Recap

$$Y = X\beta^0 + \varepsilon, \quad p \gg n$$

for estimation of β^0':
we need some identifiability conditions on X

Oracle inequality

consider (again)

$$\mathcal{T} = \{\max_{j \in \{1,\ldots,p\}} 2|\varepsilon^T X^{(j)}|/n \leq \lambda_0\}$$

Theorem 6.1 in Bühlmann and van de Geer (2011)
assume: compatibility condition holds with compatibility constant $\phi_0^2 \geq L > 0$
Then, on $\mathcal{T}$ and for $\lambda \geq 2\lambda_0$:

$$\|X(\hat{\beta} - \beta^0\|_2^2/n + \lambda\|\hat{\beta} - \beta^0\|_1 \leq 4\lambda^2 s_0/\phi_0^2$$

- we will "derive" the compatibility condition
- for e.g. $\varepsilon \sim \mathcal{N}_n(0, \sigma^2 I) \rightsquigarrow \mathbb{P}[\mathcal{T}] \to 1$ if $\lambda \geq 2\lambda_0 \sim C\sqrt{\log(p)/n}$ for $C > 0$ sufficiently large

for estimating/identifying β^0, we clearly need some assumptions on X

for $p > n$ and $\mathrm{rank}(X) = n$, the null-space of X is not only the zero vector:

$$X\xi = 0 \text{ for infinitely many } \xi \neq 0$$

$\rightsquigarrow X\beta^0 = X\theta$ for $\theta = \beta^0 + \xi$ with any ξ such that $X\xi = 0$.

we cannot identify the true parameter β^0 from (infinitely many) data
$\rightsquigarrow$ we have to make an assumption on X

Compatibility condition

the compatibility condition holds for the true active set S_0 with compatibility constant $\phi_0^2 > 0$ if:

$$\forall \beta \text{ satisfying } \|\beta_{S_0^c}\|_1 \leq 3\|\beta_{S_0}\|_1 :$$
$$\|\beta_{S_0}\|_1^2 \leq (\beta^T \hat{\Sigma} \beta) s_0 / \phi_0^2$$

see p. 106 in Bühlmann and van de Geer (2011)

$\rightsquigarrow$

Theorem 6.1 in Bühlmann and van de Geer (2011)
assume: compatibility condition holds with compatibility constant ϕ_0^2 ($\geq L$) > 0
Then, on $\mathcal{T}$ and for $\lambda \geq 2\lambda_0$:

$$\|X(\hat{\beta} - \beta^0\|_2^2/n + \lambda\|\hat{\beta} - \beta^0\|_1 \leq 4\lambda^2 s_0/\phi_0^2$$

Variable screening and $\|\hat{\beta} - \beta^0\|_q$-norms

estimation of parameters: thanks to the oracle inequality

$$\|\hat{\beta} - \beta^0\|_1 = O_P(s_0 \sqrt{\log(p)/n}\,(n \to \infty)$$

assuming

- compatibility condition on the (fixed) design X
- Gaussian errors (can be relaxed)

$\rightsquigarrow$ convergence to zero if sparsity $s_0 = o(\sqrt{n/\log(p)})$

under restricted eigenvalue assumption (slightly stronger than compatibility condition): one can show that

$$\|\hat{\beta} - \beta^0\|_2 = O_P(\sqrt{s_0 \log(p)/n})\,(n \to \infty)$$

$\rightsquigarrow$ convergence to zero under weaker ass. $s_0 = o(n/\log(p))$

Variable screening

active set (of variables): $S_0 = \{j;\ \beta_j^0 \neq 0\}$

estimated active set: $\hat{S}_0 = \{j;\ \hat{\beta}_j \neq 0\}$

Question 1: is $\hat{S}_0 = S_0$ with high probability?
$\rightsquigarrow$ often too ambitious goal

Question 2: can we do variable screening $\hat{S} \supseteq S_0$ with high probability?
still very relevant in practice: dimensionality reduction!

need to make an assumption that true regression coefficients are not too small

$$\text{"beta-min condition"}: \min_{j \in S_0} |\beta_j^0| \gg s_0 \sqrt{\log(p)/n}$$

$$\implies \mathbb{P}[\hat{S} \supseteq S_0] \to 1 \text{ if } \|\hat{\beta} - \beta^0\|_1 = O_P(s_0 \sqrt{\log(p)/n})$$

Proof: suppose that $j^* \in S_0$ but $j^* \notin \hat{S}$

$$\|\hat{\beta} - \beta^0\|_1 \geq |\hat{\beta}_{j^*} - \beta_{j^*}^0| = |\beta_{j^*}^0| \gg s_0 \sqrt{\log(p)/n}$$

which is a contradiction $\square$

analogously: if

- beta-min condition $\min_{j \in S_0} |\beta_j^0| \gg \sqrt{s_0 \log(p)/n}$
- $\|\hat{\beta} - \beta^0\|_2 = O_P(\sqrt{s_0 \log(p)/n})$

$\rightsquigarrow \mathbb{P}[\hat{S} \supseteq S_0] \to 1$

Theory versus Practice

theory:

$$\mathbb{P}[\hat{S} \supseteq S_0] \to 1$$

if the following hold:

- compatibility condition for the (fixed) design X
- beta-min condition
- i.i.d. Gaussian errors (can be relaxed)

in addition: $|\hat{S}| \leq \min(n, p)$
hence: huge dimensionality reduction if $p \gg n$

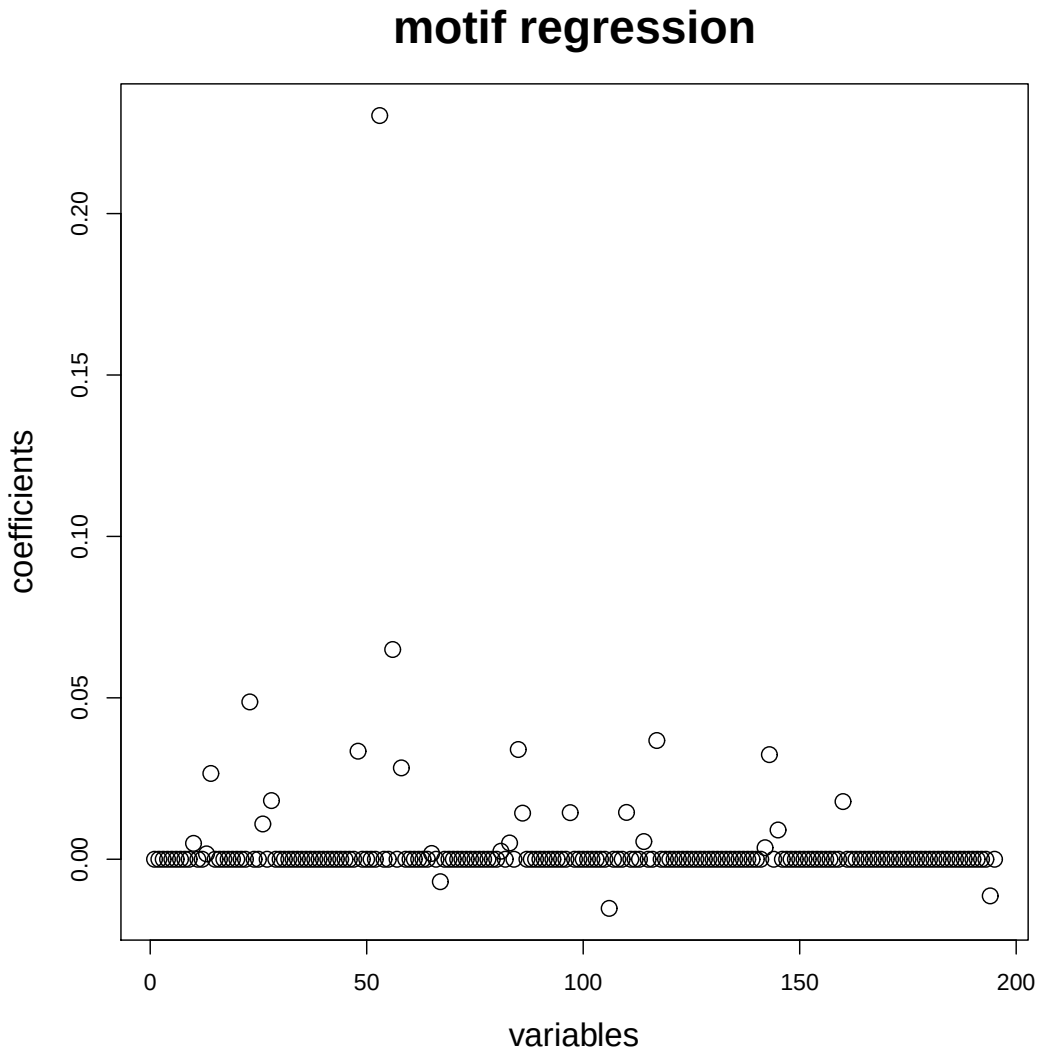
in practice: $\mathbb{P}[\hat{S} \supseteq S_0]$ may not be SO large...
even if one chooses λ very small which results in a typically larger set $\hat{S}$...

possible reasons to explain with theory:

- compatibility constant ϕ_0^2 might be very small (due to highly correlated columns in X or near linear dependence among a few columns of X)
  $\rightsquigarrow \|\hat{\beta} - \beta^0\|_1 \leq 4\lambda s_0/\phi_0^2$
  $\rightsquigarrow$ requires a stronger beta-min condition!
- errors are non-Gaussian (heavy tailed)

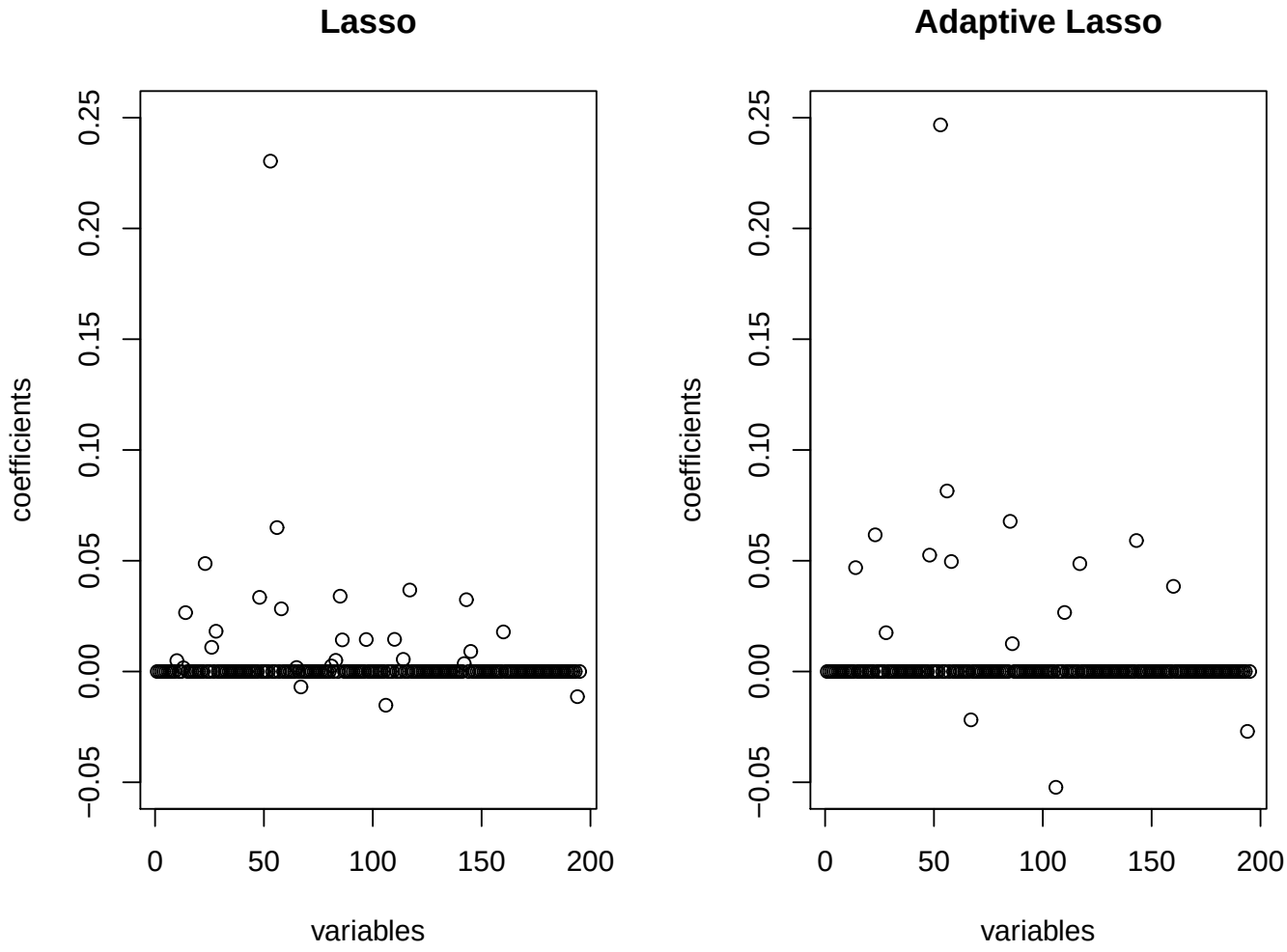
it is "empirically evident" though: $\mathbb{P}[\hat{S} \supseteq S_{\text{substantial}(C)}]$ large
where $S_{\text{substantial}(C)} = \{j;\ |\beta_j^0| \geq \underbrace{C}_{\text{large}}\}$

The Lasso workhorse



$$p = 195, n = 143, |\hat{S}(\lambda_{CV})| = 26$$

The (adaptive) Lasso workhorse



$$p = 195, n = 143, |\hat{S}_{\text{ada}-\text{Lasso}}(\lambda_{CV})| = 16$$

When does the compatibility condition hold?

have seen that the compatibility condition plays a major role for estimating β^0 and for fast convergence rate for prediction

Corollary 6.8 from Bühlmann and van de Geer (2011) – modified form
Assume that the row vectors of X are i.i.d. sampled from a sub-Gaussian distribution with mean zero and covariance matrix Σ. Assume that

- $\lambda^2_{min}(\Sigma) > 0$
- $s_0 = |S_0| = O(\sqrt{n/\log(p)})$

Then: $\phi_0^2 \geq \lambda^2_{\text{min}}(\Sigma) > 0$ with probability $\to 1$ $(n \to \infty)$

Example: Toeplitz matrix $\Sigma_{ij} = \rho^{|i-j|}$ $(0 \leq \rho < 1)$:
$\lambda^2_{min}(\Sigma) \geq L > 0$ where L is independent of p

Variable selection

under more restrictive irrepresentable condition or
neighborhood stability condition on the design X
and assuming beta-min condition $\min_{j \in S_0} |\beta_j^0| \gg \sqrt{s_0 \log(p)/n}$:

$$\mathbb{P}[\hat{S} = S_0] \to 1 \ (n \to \infty)$$

the irrepresentable condition is sufficient and essentially
necessary for consistent variable selection

this condition is often not fulfilled in practice
(and choosing the correct λ would be difficult as well)

$\rightsquigarrow$ variable screening is realistic ("choose λ by CV")
variable selection is not very realistic

better "translation":
LASSO = Least Absolute Shrinkage and Screening Operator

Variable selection

under more restrictive irrepresentable condition or
neighborhood stability condition on the design X
and assuming beta-min condition $\min_{j \in S_0} |\beta_j^0| \gg \sqrt{s_0 \log(p)/n}$:

$$\mathbb{P}[\hat{S} = S_0] \to 1 \ (n \to \infty)$$

the irrepresentable condition is sufficient and essentially
necessary for consistent variable selection

this condition is often not fulfilled in practice
(and choosing the correct λ would be difficult as well)

$\rightsquigarrow$ variable screening is realistic ("choose λ by CV")
variable selection is not very realistic
better "translation":
LASSO = Least Absolute Shrinkage and Screening Operator

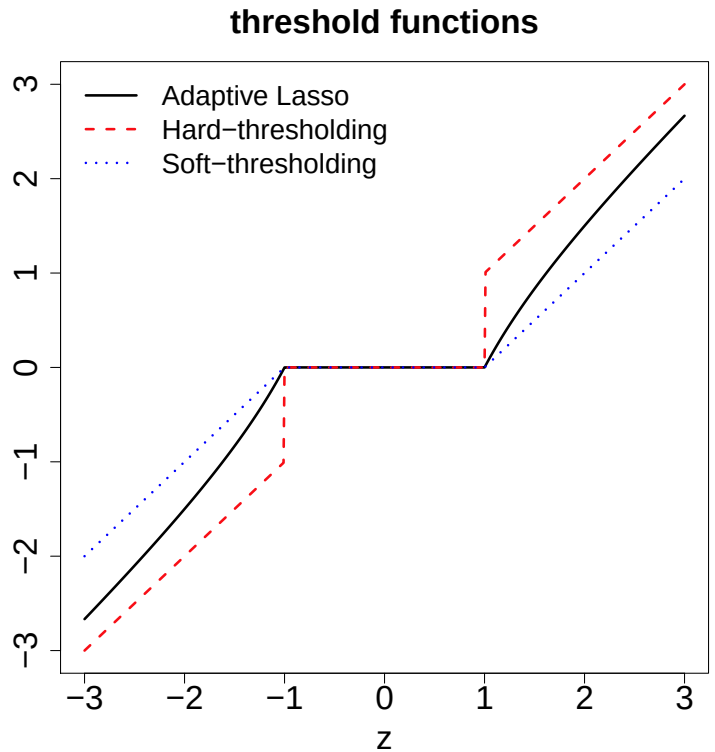
version of Table 2.2 in the book:

property	design condition	size of non-zero coeff.
slow prediction conv. rate	no requirement	no requirement
fast prediction conv. rate	compatibility	no requirement
estimation error bound $\|\hat{\beta} - \beta^0\|_1$	compatibility	no requirement
variable screening	compatibility or restricted eigenvalue	beta-min condition weaker beta-min cond.
variable selection	neighborhood stability $\Leftrightarrow$ irrepresentable cond.	beta-min condition

Adaptive Lasso

is a good way to address the bias problems of the Lasso

for orthonormal design



two-stage procedure:

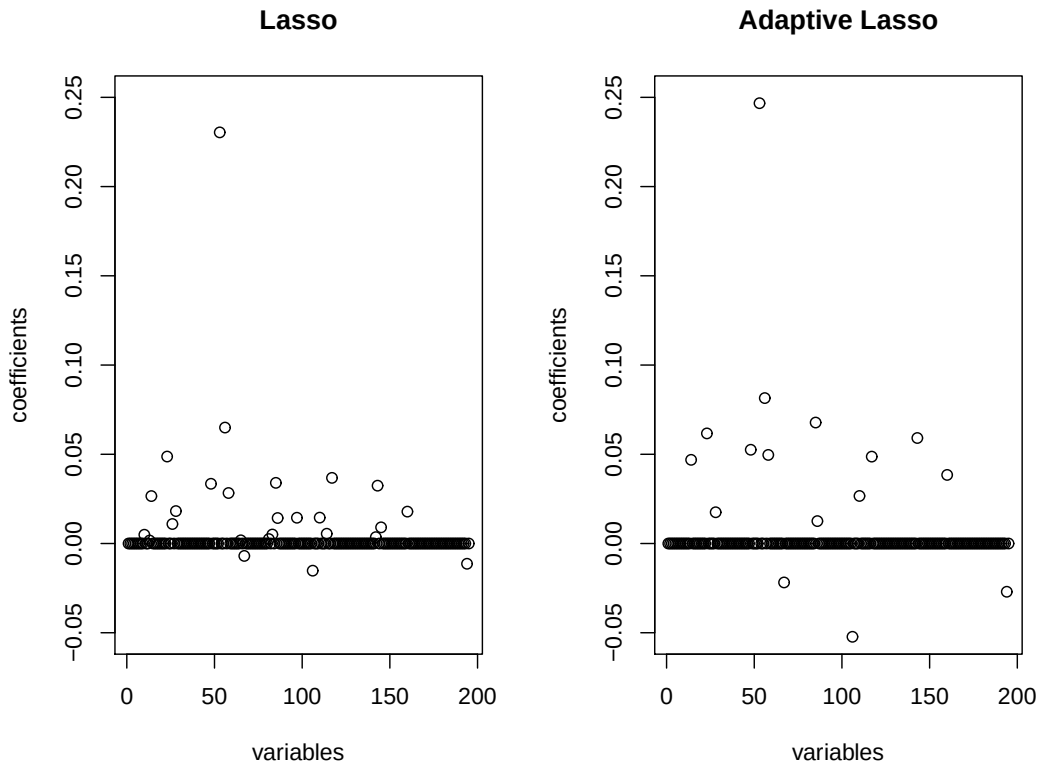
- initial estimator $\hat{\beta}_{\text{init}}$, e.g., the Lasso
- re-weighted ℓ_1-penalty

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

adaptive Lasso often works well in practice (more sparse than Lasso) and has better theoretical properties than Lasso for variable screening (and selection) if the truth is assumed to be sparse

alternatives: thresholding the Lasso; Relaxed Lasso

The adaptive Lasso workhorse



$$p = 195, n = 143, |\hat{S}_{\text{ada}-\text{Lasso}}(\lambda_{CV})| = 16$$

we will discuss later in the course the issue of assigning "significance of selected variables"

should we always use the adaptive Lasso?

- it's slightly more complicated – need two Lasso fits
- the differences in large-scale data are perhaps not so large
- I tend to say:
  "Yes, often the adaptive Lasso is perhaps a bit better"

Computational algorithm for Lasso

can use a very generic coordinate descent algorithm (not gradient descent)

motivation of the algorithm:
consider the objective function and the corresponding Karush-Kuhn-Tucker (KKT) conditions by taking the sub-differential:

$$
\begin{aligned}
& \frac{\partial}{\partial j}(\|Y - X\beta\|_2^2/n + \lambda\|\beta\|_1) \\
= \quad & G_j(\beta) + \lambda e_j, \\
& G(\beta) = -2X^T(Y - X\beta)/n, \\
& e_j = \text{sign}(\beta_j) \text{ if } \beta_j \neq 0, \quad e_j \in [-1, 1] \text{ if } \beta_j = 0
\end{aligned}
$$

this implies (by setting the sub-differential to zero) the KKT-conditions (Lemma 2.1, Bühlmann and van de Geer (2011):

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

an interesting characterization of the Lasso solution!

in abbreviated form:

1: Let $\beta^{[0]} \in \mathbb{R}^p$ be an initial parameter vector. For $m = 1, 2, \ldots$

2: **repeat**

3: Proceed componentwise $j = 1, 2, \ldots, p, 1, 2, \ldots p, 1, 2, \ldots$

update:

if $|G_j(\underbrace{\beta_{-j}^{[m-1]}}_{\text{prev. parameter with } j\text{th comp}=0})| \leq \lambda$: set $\beta_j^{[m]} = 0$,

otherwise: $\beta_j^{[m]}$ is the minimizer of the objective function with respect to the jth component but keeping all others fixed

4: **until** numerical convergence

1: Let $\beta^{[0]} \in \mathbb{R}^p$ be an initial parameter vector. Set $m = 0$.
2: **repeat**
3: Increase m by one: $m \leftarrow m + 1$.
Denote by $\mathcal{S}^{[m]}$ the index cycling through the coordinates $\{1, \ldots, p\}$:
$\mathcal{S}^{[m]} = \mathcal{S}^{[m-1]} + 1 \bmod p$. Abbreviate by $j = \mathcal{S}^{[m]}$ the value of $\mathcal{S}^{[m]}$.
4: if $|G_j(\beta_{-j}^{[m-1]})| \leq \lambda$: set $\beta_j^{[m]} = 0$,
otherwise: $\beta_j^{[m]} = \text{argmin}_{\beta_j} Q_\lambda(\beta_{+j}^{[m-1]})$,
where $\beta_{-j}^{[m-1]}$ is the parameter vector where the jth component is set to zero and $\beta_{+j}^{[m-1]}$ is the parameter vector which equals $\beta^{[m-1]}$ except for the jth component where it is equal to β_j (i.e. the argument we minimize over).
5: **until** numerical convergence

for the squared error loss: the update in Step 4 is explicit (a soft-thresholding operation)

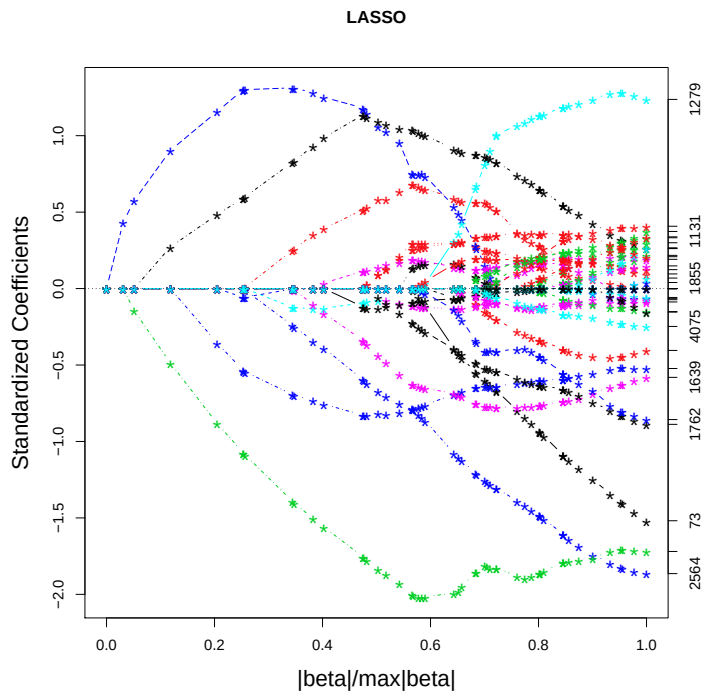
active set strategy can speed up the algorithm for sparse cases: mainly work on the non-zero coordinates and up-date all coordinates e.g. every 20th times

`R`-package `glmnet`

The Lasso regularization path

compute $\hat{\beta}(\lambda)$ over "all" λ

- just a grid of λ-values and interpolate linearly (the true solution path over all λ is piecewise linear)
- for $\lambda_{\max} = |2X^T Y/n|$: $\hat{\beta}(\lambda_{\max}) = 0$
  (because of KKT conditions!)



plot against $\|\hat{\beta}(\lambda)\|_1 / \max_\lambda \|\hat{\beta}(\lambda)\|_1$ (λ small is to the right)

Generalized linear models (GLMs)

univariate response Y, covariate $X \in \mathcal{X} \subseteq \mathbb{R}^p$

GLM: $Y_1, \ldots, Y_n$ independent

$$g(\mathbb{E}[Y_i|X_i = x]) = \underbrace{\mu + \sum_{j=1}^{p} \beta_j x^{(j)}}_{=f(x)=f_{\mu,\beta}(x)}$$

$g(\cdot)$ real-valued, known link function
μ an intercept term: the intercept is important: we cannot simply center the response and ignore an intercept...

Lasso: defined as ℓ_1-norm penalized negative log-likelihood (where μ is not penalized)

software: `glmnet` in `R`

Example: logistic (penalized) regression

$Y \in \{0, 1\}$

$\pi(x) = \mathbb{E}[Y|X = x] = \mathbb{P}[Y = 1|X = x]$

logistic link function: $g(\pi) = \log(\pi/(1-\pi))$ $(\pi \in (0, 1))$

denote by $\pi_i = \mathbb{P}[Y_1 = 1|X_i]$

$\log(\pi_i/(1-\pi_i)) = \exp(\mu + X_i^T\beta)$, $\pi_i = \frac{\exp(\mu+X_i^T\beta)}{1+\exp(\mu+X_i^T\beta)}$

log-likelihood

$$\sum_{i=1}^{n} \log(\pi_i^{Y_i}(1-\pi_i)^{1-Y_i}) = \sum_{i=1}^{n}(Y_i \log(\pi_i) + (1-Y_i)\log(1-\pi_i)$$

$$= \sum_{i=1}^{n}(Y_i \underbrace{\log(\pi_i/(1-\pi_i))}_{\mu+X_i^T\beta} + \underbrace{\log(1-\pi_i)}_{\log(1+\exp(\mu+X_i^T\beta))})$$

negative log-likelihood

$$-\ell(\mu, \beta) = \sum_{i=1}^{n}(-Y_i(\mu + X_i^T\beta) + \log(1 + \exp(\mu + X_i^T\beta)))$$

which is a convex function in μ, β

Lasso for linear logistic regression:

$$\hat{\mu}, \hat{\beta} = \text{argmin}_{\mu,\beta}(-\ell(\mu, \beta) + \lambda\|\beta\|_1)$$

note: often used nowadays for classification with deep neural networks

$$\log(\pi_i/(1-\pi_i)) = \mu + \underbrace{X^T\beta^{(1)}}_{\text{NN with linear connection}} + \beta^{(2)} \underbrace{w_\theta(X)}_{\text{features from last NN layer}}$$

estimator:

$$\hat{\mu}, \hat{\beta}^{(1)}, \hat{\beta}^{(2)}, \hat{\theta} = \text{argmin} - \ell\left(\mu, \beta^{(1)}, \beta^{(2)}, \theta) + \lambda(\|\beta^{(1)}\|_1 + \|\beta^{(2)}\|_1)\right)$$

this is now a highly non-convex function in θ...!

if somebody gives you the feature mapping $w_\theta(\cdot)$ (e.g. trained on large image database), then one can use logistic Lasso