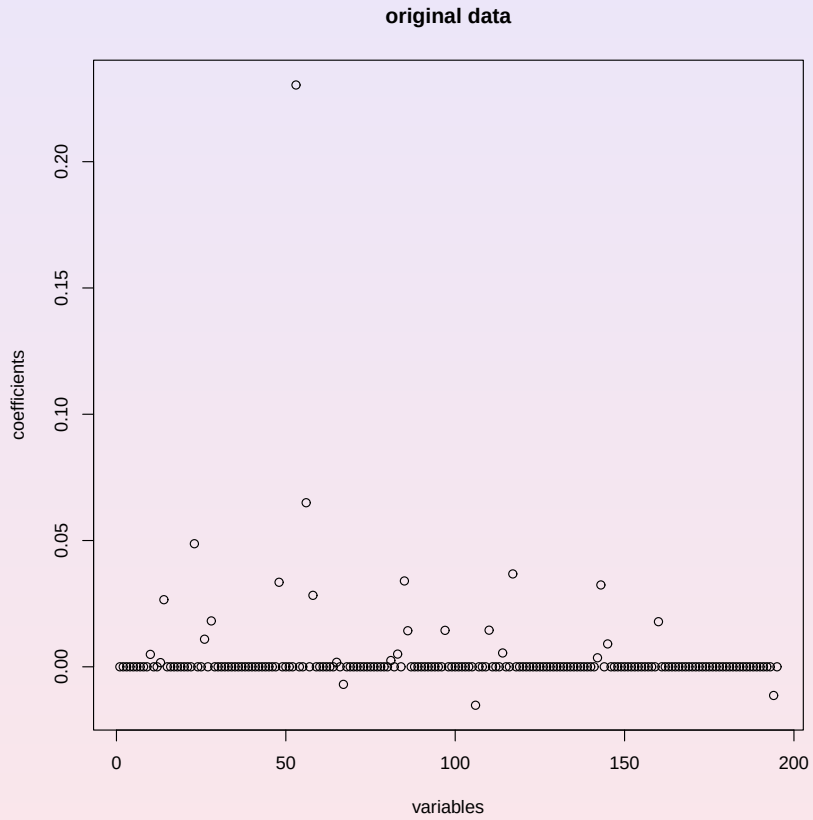


44 selected variables

36 selected variables

Motif regression ($p = 195,\ n = 287$)

26 selected covariates when using $\hat{\lambda}_{CV}$



presumably: the truly relevant variables are among the 26 selected covariates

A first conclusion for Lasso

Lasso is a good screening method: with high probability

$$\hat{S} \supseteq S_0$$

and two or multi-stage methods can be used
$\rightsquigarrow$ re-estimation on much smaller model with variables from $\hat{S}$

- OLS on $\hat{S}$ with e.g. BIC variable selection
- thresholding coefficients and maybe OLS re-estimation
- adaptive Lasso (Zou, 2006)

A first conclusion for Lasso

Lasso is a good screening method: with high probability

$$\hat{S} \supseteq S_0$$

and two or multi-stage methods can be used
$\rightsquigarrow$ re-estimation on much smaller model with variables from $\hat{S}$

- OLS on $\hat{S}$ with e.g. BIC variable selection
- thresholding coefficients and maybe OLS re-estimation
- adaptive Lasso (Zou, 2006)

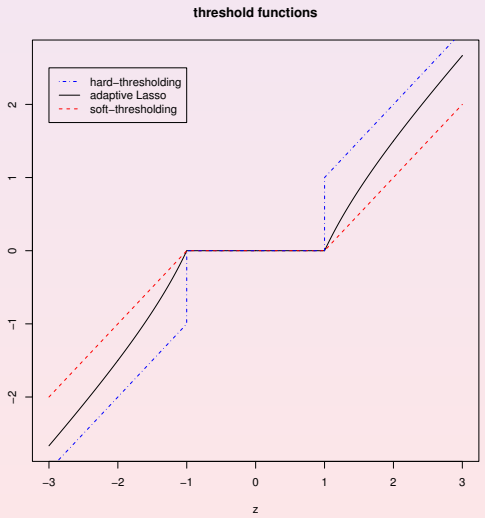
Adaptive Lasso (Zou, 2006)

re-weighting the penalty function

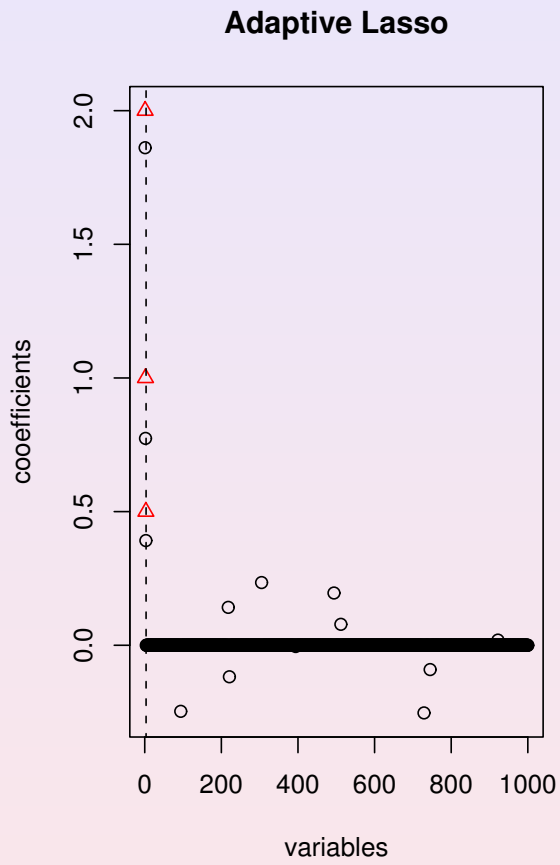
$$\hat{\beta} = \text{argmin}_{\beta}(\|\mathbf{Y} - \mathbf{X}\beta\|_2^2/n + \lambda \sum_{j=1}^{p} \frac{|\beta_j|}{|\hat{\beta}_{init,j}|}),$$

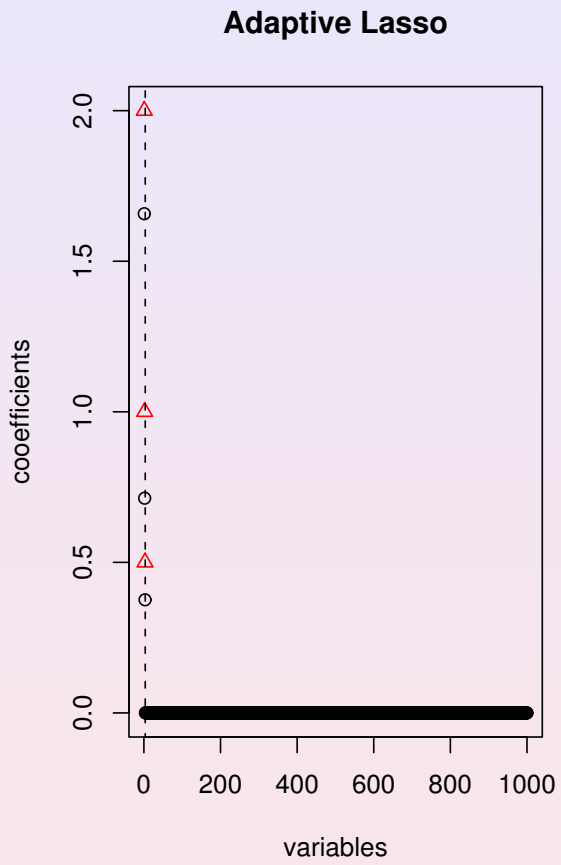
$\hat{\beta}_{init,j}$ from Lasso in first stage $\underbrace{\text{(or OLS if } p < n\text{)}}_{\text{Zou (2006)}}$

for orthogonal design,
if $\hat{\beta}_{init}$ = OLS:
Adaptive Lasso = NN-garrote
$\rightsquigarrow$ less bias than Lasso



$s_0 = 3,\ p = 1'000,\ n = 50$
same 2 independent realizations from before

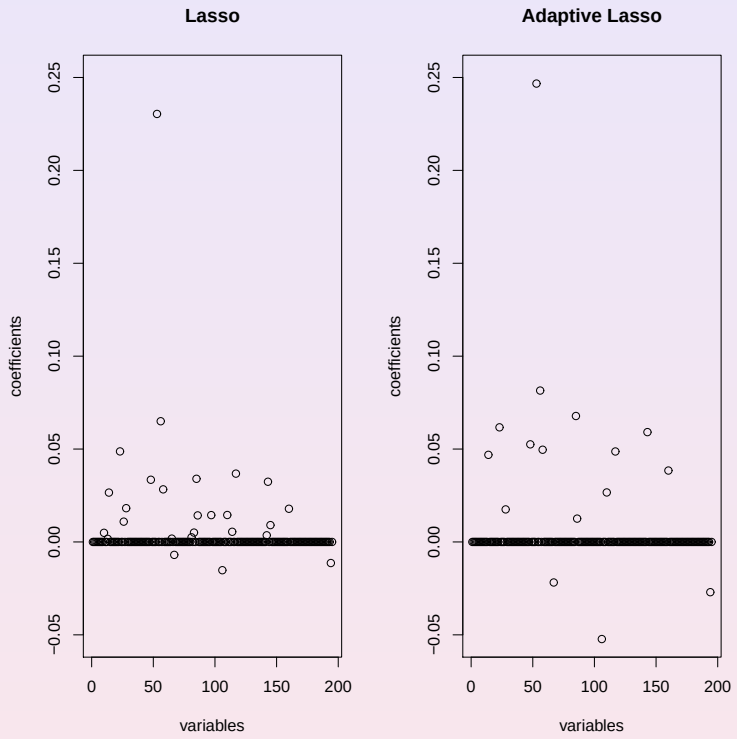




13 selected variables
(44 with Lasso)

3 selected variables
(36 with Lasso)

Motif regression: $n = 287,\ p = 195$



26 selected variables

16 selected variables

trivial property

$$\hat{\beta}_{init,j} = 0 \Rightarrow \hat{\beta}_j = 0$$

since

$$\hat{\beta} = \text{argmin}_\beta(\|\mathbf{Y} - \mathbf{X}\beta\|_2^2/n + \lambda \sum_{j=1}^{p} \frac{|\beta_j|}{|\hat{\beta}_{init,j}|})$$

another motif regression (linear model): $n = 2587$, $p = 666$

	Lasso	1-Step	2-Step
test set squared prediction error	0.6193	0.6230	0.6226
number of selected variables	91	42	28

$\rightsquigarrow$ substantially sparser model fit with
twice-iterated adaptive Lasso (three-stage procedure)

Some perspective from theory

Adaptive Lasso is consistent for variable selection under typically weaker assumptions than $\underbrace{\text{irrepresentable condition}}_{\text{necess. and suff. for Lasso}}$

Computation and KKT for Lasso

important characterization of the Lasso solution $\hat{\beta} = \hat{\beta}(\lambda)$

Lemma

Denote: $G(\beta) = -2\mathbf{X}^T(\mathbf{Y} - \mathbf{X}\beta)/n$ (gradient of $n^{-1}\|\mathbf{Y} - \mathbf{X}\beta\|_2^2$)

Then: a necessary and sufficient condition for Lasso solution

$$G_j(\hat{\beta}) = -\text{sign}(\hat{\beta}_j)\lambda \text{ if } \hat{\beta}_j \neq 0,$$
$$|G_j(\hat{\beta})| \leq \lambda \text{ if} \hat{\beta}_j = 0.$$

Moreover:
if Lasso solution is not unique (e.g. if $p > n$) and $G_j(\hat{\beta}) < \lambda$ for some solution $\hat{\beta}$, then $\hat{\beta}_j = 0$ for all Lasso solutions
i.e. the zeroes are unique (and hence estimated variable selection is “well-defined”)

Karush-Kuhn-Tucker (KKT) conditions

$$Q_\lambda(\beta) = \|\mathbf{Y} - \mathbf{X}\beta\|_2^2/n + \lambda\|\beta\|_1.$$

for a minimizer $\hat{\beta}(\lambda)$ of $Q_\lambda(\cdot)$:
necessary and sufficient that the subdifferential at $\hat{\beta}(\lambda)$ is zero

case I: jth component $\hat{\beta}_j(\lambda) \neq 0$
$\rightsquigarrow$ ordinary first derivative at $\hat{\beta}(\lambda)$ has to be zero:

$$\frac{\partial Q_\lambda(\beta)}{\partial \beta_j}|_{\beta=\hat{\beta}(\lambda)} = -2\mathbf{X}_j^T(\mathbf{Y} - \mathbf{X}\beta) + \lambda \text{sign}(\beta_j)|_{\beta=\hat{\beta}(\lambda)} = 0$$

$\Leftrightarrow$

$$G_j(\hat{\beta}(\lambda) = -2\mathbf{X}_j^T(\mathbf{Y} - \mathbf{X}\beta(\lambda) = -\lambda \text{sign}(\hat{\beta}_j(\lambda)) \text{ if } \hat{\beta}_j(\lambda) \neq 0$$

case II: if $\hat{\beta}_j(\lambda) = 0$
$\rightsquigarrow$ the subdifferential at $\hat{\beta}(\lambda)$ has to include the zero element, i.e.:

$$G_j(\hat{\beta}(\lambda)) + \lambda e = 0 \text{ for some } e \in [-1, 1], \text{ and if } \hat{\beta}_j(\lambda) = 0.$$

$\Leftrightarrow$

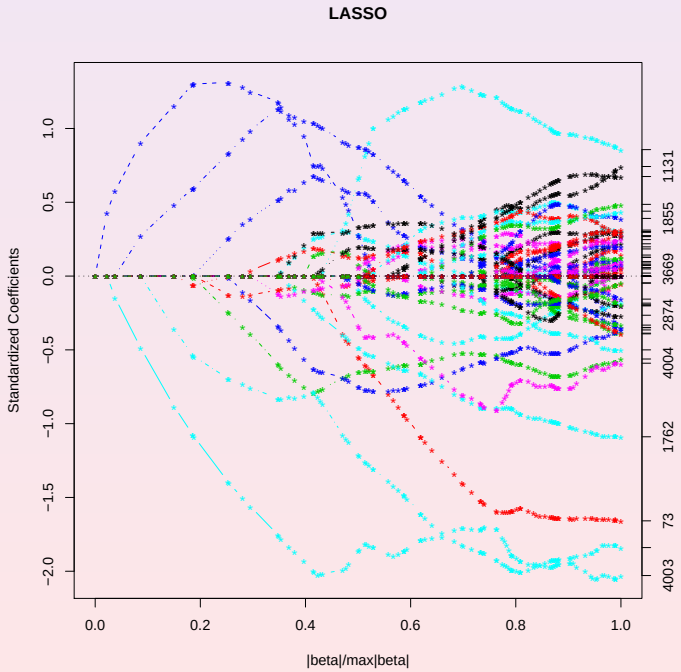
$$|G_j(\hat{\beta}(\lambda))| \leq \lambda \text{ if } \hat{\beta}_j(\lambda) = 0.$$

Path-following computation

goal: compute the Lasso-estimator $\hat{\beta}(\lambda)$ for many values of λ
e.g. when using cross-validation and searching for optimal λ

regularized solution path over all values of λ is piecewise linear
computat. complexity of whole path: $O(np\min(n,p)) \underbrace{=}_{p \gg n} O(p)$

riboflavin production example: $n = 71,\ p = 4088$



Coordinatewise optimization and shooting algorithms

general idea is to compute a solution $\hat{\beta}(\lambda_{\text{grid},k})$ and use it as a starting value for the computation of $\hat{\beta}(\underbrace{\lambda_{\text{grid},k-1}}_{<\lambda_{\text{grid},k}})$

Coordinatewise algorithm

$\beta^{(0)} \in \mathbb{R}^p$ an initial parameter vector. Set $m = 0$.

REPEAT:
Increase m by one: $m \leftarrow m + 1$.
For $j = 1, \ldots, p$:

$$\text{if } |G_j(\beta_{-j}^{(m-1)})| \leq \lambda : \text{ set } \beta_j^{(m)} = 0,$$
$$\text{otherwise: } \beta_j^{(m)} = \text{argmin}_{\beta_j} Q_\lambda(\beta_{+j}^{(m-1)}),$$

β_{-j}: parameter vector setting jth component to zero
$\beta_{+j}^{(m-1)}$: parameter vector which equals $\beta^{(m-1)}$ except for jth component equalling β_j
UNTIL numerical convergence

for squared error loss: explicit up-dating formulaa

$$G_j(\beta) = -2\mathbf{X}_j^T(y - \mathbf{X}\beta)$$
$$\beta_j^{(m)} = \frac{\text{sign}(Z_j)(|Z_j| - \lambda/2)_+}{\hat{\Sigma}_{jj}},$$
$$Z_j = \mathbf{X}_j^T(\mathbf{Y} - \mathbf{X}\beta_{-j}), \quad \hat{\Sigma} = n^{-1}\mathbf{X}^T\mathbf{X}.$$

$\rightsquigarrow$ componentwise soft-thresholding

this is very fast if true problem is sparse (can do non-systematic cycling, visiting mainly the active (non-zero) components)

riboflavin example, n=71, p=4088 0.33 secc. CPU using `glmnet`-package in `R` (Friedman, Hastie & Tibshirani, 2008)

The Group Lasso (Yuan & Lin, 2006)

high-dimensional parameter vector is structured into q groups or partitions (known a-priori):

$$\mathcal{G}_1, \ldots, \mathcal{G}_q \subseteq \{1, \ldots, p\}, \text{ disjoint and } \cup_g \mathcal{G}_g = \{1, \ldots, p\}$$

corresponding coefficients: $\beta_{\mathcal{G}} = \{\beta_j;\ j \in \mathcal{G}\}$

Example: categorical covariates

$X^{(1)}, \ldots, X^{(p)}$ are factors (categorical variables)
each with 4 levels (e.g. "letters" from DNA)

for encoding a main effect: 3 parameters
for encoding a first-order interaction: 9 parameters
and so on ...

parameterization (e.g. sum contrasts) is structured as follows:

- intercept: no penalty
- main effect of $X^{(1)}$: group $\mathcal{G}_1$ with $df = 3$
- main effect of $X^{(2)}$: group $\mathcal{G}_2$ with $df = 3$
- ...
- first-order interaction of $X^{(1)}$ and $X^{(2)}$: $\mathcal{G}_{p+1}$ with $df = 9$
- ...

often, we want sparsity on the group-level
either all parameters of an effect are zero or not

often, we want sparsity on the group-level
either all parameters of an effect are zero or not

this can be achieved with the Group-Lasso penalty

$$\lambda \sum_{g=1}^{q} s(df_g) \underbrace{\|\beta_{\mathcal{G}_g}\|_2}_{\sqrt{\|\cdot\|_2^2}}$$

typically $s(df_{\mathcal{G}_g}) = \sqrt{df_{\mathcal{G}_g}}$ so that $s(df_{\mathcal{G}_g})\|\beta_{\mathcal{G}_g}\|_2 = O(df_g)$

properties of Group-Lasso penalty

- for group-sizes $|\mathcal{G}_g| \equiv 1 \rightsquigarrow$ standard Lasso-penalty
- convex penalty $\rightsquigarrow$ convex optimization for standard likelihoods (exponential family models)
- either $(\hat{\beta}_{\mathcal{G}}(\lambda))_j = 0$ or $\neq 0$ for all $j \in \mathcal{G}$
- penalty is invariant under orthonormal transformation
  e.g. invariant when requiring orthonormal parameterization for factors

Some aspects from theory

"again":

- optimal prediction and estimation (oracle inequality)
- group screening: $\hat{S} \supseteq \underbrace{S_0}_{\text{set of active groups}}$ with high prob.

most interesting case:

- $\mathcal{G}_j$'s are "large"
- $\beta_{\mathcal{G}_j}$'s are "smooth"

example: high-dimensional additive model

$$Y = \sum_{j=1}^{p} f_j(X^{(j)}) + \epsilon$$

and expand $f_j(x^{(j)}) = \sum_{k=1}^{n} \underbrace{\beta_k^{(j)}}_{(\beta_{\mathcal{G}_j})_k} \underbrace{B_k^{(j)}}_{\text{basis fct.s}} (x^{(j)})$

$f_j(\cdot)$ smooth $\Rightarrow$ "smoothness" of $\beta_{\mathcal{G}_j}$

Computation and KKT

criterion function

$$Q_\lambda(\beta) = n^{-1} \sum_{i=1}^{n} \underbrace{\rho_\beta(x_i, Y_i)}_{\text{loss fct.}} + \lambda \sum_{g=1}^{G} s(df_g) \|\beta_g\|_2,$$

loss function $\rho_\beta(.,.)$ convex in β

KKT conditions:

$$\nabla \rho(\hat{\beta})_g + \lambda s(df_g) \frac{\hat{\beta}_{\mathcal{G}_g}}{\|\hat{\beta}_{\mathcal{G}_g}\|_2} = 0 \text{ if } \hat{\beta}_{\mathcal{G}_g} \neq 0 \text{ (not the 0-vector)},$$

$$\|\nabla \rho(\hat{\beta})_g\|_2 \leq \lambda s(df_g) \text{ if } \hat{\beta}_{\mathcal{G}_g} \equiv 0.$$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \ldots, p, 1, 2, \ldots$
  or $j = 1, \ldots, q, 1, 2, \ldots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$)
  keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

$$\text{Lasso: } (\beta_1, \beta_2 = \beta_2^{(0)}, \ldots, \beta_j = \beta_j^{(0)}, \ldots, \beta_p = \beta_p^{(0)})$$

$$\uparrow$$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \ldots, p, 1, 2, \ldots$
  or $j = 1, \ldots, q, 1, 2, \ldots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$)
  keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

$$\text{Lasso: } (\beta_1 = \beta_1^{(1)}, \beta_2, \ldots, \beta_j = \beta_j^{(0)}, \ldots, \beta_p = \beta_p^{(0)})$$

$$\uparrow$$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \ldots, p, 1, 2, \ldots$
  or $j = 1, \ldots, q, 1, 2, \ldots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$) keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

$$\text{Lasso: } (\beta_1 = \beta_1^{(1)}, \beta_2 = \beta_2^{(1)}, \ldots, \beta_j, \ldots, \beta_p = \beta_p^{(0)})$$
$$\uparrow$$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \ldots, p, 1, 2, \ldots$
  or $j = 1, \ldots, q, 1, 2, \ldots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$) keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

$$\text{Lasso: } (\beta_1 = \beta_1^{(1)}, \beta_2 = \beta_2^{(1)}, \ldots, \beta_j = \beta_j^{(1)}, \ldots, \beta_p)$$

$$\uparrow$$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \ldots, p, 1, 2, \ldots$
  or $j = 1, \ldots, q, 1, 2, \ldots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$) keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

$$\text{Lasso: } (\beta_1, \beta_2 = \beta_2^{(1)}, \ldots, \beta_j = \beta_j^{(1)}, \ldots, \beta_p = \beta_p^{(1)})$$

$$\uparrow$$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \ldots, p, 1, 2, \ldots$
  or $j = 1, \ldots, q, 1, 2, \ldots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$)
  keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

Group Lasso: $(\beta_{\mathcal{G}_1}, \beta_{\mathcal{G}_2} = \beta^{(0)}_{\mathcal{G}_2}, \ldots, \beta_{\mathcal{G}_j} = \beta^{(0)}_{\mathcal{G}_j}, \ldots, \beta_{\mathcal{G}_q} = \beta^{(0)}_{\mathcal{G}_q})$

$\uparrow$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \ldots, p, 1, 2, \ldots$
  or $j = 1, \ldots, q, 1, 2, \ldots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$)
  keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

Group Lasso: $(\beta_{\mathcal{G}_1} = \beta_{\mathcal{G}1}^{(1)}, \beta_{\mathcal{G}_2}, \ldots, \beta_{\mathcal{G}_j} = \beta_{\mathcal{G}_j}^{(0)}, \ldots, \beta_{\mathcal{G}_q} = \beta_{\mathcal{G}_q}^{(0)})$

$\uparrow$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \ldots, p, 1, 2, \ldots$
  or $j = 1, \ldots, q, 1, 2, \ldots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$)
  keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

Group Lasso: $(\beta_{\mathcal{G}_1} = \beta^{(1)}_{\mathcal{G}1}, \beta_{\mathcal{G}_2} = \beta^{(1)}_{\mathcal{G}_2}, \ldots, \beta_{\mathcal{G}_j}, \ldots, \beta_{\mathcal{G}_q} = \beta^{(0)}_{\mathcal{G}_q})$

$\uparrow$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \ldots, p, 1, 2, \ldots$
  or $j = 1, \ldots, q, 1, 2, \ldots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$)
  keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

Group Lasso: $(\beta_{\mathcal{G}_1} = \beta^{(1)}_{\mathcal{G}1}, \beta_{\mathcal{G}_2} = \beta^{(1)}_{\mathcal{G}_2}, \ldots, \beta_{\mathcal{G}_j} = \beta^{(1)}_{\mathcal{G}_j}, \ldots, \beta_{\mathcal{G}_q})$

$\uparrow$

Block coordinate descent algorithm

generic description for both, Lasso or Group-Lasso problems:

- cycle through all coordinates $j = 1, \ldots, p, 1, 2, \ldots$ or $j = 1, \ldots, q, 1, 2, \ldots$
- optimize the penalized log-likelihood w.r.t. β_j (or $\beta_{\mathcal{G}_j}$) keeping all other coefficients $\beta_k, k \neq j$ (or $k \neq \mathcal{G}_j$) fixed

Group Lasso: $(\beta_{\mathcal{G}_1}, \beta_{\mathcal{G}_2} = \beta^{(1)}_{\mathcal{G}_2}, \ldots, \beta_{\mathcal{G}_j} = \beta^{(1)}_{\mathcal{G}_j}, \ldots, \beta_{\mathcal{G}_q} = \beta^{(1)}_{\mathcal{G}_q})$

$\uparrow$

for Gaussian log-likelihood (squared error loss):
blockwise up-dates are easy and closed-form solutions exist (use KKT)

for other loss functions (e.g. logistic loss):
blockwise up-dates: no closed-form solution

$\rightsquigarrow$

strategy which is fast: improve every coordinate/group numerically, but not until numerical convergence
(by using quadratic approximation of log-likelihood function for improving/optimization of a single block)

and further tricks... (still allowing provable numerical convergence)

How fast?

logistic case: $p = 10^6,\ n = 100$
group-size = 20, sparsity: 2 active groups = 40 parameters

for 10 different λ-values
CPU using `grplasso`: 203.16 seconds $\approx$ 3.5 minutes
(dual core processor with 2.6 GHz and 32 GB RAM)

we can easily deal today with predictors in the Mega's
i.e. $p \approx 10^6 - 10^7$

How fast?

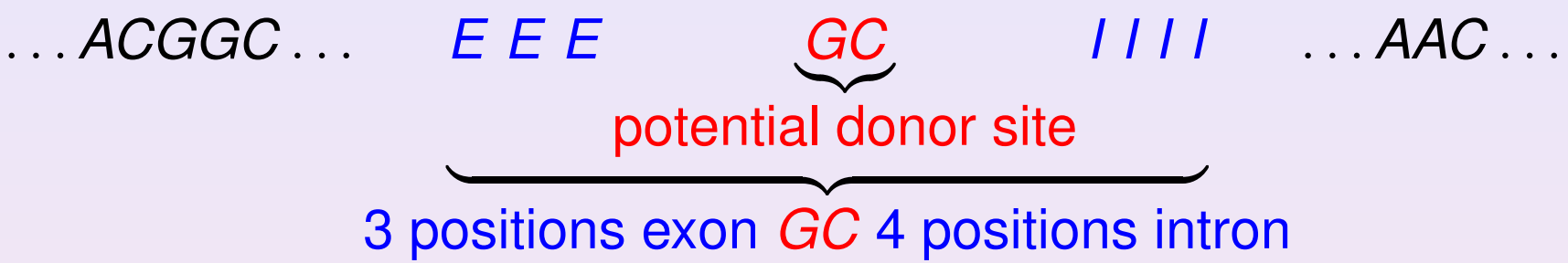
logistic case: $p = 10^6,\ n = 100$
group-size = 20, sparsity: 2 active groups = 40 parameters

for 10 different λ-values
CPU using `grplasso`: 203.16 seconds $\approx$ 3.5 minutes
(dual core processor with 2.6 GHz and 32 GB RAM)

we can easily deal today with predictors in the Mega's
i.e. $p \approx 10^6 - 10^7$

DNA splice site detection: (mainly) prediction problem

DNA sequence



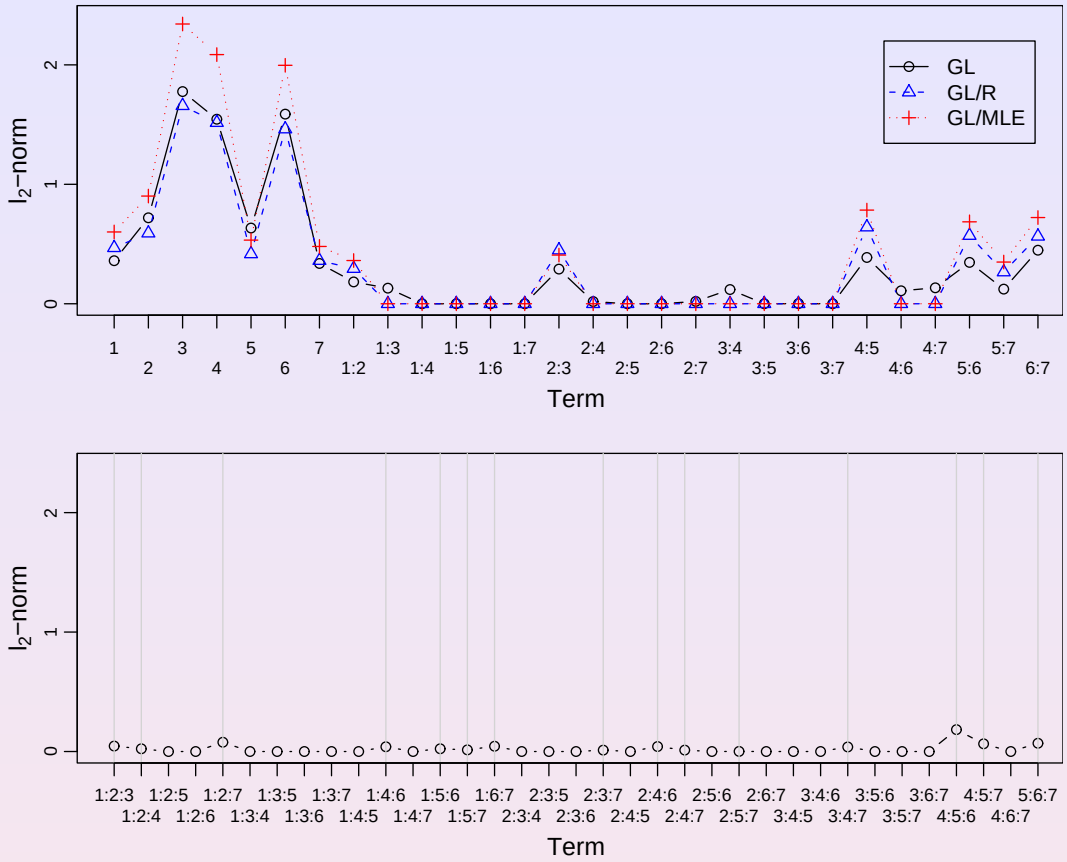
response $Y \in \{0, 1\}$: splice or non-splice site
predictor variables: 7 factors each having 4 levels
(full dimension: $4^7 = 16'384$)
data:

training:	5′610 true splice sites
	5′610 non-splice sites
	plus an unbalanced validation set
test data:	4′208 true splice sites
	89′717 non-splice sites

logistic regression:

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \text{ main effects} + \text{first order interactions} + \ldots$$

use the Group-Lasso which selects whole terms



- mainly neighboring DNA positions show interactions (has been “known” and “debated”)
- no interaction among exons and introns (with Group Lasso method)
- no second-order interactions (with Group Lasso method)

predictive power:
competitive with “state to the art” maximum entropy modeling from Yeo and Burge (2004)

correlation between true and predicted class

Logistic Group Lasso	0.6593
max. entropy (Yeo and Burge)	0.6589

our model (not necessarily the method/algorithm) is simple and has clear interpretation

The sparsity-smoothness penalty (SSP)

(whose corresponding optimization becomes again a Group-Lasso problem...)

for additive modeling in high dimensions

$$Y_i = \sum_{j=1}^{p} f_j(x_i^{(j)}) + \varepsilon_i \ (i = 1, \ldots, n)$$

$f_j : \mathbb{R} \to \mathbb{R}$ smooth univariate functions

$p \gg n$

in principle: basis expansion for every $f_j(\cdot)$ with basis functions

$$B_1^{(j)}, \ldots, B_m^{(j)} \text{ where } m = O(n) \quad (\text{or e.g. } m = O(n^{1/2}))$$
$$j = 1, \ldots, p$$

$\rightsquigarrow$ represent

$$\sum_{j=1}^{p} f_j(x^{(j)}) = \sum_{j=1}^{p} \sum_{k=1}^{m} \beta_k^{(j)} B_k^{(j)}(x^{(j)})$$

$\rightsquigarrow$ high-dimensional parametric problem

and use the Group-Lasso penalty to ensure sparsity of whole functions

$$\lambda \sum_{g=1}^{p} \| \underbrace{\beta_{\mathcal{G}_j}}_{(\beta_1^{(j)}, \ldots, \beta_m^{(j)})^T} \|_2$$

drawback:
does not exploit smoothness
(except when choosing appropriate m which is “bad” if different f_j’s have different complexity)

when using a large number of basis functions (large m) for achieving a high degree of flexibility
$\rightsquigarrow$ need additional control for smoothness

Sparsity-Smoothness Penalty (SSP)

(Meier, van de Geer & PB, 2008)

$$\lambda_1 \sum_{j=1}^{p} \sqrt{\|f_j\|_2^2 + \lambda_2 I^2(f_j)}$$

$$I^2(f_j) = \int (f_j''(x))^2 dx$$

where $f_j = (f_j(X_1^{(j)}), \ldots, f_j(X_n^{(j)}))^T$

$\rightsquigarrow$ SSP-penalty does variable selection ($\hat{f}_j \equiv 0$ for some j)

and SSP-penalty is "oracle optimal"

for additive modeling:

$$\hat{f}_1, \ldots, \hat{f}_p = \text{argmin}_{f_1,\ldots,f_p} \| Y - \sum_{j=1}^{p} f_j \|_2^2 + \lambda_1 \sum_{j=1}^{p} \sqrt{\| f_j \|_2^2 + \lambda_2 I^2(f_j)}$$

assuming f_j is twice differentiable

$\rightsquigarrow$ solution is a natural cubic spline with knots at $X_i^{(j)}$

$\rightsquigarrow$ finite-dimensional parameterization with e.g. B-splines:

$$f = \sum_{j=1}^{p} f_j, \quad f_j = \underbrace{B_j}_{n \times m} \underbrace{\beta_j}_{m \times 1}$$

penalty becomes:

$$\lambda_1 \sum_{j=1}^{p} \sqrt{\|f_j\|_2^2 + \lambda_2 I^2(f_j)}$$

$$= \lambda_1 \sum_{j=1}^{p} \sqrt{\beta_j^T \underbrace{B_j^T B_j}_{\Sigma_j} \beta_j + \lambda_2 \beta_j^T \underbrace{\Omega_j}_{\text{integ. 2nd derivatives}} \beta_j}$$

$$= \lambda_1 \sum_{j=1}^{p} \sqrt{\beta_j^T \underbrace{(\Sigma_j + \lambda_2 \Omega_j)}_{A_j = A_j(\lambda_2)} \beta}$$

$\rightsquigarrow$ re-parameterize $\tilde{\beta}_j = \tilde{\beta}_j(\lambda_2) = R_j \beta_j, \qquad R_j^T R_j = A_j = A_j(\lambda_2)$ (Choleski)
penalty becomes

$$\lambda_1 \sum_{j=1}^{p} \underbrace{\|\tilde{\beta}_j\|_2}_{\text{depending on } \lambda_2}$$

i.e., a Group-Lasso penalty

HIF1α motif additive regression

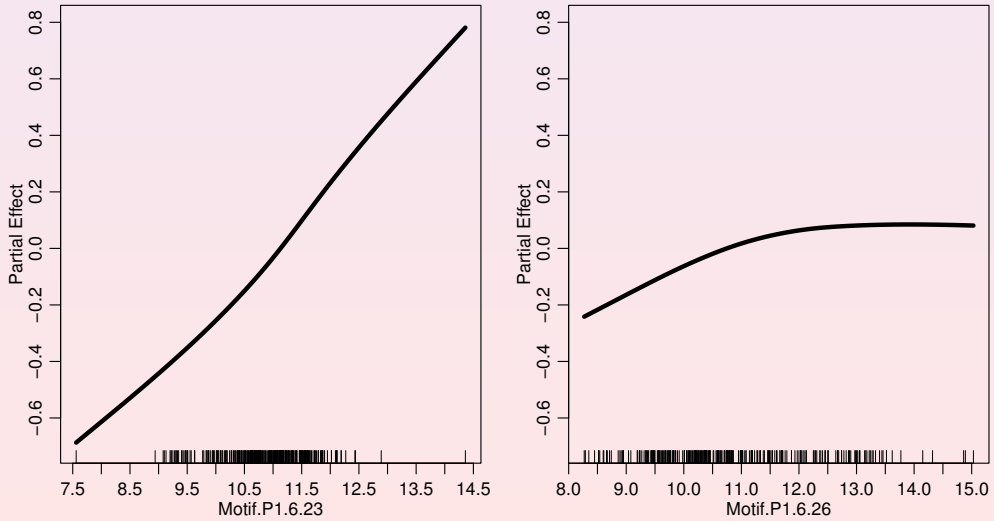
for finding HIF1α transcription factor binding sites on DNA sequences

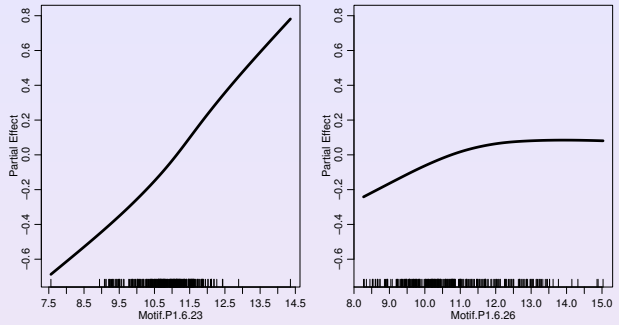
$n = 287$, $p = 196$

additive model with SSP has $\approx$ 20% better prediction performance than linear model with Lasso

bootstrap stability analysis: select the variables (functions) which have occurred at least in 50% among all bootstrap runs

$\rightsquigarrow$ only 2 stable variables /candidate motifs remain





right panel: variable corresponds to a true, known motif

variable/motif corresponding to left panel:
good additional support for relevance (nearness to transcriptional start-site of important genes, ...)
ongoing validation with Ricci and Krek labs, ETH Zurich

P-values for high-dimensional regression

Motif regression
for finding HIF1α transcription factor binding sites in DNA seq.

variable selection in linear model $\mathbf{Y} = \mathbf{X}\beta + \epsilon$,
$n = 287,\ p = 195$

$\rightsquigarrow$ Lasso selects 26 covariates and $R^2 \approx 50\%$
i.e. 26 interesting candidate motifs
and hence report these findings to the biologists...

really?
do we trust our selection algorithm?
how stable are the findings?

P-values for high-dimensional regression

Motif regression
for finding HIF1α transcription factor binding sites in DNA seq.

variable selection in linear model $\mathbf{Y} = \mathbf{X}\beta + \epsilon$,
$n = 287,\ p = 195$

$\rightsquigarrow$ Lasso selects 26 covariates and $R^2 \approx 50\%$
i.e. 26 interesting candidate motifs
and hence report these findings to the biologists...

really?
do we trust our selection algorithm?
how stable are the findings?

P-values for high-dimensional regression

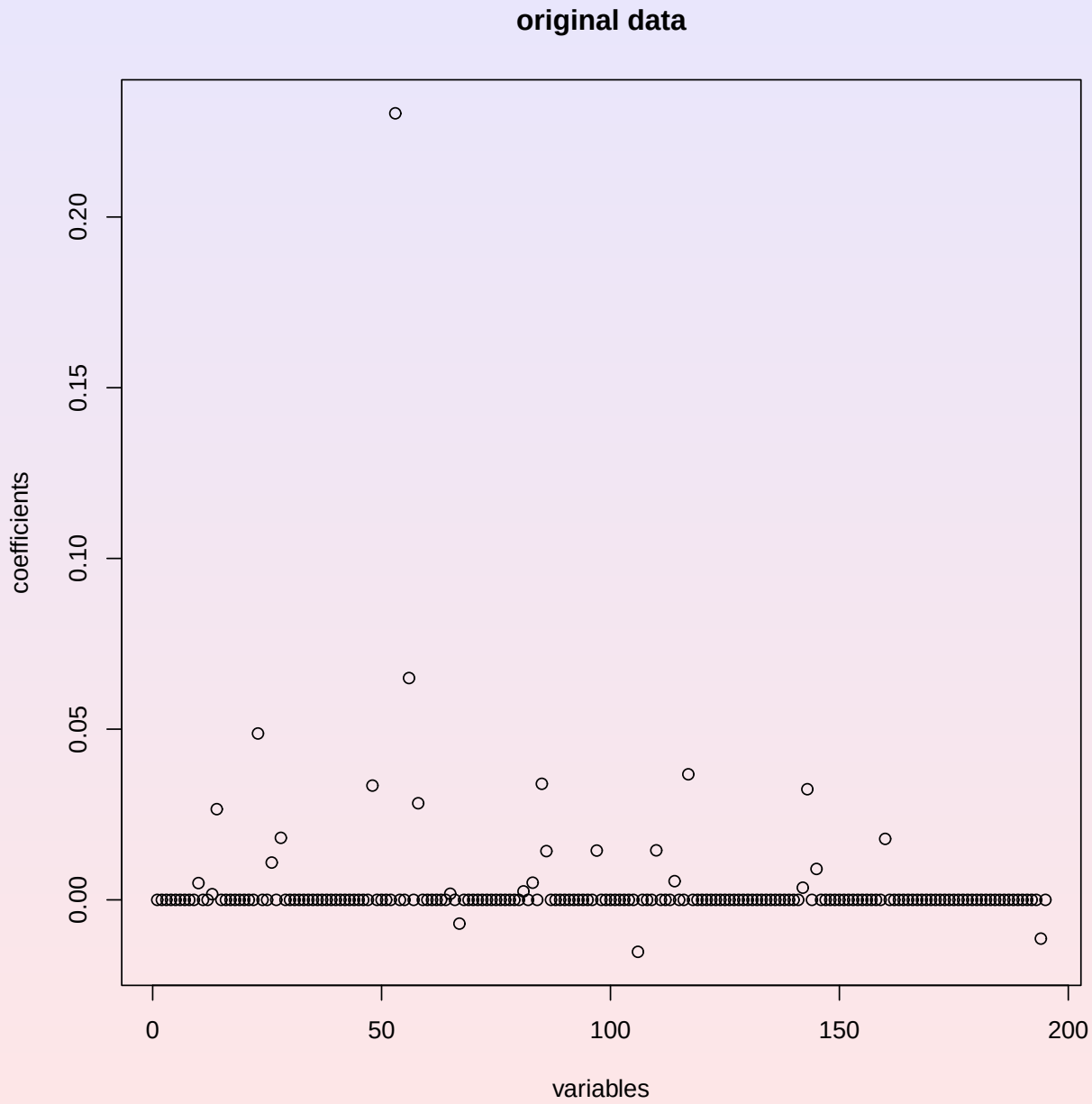
Motif regression
for finding HIF1α transcription factor binding sites in DNA seq.

variable selection in linear model $\mathbf{Y} = \mathbf{X}\beta + \epsilon$,
$n = 287,\ p = 195$

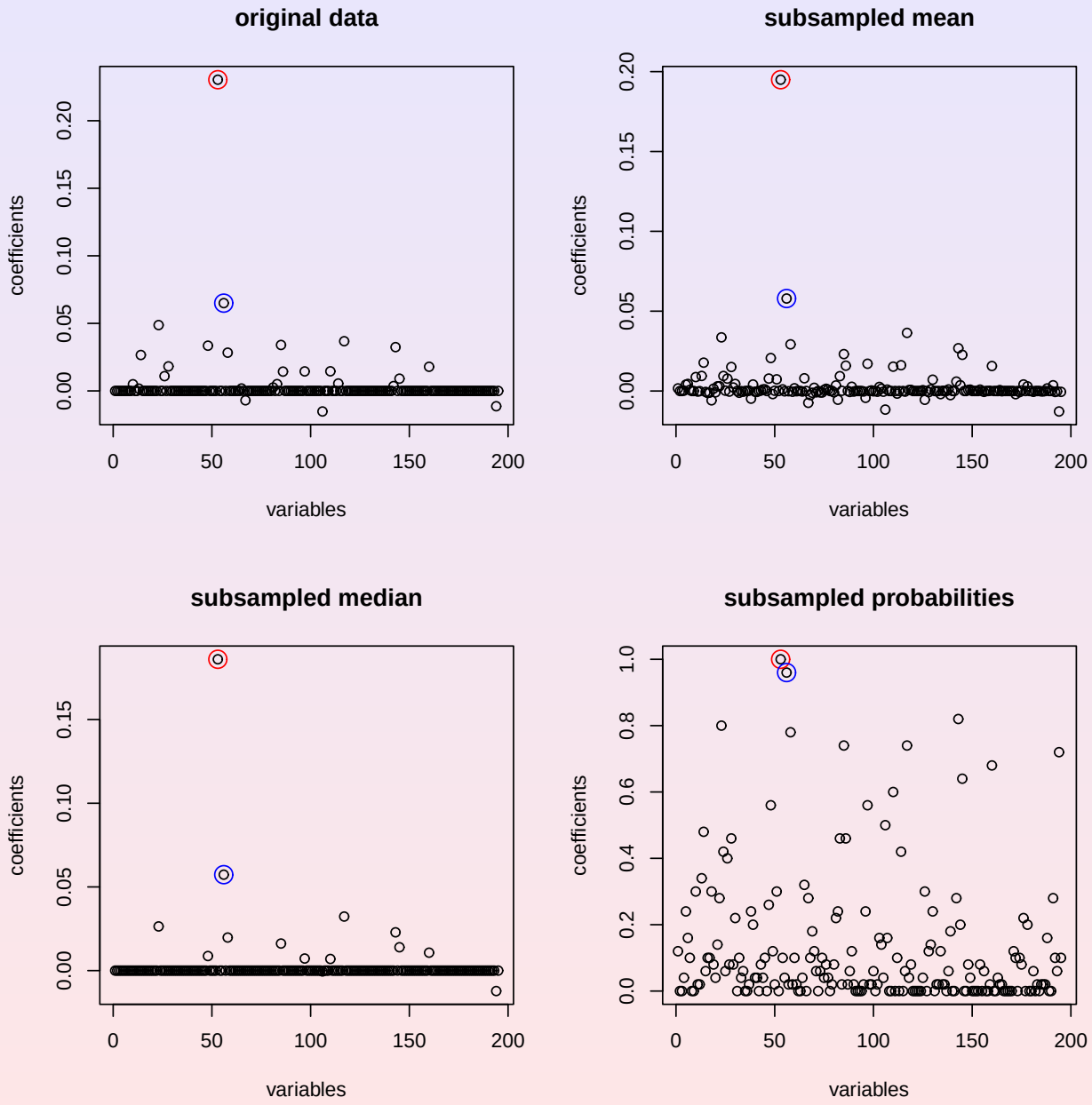
$\rightsquigarrow$ Lasso selects 26 covariates and $R^2 \approx 50\%$
i.e. 26 interesting candidate motifs
and hence report these findings to the biologists...

really?
do we trust our selection algorithm?
how stable are the findings?

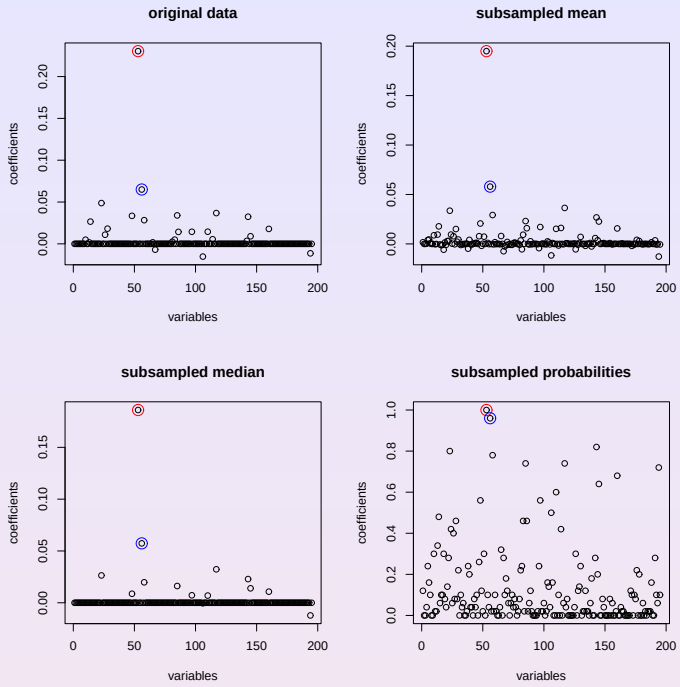
estimated coefficients $\hat{\beta}(\hat{\lambda}_{\mathrm{CV}})$



stability check: subsampling with subsample size $\lfloor n/2 \rfloor$



$\rightsquigarrow$ only 2 “stable” findings
($\neq$ 26)



one variable (○):
corresponds to true, known motif

other variable (○): good additional support for relevance
(nearness to transcriptional start-site of important genes, ...)
ongoing biological validation with Ricci lab (ETH Zurich)

and we would like to have a P-value!

P-values (Meinshausen, Meier & PB, 2008)

for simplicity: focus on P-values for regression coefficients
$H_0^{(j)} : \beta_j = 0$

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

A first idea: sample splitting with sub-samples of sizes $\lfloor n/2 \rfloor$

related to subsampling with sub-sample size $\lfloor n/2 \rfloor$

- select variables on first half of the sample $\rightsquigarrow \hat{S}$
- compute OLS for variables in $\hat{S}$ on second half of the sample
  $\rightsquigarrow$ P-values $P^{(j)}$ based on Gaussian linear model

$$\begin{aligned} &\text{if } j \in \hat{S}: \quad P^{(j)} \text{ from } t\text{-statistics} \\ &\text{if } j \notin \hat{S}: \quad P^{(j)} = 1 \ \ (\text{i.e. if } \hat{\beta}^{(j)} = 0) \end{aligned}$$

Bonferroni-corrected P-values:

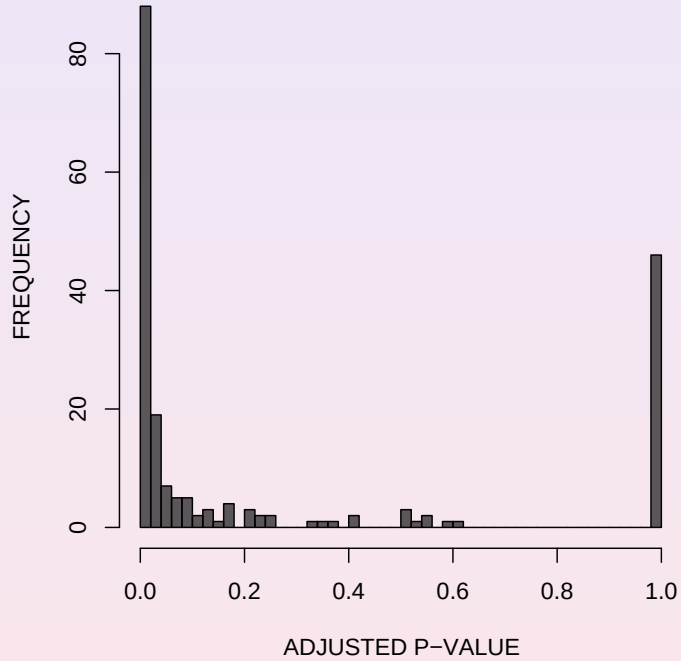
$$P_{\text{corr}}^{(j)} = \min(P^{(j)} \cdot |\hat{S}|, 1)$$

$\rightsquigarrow$ (conserv.) familywise error control with $P_{\text{corr}}^{(j)}$ $(j = 1, \ldots, p)$

(Wasserman & Roeder, 2008)

this is a “P-value lottery”
motif regression example: $p = 195,\ n = 287$

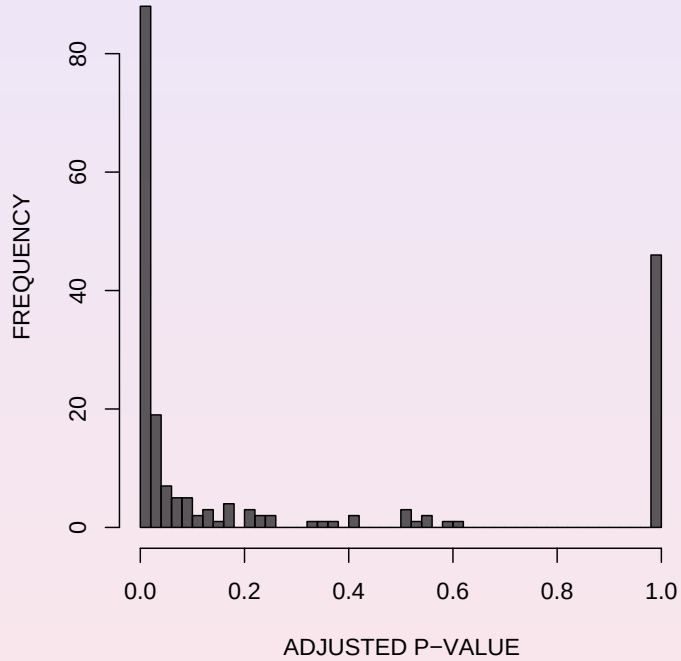
adjusted P-values for same important variable
over different random sample-splits



in addition: bad “efficiency”
$\rightsquigarrow$ improve by aggregating over many sample-splits

this is a “P-value lottery”
motif regression example: $p = 195,\ n = 287$

adjusted P-values for same important variable
over different random sample-splits



in addition: bad “efficiency”
$\rightsquigarrow$ improve by aggregating over many sample-splits

Multi sample-split P-values and aggregation

run the sample-splitting procedure B times:

$$\text{P-values: } P^{(j)}_{\text{corr},1}, \ldots, P^{(j)}_{\text{corr},B}$$

(assuming a Gaussian linear model with fixed design)

goal:
aggregation of $P^{(j)}_{\text{corr},1}, \ldots, P^{(j)}_{\text{corr},B}$ to a single P-value $P^{(j)}_{\text{final}}$

problem: dependence among $P^{(j)}_{\text{corr},1}, \ldots, P^{(j)}_{\text{corr},B}$

define

$$Q^{(j)}(\gamma) = \underbrace{q_\gamma}_{\text{emp. } \gamma\text{-quantile fct.}}(P^{(j)}_{\text{corr},b}/\gamma;\ b = 1, \ldots B)$$

e.g: $\gamma = 1/2$, aggregation with the median
$\rightsquigarrow$ (conserv.) familywise error control for any fixed value of γ

what is the best γ? it really matters
$\rightsquigarrow$ can "search" for it an correct with an additional factor

"adaptively" aggregated P-value:

$$P^{(j)}_{\text{final}} = (1 - \log(\gamma_{\min})) \cdot \inf_{\gamma \in (\gamma_{\min}, 1)} Q^{(j)}(\gamma)$$

$$Q^{(j)}(\gamma) = q_\gamma(P^{(j)}_{\text{corr},b}/\gamma;\ b = 1, \ldots B)$$

$$\rightsquigarrow \text{reject } H_0^{(j)} : \ \beta_j = 0 \iff P^{(j)}_{\text{final}} \leq \alpha$$

$P^{(j)}_{\text{final}}$ equals roughly a raw P-value based on sample size $\lfloor n/2 \rfloor$, multiplied by

$$\text{a factor} \approx (5 - 10) \cdot |\hat{\mathcal{S}}|$$

(which is to be compared with p)

for familywise error rate (FWER) =
$\mathbb{P}$[at least one false positive selection]

Theorem (Meinshausen, Meier & PB, 2008)
assumptions: Gaussian linear model (with fixed design) and

- $\lim_{n\to\infty} \mathbb{P}[\hat{S} \supseteq S] = 1$ screening property
- $|\hat{S}| < \lfloor n/2 \rfloor$ sparsity property

Then:

$P^{(j)}_{\text{final}}$'s yield asymptotic FWER control

$$\limsup_{n\to\infty} \mathbb{P}(\min_{j\in S^c} P^{(j)}_{\text{final}} \leq \alpha) \leq \alpha$$

i.e. (conservative) familywise error control

False discovery rate (FDR) (Benjamini & Hochberg, 1995)

based on ordered $P_{\text{final}}^{(j)}$'s from before

$\rightsquigarrow$ control of FDR for multiple testing of regression coefficients with $p \gg n$

(Meinshausen, Meier & PB, 2008)

assumptions for selector $\hat{\mathcal{S}}$:
are satisfied for

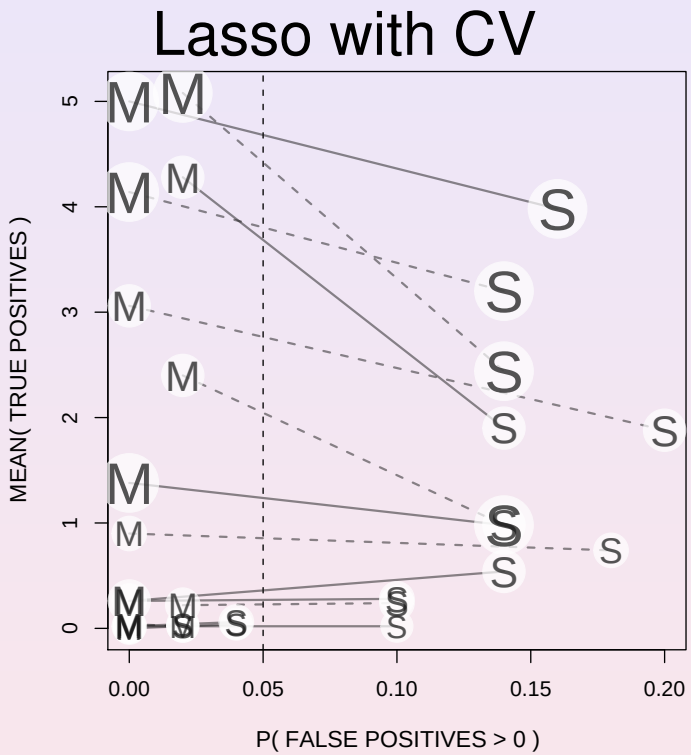
- Lasso
  - assuming compatibility conditions on the design **X**
  - assuming sparsity of true regression coefficients

- L_2Boosting, Sure Independence Screening, PC-algorithm,...
  - assuming reasonable conditions on the design
  - assuming sparsity of true regression coefficients

Simulations for FWER: $p = 1000,\ n = 100$

design matrix from multivariate Gaussian with $\Sigma_{j,k} = 0.5^{|j-k|}$

signal to noise ratio $\in \{0.25, 1, 4, 16\}$



multi sample-split method (M) has

- much better error control than single sample-split method
- (slightly) more power than single split method

Motif regression

$p = 195,\ n = 287$
for $\alpha = 0.05$, only one variable/motif $\tilde{j}$ remains

$$P_{\text{final}}^{(\tilde{j})} = 0.0059 \ \ (= 0.59\%)$$

and also with FDR control: only this one variable

in this application:
we are rather concerned about false positive findings
$\rightsquigarrow$ (conservative) P-values are very useful

Motif regression

$p = 195,\ n = 287$
for $\alpha = 0.05$, only one variable/motif $\tilde{j}$ remains

$$P_{\text{final}}^{(\tilde{j})} = 0.0059 \ \ (= 0.59\%)$$

and also with FDR control: only this one variable

in this application:
we are rather concerned about false positive findings
$\rightsquigarrow$ (conservative) P-values are very useful

From another perspective

using Lasso:

1. on first half of the sample: with high probability,

$$\hat{S} \supseteq S_{\text{subst};C}$$
$$S_{\text{subst};C} = \{j;\ |\beta_j| \geq C\}, \quad C \geq const.\sqrt{\log(p)/n}$$

2. on second half of the sample:
to get rid of the false positives, do OLS re-estimation and threshold with a P-value controlling FWER