SD-TSIA204: Lasso

Joseph Salmon
http://josephsalmon.eu
Télécom Paristech, Institut Mines-Télécom

Outline

Table of Contents

Reminding the model

$$\mathbf{y} = X\boldsymbol{\theta}^\star + \boldsymbol{\varepsilon} \in \mathbb{R}^n$$

$$X = [\mathbf{x}_1, \ldots, \mathbf{x}_p] = \begin{pmatrix} x_{1,1} & \ldots & x_{1,p} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \ldots & x_{n,p} \end{pmatrix} \in \mathbb{R}^{n \times p}, \boldsymbol{\theta}^\star \in \mathbb{R}^p$$

Table of Contents

Motivation

Estimators $\hat{\boldsymbol{\theta}}$ with many zero coefficients are useful:

- for interpretation
- for computational efficiency if p is huge

Underlying idea: **variable selection**

Rem: also useful if $\boldsymbol{\theta}^\star$ has few non-zero coefficients

Variable selection overview

- **Screening**: remove the $\mathbf{x}_j$'s whose correlation with $\mathbf{y}$ is weak
  - pros: fast $(+++)$, *i.e.*,one pass over data, intuitive $(+++)$
  - cons: neglect variables interactions $\mathbf{x}_j$, weak theory (- - -)
- **Greedy** methods aka stagewise / stepwise
  - pros: fast $(++)$, intuitive $(++)$
  - cons: propagates wrong selection forward; weak theory (-)
- Sparsity enforcing **penalized** methods (*e.g.*,Lasso)
  - pros: better theory for convex cases $(++)$
  - cons: can be still slow (-)

The ℓ_0 pseudo-norm

Definition

The **support** of $\boldsymbol{\theta} \in \mathbb{R}^p$ is the set of indexes of non-zero coordinates:

$$\operatorname{supp}(\boldsymbol{\theta}) = \{j \in [\![1, p]\!], \theta_j \neq 0\}$$

The ℓ_0 **pseudo-norm** of a $\boldsymbol{\theta} \in \mathbb{R}^p$ is the number of non-zero coordinates:

$$\|\boldsymbol{\theta}\|_0 = \operatorname{card}\{j \in [\![1, p]\!], \theta_j \neq 0\}$$

Rem: $\|\cdot\|_0$ is not a norm, $\forall t \in \mathbb{R}^*, \|t\boldsymbol{\theta}\|_0 = \|\boldsymbol{\theta}\|_0$

Rem: $\|\cdot\|_0$ it is not even convex, $\boldsymbol{\theta}_1 = (1, 0, 1, \ldots, 0)$ $\boldsymbol{\theta}_2 = (0, 1, 1, \ldots, 0)$ and $3 = \|\frac{\boldsymbol{\theta}_1 + \boldsymbol{\theta}_2}{2}\|_0 \geqslant \frac{\|\boldsymbol{\theta}_1\|_0 + \|\boldsymbol{\theta}_2\|_0}{2} = 2$

The ℓ_0 penalty

First try to get a sparsity enforcing penalty: use ℓ_0 as a penalty (or regularization)

$$\boxed{\hat{\boldsymbol{\theta}}_\lambda = \mathop{\arg\min}_{\boldsymbol{\theta}\in\mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}_{\text{data fitting}} + \underbrace{\lambda\|\boldsymbol{\theta}\|_0}_{\text{regularization}} \right)}$$

<u>**Combinatorial problem**</u>!!!

Exact solution: require considering all sub-models, *i.e.*,computing OLS for all possible support; meaning one might need 2^p least squares computation!

<u>Example</u> :
$p = 10$ possible: $\approx 10^3$ least squares
$p = 30$ impossible: $\approx 10^{10}$ least squares

<u>Rem</u>: problem "NP-hard", can be solved for small problems by mixed integer programming.

Le Lasso: penalty point of view

Lasso: *Least Absolute Shrinkage and Selection Operator* Tibshirani (1996)

$$\boxed{\hat{\boldsymbol{\theta}}_\lambda^{\text{Lasso}} = \mathop{\arg\min}_{\boldsymbol{\theta} \in \mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}_{\textbf{data fitting}} + \underbrace{\lambda\|\boldsymbol{\theta}\|_1}_{\textbf{regularization}} \right)}$$

où $\|\boldsymbol{\theta}\|_1 = \sum_{j=1}^{p} |\theta_j|$ sum of absolute values of the coefficients)

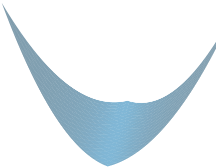
- We recover the limiting cases:

$$\lim_{\lambda \to 0} \hat{\boldsymbol{\theta}}_\lambda^{\text{Lasso}} = \hat{\boldsymbol{\theta}}^{\text{OLS}}$$

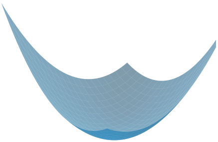
$$\lim_{\lambda \to +\infty} \hat{\boldsymbol{\theta}}_\lambda^{\text{Lasso}} = 0 \in \mathbb{R}^p$$

Beware: the Lasso estimator is not always **unique** for a fixed λ (consider cases with two equals columns in X)

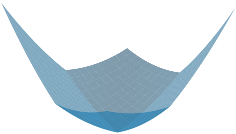
Optimization in $\mathbb{R}^d$



OLS

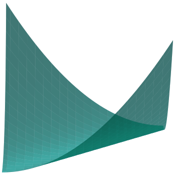


Ridge

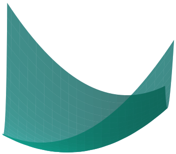


Lasso

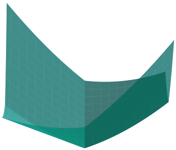
Optimization in $\mathbb{R}^d$



OLS

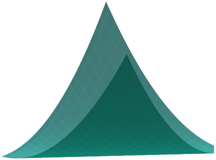


Ridge

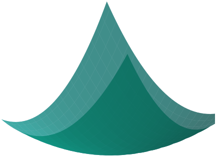


Lasso

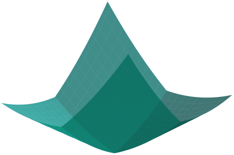
Optimization in $\mathbb{R}^d$



OLS

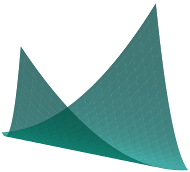


Ridge

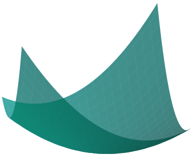


Lasso

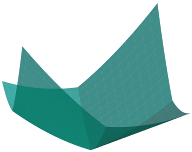
Optimization in $\mathbb{R}^d$



OLS

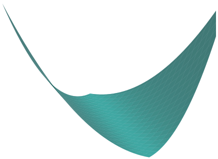


Ridge

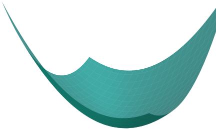


Lasso

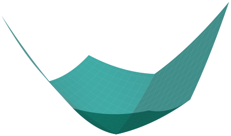
Optimization in $\mathbb{R}^d$



OLS



Ridge



Lasso

Constraint point of view

The following problem:

$$\boxed{\hat{\boldsymbol{\theta}}_\lambda^{\text{Lasso}} = \mathop{\arg\min}_{\boldsymbol{\theta}\in\mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}_{\text{data fitting}} + \underbrace{\lambda\|\boldsymbol{\theta}\|_1}_{\text{regularization}} \right)}$$

shares the same solutions as the constrained formulation:

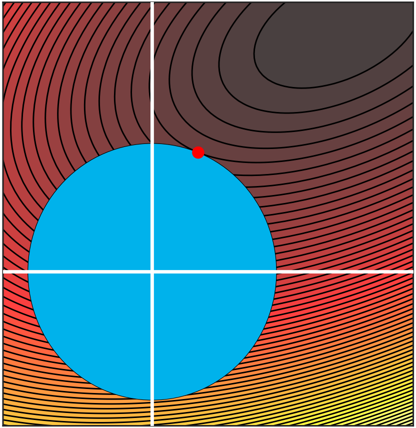
$$\begin{cases} \mathop{\arg\min}_{\boldsymbol{\theta}\in\mathbb{R}^p} \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2 \\ \text{s.t. } \|\boldsymbol{\theta}\|_1 \leqslant T \end{cases}$$

for some $T > 0$.

Rem: unfortunately the link $T \leftrightarrow \lambda$ is not explicit

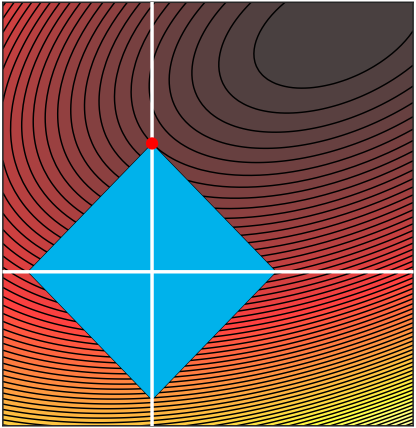
- If $T \to 0$ one recovers the null vector: $0 \in \mathbb{R}^p$
- If $T \to \infty$ one recovers $\hat{\boldsymbol{\theta}}^{\text{OLS}}$ (unconstrained)

Zeroing coefficients



Optimization under ℓ_2 constraint : non sparse solution

Zeroing coefficients



Optimization under ℓ_1 constraint : sparse solution

Sub-gradients / sub-differential

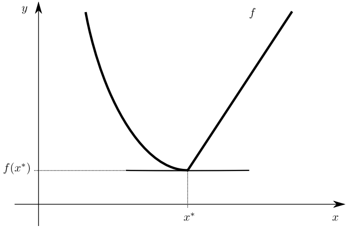
Definitions

For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, $u \in \mathbb{R}^n$ is a **sub-gradient** of f at x^*, if for any $x \in \mathbb{R}^n$,

$$f(x) \geqslant f(x^*) + \langle u, x - x^* \rangle$$

The **sub-differential** is the set
$\partial f(x^*) = \{u \in \mathbb{R}^n : \forall x \in \mathbb{R}^n, f(x) \geqslant f(x^*) + \langle u, x - x^* \rangle\}$.

Rem: if the sub-gradient is unique, one recovers the standard gradient



Sub-gradients / sub-differential

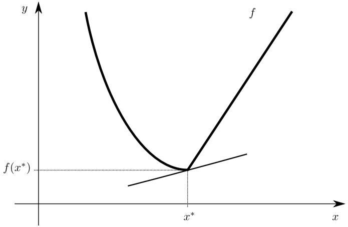
Definitions

For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, $u \in \mathbb{R}^n$ is a **sub-gradient** of f at x^*, if for any $x \in \mathbb{R}^n$,

$$f(x) \geqslant f(x^*) + \langle u, x - x^* \rangle$$

The **sub-differential** is the set
$\partial f(x^*) = \{u \in \mathbb{R}^n : \forall x \in \mathbb{R}^n, f(x) \geqslant f(x^*) + \langle u, x - x^* \rangle\}$.

Rem: if the sub-gradient is unique, one recovers the standard gradient



Sub-gradients / sub-differential

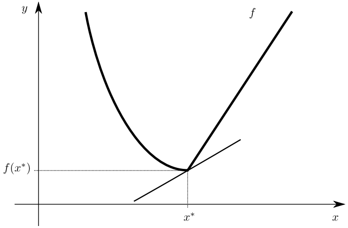
Definitions

For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, $u \in \mathbb{R}^n$ is a **sub-gradient** of f at x^*, if for any $x \in \mathbb{R}^n$,

$$f(x) \geqslant f(x^*) + \langle u, x - x^* \rangle$$

The **sub-differential** is the set
$\partial f(x^*) = \{u \in \mathbb{R}^n : \forall x \in \mathbb{R}^n, f(x) \geqslant f(x^*) + \langle u, x - x^* \rangle\}$.

Rem: if the sub-gradient is unique, one recovers the standard gradient



Sub-gradients / sub-differential

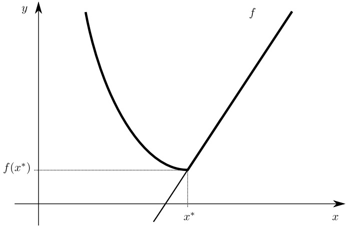
Definitions

For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, $u \in \mathbb{R}^n$ is a **sub-gradient** of f at x^*, if for any $x \in \mathbb{R}^n$,

$$f(x) \geqslant f(x^*) + \langle u, x - x^* \rangle$$

The **sub-differential** is the set
$\partial f(x^*) = \{u \in \mathbb{R}^n : \forall x \in \mathbb{R}^n, f(x) \geqslant f(x^*) + \langle u, x - x^* \rangle\}$.

Rem: if the sub-gradient is unique, one recovers the standard gradient



Fermat's Rule

Theorem

A point x^* is a minimum of a convex function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ if and only if $0 \in \partial f(x^*)$

Proof: use the sub-gradient definition:

- 0 is a sub-gradient of f at x^* if and only if $\forall x \in \mathbb{R}^n, f(x) \geqslant f(x^*) + \langle 0, x - x^* \rangle$

Fermat's Rule

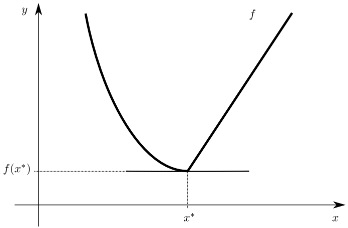
Theorem

A point x^* is a minimum of a convex function $f : \mathbb{R}^n \to \mathbb{R}$ if and only if $0 \in \partial f(x^*)$

Proof: use the sub-gradient definition:

- 0 is a sub-gradient of f at x^* if and only if $\forall x \in \mathbb{R}^n, f(x) \geqslant f(x^*) + \langle 0, x - x^* \rangle$

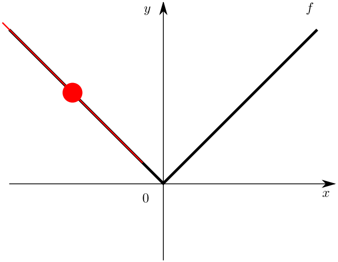
Rem:Visually, it corresponds to a horizontal tangent



Absolute value sub-differential

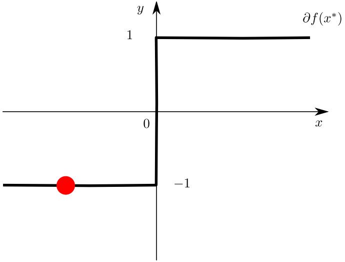
Function (abs):

$$f : \begin{cases} \mathbb{R} & \to \mathbb{R} \\ x & \mapsto |x| \end{cases}$$



Sub-differential (sign)

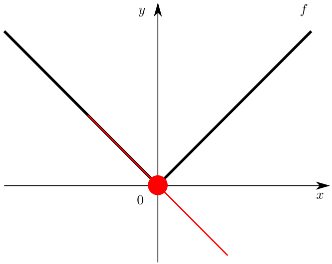
$$\partial f(x^*) = \begin{cases} \{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0 \end{cases}$$



Absolute value sub-differential

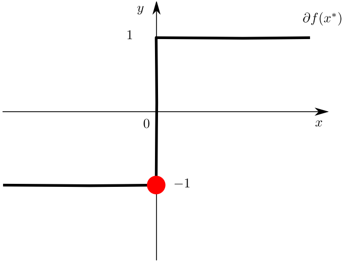
Function (abs):

$$f : \begin{cases} \mathbb{R} & \to \mathbb{R} \\ x & \mapsto |x| \end{cases}$$



Sub-differential (sign)

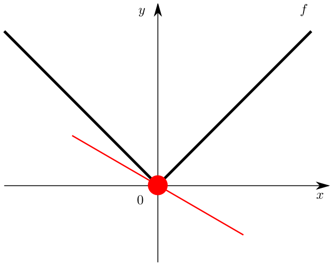
$$\partial f(x^*) = \begin{cases} \{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0 \end{cases}$$



Absolute value sub-differential

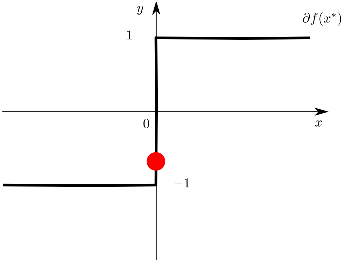
Function (abs):

$$f : \begin{cases} \mathbb{R} & \to \mathbb{R} \\ x & \mapsto |x| \end{cases}$$



Sub-differential (sign)

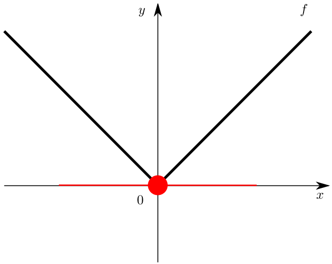
$$\partial f(x^*) = \begin{cases} \{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0 \end{cases}$$



Absolute value sub-differential

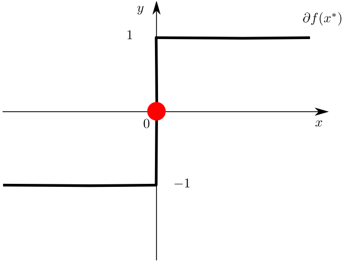
Function (abs):

$$f : \begin{cases} \mathbb{R} & \to \mathbb{R} \\ x & \mapsto |x| \end{cases}$$



Sub-differential (sign)

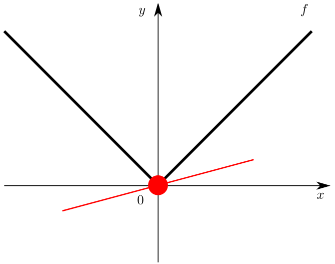
$$\partial f(x^*) = \begin{cases} \{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0 \end{cases}$$



Absolute value sub-differential

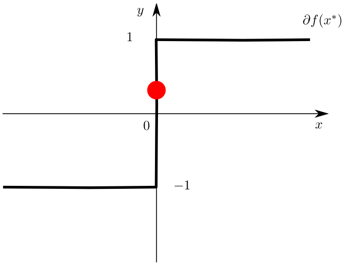
Function (abs):

$$f : \begin{cases} \mathbb{R} & \to \mathbb{R} \\ x & \mapsto |x| \end{cases}$$



Sub-differential (sign)

$$\partial f(x^*) = \begin{cases} \{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0 \end{cases}$$



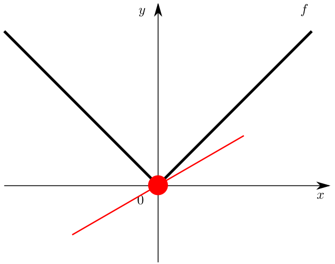
Absolute value sub-differential

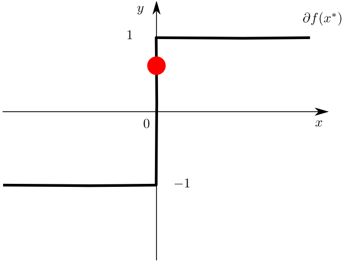
Function (abs):

$$f : \begin{cases} \mathbb{R} & \to \mathbb{R} \\ x & \mapsto |x| \end{cases}$$

Sub-differential (sign)

$$\partial f(x^*) = \begin{cases} \{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0 \end{cases}$$

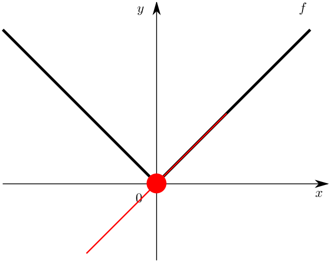




Absolute value sub-differential

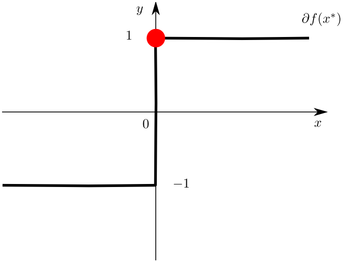
Function (abs):

$$f : \begin{cases} \mathbb{R} & \to \mathbb{R} \\ x & \mapsto |x| \end{cases}$$



Sub-differential (sign)

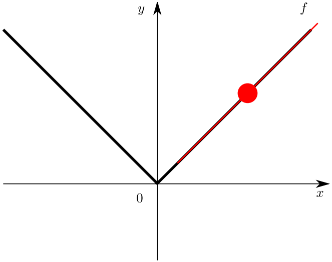
$$\partial f(x^*) = \begin{cases} \{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0 \end{cases}$$



Absolute value sub-differential

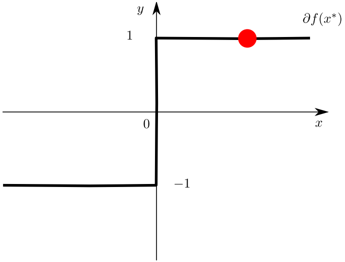
Function (abs):

$$f:\begin{cases}\mathbb{R} & \to \mathbb{R}\\ x & \mapsto |x|\end{cases}$$



Sub-differential (sign)

$$\partial f(x^*) = \begin{cases}\{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0\end{cases}$$



Fermat's rule for the Lasso

$$\boxed{\hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Lasso}} = \mathop{\arg\min}_{\boldsymbol{\theta}\in\mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}_{\text{data fitting}} + \underbrace{\lambda\|\boldsymbol{\theta}\|_1}_{\text{regularization}} \right)}$$

Necessary and sufficient optimality (Fermat):

$$\boxed{\forall j \in [p],\ \mathbf{x}_j^\top \left(\frac{y - X\hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Lasso}}}{\lambda} \right) \in \begin{cases} \{\mathrm{sign}(\hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Lasso}})_j\} & \text{if} \quad (\hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Lasso}})_j \neq 0, \\ [-1, 1] & \text{if} \quad (\hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Lasso}})_j = 0. \end{cases}}$$

Rem: If $\lambda > \lambda_{\max} := \max_{j \in [\![1,p]\!]} |\langle \mathbf{x}_j, \mathbf{y} \rangle|$, then $\hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Lasso}} = 0$

Orthogonal case: soft thresholding

Orthogonal design case: $X^\top X = \mathrm{Id}_p$ (X is an isometry)
$$\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2 = \|X^\top \mathbf{y} - X^\top X\boldsymbol{\theta}\|_2^2 = \|X^\top \mathbf{y} - \boldsymbol{\theta}\|_2^2$$

Lasso objective reformulation:
$$\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2 + \lambda\|\boldsymbol{\theta}\|_1 = \sum_{j=1}^{p}\left(\frac{1}{2}(\mathbf{x}_j^\top \mathbf{y} - \theta_j)^2 + \lambda|\theta_j|\right)$$

Separable problem: problem that can be reduced to minimizing coordinate by coordinate (independently)

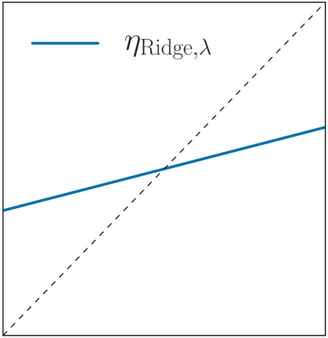
One needs to minimize: $x \mapsto \frac{1}{2}(z - x)^2 + \lambda|x|$ for $z = \mathbf{x}_j^\top \mathbf{y}$

Rem: this function is called the **proximal operator** at z of the function $x \mapsto \lambda|x|$
cf. Parikh and Boyd (2013), for more details on proximal methods

1D Regularization: Ridge

Solve: $\eta_\lambda(z) = \underset{x\in\mathbb{R}}{\arg\min}\, x \mapsto \frac{1}{2}(z-x)^2 + \frac{\lambda}{2}x^2$

$$\eta_\lambda(z) = \frac{z}{1+\lambda}$$

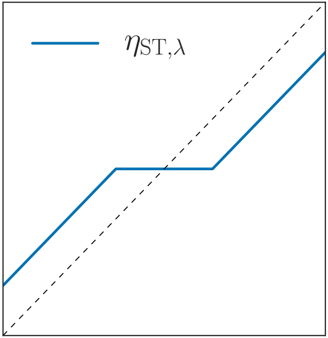


ℓ_2 shrinkage : Ridge

1D Regularization: Lasso

Solve: $\eta_\lambda(z) = \arg\min_{x\in\mathbb{R}} x \mapsto \frac{1}{2}(z-x)^2 + \lambda|x|$

$\eta_\lambda(z) = \text{sign}(z)(|z|-\lambda)_+$ (Exercise)

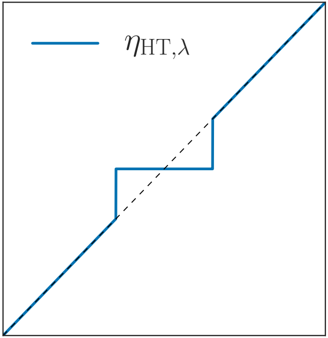


ℓ_1 shrinkage: soft thresholding

1D Regularization: ℓ_0

Solve: $\eta_\lambda(z) = \arg\min_{x \in \mathbb{R}} x \mapsto \frac{1}{2}(z-x)^2 + \lambda \mathbb{1}_{x \neq 0}$

$$\eta_\lambda(z) = z \mathbb{1}_{|z| \geqslant \sqrt{2\lambda}}$$

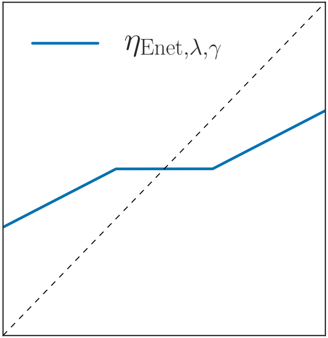


ℓ_0 shrinkage: hard thresholding

1D Regularization: enet

Solve: $\eta_\lambda(z) = \underset{x\in\mathbb{R}}{\arg\min}\, x \mapsto \frac{1}{2}(z-x)^2 + \lambda(\gamma|x| + (1-\gamma)\frac{x^2}{2})$

$$\eta_\lambda(z) = \text{Exercise}$$



ℓ_1/ℓ_2

Soft thresholding: closed form solution

$$\eta_{\mathrm{Lasso},\lambda}(z) = \begin{cases} z + \lambda & \text{if } \ z \ < -\lambda \\ 0 & \text{if } |z| \leqslant \lambda \\ z - \lambda & \text{if } \ z \ > \lambda \end{cases}$$

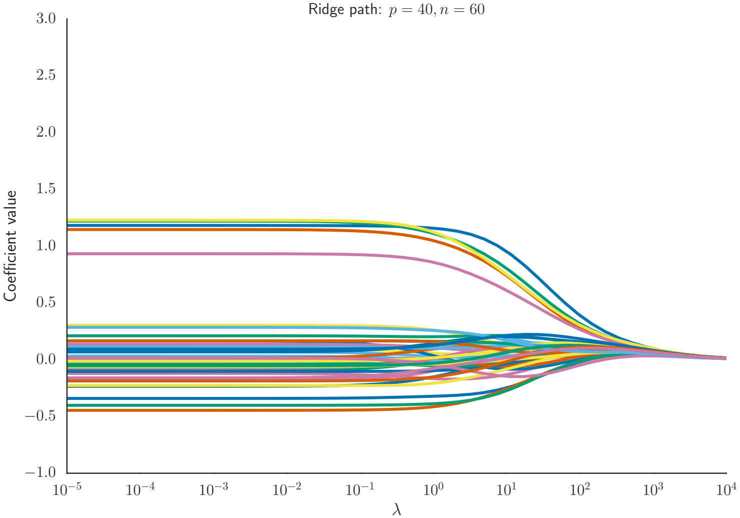
Exo: Use sub-gradients to prove the previous result

Numerical example on simulated data

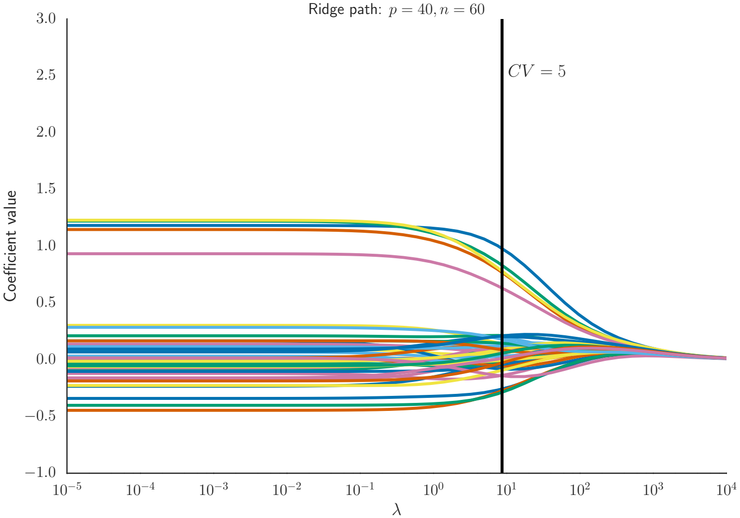
- $\boldsymbol{\theta}^\star = (1, 1, 1, 1, 1, 0, \ldots, 0) \in \mathbb{R}^p$ (5 non-zero coefficients)
- $X \in \mathbb{R}^{n \times p}$ has columns drawn according to a Gaussian distribution
- $y = X\boldsymbol{\theta}^\star + \boldsymbol{\varepsilon} \in \mathbb{R}^n$ with $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 \operatorname{Id}_n)$
- We use a grid of 50 λ values

For this example : $n = 60, p = 40, \sigma = 1$

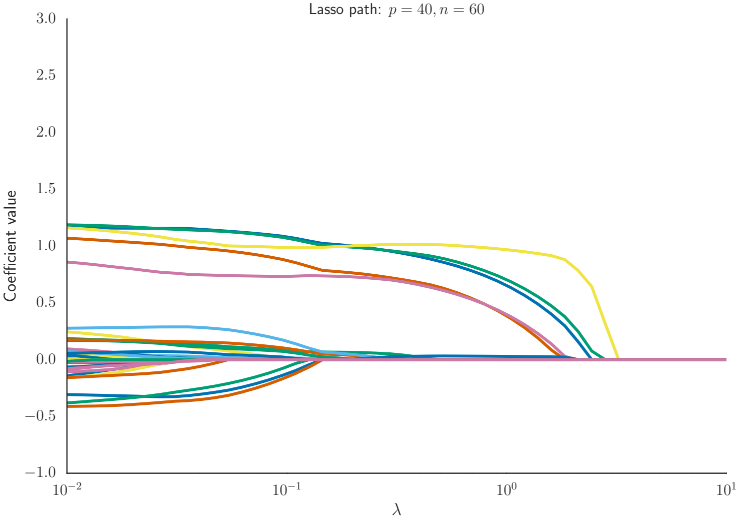
Lasso vs Ridge



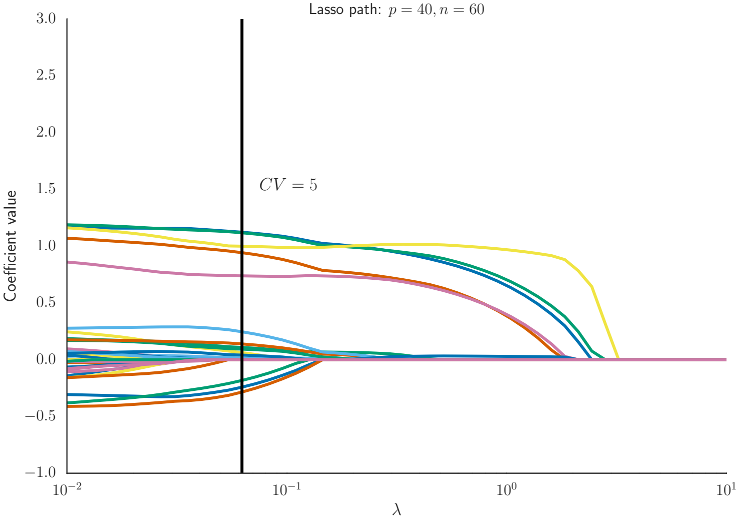
Lasso vs Ridge



Lasso vs Ridge



Lasso vs Ridge



Lasso properties

- Numerical aspect: the Lasso is a **convex** problem
- Variable selection / sparse solutions: $\hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Lasso}}$ has potentially many zeroed coefficients. The λ parameter controls the sparsity level: if λ is large, solutions are very sparse.

  Example : We got 17 non-zero coefficients for `LassoCV` in the previous simulated example

  Rem: RidgeCV has no zero coefficients

Lasso analysis

Theory : more involved for the Lasso than for least squares / Ridge
Recent reference : Bühlmann and van de Geer (2011)

In a nutshell: add bias to the standard least squares to perform variance reduction

Table of Contents

Elastic-net : ℓ_1/ℓ_2 regularization

The Elastic-Net, introduced by Zou and Hastie (2005) is the (unique) solution of

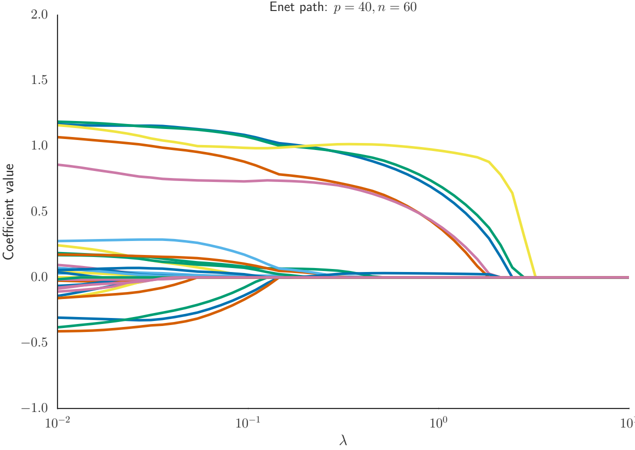
$$\hat{\boldsymbol{\theta}}_\lambda = \underset{\boldsymbol{\theta}\in\mathbb{R}^p}{\arg\min}\left[\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2 + \lambda\left(\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\frac{\|\boldsymbol{\theta}\|_2^2}{2}\right)\right]$$

Motivation: help selecting all relevant but correlated variable (not only one as for the Lasso)

Rem: two parameters needed, one for global regularization, one trading-off Ridge vs. Lasso

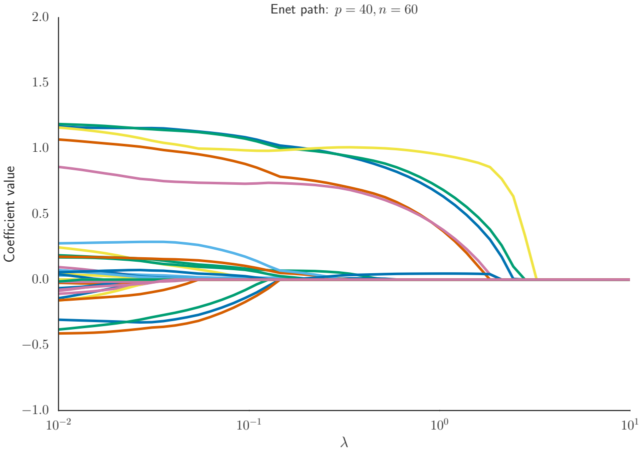
Rem: the solution is unique and the size of the Elastic-Net support is smaller than $\min(n, p)$

Elastic-Net: $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



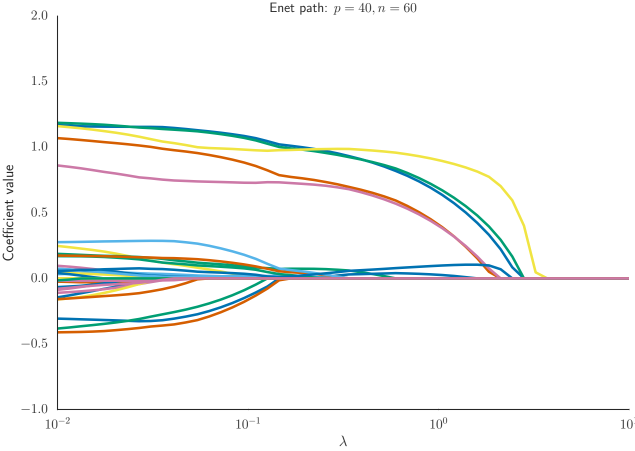
$$\gamma = 1.00$$

Elastic-Net: $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



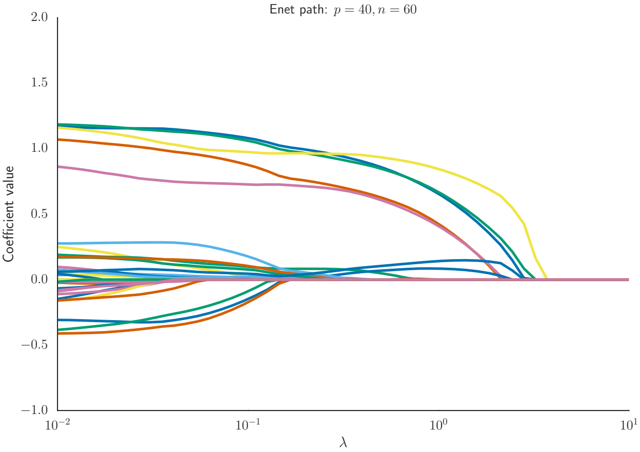
$$\gamma = 0.99$$

Elastic-Net: $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



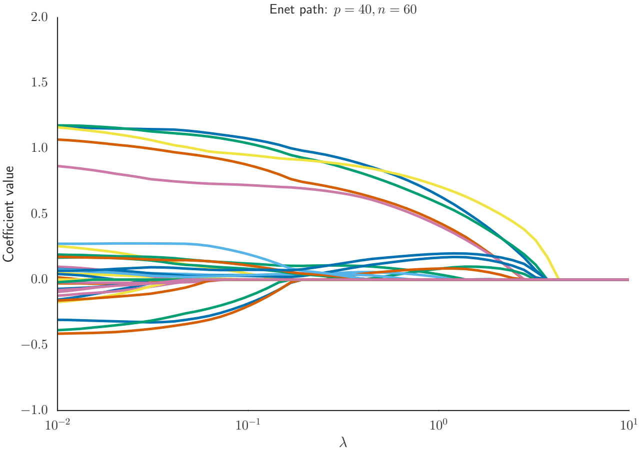
$$\gamma = 0.95$$

Elastic-Net: $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



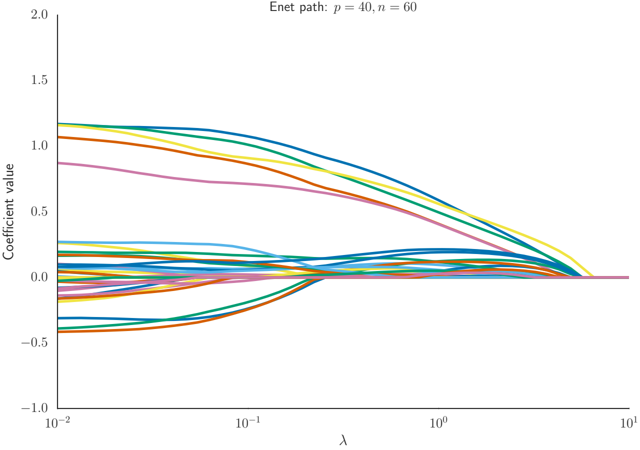
$$\gamma = 0.90$$

Elastic-Net: $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



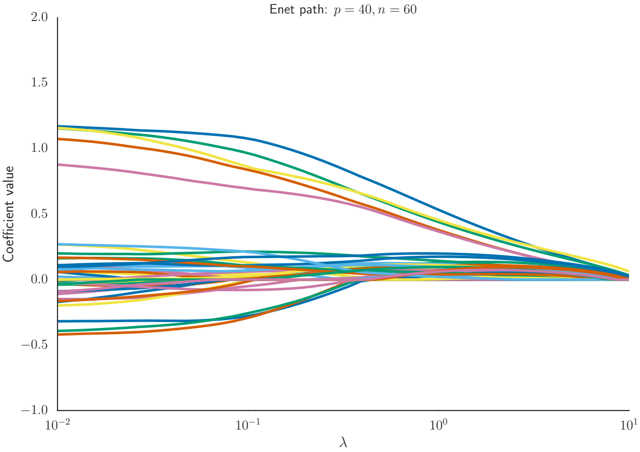
$$\gamma = 0.75$$

Elastic-Net: $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



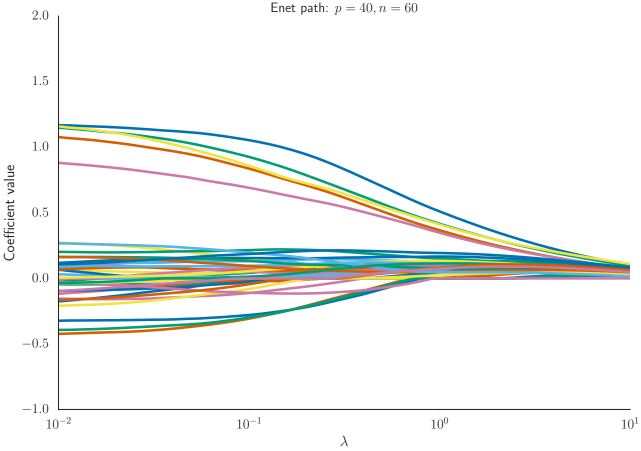
$$\gamma = 0.50$$

Elastic-Net: $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



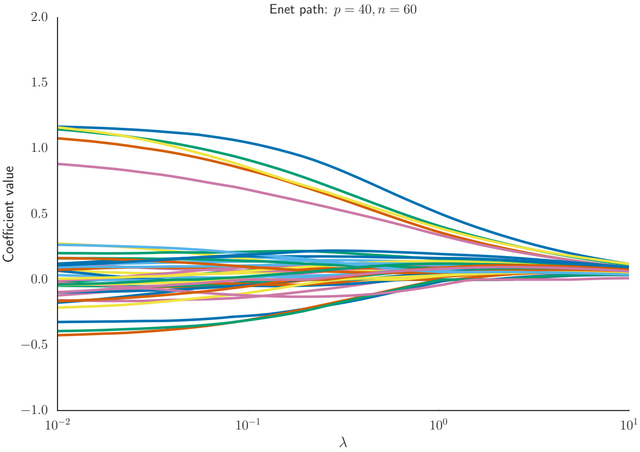
$$\gamma = 0.25$$

Elastic-Net: $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



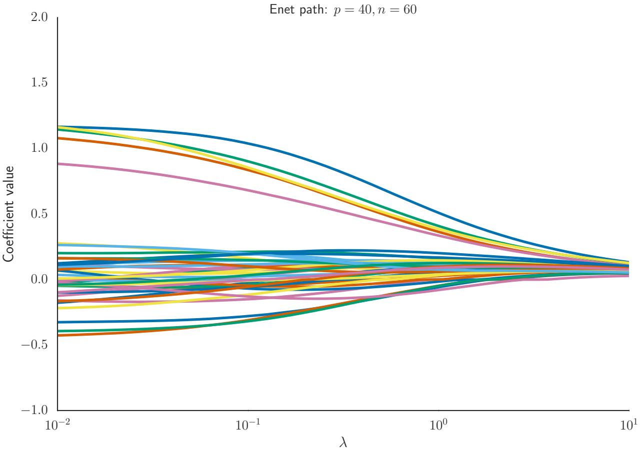
$$\gamma = 0.1$$

Elastic-Net: $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



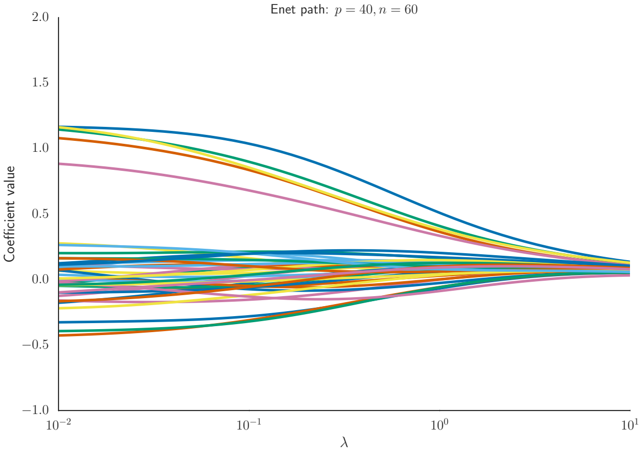
$$\gamma = 0.05$$

Elastic-Net: $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



$$\gamma = 0.01$$

Elastic-Net: $\gamma\|\boldsymbol{\theta}\|_1 + (1-\gamma)\|\boldsymbol{\theta}\|_2^2/2$



$$\gamma = 0.00$$

The Lasso bias

The Lasso is biased: it shrinks large coefficients towards 0

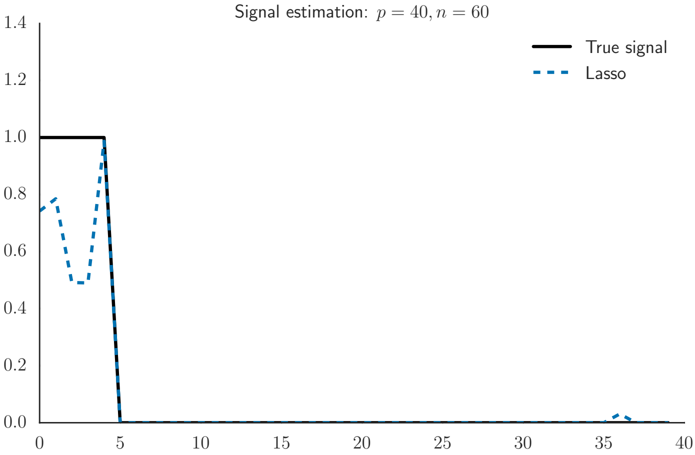


Illustration over the previous example

The Lasso bias

The Lasso is biased: it shrinks large coefficients towards 0

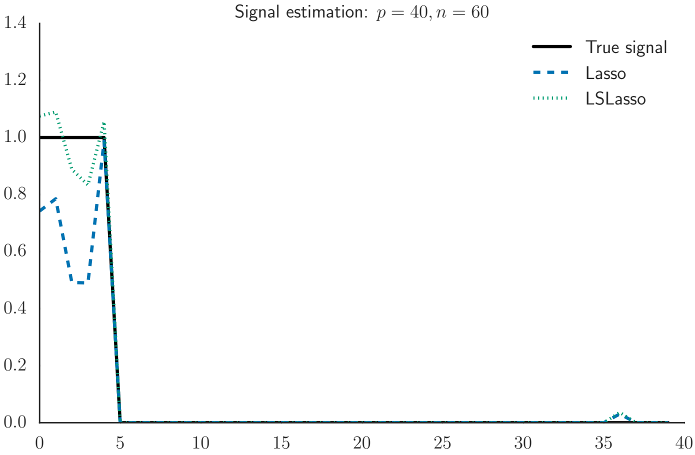


Illustration over the previous example

The Lasso bias: a simple remedy

How to rescale shrunk coefficients?

LSLasso (Least Square Lasso)

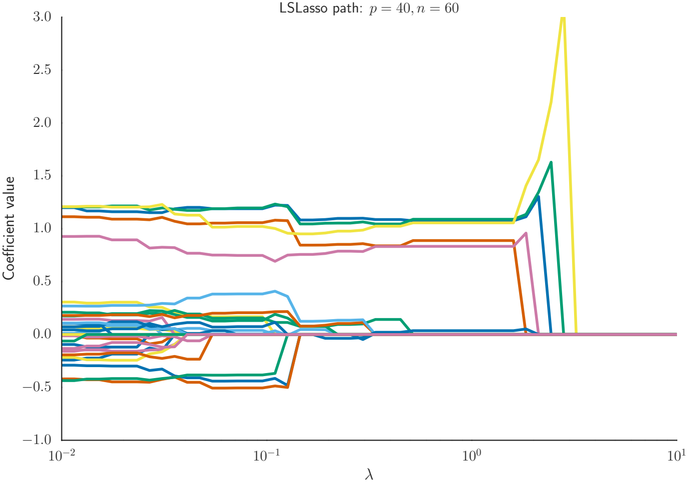
1. Lasso : compute $\hat{\boldsymbol{\theta}}_\lambda^{\text{Lasso}}$
2. Perform least squares over selected variables: $\operatorname{supp}(\hat{\boldsymbol{\theta}}_\lambda^{\text{Lasso}})$

$$\hat{\boldsymbol{\theta}}_\lambda^{\text{LSLasso}} = \mathop{\arg\min}_{\substack{\boldsymbol{\theta}\in\mathbb{R}^p \\ \operatorname{supp}(\boldsymbol{\theta})=\operatorname{supp}(\hat{\boldsymbol{\theta}}_\lambda^{\text{Lasso}})}} \frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2$$

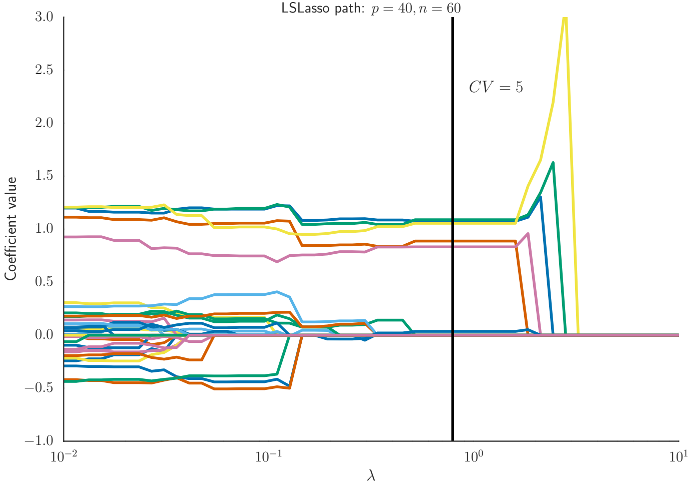
Rem: perform CV for the double step procedure; choosing λ by LassoCV and then performing OLS keeps too many variables

Rem: LSLasso is not coded in standard packages

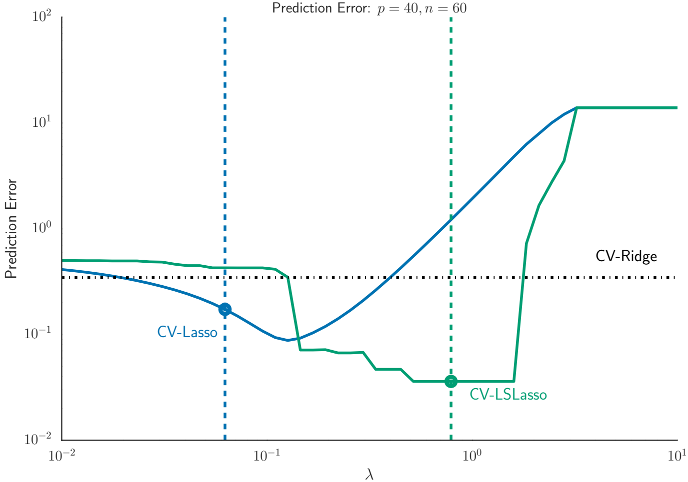
De-biasing


LSLasso path: $p = 40, n = 60$
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
Coefficient value
10^{-2}
10^{-1}
10^{0}
10^{1}
λ

De-biasing


LSLasso path: $p = 40, n = 60$
$CV = 5$
Coefficient value
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
10^{-2}
10^{-1}
10^{0}
10^{1}
λ

Prediction: Lasso vs. LSLasso



LSLasso evaluation

Pros

- the “true” large coefficients are less shrunk
- CV recovers less “parasite” variables (improve interpretability) *e.g.*,in the previous example the LSLassoCV recovers exactly the 5 “true” non zero variables, up to a single false positive

LSLasso: especially useful for <u>estimation</u>

Cons

- the difference in term of prediction is not always striking
- requires (slightly) more computation: needs to compute as many OLS as λ's

Non-convex penalties

Use a (smooth) penalty approximating better $\|\cdot\|_0$, choosing a non-convex$t \to \mathrm{pen}_{\lambda,\gamma}(t)$

$$\boxed{\hat{\boldsymbol{\theta}}^{\mathrm{pen}}_{\lambda,\gamma} = \mathop{\arg\min}_{\boldsymbol{\theta}\in\mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}_{\textbf{data fitting}} + \underbrace{\sum_{j=1}^{p} \mathrm{pen}_{\lambda,\gamma}(|\theta_j|)}_{\textbf{regularization}} \right)}$$

<u>Rem</u>: algorithmic difficulties (local minima), less theory

Non-convex penalties

Use a (smooth) penalty approximating better $\| \cdot \|_0$, choosing a non-convex$t \to \mathrm{pen}_{\lambda,\gamma}(t)$

$$\boxed{\hat{\boldsymbol{\theta}}^{\mathrm{pen}}_{\lambda,\gamma} = \mathop{\arg\min}_{\boldsymbol{\theta} \in \mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}_{\text{data fitting}} + \underbrace{\sum_{j=1}^{p} \mathrm{pen}_{\lambda,\gamma}(|\theta_j|)}_{\text{regularization}} \right)}$$

<u>Rem</u>: algorithmic difficulties (local minima), less theory

- Adaptive-Lasso Zou (2006) / re-weighted ℓ_1 Candès *et al.*(2008)

$$\mathrm{pen}_{\lambda,\gamma}(t) = \lambda |t|^q \text{ with } 0 < q < 1$$

Non-convex penalties

Use a (smooth) penalty approximating better $\|\cdot\|_0$, choosing a non-convex $t \to \mathrm{pen}_{\lambda,\gamma}(t)$

$$\boxed{\hat{\boldsymbol{\theta}}^{\mathrm{pen}}_{\lambda,\gamma} = \mathop{\arg\min}_{\boldsymbol{\theta}\in\mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}_{\textbf{data fitting}} + \underbrace{\sum_{j=1}^{p} \mathrm{pen}_{\lambda,\gamma}(|\theta_j|)}_{\textbf{regularization}} \right)}$$

Rem: algorithmic difficulties (local minima), less theory

- re-weighted ℓ_1 Candès *et al.*(2008)

$$\mathrm{pen}_{\lambda,\gamma}(t) = \lambda \log(1 + |t|/\gamma)$$

Non-convex penalties

Use a (smooth) penalty approximating better $\|\cdot\|_0$, choosing a non-convex$t \rightarrow \mathrm{pen}_{\lambda,\gamma}(t)$

$$\boxed{\hat{\boldsymbol{\theta}}^{\mathrm{pen}}_{\lambda,\gamma} = \mathop{\arg\min}_{\boldsymbol{\theta}\in\mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}_{\textbf{data fitting}} + \underbrace{\sum_{j=1}^{p} \mathrm{pen}_{\lambda,\gamma}(|\theta_j|)}_{\textbf{regularization}} \right)}$$

<u>Rem</u>: algorithmic difficulties (local minima), less theory

- MCP (*minimax concave penalty*) Zhang (2010) for $\lambda > 0$ and $\gamma > 1$

$$\mathrm{pen}_{\lambda,\gamma}(t) = \begin{cases} \lambda|t| - \frac{t^2}{2\gamma}, & \text{if } |t| \leqslant \gamma\lambda \\ \frac{1}{2}\gamma\lambda^2, & \text{if } |t| > \gamma\lambda \end{cases}$$

Non-convex penalties

Use a (smooth) penalty approximating better $\|\cdot\|_0$, choosing a non-convex$t \to \mathrm{pen}_{\lambda,\gamma}(t)$

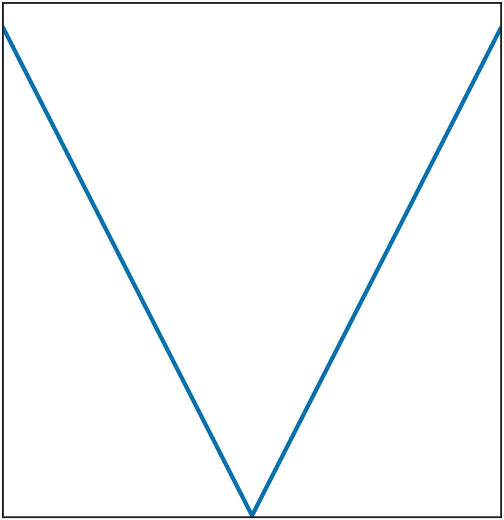
$$\boxed{\hat{\boldsymbol{\theta}}^{\mathrm{pen}}_{\lambda,\gamma} = \operatorname*{arg\,min}_{\boldsymbol{\theta}\in\mathbb{R}^p} \Bigg(\underbrace{\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}_{\text{data fitting}} + \underbrace{\sum_{j=1}^{p} \mathrm{pen}_{\lambda,\gamma}(|\theta_j|)}_{\text{regularization}} \Bigg)}$$

<u>Rem</u>: algorithmic difficulties (local minima), less theory

- SCAD (*Smoothly Clipped Absolute Deviation*) Fan and Li (2001) for $\lambda > 0$ and $\gamma > 2$

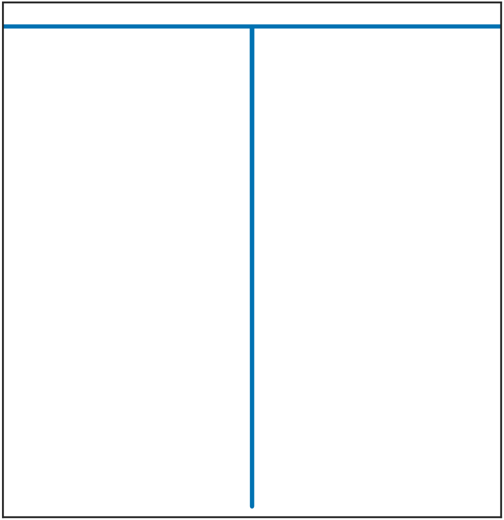
$$\mathrm{pen}_{\lambda,\gamma}(t) = \begin{cases} \lambda|t|, & \text{if } |t| \leqslant \lambda \\ \frac{\gamma\lambda|t| - (t^2+\lambda^2)/2}{\gamma-1}, & \text{if } \lambda < |t| \leqslant \gamma\lambda \\ \frac{\lambda^2(\gamma^2-1)}{2(\gamma-1)}, & \text{if } |t| > \gamma\lambda \end{cases}$$

Standard non-convex penalties



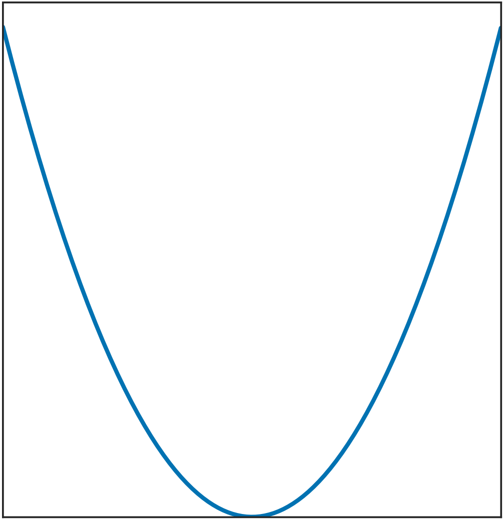
l1

Standard non-convex penalties



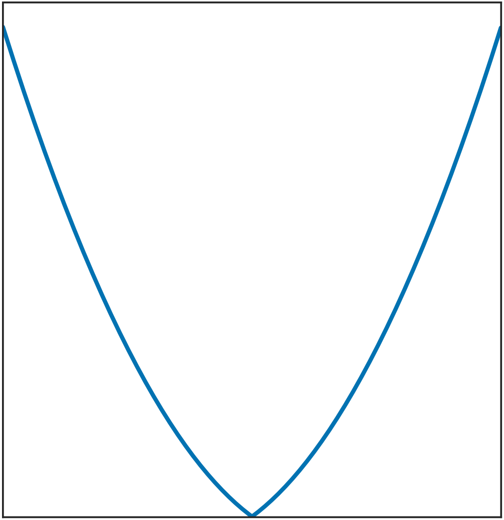
l0

Standard non-convex penalties



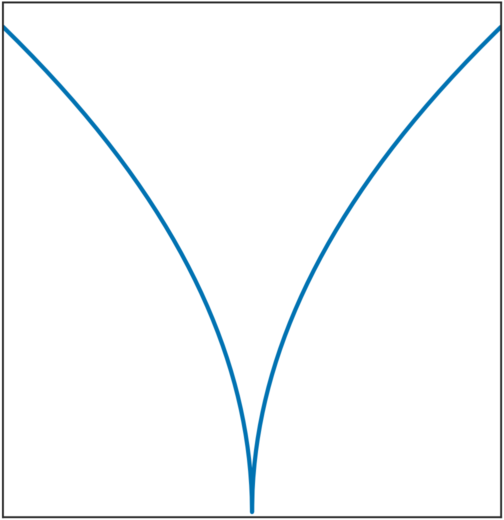
l22

Standard non-convex penalties



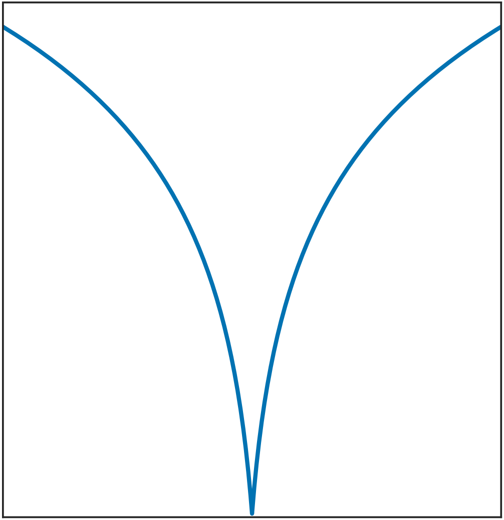
enet

Standard non-convex penalties



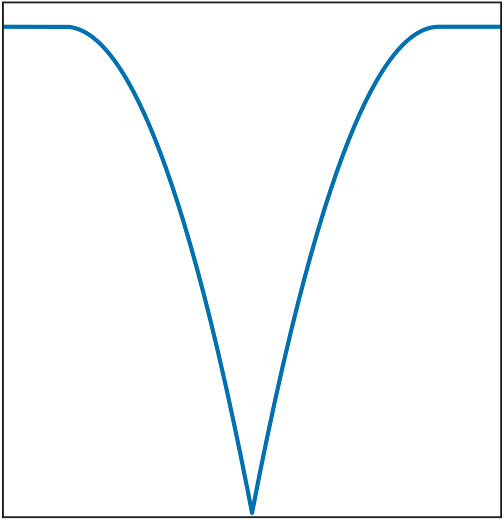
sqrt

Standard non-convex penalties



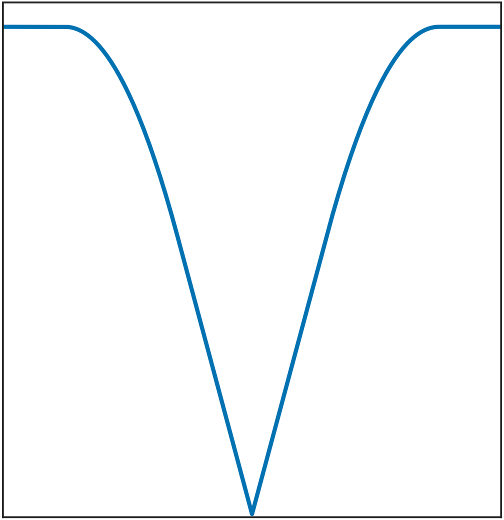
log

Standard non-convex penalties



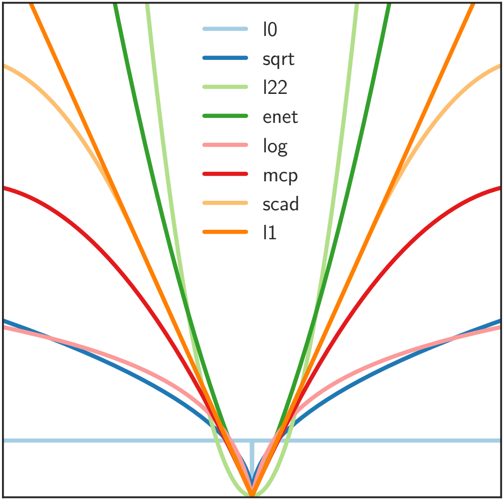
mcp

Standard non-convex penalties



scad

Standard non-convex penalties



Adaptive-Lasso

Several names for the same idea:

- Adaptive-Lasso Zou (2006)
- re-weighted ℓ_1 Candès *et al.*(2008)
- DC-programming approach (for *Difference of Convex Programming*) Gasso *et al.*(2008)

Underlying idea: **Majorization-Minorization** (MM) method in optimization:

- find an upper bound of the target function to optimize
- optimize this proxy
- repeat

Adaptive-Lasso

Several names for the same idea:

- Adaptive-Lasso Zou (2006)
- re-weighted ℓ_1 Candès *et al.*(2008)
- DC-programming approach (for *Difference of Convex Programming*) Gasso *et al.*(2008)

Underlying idea: **Majorization-Minorization** (MM) method in optimization:

- find an upper bound of the target function to optimize
- optimize this proxy
- repeat

Adaptive-Lasso

Example : take $\mathrm{pen}_{\lambda,\gamma}(t) = \lambda|t|^q$ with $q = 1/2$

Algorithm: Adaptive Lasso ($q = 1/2$ case)

Input : $X, \mathbf{y}$, maximum number of iterations K, λ (regularization)

Initialization: $\hat{w} \leftarrow (1, \ldots, 1)^\top$

Adaptive-Lasso

Example : take $\mathrm{pen}_{\lambda,\gamma}(t) = \lambda |t|^q$ with $q = 1/2$

Algorithm: Adaptive Lasso ($q = 1/2$ case)

Input : $X, \mathbf{y}$, maximum number of iterations K, λ (regularization)

Initialization: $\hat{w} \leftarrow (1, \ldots, 1)^\top$

for $k = 1, \ldots, K$ **do**

Adaptive-Lasso

Example : take $\mathrm{pen}_{\lambda,\gamma}(t) = \lambda |t|^q$ with $q = 1/2$

Algorithm: Adaptive Lasso ($q = 1/2$ case)

Input : $X, \mathbf{y}$, maximum number of iterations K, λ (regularization)

Initialization: $\hat{w} \leftarrow (1, \dots, 1)^\top$

for $k = 1, \dots, K$ **do**

$$\hat{\boldsymbol{\theta}} \leftarrow \arg\min_{\boldsymbol{\theta}} \left(\frac{\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}{2} + \lambda \sum_{j=1}^{p} \hat{w}_j |\theta_j| \right)$$

Adaptive-Lasso

Example : take $\mathrm{pen}_{\lambda,\gamma}(t) = \lambda|t|^q$ with $q = 1/2$

Algorithm: Adaptive Lasso ($q = 1/2$ case)

Input : $X, \mathbf{y}$, maximum number of iterations K, λ (regularization)

Initialization: $\hat{w} \leftarrow (1, \ldots, 1)^\top$

for $k = 1, \ldots, K$ **do**

$$\hat{\boldsymbol{\theta}} \leftarrow \arg\min_{\boldsymbol{\theta}} \left(\frac{\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}{2} + \lambda \sum_{j=1}^{p} \hat{w}_j |\theta_j| \right)$$

$$\hat{w}_j \leftarrow \frac{1}{|\hat{\theta}_j|^{\frac{1}{2}}}, \ \forall j \in [\![1, p]\!]$$

Adaptive-Lasso

Example : take $\text{pen}_{\lambda,\gamma}(t) = \lambda|t|^q$ with $q = 1/2$

Algorithm: Adaptive Lasso ($q = 1/2$ case)

Input : $X, \mathbf{y}$, maximum number of iterations K, λ (regularization)

Initialization: $\hat{w} \leftarrow (1, \dots, 1)^\top$

for $k = 1, \dots, K$ **do**

$$\hat{\boldsymbol{\theta}} \leftarrow \arg\min_{\boldsymbol{\theta}} \left(\frac{\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}{2} + \lambda \sum_{j=1}^{p} \hat{w}_j |\theta_j| \right)$$

$$\hat{w}_j \leftarrow \frac{1}{|\hat{\theta}_j|^{\frac{1}{2}}}, \ \forall j \in [\![1, p]\!]$$

Rem: in practice few iterations needed (about 5/10)

Adaptive-Lasso

Example : take $\text{pen}_{\lambda,\gamma}(t) = \lambda|t|^q$ with $q = 1/2$

Algorithm: Adaptive Lasso ($q = 1/2$ case)

Input : $X, \mathbf{y}$, maximum number of iterations K, λ (regularization)

Initialization: $\hat{w} \leftarrow (1, \dots, 1)^\top$

for $k = 1, \dots, K$ **do**

$$\hat{\boldsymbol{\theta}} \leftarrow \arg\min_{\boldsymbol{\theta}} \left(\frac{\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2}{2} + \lambda \sum_{j=1}^{p} \hat{w}_j |\theta_j| \right)$$

$$\hat{w}_j \leftarrow \frac{1}{|\hat{\theta}_j|^{\frac{1}{2}}}, \ \forall j \in [\![1, p]\!]$$

Rem: in practice few iterations needed (about 5/10)

Rem: use a Lasso solver to update $\hat{\boldsymbol{\theta}}$, by rescaling the design matrix

Support structure

Suppose a known group structure on the variables (prior the experiment) : $[\![1, p]\!] = \bigcup_{g \in \mathcal{G}} g$

Active coordinates (in orange):

Sparse support: any

Possible penalties: Lasso

$$\|\theta\|_1 = \sum_{j=1}^{p} |\theta_j|$$

Support structure

Suppose a known group structure on the variables (prior the experiment) : $[\![1, p]\!] = \bigcup_{g \in \mathcal{G}} g$

Active coordinates (in orange):

Sparse support: group

Possible penalties: Group-Lasso

$$\|\theta\|_{2,1} = \sum_{g \in G} \|\theta_g\|_2$$

Support structure

Suppose a known group structure on the variables (prior the experiment) : $[\![1, p]\!] = \bigcup_{g \in \mathcal{G}} g$

Active coordinates (in orange):

Sparse support: group + sub-groups

Possible penalties: Sparse-Group-Lasso

$$\alpha \|\theta\|_1 + (1-\alpha)\|\theta\|_{2,1} = \alpha \textstyle\sum_{j=1}^{p} |\theta_j| + (1-\alpha) \displaystyle\sum_{g \in G} \|\theta_g\|_2$$

Group-Lasso

ℓ_1 penalty : ensures few active coefficients, but other structures could be enforced similarly

One can aim at:

- group/block wise sparsity: Group-Lasso Yuan and Lin (2006)
- individual and group wise : Sparse Group-Lasso Simon, Friedman, Hastie and Tibshirani (2012)

Group-Lasso

ℓ_1 penalty : ensures few active coefficients, but other structures could be enforced similarly

One can aim at:

- group/block wise sparsity: Group-Lasso Yuan and Lin (2006)
- individual and group wise : Sparse Group-Lasso Simon, Friedman, Hastie and Tibshirani (2012)
- hierarchical structures, *e.g.*,for higher order interactions Bien, Taylor and Tibshirani (2013)

Group-Lasso

ℓ_1 penalty : ensures few active coefficients, but other structures could be enforced similarly

One can aim at:

- group/block wise sparsity: Group-Lasso Yuan and Lin (2006)
- individual and group wise : Sparse Group-Lasso Simon, Friedman, Hastie and Tibshirani (2012)
- hierarchical structures, *e.g.*,for higher order interactions Bien, Taylor and Tibshirani (2013)
- graph structures, gradient structures (aka Total Variation)

Group-Lasso

ℓ_1 penalty : ensures few active coefficients, but other structures could be enforced similarly

One can aim at:

- group/block wise sparsity: Group-Lasso Yuan and Lin (2006)
- individual and group wise : Sparse Group-Lasso Simon, Friedman, Hastie and Tibshirani (2012)
- hierarchical structures, *e.g.*,for higher order interactions Bien, Taylor and Tibshirani (2013)
- graph structures, gradient structures (aka Total Variation)
- ...

Group-Lasso

ℓ_1 penalty : ensures few active coefficients, but other structures could be enforced similarly

One can aim at:

- group/block wise sparsity: Group-Lasso Yuan and Lin (2006)
- individual and group wise : Sparse Group-Lasso Simon, Friedman, Hastie and Tibshirani (2012)
- hierarchical structures, *e.g.*,for higher order interactions Bien, Taylor and Tibshirani (2013)
- graph structures, gradient structures (aka Total Variation)
- . . .

Lasso stability

The Lasso can be **instable**: when non-unique solutions (*e.g.*,when $p > n$) depending on the numerical solver and the required precision, there might be errors in the variable selection process.

Re-sampling techniques : designed to limit such drawbacks

- Bolasso Bach (2008)
- Stability Selection Meinshausen and Buhlmann (2010)

Bolasso Bach (2008)

Algorithm: Bootstrap Lasso

Input : $X, \mathbf{y}$, replications number B, λ regularization

Exo: code the Bolasso in `Python` using `sklearn`

Bolasso Bach (2008)

Algorithm: Bootstrap Lasso

Input : $X, \mathbf{y}$, replications number B, λ regularization

for $k = 1, \dots, B$ **do**

Exo: code the Bolasso in `Python` using `sklearn`

Bolasso Bach (2008)

Algorithm: Bootstrap Lasso

Input : $X, \mathbf{y}$, replications number B, λ regularization

for $k = 1, \ldots, B$ **do**

Draw a *bootstrap* sample: $X^{(k)}, y^{(k)}$

Exo: code the Bolasso in `Python` using `sklearn`

Bolasso Bach (2008)

Algorithm: Bootstrap Lasso

Input : $X, \mathbf{y}$, replications number B, λ regularization

for $k = 1, \ldots, B$ **do**

Draw a *bootstrap* sample: $X^{(k)}, y^{(k)}$

Compute the Lasso for this sample: $\hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Lasso},(k)}$

Exo: code the Bolasso in `Python` using `sklearn`

Bolasso Bach (2008)

Algorithm: Bootstrap Lasso

Input : $X, \mathbf{y}$, replications number B, λ regularization

for $k = 1, \dots, B$ **do**

Draw a *bootstrap* sample: $X^{(k)}, y^{(k)}$

Compute the Lasso for this sample: $\hat{\boldsymbol{\theta}}_{\lambda}^{\mathrm{Lasso},(k)}$

Compute the associated support: $S_k = \mathrm{supp}\left(\hat{\boldsymbol{\theta}}_{\lambda}^{\mathrm{Lasso},(k)}\right)$

Exo: code the Bolasso in `Python` using `sklearn`

Bolasso Bach (2008)

Algorithm: Bootstrap Lasso

Input : $X, \mathbf{y}$, replications number B, λ regularization

for $k = 1, \dots, B$ **do**

Draw a *bootstrap* sample: $X^{(k)}, y^{(k)}$

Compute the Lasso for this sample: $\hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Lasso},(k)}$

Compute the associated support: $S_k = \operatorname{supp}\left(\hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Lasso},(k)}\right)$

Compute: $S := \bigcap_{k=1}^{B} S_k$

Exo: code the Bolasso in `Python` using `sklearn`

Bolasso Bach (2008)

Algorithm: Bootstrap Lasso

Input : $X, \mathbf{y}$, replications number B, λ regularization

for $k = 1, \dots, B$ **do**

- Draw a *bootstrap* sample: $X^{(k)}, y^{(k)}$
- Compute the Lasso for this sample: $\hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Lasso},(k)}$
- Compute the associated support: $S_k = \operatorname{supp}\left(\hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Lasso},(k)}\right)$

Compute: $S := \displaystyle\bigcap_{k=1}^{B} S_k$

Compute: $\hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Bolasso}} \in \displaystyle\mathop{\arg\min}_{\substack{\boldsymbol{\theta} \in \mathbb{R}^p \\ \operatorname{supp}(\boldsymbol{\theta}) = S}} \frac{1}{2} \|\mathbf{y} - X\boldsymbol{\theta}\|_2^2$

Exo: code the Bolasso in `Python` using `sklearn`

Bolasso Bach (2008)

Algorithm: Bootstrap Lasso

Input : $X, \mathbf{y}$, replications number B, λ regularization

for $k = 1, \dots, B$ **do**

  Draw a *bootstrap* sample: $X^{(k)}, y^{(k)}$

  Compute the Lasso for this sample: $\hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Lasso},(k)}$

  Compute the associated support: $S_k = \operatorname{supp}\left(\hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Lasso},(k)}\right)$

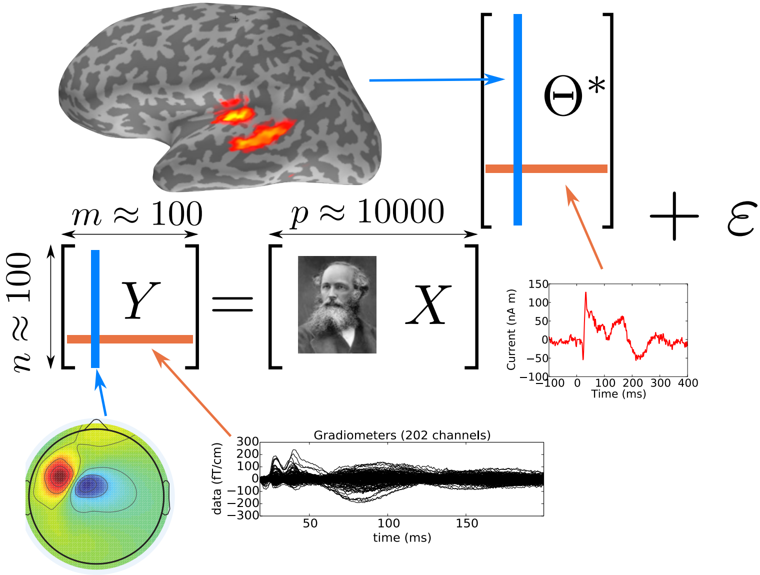
Compute: $S := \bigcap_{k=1}^{B} S_k$

Compute: $\hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Bolasso}} \in \underset{\substack{\boldsymbol{\theta}\in\mathbb{R}^p \\ \operatorname{supp}(\boldsymbol{\theta})=S}}{\arg\min} \frac{1}{2}\|\mathbf{y} - X\boldsymbol{\theta}\|_2^2$

Output : support S, and a vector $\hat{\boldsymbol{\theta}}_\lambda^{\mathrm{Bolasso}}$

Exo: code the Bolasso in `Python` using `sklearn`

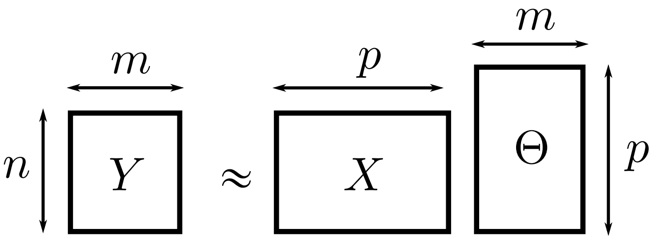
Example



Θ^*
$+ \ \varepsilon$
$m \approx 100$
$p \approx 10000$
$n \approx 100$
Y
$=$
X
Current (nA m)
150
100
50
0
−50
−100
−100 0 100 200 300 400
Time (ms)
Gradiometers (202 channels)
data (fT/cm)
300
200
100
0
−100
−200
−300
50 100 150
time (ms)

Multi-task regression

One aims at jointly solving m linear regression: $Y \approx X\Theta$



with

- $Y \in \mathbb{R}^{n \times m}$: observation matrix
- $X \in \mathbb{R}^{n \times p}$: design matrix (known)
- $\Theta \in \mathbb{R}^{p \times m}$: coefficient matrix (unknown)

Example : several observed signals through time (*e.g.*,several captors for the same phenomenon)

Rem:*cf.*`MultiTaskLasso` in `sklearn` for a solver

Multi-task and regularization

In multi-task settings penalties can also be helpful:

$$\boxed{\hat{\Theta}_\lambda = \underset{\Theta \in \mathbb{R}^{p \times m}}{\arg\min} \left(\underbrace{\frac{1}{2}\|Y - X\Theta\|_F^2}_{\text{data fitting}} + \underbrace{\lambda\Omega(\Theta)}_{\text{regularization}} \right)}$$

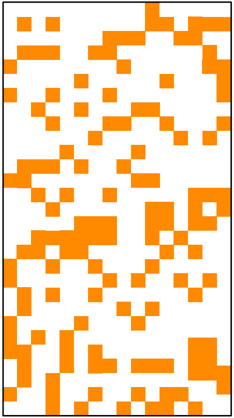
where Ω is a penalty / regularization

Rem: the Frobenius norm $\|\cdot\|_F$ is defined for any matrix $A \in \mathbb{R}^{n_1 \times n_2}$ by

$$\|A\|_F^2 = \sum_{j_1=1}^{n_1} \sum_{j_2=1}^{n_2} A_{j_1,j_2}^2$$

Multi-tasks penalties

Vectorial penalties need to be adapted:



Parameter $\Theta \in \mathbb{R}^{p \times m}$

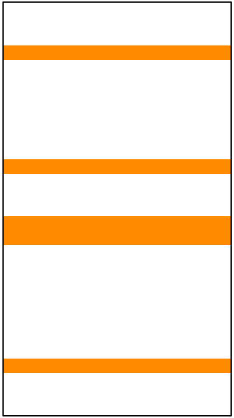
Sparse support:
any

Penalty: Lasso

$$\|\Theta\|_1 = \sum_{j=1}^{p} \sum_{k=1}^{m} |\Theta_{j,k}|$$

Multi-tasks penalties

Vectorial penalties need to be adapted:



Parameter $\Theta \in \mathbb{R}^{p \times m}$

Sparse support:
group

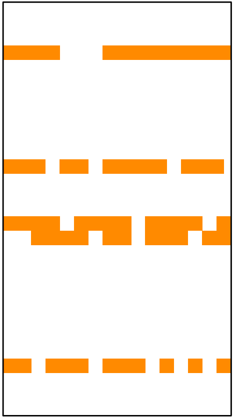
Penalty: Group-Lasso

$$\|\Theta\|_{2,1} = \sum_{j=1}^{p} \|\Theta_{j:}\|_2$$

where $\Theta_{j,:}$ the j-th line of Θ

Multi-tasks penalties

Vectorial penalties need to be adapted:



Parameter $\Theta \in \mathbb{R}^{p \times m}$

Sparse support:
group + sub-groups

Penalty: Sparse-Group-Lasso

$$\alpha \|\Theta\|_1 + (1 - \alpha)\|\Theta\|_{2,1}$$

References I

- F. Bach.
  Bolasso: model consistent Lasso estimation through the bootstrap.
  In *ICML*, 2008.
- P. Bühlmann and S. van de Geer.
  Statistics for high-dimensional data.
  Springer Series in Statistics. Springer, Heidelberg, 2011.
  Methods, theory and applications.
- E. J. Candès, M. B. Wakin, and S. P. Boyd.
  Enhancing sparsity by reweighted l_1 minimization.
  J. Fourier Anal. Applicat., 14(5-6):877–905, 2008.
- J. Fan and R. Li.
  Variable selection via nonconcave penalized likelihood and its oracle properties.
  J. Amer. Statist. Assoc., 96(456):1348–1360, 2001.

References II

- G. Gasso, A. Rakotomamonjy, and S. Canu.
  Recovering sparse signals with non-convex penalties and DC programming.
  IEEE Trans. Sig. Process., 57(12):4686–4698, 2009.

- Bien J, J. Taylor, and R. Tibshirani.
  A lasso for hierarchical interactions.
  Ann. Statist., 41(3):1111–1141, 2013.

- N. Meinshausen and P. Bühlmann.
  Stability selection.
  Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(4):417–473, 2010.

- N. Parikh, S. Boyd, E. Chu, B. Peleato, and J. Eckstein.
  Proximal algorithms.
  Foundations and Trends in Machine Learning, 1(3):1–108, 2013.

- N. Simon, J. Friedman, T. Hastie, and R. Tibshirani.
  A sparse-group lasso.
  J. Comput. Graph. Statist., 22(2):231–245, 2013.

References III

- R. Tibshirani.
  Regression shrinkage and selection via the lasso.
  J. Roy. Statist. Soc. Ser. B, 58(1):267–288, 1996.
- M. Yuan and Y. Lin.
  Model selection and estimation in regression with grouped variables.
  J. Roy. Statist. Soc. Ser. B, 68(1):49–67, 2006.
- H. Zou and T. Hastie.
  Regularization and variable selection via the elastic net.
  J. Roy. Statist. Soc. Ser. B, 67(2):301–320, 2005.
- C.-H. Zhang.
  Nearly unbiased variable selection under minimax concave penalty.
  Ann. Statist., 38(2):894–942, 2010.
- H. Zou.
  The adaptive lasso and its oracle properties.
  J. Am. Statist. Assoc., 101(476):1418–1429, 2006.