Statistics for high-dimensional data: Group Lasso and additive models

Peter Bühlmann and Sara van de Geer

Seminar für Statistik, ETH Zürich

May 2012

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

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

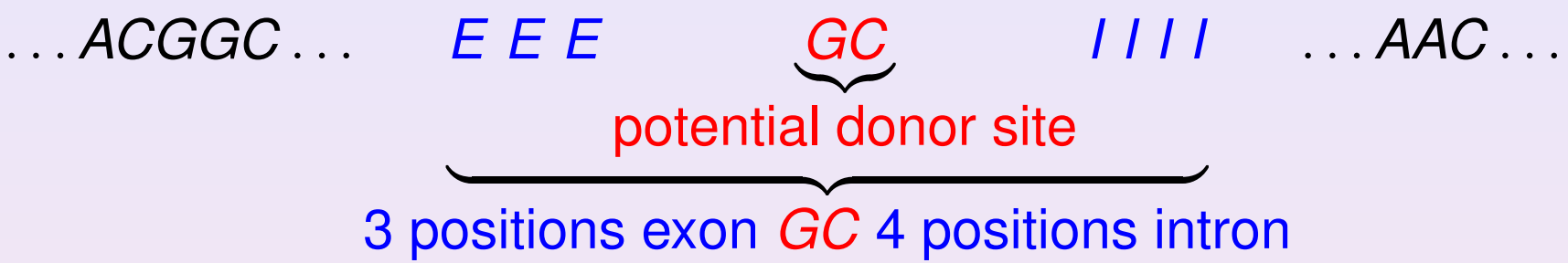
typically $m_g = \sqrt{|\mathcal{G}_g|}$

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

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:

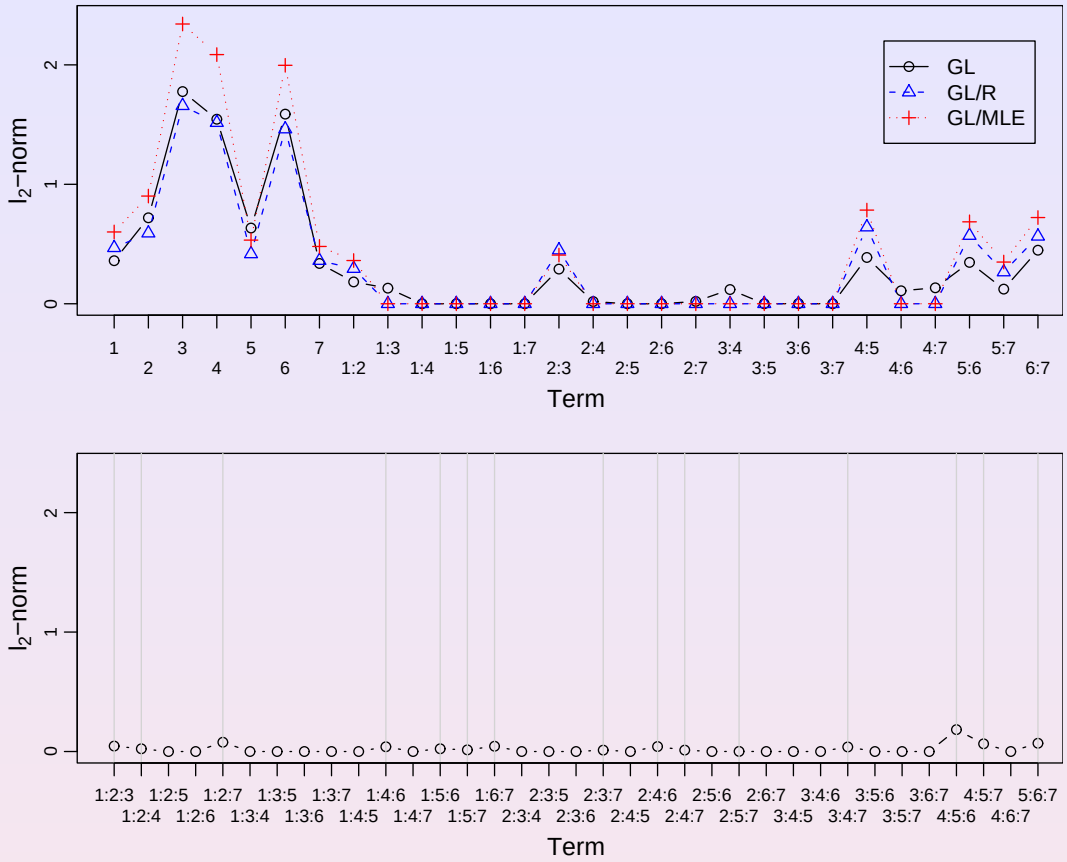
training:	5′610 true splice sites
	5′610 non-splice sites
	plus an unbalanced validation set
test data:	4′208 true splice sites
	89′717 non-splice sites

logistic regression:

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

up to second oreder interactions: 1156 parameters

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

Generalized group Lasso penalty

$$\lambda \sum_{j=1}^{q} m_j \sqrt{\beta_{\mathcal{G}_j}^T A_j \beta_{\mathcal{G}_j}},$$

where A_j are $T_j \times T_j$ positive definite matrices

$\rightsquigarrow$ generalized group Lasso:

$$\hat{\beta} = \text{argmin}_\beta (\|\mathbf{Y} - \mathbf{X}\beta\|_2^2/n + \lambda \sum_{j=1}^{q} m_j \sqrt{\beta_{\mathcal{G}_j}^T A_j \beta_{\mathcal{G}_j}})$$

reparameterize

$$\tilde{\beta}_{\mathcal{G}_j} = A_j^{1/2} \beta_{\mathcal{G}_j},$$
$$\tilde{\mathbf{X}}_{\mathcal{G}_j} = \mathbf{X}_{\mathcal{G}_j} A_j^{-1/2}$$

$$\hat{\beta} = \text{argmin}_\beta (\|\mathbf{Y} - \mathbf{X}\beta\|_2^2/n + \lambda \sum_{j=1}^{q} m_j \sqrt{\beta_{\mathcal{G}_j}^T A_j \beta_{\mathcal{G}_j}})$$

can be derived from

$$\hat{\beta}_{\mathcal{G}_j} = A_j^{-1/2} \hat{\tilde{\beta}}_{\mathcal{G}_j},$$
$$\hat{\tilde{\beta}} = \text{argmin}_{\tilde{\beta}} (\|\mathbf{Y} - \tilde{\mathbf{X}}\tilde{\beta}\|_2^2/n + \lambda \sum_{j=1}^{q} m_j \|\tilde{\beta}_{\mathcal{G}_j}\|_2)$$

Groupwise prediction penalty and parameterization invariance

$$\lambda \sum_{j=1}^{q} m_j \|\mathbf{X}_{\mathcal{G}_j} \beta_{\mathcal{G}_j}\|_2 = \lambda \sum_{j=1}^{q} m_j \sqrt{\beta_{\mathcal{G}_j}^T \mathbf{X}_{\mathcal{G}_j}^T \mathbf{X}_{\mathcal{G}_j} \beta_{\mathcal{G}_j}}$$

is a generalized group Lasso penalty if $\mathbf{X}_{\mathcal{G}_j}^T \mathbf{X}_{\mathcal{G}_j}$ are positive definite (i.e. necessarily $|\mathcal{G}_j| \leq n$)

this penalty is invariant under any (invertible) transformations within groups
i.e. can use $\tilde{\mathcal{G}}_j = B_j \beta_{\mathcal{G}_j}$ where B_j is any $T_j \times T_j$ invertible matrix
$\rightsquigarrow$

$$\mathbf{X}_{\mathcal{G}_j} \hat{\beta}_{\mathcal{G}_j} = \tilde{\mathbf{X}}_{\mathcal{G}_j} \hat{\tilde{\beta}}_{\mathcal{G}_j},$$
$$\{j;\ \hat{\beta}_{\mathcal{G}_j} \neq 0\} = \{j;\ \hat{\tilde{\beta}}_{\mathcal{G}_j} \neq 0\}$$

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.

but listen to Sara
in $\approx$ "a few" minutes

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

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

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

KKT conditions:

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

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

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. $p \approx 10^6 - 10^7$

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

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

$\rightsquigarrow$ represent

$$\sum_{j=1}^{p} f_j(x^{(j)}) = \sum_{j=1}^{p} \sum_{k=1}^{K} \beta_{j,k} h_{j,k}(x^{(j)})$$

$\rightsquigarrow$ high-dimensional parametric problem

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

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

drawback:
does not exploit smoothness
(except when choosing appropriate K which is "bad" if different f_j's have different complexity)

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

Sparsity-Smoothness Penalty (SSP)

$$\lambda_1 \sum_{j=1}^{p} \underbrace{\|f_j\|_n}_{\|H_j\beta_j\|_2/\sqrt{n}} + \lambda_2 \sum_{j=1}^{p} I(f_j)$$

$$I^2(f_j) = \int (f_j''(x))^2 dx = \beta_j^T W_j \beta_j$$

where $f_j = (f_j(X_1^{(j)}), \ldots, f_j(X_n^{(j)}))^T$,
and $W_j = \int h_{j,k}''(x) h_{j,\ell}''(x) dx$

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

Orthogonal basis and diagonal smoothing matrices

$n^{-1} H_j^T H_j = I$ and
$W_j \equiv \text{diag}(d_1^2, \ldots, d_K^2) := D^2,\ d_k = k^m\ (m > 1/2)$
then, the penalty becomes

$$\lambda_1 \sum_{j=1}^{p} \|\beta_j\|_2 + \lambda_2 \sum_{j=1}^{p} \|D\beta_j\|_2$$

$\rightsquigarrow$

$$\hat{\beta}(\lambda_1, \lambda_2) = \text{argmin}_\beta \|\mathbf{Y} - \sum_{j=1}^{p} H_j \beta_j\|_2^2 / n + \lambda_1 \sum_{j=1}^{p} \|\beta_j\|_2 + \lambda_2 \sum_{j=1}^{p} \|D\beta_j\|_2$$

the difficulty is the computation, although still a convex optimization problem

see Section 5.3.3 in the book

A modified SSP-penalty

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

for additive modeling:

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

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:

$$f = \sum_{j=1}^{p} f_j, \quad f_j = H_j \beta_j$$

penalty becomes:

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

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

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

$\rightsquigarrow$ 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

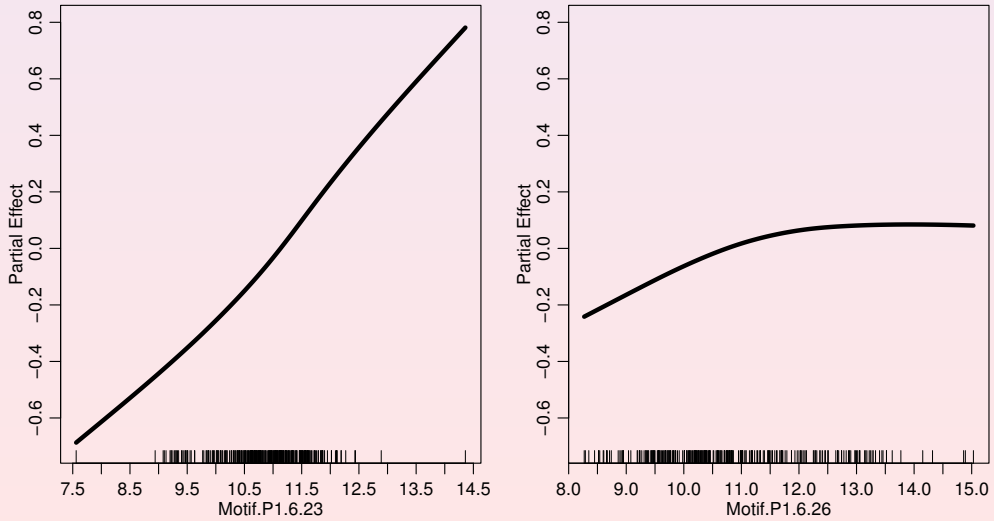
for finding HIF1α transcription factor binding sites on DNA sequences

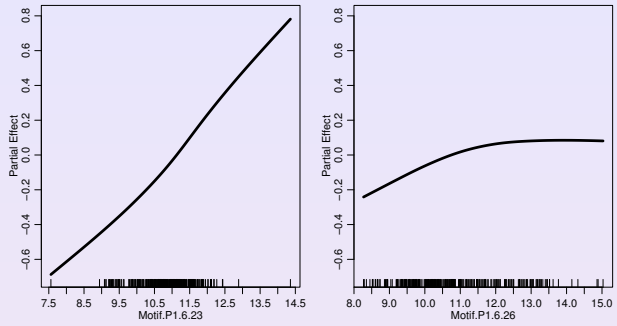
$n = 287$, $p = 196$

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, ...)