(Gap) Safe screening rules to speed-up sparse regression solvers

Joseph Salmon

http://josephsalmon.eu

LTCI, Télécom Paristech, ~~Université Paris-Saclay~~ NewUni

Joint work with:

Eugene Ndiaye (Télécom ParisTech)

Olivier Fercoq (Télécom ParisTech)

Alexandre Gramfort (INRIA, Parietal Team)

Table of Contents

Motivation - notation

Optimization and convexity reminders

Optimization property for the Lasso

Safe rules

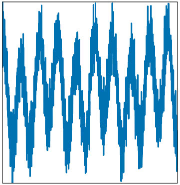
Gap safe rules

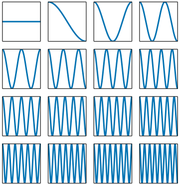
Coordinate descent implementation

Sparsity is all around

Signals can often be represented through a combination of a few **atoms** / **features** :

- Fourier decomposition for sounds

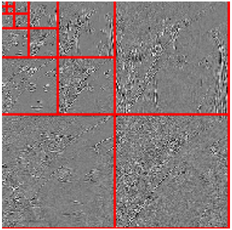




Sparsity is all around

Signals can often be represented through a combination of a few **atoms** / **features** :

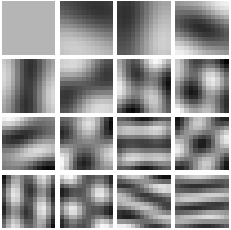
- Fourier decomposition for sounds
- Wavelet for images (1990's)



Sparsity is all around

Signals can often be represented through a combination of a few **atoms** / **features** :

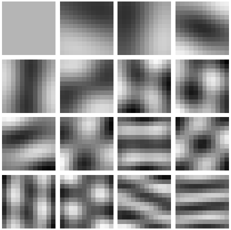
- Fourier decomposition for sounds
- Wavelet for images (1990's)
- Dictionary learning for images (late 2000's)



Sparsity is all around

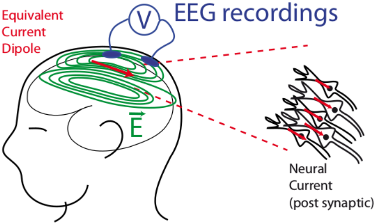
Signals can often be represented through a combination of a few **atoms** / **features** :

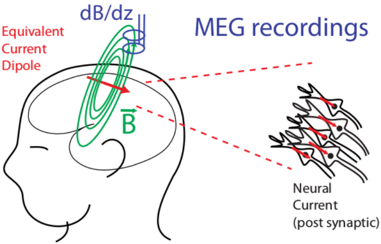
- Fourier decomposition for sounds
- Wavelet for images (1990's)
- Dictionary learning for images (late 2000's)
- More inverse problems



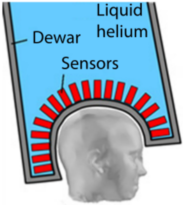
Another motivation: M/EEG inverse problem

- sensors: magneto- and electro-encephalogram measurements during a cognitive experiment (*e.g.*, sensory or memory)
- sources: brain locations



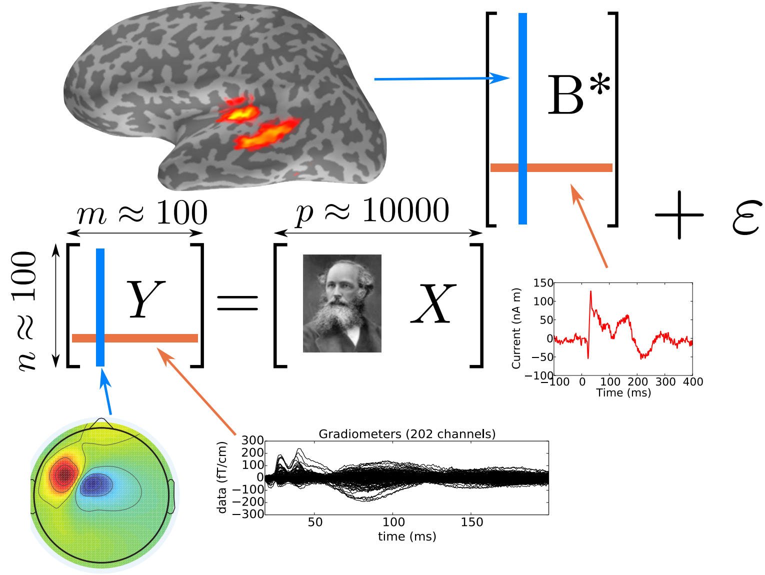


First EEG
recordings
in 1929
by H. Berger



Hôpital La Timone
Marseille, France

Modeling for this problem



Simplest model: standard sparse regression

$y \in \mathbb{R}^n$: a signal

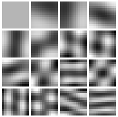
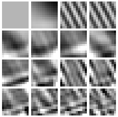
$X = [\mathbf{x}_1, \dots, \mathbf{x}_p] \in \mathbb{R}^{n \times p}$:
dictionary of atoms/features

Assumption : signal well approximated by a **sparse** combination $\beta^* \in \mathbb{R}^p$: $y \approx X\beta^*$

Objective(s): find $\hat{\beta}$

- Estimation: $\hat{\beta} \approx \hat{\beta}^*$
- Prediction: $X\hat{\beta} \approx X\hat{\beta}^*$
- Support recovery: $\mathrm{supp}(\hat{\beta}) \approx \mathrm{supp}(\beta^*)$

Constraints: large p, sparse β^*

$$\underbrace{\begin{bmatrix} y \end{bmatrix}}_{y \in \mathbb{R}^n} \approx \underbrace{\begin{bmatrix} \mathbf{x}_1 & \Big| & \dots & \Big| & \mathbf{x}_p \end{bmatrix}}_{X \in \mathbb{R}^{n \times p}} \cdot \underbrace{\begin{bmatrix} \beta_1^* \\ \vdots \\ \beta_p^* \end{bmatrix}}_{\beta \in \mathbb{R}^p}$$

$$y \approx \sum_{j=1}^{p} \beta_j^* \mathbf{x}_j$$

The ℓ_0 penalty

Objective: use Least-Squares with an ℓ_0 penalty to enforce sparsity

$$\boxed{\operatorname*{arg\,min}_{\beta \in \mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|y - X\beta\|_2^2}_{\textbf{data fitting}} + \underbrace{\lambda \|\beta\|_0}_{\textbf{regularization}} \right)}$$

where $\|\beta\|_0 = \operatorname{card}(\{j \in [\![1, p]\!], \beta_j \neq 0\}) = \operatorname{card}(\operatorname{supp}(\beta))$

Combinatorial problem; "NP-hard" Natarajan (1995)

$\hookrightarrow$ Exact resolution requires Least-Squares (LS) solutions for all sub-models, *i.e.*, compute LS for all possible supports (up to 2^p)

- $p = 10$ possible: $\approx 10^3$ least squares
- $p = 30$ impossible: $\approx 10^{10}$ least squares

Rem: for "small" problems mixed integer programming (MIP) well suited Bertsimas et al. (2015)

The ℓ_1 penalty: Lasso and variants

Vocabulary: the "Modern least square" Candès *et al.* (2008)

- Statistics: **Lasso** Tibshirani (1996)
- Signal processing variant: **Basis Pursuit** Chen *et al.* (1998)

$$\boxed{\hat{\beta}^{(\lambda)} \in \mathop{\arg\min}_{\beta \in \mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|y - X\beta\|^2}_{\textbf{data fitting term}} + \underbrace{\lambda \|\beta\|_1}_{\textbf{sparsity-inducing penalty}} \right)}$$

- Solutions are **sparse** (sparsity level controlled by λ)
- Need to tune/choose λ (standard is Cross-Validation)

The ℓ_1 penalty: Lasso and variants

Vocabulary: the "Modern least square" Candès *et al.* (2008)

- Statistics: **Lasso** Tibshirani (1996)
- Signal processing variant: **Basis Pursuit** Chen *et al.* (1998)

$$\boxed{\hat{\beta}^{(\lambda)} \in \operatorname*{arg\,min}_{\beta \in \mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|y - X\beta\|^2}_{\textbf{data fitting term}} + \underbrace{\lambda\|\beta\|_1}_{\textbf{sparsity-inducing penalty}} \right)}$$

- Solutions are **sparse** (sparsity level controlled by λ)
- Need to tune/choose λ (standard is Cross-Validation)
- Theoretical guaranties Bickel *et al.* (2009)

The ℓ_1 penalty: Lasso and variants

Vocabulary: the "Modern least square" Candès *et al.* (2008)

- Statistics: **Lasso** Tibshirani (1996)
- Signal processing variant: **Basis Pursuit** Chen *et al.* (1998)

$$\boxed{\hat{\beta}^{(\lambda)} \in \operatorname*{arg\,min}_{\beta \in \mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|y - X\beta\|^2}_{\textbf{data fitting term}} + \underbrace{\lambda\|\beta\|_1}_{\textbf{sparsity-inducing penalty}} \right)}$$

- Solutions are **sparse** (sparsity level controlled by λ)
- Need to tune/choose λ (standard is Cross-Validation)
- Theoretical guaranties Bickel *et al.* (2009)
- Uniqueness not automatic, see discussion in Tibshirani (2013)

The ℓ_1 penalty: Lasso and variants

Vocabulary: the "Modern least square" Candès *et al.* (2008)

- Statistics: **Lasso** Tibshirani (1996)
- Signal processing variant: **Basis Pursuit** Chen *et al.* (1998)

$$\boxed{\hat{\beta}^{(\lambda)} \in \underset{\beta \in \mathbb{R}^p}{\arg\min} \left(\underbrace{\frac{1}{2}\|y - X\beta\|^2}_{\textbf{data fitting term}} + \underbrace{\lambda\|\beta\|_1}_{\textbf{sparsity-inducing penalty}} \right)}$$

- Solutions are **sparse** (sparsity level controlled by λ)
- Need to tune/choose λ (standard is Cross-Validation)
- Theoretical guaranties Bickel *et al.* (2009)
- Uniqueness not automatic, see discussion in Tibshirani (2013)
- Refinements: non-convex approaches Adaptive Lasso Zou (2006), scaled invariance Zhang and Zhang (2012), etc.

The ℓ_1 penalty: Lasso and variants

Vocabulary: the "Modern least square" Candès *et al.* (2008)

- Statistics: **Lasso** Tibshirani (1996)
- Signal processing variant: **Basis Pursuit** Chen *et al.* (1998)

$$\hat{\beta}^{(\lambda)} \in \operatorname*{arg\,min}_{\beta \in \mathbb{R}^p} \left(\underbrace{\frac{1}{2}\|y - X\beta\|^2}_{\text{data fitting term}} + \underbrace{\lambda\|\beta\|_1}_{\text{sparsity-inducing penalty}} \right)$$

- Solutions are **sparse** (sparsity level controlled by λ)
- Need to tune/choose λ (standard is Cross-Validation)
- Theoretical guaranties Bickel *et al.* (2009)
- Uniqueness not automatic, see discussion in Tibshirani (2013)
- Refinements: non-convex approaches Adaptive Lasso Zou (2006), scaled invariance Zhang and Zhang (2012), etc.

More constraints: many Lasso's are needed

Reminder: $\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|_2^2 + \lambda\|\beta\|_1$

- Additional constraint: λ hard to "guess" in practice
- Common strategy: compute solutions over a grid, *i.e.*, get $\hat{\beta}^{(\lambda_0)}, \ldots, \hat{\beta}^{(\lambda_{T-1})}$, with $\lambda_0 > \cdots > \lambda_{T-1}$ for many T's, then pick the "best" one
  Standard grid (R-`glmnet` / Python-`sklearn`) : geometric with $\lambda_0 = \|X^\top y\|_\infty$, $\lambda_{T-1} = \alpha\lambda_{\max}$, $T = 100$ and $\alpha = 0.001$

What follows is **not** addressed in this talk:

- Grid choice
- Criterion to pick a "best" λ parameter : cross-validation, SURE (Stein Unbiased Risk Estimation), etc.

Contributions: speeding-up full grid evaluation

Take home message: screen/detect "early" zero-coefficients of $\hat{\beta}^{(\lambda)}$; remove associated features to speed-up numerical solver

- Safe screening rules can help:
  1. prior any computation (**static**)
  2. thanks to solutions already obtained for close λ's (**sequential**)
  3. along iterative steps of the algorithm (**dynamic**)

Contributions: speeding-up full grid evaluation

Take home message: screen/detect "early" zero-coefficients of $\hat{\beta}^{(\lambda)}$; remove associated features to speed-up numerical solver

- Safe screening rules can help:
  1. prior any computation (**static**)
  2. thanks to solutions already obtained for close λ's (**sequential**)
  3. along iterative steps of the algorithm (**dynamic**)

- Flexible : well suited for most iterative solvers, particularly for coordinate descent (more on that later) or active sets methods

Contributions: speeding-up full grid evaluation

Take home message: screen/detect "early" zero-coefficients of $\hat{\beta}^{(\lambda)}$; remove associated features to speed-up numerical solver

- Safe screening rules can help:
  1. prior any computation (**static**)
  2. thanks to solutions already obtained for close λ's (**sequential**)
  3. along iterative steps of the algorithm (**dynamic**)

- Flexible : well suited for most iterative solvers, particularly for coordinate descent (more on that later) or active sets methods

- Guaranteed convergence: when using a (proved) converging solvers, adding a safe screening step maintains convergence

Contributions: speeding-up full grid evaluation

Take home message: screen/detect "early" zero-coefficients of $\hat{\beta}^{(\lambda)}$; remove associated features to speed-up numerical solver

- Safe screening rules can help:
  1. prior any computation (**static**)
  2. thanks to solutions already obtained for close λ's (**sequential**)
  3. along iterative steps of the algorithm (**dynamic**)

- Flexible : well suited for most iterative solvers, particularly for coordinate descent (more on that later) or active sets methods

- Guaranteed convergence: when using a (proved) converging solvers, adding a safe screening step maintains convergence

- Simplicity: easy to incorporate in standard solvers, contrarily to non-safe methods like **Strong rules** Tibshirani *et al.* (2012)

Contributions: speeding-up full grid evaluation

Take home message: screen/detect "early" zero-coefficients of $\hat{\beta}^{(\lambda)}$; remove associated features to speed-up numerical solver

- Safe screening rules can help:
  1. prior any computation (**static**)
  2. thanks to solutions already obtained for close λ's (**sequential**)
  3. along iterative steps of the algorithm (**dynamic**)

- Flexible : well suited for most iterative solvers, particularly for coordinate descent (more on that later) or active sets methods

- Guaranteed convergence: when using a (proved) converging solvers, adding a safe screening step maintains convergence

- Simplicity: easy to incorporate in standard solvers, contrarily to non-safe methods like **Strong rules** Tibshirani *et al.* (2012)

Table of Contents

Motivation - notation

Optimization and convexity reminders

Optimization property for the Lasso

Safe rules

Gap safe rules

Coordinate descent implementation

The Lasso: algorithmic point of view

Commonly used algorithms for solving this **convex** program:

- Homotopy method - LARS:
  efficient for small p Osborne *et al.* (2000), Efron *et al.* (2004) and to get full path (*i.e.*, the full $\lambda \to \hat{\beta}^{(\lambda)}$)
  Limitation: do not generalize to other data-fitting term, potentially too many kinks Mairal and Yu (2012) (up to 3^p)

- (F)ISTA, Forward - Backward, proximal algorithm:
  useful in signal processing where $r \to X^\top r$ is cheap to compute (*e.g.*, FFT, Fast Wavelet Transform, etc.) Beck and Teboulle (2009)
  Limitation: unstructured X in statistics / machine learning

The Lasso: algorithmic point of view

Commonly used algorithms for solving this **convex** program:

- Homotopy method - LARS:
  efficient for small p Osborne *et al.* (2000), Efron *et al.* (2004) and to get full path (*i.e.*, the full $\lambda \to \hat{\beta}^{(\lambda)}$)
  Limitation: do not generalize to other data-fitting term, potentially too many kinks Mairal and Yu (2012) (up to 3^p)

- (F)ISTA, Forward - Backward, proximal algorithm:
  useful in signal processing where $r \to X^\top r$ is cheap to compute (*e.g.*, FFT, Fast Wavelet Transform, etc.) Beck and Teboulle (2009)
  Limitation: unstructured X in statistics / machine learning

- Coordinate descent:
  useful for large p and (unstructured) sparse matrix X, *e.g.*, for text encoding Friedman *et al.* (2007)
  Conclusion: standard approach in machine learning/statistics

The Lasso: algorithmic point of view

Commonly used algorithms for solving this **convex** program:

- Homotopy method - LARS:
  efficient for small p Osborne *et al.* (2000), Efron *et al.* (2004) and to get full path (*i.e.*, the full $\lambda \to \hat{\beta}^{(\lambda)}$)
  Limitation: do not generalize to other data-fitting term, potentially too many kinks Mairal and Yu (2012) (up to 3^p)

- (F)ISTA, Forward - Backward, proximal algorithm:
  useful in signal processing where $r \to X^\top r$ is cheap to compute (*e.g.*, FFT, Fast Wavelet Transform, etc.) Beck and Teboulle (2009)
  Limitation: unstructured X in statistics / machine learning

- Coordinate descent:
  useful for large p and (unstructured) sparse matrix X, *e.g.*, for text encoding Friedman *et al.* (2007)
  Conclusion: standard approach in machine learning/statistics

Coordinate Descent

<u>Goal</u>: find a solution for $\arg\min_{\beta \in \mathbb{R}^p} f(\beta) := \|y - X\beta\|^2/2 + \lambda \|\beta\|_1$

Algorithm: (Block) coordinate descent

Input : f, number or epochs K (or pass over the data)

Initialization: $k = 0$ and $\beta^{(k)} = 0 \in \mathbb{R}^p$ (or warm start)

Coordinate Descent

Goal: find a solution for $\underset{\beta \in \mathbb{R}^p}{\arg\min}\, f(\beta) := \|y - X\beta\|^2/2 + \lambda\|\beta\|_1$

Algorithm: (Block) coordinate descent

Input : f, number or epochs K (or pass over the data)
Initialization: $k = 0$ and $\beta^{(k)} = 0 \in \mathbb{R}^p$ (or warm start)
for $k = 1, \ldots, K$ **do**

Coordinate Descent

Goal: find a solution for $\arg\min_{\beta \in \mathbb{R}^p} f(\beta) := \|y - X\beta\|^2/2 + \lambda \|\beta\|_1$

Algorithm: (Block) coordinate descent

Input : f, number or epochs K (or pass over the data)

Initialization: $k = 0$ and $\beta^{(k)} = 0 \in \mathbb{R}^p$ (or warm start)

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

$$\beta_1^{(k)} \leftarrow \arg\min_{\beta_1 \in \mathbb{R}} f(\beta_1, \beta_2^{(k-1)}, \beta_3^{(k-1)}, \ldots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

Coordinate Descent

<u>Goal</u>: find a solution for $\underset{\beta \in \mathbb{R}^p}{\arg\min} f(\beta) := \|y - X\beta\|^2 / 2 + \lambda \|\beta\|_1$

Algorithm: (Block) coordinate descent

Input : f, number or epochs K (or pass over the data)

Initialization: $k = 0$ and $\beta^{(k)} = 0 \in \mathbb{R}^p$ (or warm start)

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

$$\beta_1^{(k)} \leftarrow \underset{\beta_1 \in \mathbb{R}}{\arg\min} f(\beta_1 \quad, \beta_2^{(k-1)}, \beta_3^{(k-1)}, \dots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$$\beta_2^{(k)} \leftarrow \underset{\beta_2 \in \mathbb{R}}{\arg\min} f(\beta_1^{(k)}, \beta_2 \qquad, \beta_3^{(k-1)}, \dots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

Coordinate Descent

Goal: find a solution for $\arg\min_{\beta \in \mathbb{R}^p} f(\beta) := \|y - X\beta\|^2/2 + \lambda \|\beta\|_1$

Algorithm: (Block) coordinate descent

Input : f, number or epochs K (or pass over the data)

Initialization: $k = 0$ and $\beta^{(k)} = 0 \in \mathbb{R}^p$ (or warm start)

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

$$\beta_1^{(k)} \leftarrow \arg\min_{\beta_1 \in \mathbb{R}} f(\beta_1, \beta_2^{(k-1)}, \beta_3^{(k-1)}, \ldots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$$\beta_2^{(k)} \leftarrow \arg\min_{\beta_2 \in \mathbb{R}} f(\beta_1^{(k)}, \beta_2, \beta_3^{(k-1)}, \ldots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$$\beta_3^{(k)} \leftarrow \arg\min_{\beta_3 \in \mathbb{R}} f(\beta^{(k)}, \beta_2^{(k)}, \beta_3, \ldots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

Coordinate Descent

Goal: find a solution for $\underset{\beta \in \mathbb{R}^p}{\arg\min}\, f(\beta) := \|y - X\beta\|^2/2 + \lambda \|\beta\|_1$

Algorithm: (Block) coordinate descent

Input : f, number or epochs K (or pass over the data)
Initialization: $k = 0$ and $\beta^{(k)} = 0 \in \mathbb{R}^p$ (or warm start)
for $k = 1, \dots, K$ **do**

$$\beta_1^{(k)} \leftarrow \underset{\beta_1 \in \mathbb{R}}{\arg\min}\, f(\beta_1, \beta_2^{(k-1)}, \beta_3^{(k-1)}, \dots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$$\beta_2^{(k)} \leftarrow \underset{\beta_2 \in \mathbb{R}}{\arg\min}\, f(\beta_1^{(k)}, \beta_2, \beta_3^{(k-1)}, \dots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$$\beta_3^{(k)} \leftarrow \underset{\beta_3 \in \mathbb{R}}{\arg\min}\, f(\beta^{(k)}, \beta_2^{(k)}, \beta_3, \dots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$\vdots$

Coordinate Descent

Goal: find a solution for $\arg\min_{\beta \in \mathbb{R}^p} f(\beta) := \|y - X\beta\|^2/2 + \lambda \|\beta\|_1$

Algorithm: (Block) coordinate descent

Input : f, number or epochs K (or pass over the data)

Initialization: $k = 0$ and $\beta^{(k)} = 0 \in \mathbb{R}^p$ (or warm start)

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

$$\beta_1^{(k)} \leftarrow \arg\min_{\beta_1 \in \mathbb{R}} f(\beta_1, \beta_2^{(k-1)}, \beta_3^{(k-1)}, \dots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$$\beta_2^{(k)} \leftarrow \arg\min_{\beta_2 \in \mathbb{R}} f(\beta_1^{(k)}, \beta_2, \beta_3^{(k-1)}, \dots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$$\beta_3^{(k)} \leftarrow \arg\min_{\beta_3 \in \mathbb{R}} f(\beta^{(k)}, \beta_2^{(k)}, \beta_3, \dots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$\vdots$

$$\beta_p^{(k)} \leftarrow \arg\min_{\beta_p \in \mathbb{R}} f(\beta_1^{(k)}, \beta_2^{(k)}, \beta_3^{(k)}, \dots, \beta_{p-1}^{(k)}, \beta_p)$$

Coordinate Descent

<u>Goal</u>: find a solution for $\underset{\beta \in \mathbb{R}^p}{\arg\min}\, f(\beta) := \|y - X\beta\|^2/2 + \lambda \|\beta\|_1$

Algorithm: (Block) coordinate descent

Input : f, number or epochs K (or pass over the data)

Initialization: $k = 0$ and $\beta^{(k)} = 0 \in \mathbb{R}^p$ (or warm start)

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

$$\beta_1^{(k)} \leftarrow \underset{\beta_1 \in \mathbb{R}}{\arg\min}\, f(\beta_1, \beta_2^{(k-1)}, \beta_3^{(k-1)}, \dots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$$\beta_2^{(k)} \leftarrow \underset{\beta_2 \in \mathbb{R}}{\arg\min}\, f(\beta_1^{(k)}, \beta_2, \beta_3^{(k-1)}, \dots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$$\beta_3^{(k)} \leftarrow \underset{\beta_3 \in \mathbb{R}}{\arg\min}\, f(\beta_1^{(k)}, \beta_2^{(k)}, \beta_3, \dots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$\vdots$

$$\beta_p^{(k)} \leftarrow \underset{\beta_p \in \mathbb{R}}{\arg\min}\, f(\beta_1^{(k)}, \beta_2^{(k)}, \beta_3^{(k)}, \dots, \beta_{p-1}^{(k)}, \beta_p)$$

Output : $\beta^{(K)}$

Coordinate Descent

<u>Goal</u>: find a solution for $\arg\min_{\beta \in \mathbb{R}^p} f(\beta) := \|y - X\beta\|^2/2 + \lambda \|\beta\|_1$

Algorithm: (Block) coordinate descent

Input : f, number or epochs K (or pass over the data)

Initialization: $k = 0$ and $\beta^{(k)} = 0 \in \mathbb{R}^p$ (or warm start)

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

$$\beta_1^{(k)} \leftarrow \arg\min_{\beta_1 \in \mathbb{R}} f(\beta_1, \beta_2^{(k-1)}, \beta_3^{(k-1)}, \dots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$$\beta_2^{(k)} \leftarrow \arg\min_{\beta_2 \in \mathbb{R}} f(\beta_1^{(k)}, \beta_2, \beta_3^{(k-1)}, \dots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$$\beta_3^{(k)} \leftarrow \arg\min_{\beta_3 \in \mathbb{R}} f(\beta^{(k)}, \beta_2^{(k)}, \beta_3, \dots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$\vdots$

$$\beta_p^{(k)} \leftarrow \arg\min_{\beta_p \in \mathbb{R}} f(\beta_1^{(k)}, \beta_2^{(k)}, \beta_3^{(k)}, \dots, \beta_{p-1}^{(k)}, \beta_p)$$

Output : $\beta^{(K)}$

Break if : stable iterates/objective, small duality gap,...

Illustration of convergence (convex case)

- Convergence toward global minimum for **smooth** (gradient Lipschitz) functions, *cf.* Tseng (2001)

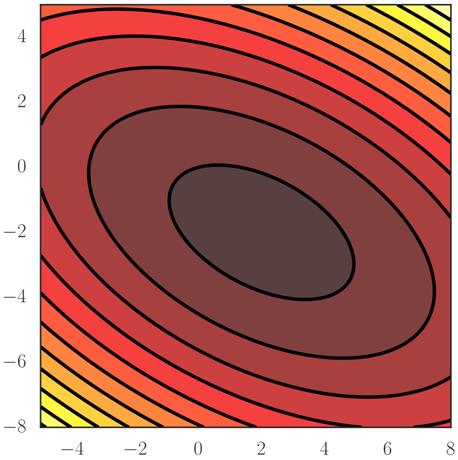


Illustration of convergence (convex case)

- Convergence toward global minimum for **smooth** (gradient Lipschitz) functions, *cf.* Tseng (2001)

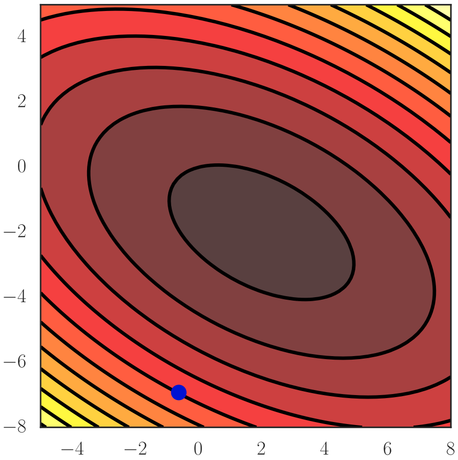


Illustration of convergence (convex case)

- Convergence toward global minimum for **smooth** (gradient Lipschitz) functions, *cf.* Tseng (2001)

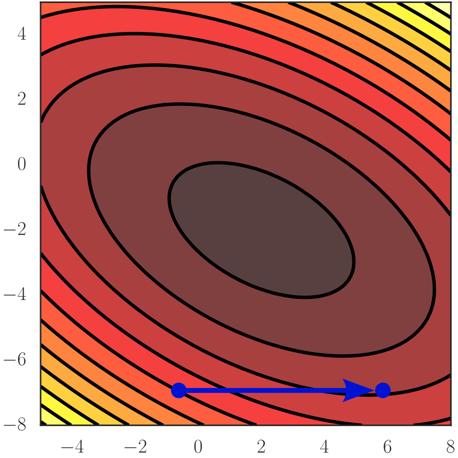


Illustration of convergence (convex case)

- Convergence toward global minimum for **smooth** (gradient Lipschitz) functions, *cf.* Tseng (2001)

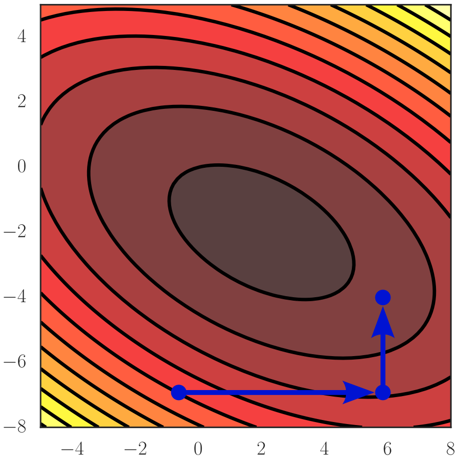


Illustration of convergence (convex case)

- Convergence toward global minimum for **smooth** (gradient Lipschitz) functions, *cf.* Tseng (2001)

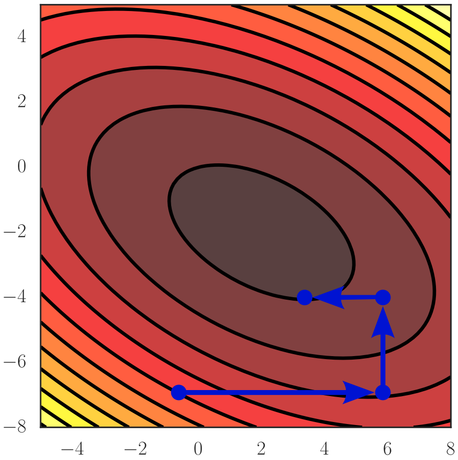


Illustration of convergence (convex case)

- Convergence toward global minimum for **smooth** (gradient Lipschitz) functions, *cf.* Tseng (2001)

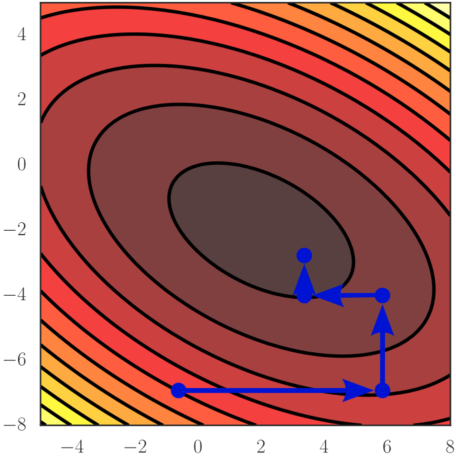


Illustration of convergence (convex case)

- Convergence toward global minimum for **separable** functions, *i.e.*, $f(\beta) = \sum_{j=1}^{p} f_j(\beta_j)$, *cf.* Tseng (2001)

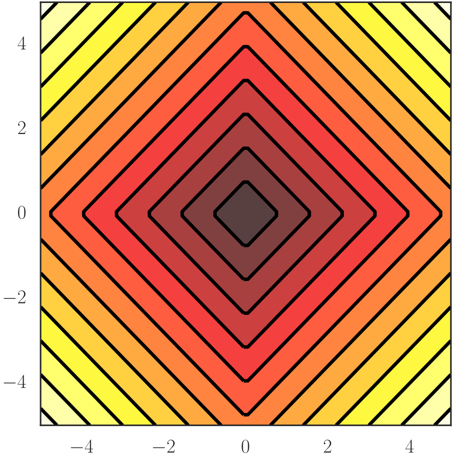


Illustration of convergence (convex case)

- Convergence toward global minimum for **separable** functions, *i.e.*, $f(\beta) = \sum_{j=1}^{p} f_j(\beta_j)$, *cf.* Tseng (2001)

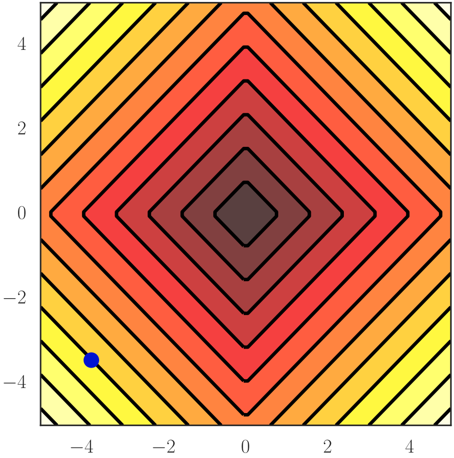


Illustration of convergence (convex case)

- Convergence toward global minimum for **separable** functions, *i.e.*, $f(\beta) = \sum_{j=1}^{p} f_j(\beta_j)$, *cf.* Tseng (2001)

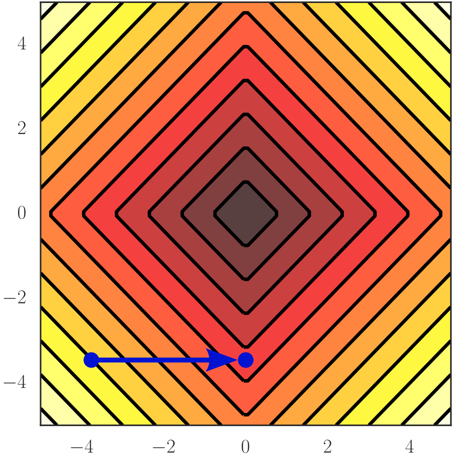


Illustration of convergence (convex case)

- Convergence toward global minimum for **separable** functions, *i.e.*, $f(\beta) = \sum_{j=1}^{p} f_j(\beta_j)$, *cf.* Tseng (2001)

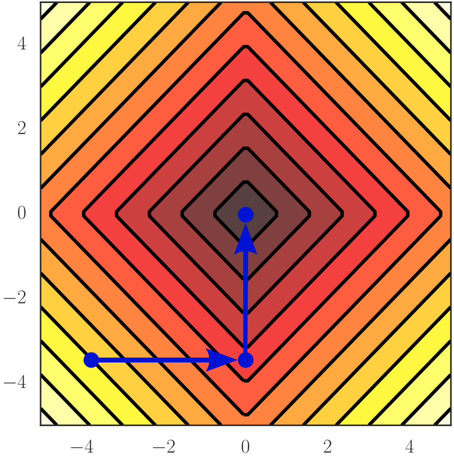


Illustration of convergence (convex case)

- Convergence toward global minimum for sums of a **smooth** function + a **separable** function, *cf.* Tseng (2001)

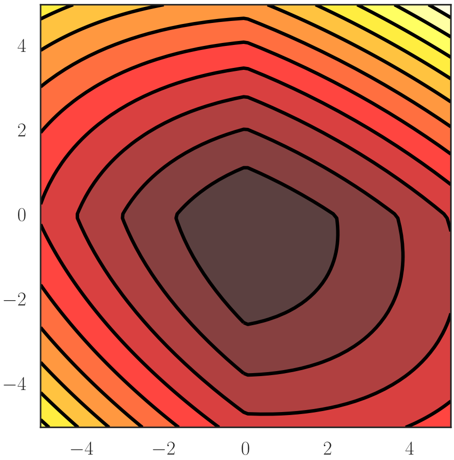


Illustration of convergence (convex case)

- Convergence toward global minimum for sums of a **smooth** function + a **separable** function, *cf.* Tseng (2001)

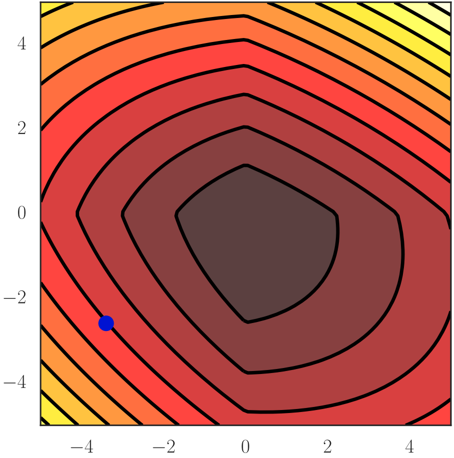


Illustration of convergence (convex case)

- Convergence toward global minimum for sums of a **smooth** function + a **separable** function, *cf.* Tseng (2001)

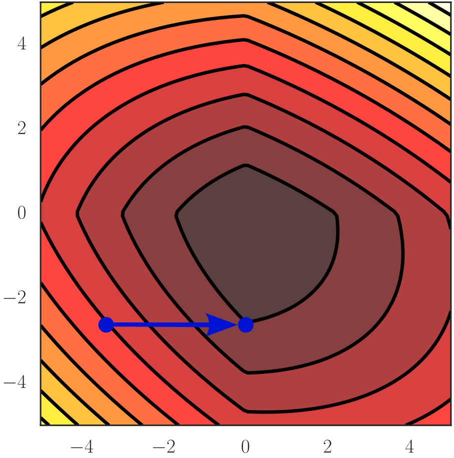


Illustration of convergence (convex case)

- Convergence toward global minimum for sums of a **smooth** function + a **separable** function, *cf.* Tseng (2001)

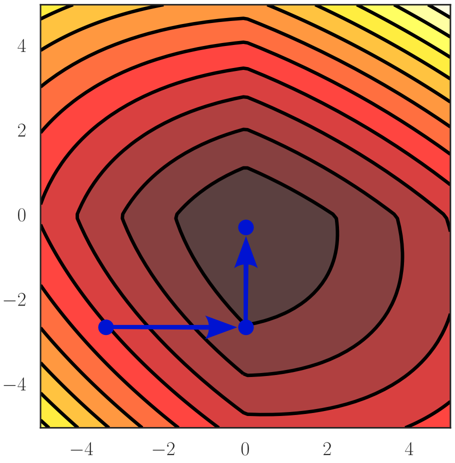


Illustration of convergence (convex case)

- Convergence toward global minimum for sums of a **smooth** function + a **separable** function, *cf.* Tseng (2001)

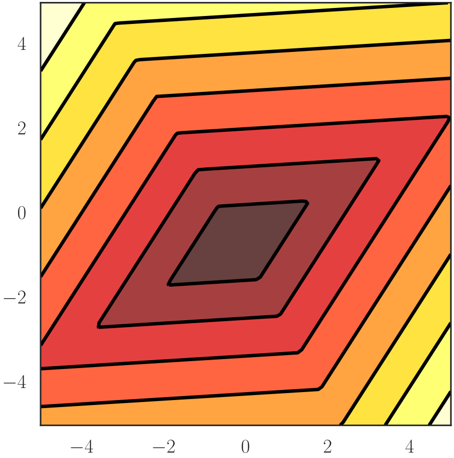


Illustration of non-convergence (convex case)

- **Beware**: otherwise convergence no longer guaranteed even for convex cases

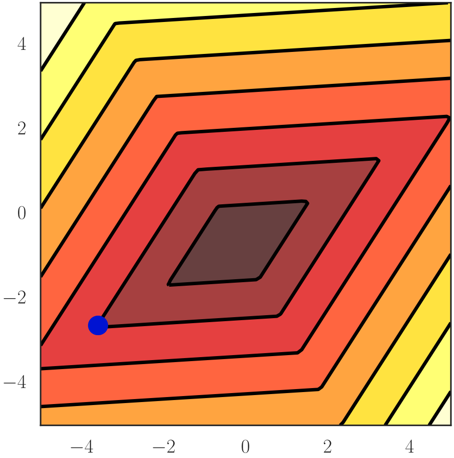


Illustration of non-convergence (convex case)

- **Beware**: otherwise convergence no longer guaranteed even for convex cases

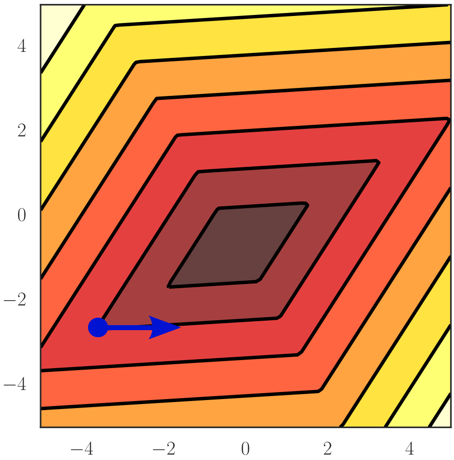
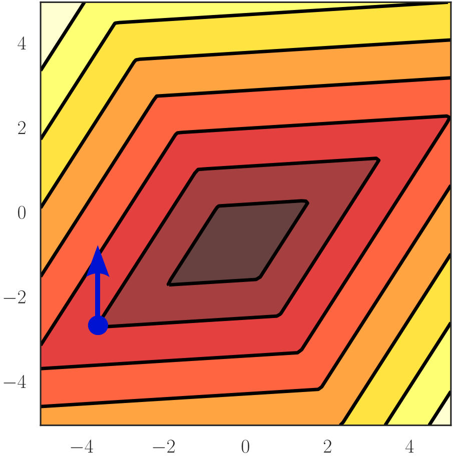
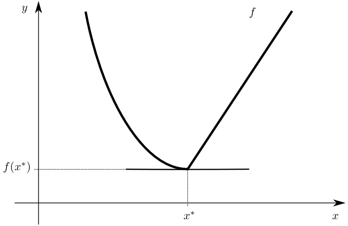


Illustration of non-convergence (convex case)

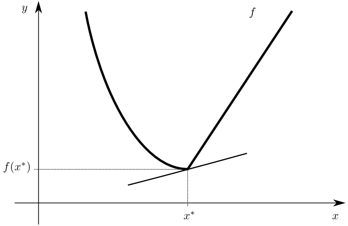
- **Beware**: otherwise convergence no longer guaranteed even for convex cases



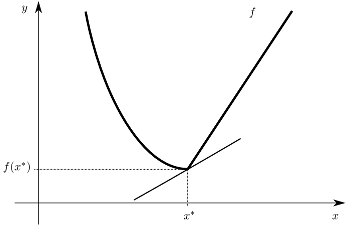
Sub-gradients / sub-differential



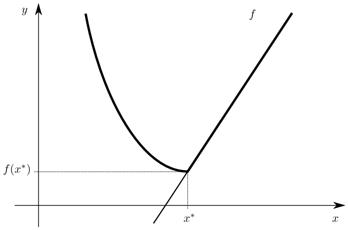
Sub-gradients / sub-differential



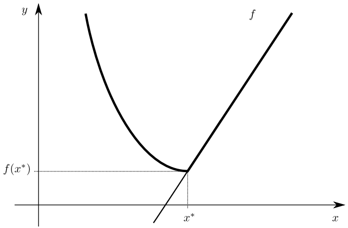
Sub-gradients / sub-differential



Sub-gradients / sub-differential



Sub-gradients / sub-differential



Definition: sub-gradient / sub-differential

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

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

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

Rem: recover the gradient when the sub-gradient is a singleton

Fermat's rule: first order condition

Theorem

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

Proof: use the definition of sub-gradients:

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

Fermat's rule: first order condition

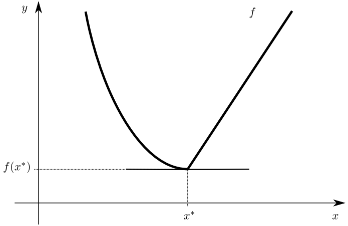
Theorem

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

Proof: use the definition of sub-gradients:

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

Rem: correspond to a "horizontal" tangent



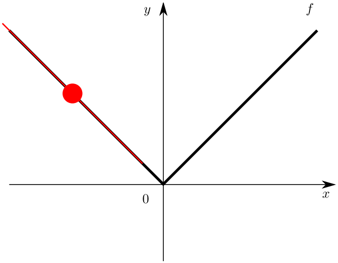
Absolute value / ℓ_1 case

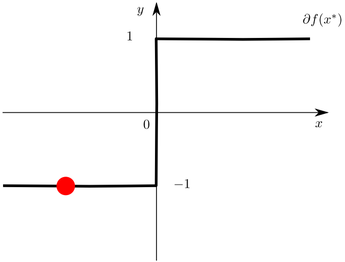
Function (abs):

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

Sub-differential (sign)

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

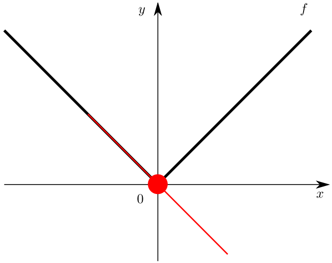




Absolute value / ℓ_1 case

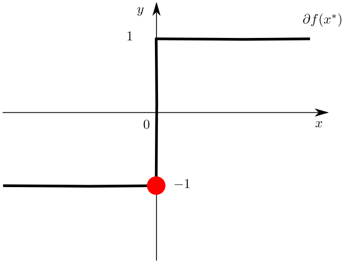
Function (abs):

$$f : \begin{cases} \mathbb{R} & \longrightarrow \mathbb{R} \\ x & \longmapsto |x| \end{cases}$$



Sub-differential (sign)

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



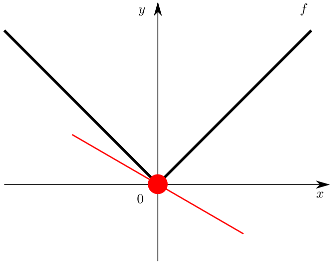
Absolute value / ℓ_1 case

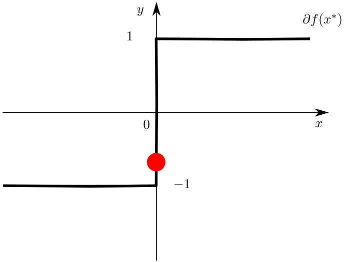
Function (abs):

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

Sub-differential (sign)

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

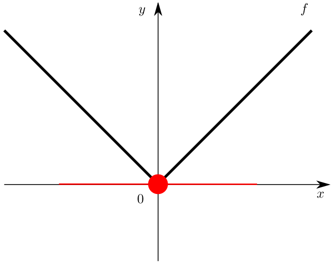




Absolute value / ℓ_1 case

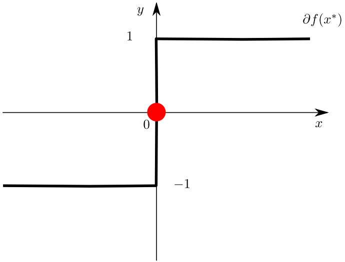
Function (abs):

$$f : \begin{cases} \mathbb{R} & \longrightarrow \mathbb{R} \\ x & \longmapsto |x| \end{cases}$$



Sub-differential (sign)

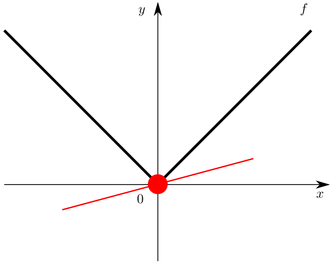
$$\partial f(x^*) = \begin{cases} \{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0 \end{cases}$$



Absolute value / ℓ_1 case

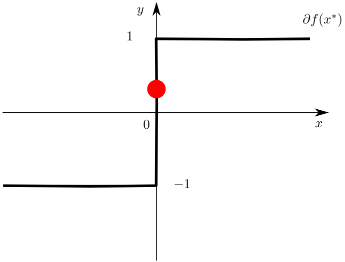
Function (abs):

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



Sub-differential (sign)

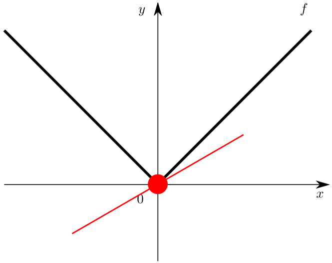
$$\partial f(x^*) = \begin{cases}\{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0\end{cases}$$



Absolute value / ℓ_1 case

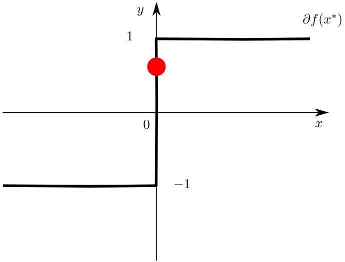
Function (abs):

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



Sub-differential (sign)

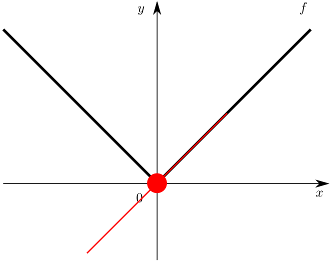
$$\partial f(x^*) = \begin{cases} \{-1\} & \text{if } x^* \in]-\infty, 0[\\ \{1\} & \text{if } x^* \in]0, \infty[\\ [-1, 1] & \text{if } x^* = 0 \end{cases}$$



Absolute value / ℓ_1 case

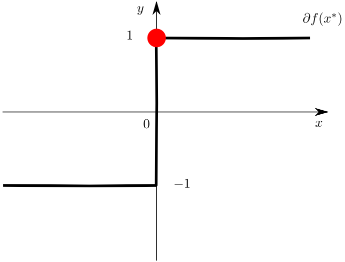
Function (abs):

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



Sub-differential (sign)

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



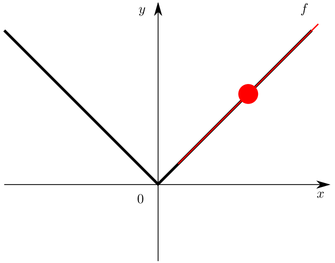
Absolute value / ℓ_1 case

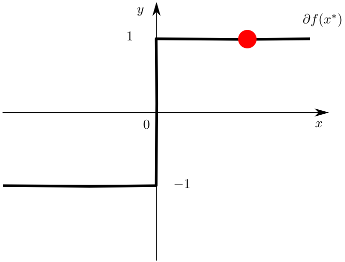
Function (abs):

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

Sub-differential (sign)

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



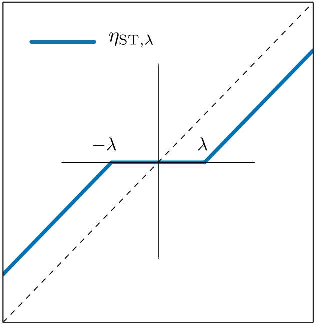


Soft-Thresholding

Closed form solution for 1D-problem $(p = 1)$: **Soft-Thresholding**

$$\eta_{\mathrm{ST},\lambda}(y) := \mathop{\arg\min}_{\beta \in \mathbb{R}} \left(\frac{(y-\beta)^2}{2} + \lambda|\beta| \right)$$
$$= \mathrm{sign}(y)(|y| - \lambda)_+$$

with $(\cdot)_+ := \max(0, \cdot)$



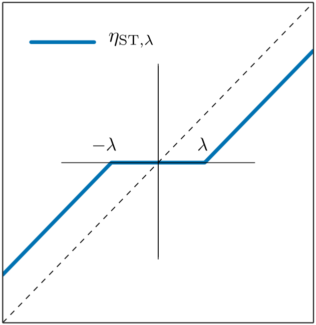
Proof: sub-differential of $|\cdot|$ + Fermat's rule

Soft-Thresholding

Closed form solution for 1D-problem ($p = 1$) : **Soft-Thresholding**

$$\eta_{\mathrm{ST},\lambda}(y) := \mathop{\arg\min}_{\beta \in \mathbb{R}} \left(\frac{(y-\beta)^2}{2} + \lambda|\beta| \right)$$
$$= \mathrm{sign}(y)(|y| - \lambda)_+$$

with $(\cdot)_+ := \max(0, \cdot)$



Proof: sub-differential of $|\cdot| +$ Fermat's rule

Coordinate descent update: (closed-form)

$$\beta_j \leftarrow \eta_{\mathrm{ST}, \frac{\lambda}{\|\mathbf{x_j}\|^2}} \left(\beta_j - \frac{\mathbf{x}_j^\top (X\beta - y)}{\|\mathbf{x}_j\|^2} \right)$$

Table of Contents

Motivation - notation

Optimization and convexity reminders

Optimization property for the Lasso

Safe rules

Gap safe rules

Coordinate descent implementation

Dual problem Kim *et al.* (2007)

Primal function : $$P_\lambda(\beta) = \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1$$

Dual problem : $$\hat{\theta}^{(\lambda)} = \mathop{\arg\max}_{\theta \in \Delta_X} \underbrace{\frac{1}{2}\|y\|^2 - \frac{\lambda^2}{2}\left\|\theta - \frac{y}{\lambda}\right\|^2}_{=D_\lambda(\theta)}$$

Dual feasible set : $$\Delta_X = \left\{\theta \in \mathbb{R}^n \ : \ |\mathbf{x}_j^\top \theta| \leqslant 1, \forall j \in [p]\right\}$$

- $\Delta_X = \left\{\theta \in \mathbb{R}^n : \|X^\top \theta\|_\infty \leqslant 1\right\}$ is a polyhedral set, *i.e.*, a finite intersection of closed half-spaces
- The (unique) dual solution is the **projection** of y/λ over Δ_X:

$$\hat{\theta}^{(\lambda)} = \mathop{\arg\min}_{\theta \in \Delta_X} \left\|\frac{y}{\lambda} - \theta\right\|^2 := \Pi_{\Delta_X}\left(\frac{y}{\lambda}\right)$$

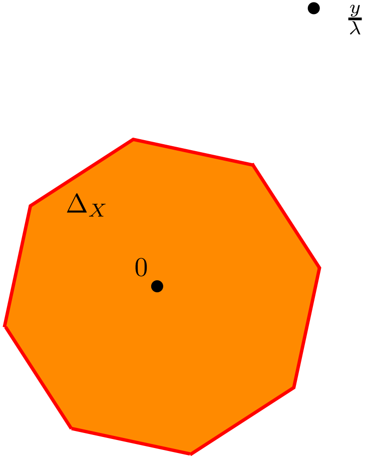
Sketch of proof (in two slides)

Geometric interpretation

The dual optimal solution is the projection of y/λ over the dual feasible set $\Delta_X = \left\{\theta \in \mathbb{R}^n : \|X^\top \theta\|_\infty \leqslant 1\right\} : \hat{\theta}^{(\lambda)} = \Pi_{\Delta_X}(y/\lambda)$

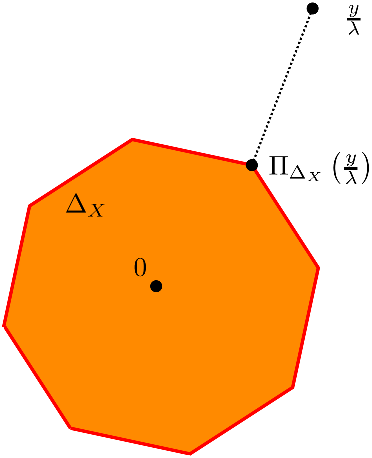
Geometric interpretation

The dual optimal solution is the projection of y/λ over the dual feasible set $\Delta_X = \left\{\theta \in \mathbb{R}^n : \|X^\top \theta\|_\infty \leqslant 1\right\}$: $\hat{\theta}^{(\lambda)} = \Pi_{\Delta_X}(y/\lambda)$



Geometric interpretation

The dual optimal solution is the projection of y/λ over the dual feasible set $\Delta_X = \left\{\theta \in \mathbb{R}^n : \|X^\top \theta\|_\infty \leqslant 1\right\}$: $\hat{\theta}^{(\lambda)} = \Pi_{\Delta_X}(y/\lambda)$



Sketch of proof for the dual formulation

$$\min_{\beta \in \mathbb{R}^p} \underbrace{\frac{1}{2}\|y - X\beta\|^2}_{g(y - X\beta)} + \lambda \underbrace{\|\beta\|_1}_{\Omega(\beta)} \Leftrightarrow \min_{\beta \in \mathbb{R}^p, z \in \mathbb{R}^n} \begin{cases} g(z) + \lambda\Omega(\beta) \\ \text{s.t.} \quad z = y - X\beta \end{cases}$$

Lagrangian : $\quad \mathcal{L}(z, \beta, \theta) := g(z) + \lambda\Omega(\beta) + \lambda\theta^\top(y - X\beta - z).$

Find a **Lagrangian** saddle point $(z^\star, \hat{\beta}^{(\lambda)}, \hat{\theta}^{(\lambda)})$ (Strong duality):

$$\min_{\beta \in \mathbb{R}^p, z \in \mathbb{R}^n} \max_{\theta \in \mathbb{R}^n} \mathcal{L}(z, \beta, \theta) = \max_{\theta \in \mathbb{R}^n} \min_{\beta \in \mathbb{R}^p, z \in \mathbb{R}^n} \mathcal{L}(z, \beta, \theta) =$$

$$\max_{\theta \in \mathbb{R}^n} \left\{ \min_{z \in \mathbb{R}^n} [g(z) - \lambda\theta^\top z] + \min_{\beta \in \mathbb{R}^p} [\lambda\Omega(\beta) - \lambda\theta^\top X\beta] + \lambda\theta^\top y \right\} =$$

$$\max_{\theta \in \mathbb{R}^n} \left\{ -g^*(\lambda\theta) - \lambda\Omega^*(X^\top\theta) + \lambda\theta^\top y \right\}$$

Provided a few conjugate properties, it is the formulation asserted

Fenchel conjugation

For any $g : \mathbb{R}^n \to \mathbb{R}$, the Fenchel conjugate g^* is defined as

$$\boxed{g^*(z) = \sup_{x \in \mathbb{R}^n} x^\top z - g(x)}$$

- If $g(\cdot) = \|\cdot\|^2/2$ then $g^*(\cdot) = g(\cdot)$
- If $g(\cdot) = \Omega(\cdot)$ is a norm, then $g^*(\cdot) = \iota_{\mathcal{B}_*(0,1)}(\cdot)$, *i.e.*, it is the indicator function of the dual norm unit ball, where the **dual norm** Ω^* is defined by:

$$\Omega^*(z) = \sup_{x:\ \Omega(x) \leqslant 1} x^\top z = \iota^*_{\mathcal{B}(0,1)}$$

and

$$\iota_{\mathcal{B}}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{B} \\ +\infty & \text{otherwise} \end{cases}, \text{ where } \mathcal{B} = \{x \in \mathbb{R}^n : \Omega(x) \leqslant 1\}$$

Fermat rule / KKT conditions

- **Primal solution :** $\hat{\beta}^{(\lambda)} \in \mathbb{R}^p$
- **Dual solution :** $\hat{\theta}^{(\lambda)} \in \Delta_X \subset \mathbb{R}^n$

Primal/Dual link: $\boxed{y = X\hat{\beta}^{(\lambda)} + \lambda\hat{\theta}^{(\lambda)}}$

Necessary and sufficient optimality conditions:

KKT/Fermat: $$\boxed{\forall j \in [p],\ \mathbf{x}_j^\top \hat{\theta}^{(\lambda)} \in \begin{cases} \{\mathrm{sign}(\hat{\beta}_j^{(\lambda)})\} & \text{if} \quad \hat{\beta}_j^{(\lambda)} \neq 0, \\ [-1, 1] & \text{if} \quad \hat{\beta}_j^{(\lambda)} = 0. \end{cases}}$$

(<u>Sketch of proof next slide</u>)

<u>"Mother" of safe rules</u>: $(0, \frac{y}{\lambda}) \in \mathbb{R}^p \times \mathbb{R}^n$ is a primal/dual solution whenever $\lambda \geqslant \|X^\top y\|_\infty =: \lambda_{\max}$, (all β_j's screened-out!)

Proof Fermat/KKT + primal/dual link

Lagrangian : $\mathcal{L}(z, \beta, \theta) := \underbrace{\frac{1}{2}\|z\|^2}_{g(z)} + \lambda \underbrace{\|\beta\|_1}_{\Omega(\beta)} + \lambda\theta^\top(y - X\beta - z).$

A saddle point $(z^\star, \hat{\beta}^{(\lambda)}, \hat{\theta}^{(\lambda)})$ of the Lagrangian satisfies:

$$\begin{cases} 0 & = & \frac{\partial \mathcal{L}}{\partial z}(z^\star, \hat{\beta}^{(\lambda)}, \hat{\theta}^{(\lambda)}) = \nabla g(z^\star) = z^\star - \lambda\hat{\theta}^{(\lambda)}, \\ 0 & \in & \partial\mathcal{L}(z^\star, \cdot, \hat{\theta}^{(\lambda)})(\hat{\beta}^{(\lambda)}) = -\lambda X^\top\hat{\theta}^{(\lambda)} + \lambda\partial\Omega(\hat{\beta}^{(\lambda)}) \\ 0 & = & \frac{\partial \mathcal{L}}{\partial \theta}(z^\star, \hat{\beta}^{(\lambda)}, \hat{\theta}^{(\lambda)}) = y - X\hat{\beta}^{(\lambda)} - z^\star. \end{cases}$$

Hence, $y - X\hat{\beta}^{(\lambda)} = z^\star = \lambda\hat{\theta}^{(\lambda)}$ and $X^\top\hat{\theta}^{(\lambda)} \in \partial\Omega(\hat{\beta}^{(\lambda)})$ so

$$\forall j \in [p], \quad \mathbf{x}_j^\top\hat{\theta}^{(\lambda)} \in \partial|\cdot|(\hat{\beta}_j^{(\lambda)}) \text{ (separability)}$$

Geometric interpretation (II)

A simple dual (feasible) point: $\frac{y}{\lambda_{\max}} \in \Delta_X$ where $\lambda_{\max} = \|X^\top y\|_\infty$

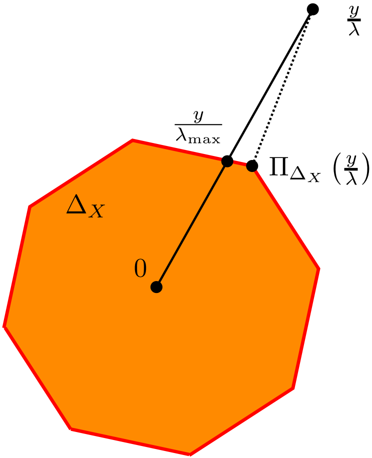


Table of Contents

Motivation - notation

Optimization and convexity reminders

Optimization property for the Lasso

Safe rules

Gap safe rules

Coordinate descent implementation

Safe screening rules El Ghaoui *et al.* (2012)

Screening thanks to Fermat's Rule: $\boxed{\text{If } |\mathbf{x}_j^\top \hat{\theta}^{(\lambda)}| < 1 \text{ then, } \hat{\beta}_j^{(\lambda)} = 0}$

Beware: $\hat{\theta}^{(\lambda)}$ is **unknown** so this not practical

Consider instead a **safe region** $\mathcal{C} \subset \mathbb{R}^n$ *i.e.*, $\mathcal{C} \ni \hat{\theta}^{(\lambda)}$:

safe rule :

$$\boxed{\text{If } \sup_{\theta \in \mathcal{C}} |\mathbf{x}_j^\top \theta| < 1 \text{ then } \hat{\beta}_j^{(\lambda)} = 0} \qquad (\star)$$

Consequence: if safe rule satisfied, $\mathbf{x}_j$ can be "safely removed"

Safe screening rules El Ghaoui *et al.* (2012)

Screening thanks to Fermat's Rule: $\boxed{\text{If } |\mathbf{x}_j^\top \hat{\theta}^{(\lambda)}| < 1 \text{ then, } \hat{\beta}_j^{(\lambda)} = 0}$

Beware: $\hat{\theta}^{(\lambda)}$ is **unknown** so this not practical

Consider instead a **safe region** $\mathcal{C} \subset \mathbb{R}^n$ *i.e.*, $\mathcal{C} \ni \hat{\theta}^{(\lambda)}$:

$$\textbf{safe rule}: \quad \boxed{\text{If } \sup_{\theta \in \mathcal{C}} |\mathbf{x}_j^\top \theta| < 1 \text{ then } \hat{\beta}_j^{(\lambda)} = 0} \qquad (\star)$$

Consequence: if safe rule satisfied, $\mathbf{x}_j$ can be "safely removed"

Goal: find $\mathcal{C}$

- as narrow as possible containing $\hat{\theta}^{(\lambda)}$
- with $\begin{cases} \mathbb{R}^n & \mapsto \mathbb{R}^+ \\ \mathbf{x} & \rightarrow \sup_{\theta \in \mathcal{C}} |\mathbf{x}^\top \theta| \end{cases}$ cheap to compute

Safe sphere rules

Let $\mathcal{C} = B(c, r)$ be a ball of **center** $c \in \mathbb{R}^n$ and **radius** $r > 0$, then

$$\sup_{\theta \in \mathcal{C}} |\mathbf{x}^\top \theta| = |\mathbf{x}^\top c| + r\|\mathbf{x}\|$$

safe sphere rule: $\boxed{\text{If } |\mathbf{x}_j^\top c| + r\|\mathbf{x}_j\| < 1 \text{ then } \hat{\beta}_j^{(\lambda)} = 0}$

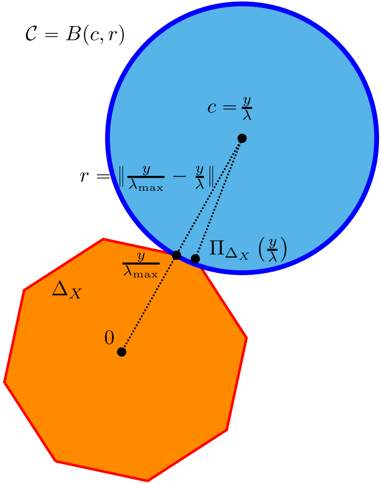
Screening cost:

- one dot product in $\mathbb{R}^n$
- norm computation "free": pre computed / normalized

New objective:

- find r as small as possible
- find c as close to $\hat{\theta}^{(\lambda)}$ as possible

Static safe rules: El Ghaoui *et al.* (2012)



Properties of static safe rules

Interest: can be useful prior any optimization (only $\lambda_{\max}$ needed)

Static safe region: $\mathcal{C} = B(c, r) = B(y/\lambda, \|y/\lambda_{\max} - y/\lambda\|)$

Static safe rule: If $|\mathbf{x}_j^\top y| < \lambda \left(1 - \left\| \frac{y}{\lambda_{\max}} - \frac{y}{\lambda} \right\| \|\mathbf{x}_j\| \right)$ then $\hat{\beta}_j^{(\lambda)} = 0$

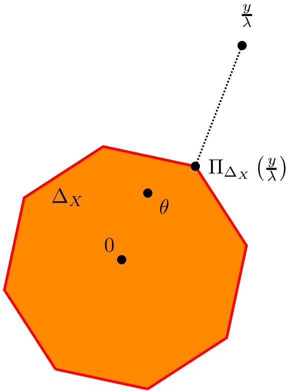
Statistical interpretation: static screening = correlation screening for **variable selection**: "If $|\mathbf{x}_j^\top y|$ small, discard $\mathbf{x}_j$" (for $\|\mathbf{x}_j\| = 1$):

$$\boxed{\text{If } |\mathbf{x}_j^\top y| < C_{X,y} \text{ then } \hat{\beta}_j^{(\lambda)} = 0}$$

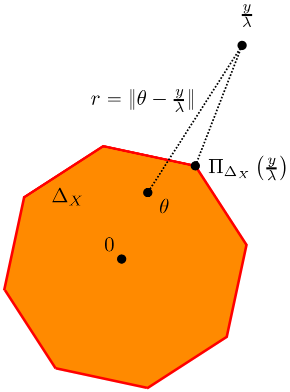
Limit: static screening **useless** for small λ's , *i.e.*, **no feature** can be screened-out

$$\frac{\lambda}{\lambda_{\max}} \leqslant C'_{X,y} = \min_{j \in [p]} \left(\frac{1 + |\mathbf{x}_j^\top y| / (\|\mathbf{x}_j\| \|y\|)}{1 + \lambda_{\max} / (\|\mathbf{x}_j\| \|y\|)} \right)$$

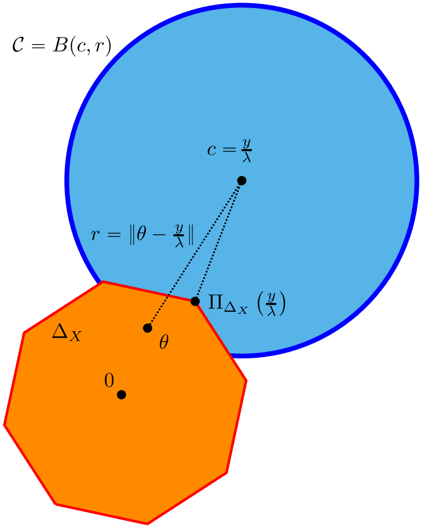
Dynamic safe rules Bonnefoy *et al.* (2014)



Dynamic safe rules Bonnefoy *et al.* (2014)



Dynamic safe rules Bonnefoy *et al.* (2014)



Dynamic safe rule

Dynamic rules: build iteratively $\theta_k \in \Delta_X$, as the solver proceeds to get refined safe rules Bonnefoy *et al.* (2014, 2015)

$$\text{Remind link at optimum:} \quad \lambda \hat{\theta}^{(\lambda)} = y - X \hat{\beta}^{(\lambda)}$$

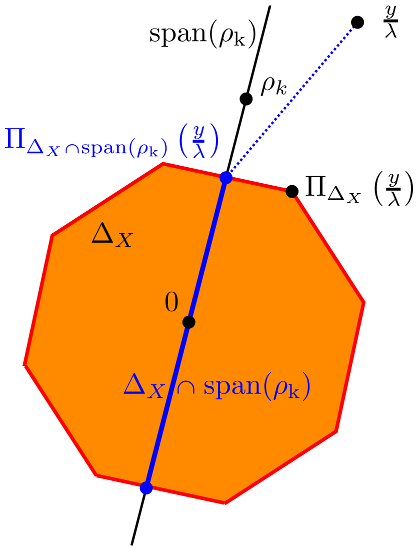
$$\text{Current } \textbf{residual} \text{ for primal point } \beta_k\text{:} \quad \rho_k = y - X\beta_k$$

Dual candidate: choose θ_k proportional to the residual

$$\begin{aligned} \theta_k &= \alpha_k \rho_k, \\ \text{where} \quad \alpha_k &= \min \left[\max \left(\frac{y^\top \rho_k}{\lambda \left\| \rho_k \right\|^2}, \frac{-1}{\| X^\top \rho_k \|_\infty} \right), \frac{1}{\| X^\top \rho_k \|_\infty} \right]. \end{aligned}$$

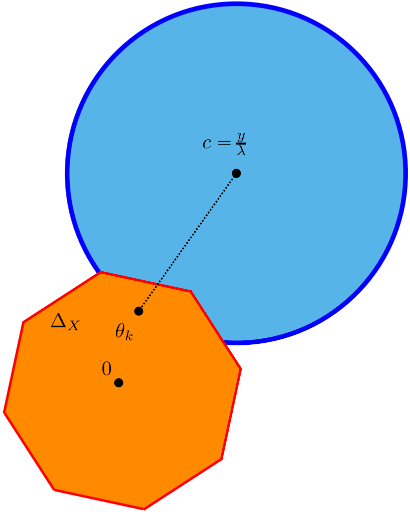
Motivation: projecting over the convex set $\Delta_X \cap \mathrm{Span}(\rho_k)$ is “relatively” cheap (cost: p dot products in $\mathbb{R}^n$)

Creating dual points: project on a segment



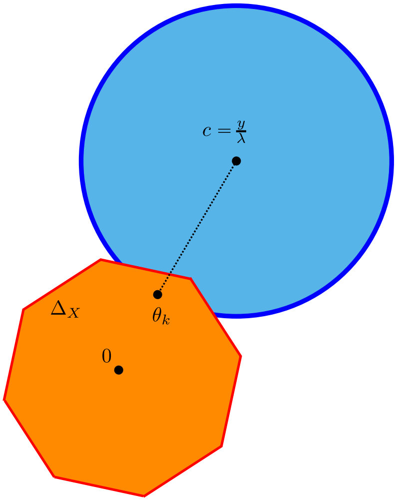
Limits of previous dynamic rules

For $B(c, r) = B(\theta_k, r_k)$ with $r_k = \|\theta_k - y/\lambda\|$, the radius does not converge to zero, even when $\beta_k \to \hat{\beta}^{(\lambda)}$ and $\theta_k \to \hat{\theta}^{(\lambda)}$ (converging solver). The limiting safe sphere is



Limits of previous dynamic rules

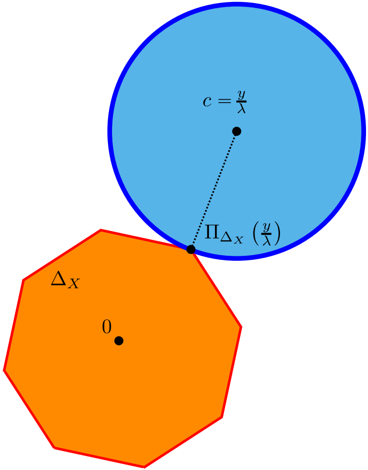
For $B(c, r) = B(\theta_k, r_k)$ with $r_k = \|\theta_k - y/\lambda\|$, the radius does not converge to zero, even when $\beta_k \to \hat{\beta}^{(\lambda)}$ and $\theta_k \to \hat{\theta}^{(\lambda)}$ (converging solver). The limiting safe sphere is



Limits of previous dynamic rules

For $B(c, r) = B(\theta_k, r_k)$ with $r_k = \|\theta_k - y/\lambda\|$, the radius does not converge to zero, even when $\beta_k \to \hat{\beta}^{(\lambda)}$ and $\theta_k \to \hat{\theta}^{(\lambda)}$ (converging solver). The limiting safe sphere is

$$\mathcal{C} = B(y/\lambda, \|\Pi_{\Delta_X}(y/\lambda) - y/\lambda\|)$$



Duality Gap properties

- Primal objective: P_λ
- Dual objective: D_λ
- Primal solution: $\hat{\beta}^{(\lambda)} \in \mathbb{R}^p$
- Primal solution: $\hat{\theta}^{(\lambda)} \in \Delta_X \subset \mathbb{R}^n$,

Duality gap: for any $\beta \in \mathbb{R}^p, \theta \in \Delta_X$, $G_\lambda(\beta, \theta) = P_\lambda(\beta) - D_\lambda(\theta)$

$$G_\lambda(\beta, \theta) = \frac{1}{2} \left\| X\beta - y \right\|^2 + \lambda \left\| \beta \right\|_1 - \left(\frac{1}{2} \left\| y \right\|^2 - \frac{\lambda^2}{2} \left\| \theta - \frac{y}{\lambda} \right\|^2 \right)$$

Strong duality: for any $\beta \in \mathbb{R}^p, \theta \in \Delta_X$,

$$D_\lambda(\theta) \leqslant D_\lambda(\hat{\theta}^{(\lambda)}) = P_\lambda(\hat{\beta}^{(\lambda)}) \leqslant P_\lambda(\beta)$$

Consequences:

- $G_\lambda(\beta, \theta) \geqslant 0$, for any $\beta \in \mathbb{R}^p, \theta \in \Delta_X$ (**weak duality**)
- $G_\lambda(\beta, \theta) \leqslant \epsilon \Rightarrow P_\lambda(\beta) - P_\lambda(\hat{\beta}^{(\lambda)}) \leqslant \epsilon$ (stopping criterion!)

Table of Contents

Motivation - notation

Optimization and convexity reminders

Optimization property for the Lasso

Safe rules

Gap safe rules

Coordinate descent implementation

Gap Safe sphere

For any $\beta \in \mathbb{R}^p, \theta \in \Delta_X$

$$G_\lambda(\beta,\theta) = \frac{1}{2}\left\|X\beta - y\right\|^2 + \lambda\left\|\beta\right\|_1 - \left(\frac{1}{2}\left\|y\right\|^2 - \frac{\lambda^2}{2}\left\|\theta - \frac{y}{\lambda}\right\|^2\right)$$

Gap Safe ball: $\boxed{B(\theta, r_\lambda(\beta,\theta)),\ \text{where } r_\lambda(\beta,\theta) = \sqrt{2G_\lambda(\beta,\theta)}/\lambda}$

Rem: If $\beta_k \to \hat{\beta}^{(\lambda)}$ and $\theta_k \to \hat{\theta}^{(\lambda)}$ then $G_\lambda(\beta_k,\theta_k) \to 0$: a converging solver leads to a converging safe rule, *i.e.*, the limiting safe sphere is $\{\hat{\theta}^{(\lambda)}\}$

Sketch of proof next slide

The Gap safe sphere is safe

- $D_\lambda(\hat{\theta}^{(\lambda)}) \leqslant P_\lambda(\beta)$ for any β (weak Duality)
- D_λ is λ^2-strongly concave so for any $\theta_1, \theta_2 \in \mathbb{R}^n$,

$$D_\lambda(\theta_1) \leqslant D_\lambda(\theta_2) + \langle \nabla D_\lambda(\theta_2), \theta_1 - \theta_2 \rangle - \frac{\lambda^2}{2} \|\theta_1 - \theta_2\|_2^2$$

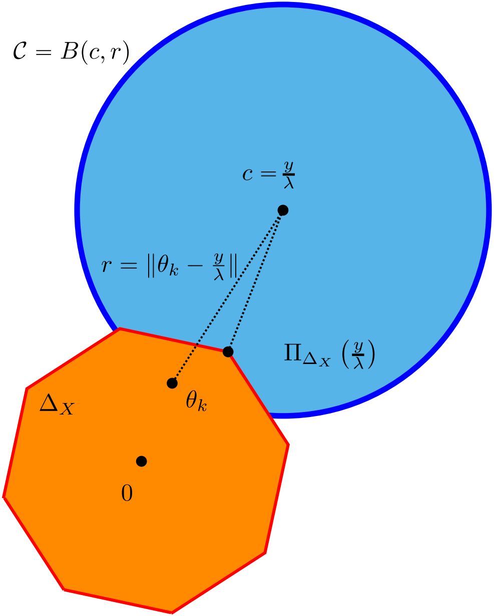
- $\hat{\theta}^{(\lambda)}$ maximizes D_λ over Δ_X, so Fermat's rule yields

$$\forall \theta \in \Delta_X, \qquad \langle \nabla D_\lambda(\hat{\theta}^{(\lambda)}), \theta - \hat{\theta}^{(\lambda)} \rangle \leqslant 0$$

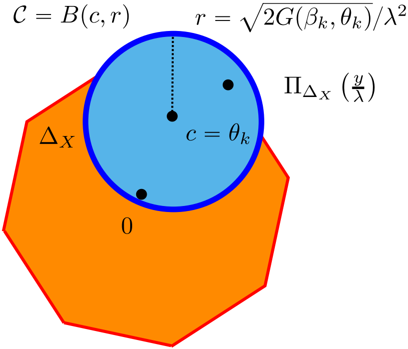
To conclude, for any $\theta \in \Delta_X$:

$$\begin{aligned} \frac{\lambda^2}{2} \left\| \theta - \hat{\theta}^{(\lambda)} \right\|_2^2 &\leqslant D_\lambda(\hat{\theta}^{(\lambda)}) - D_\lambda(\theta) + \langle \nabla D_\lambda(\hat{\theta}^{(\lambda)}), \theta - \hat{\theta}^{(\lambda)} \rangle \\ &\leqslant P_\lambda(\beta) - D_\lambda(\theta) \end{aligned}$$

Dynamic safe sphere Bonnefoy *et al.* (2014)

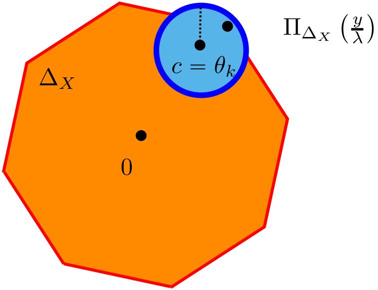


Gap safe sphere Fercoq *et al.* (2015)

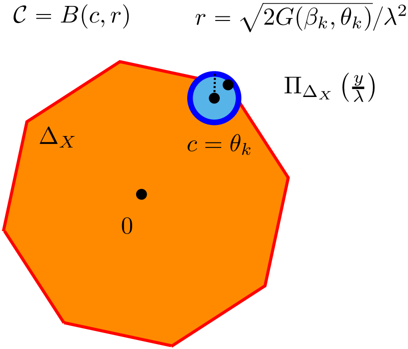


Gap safe sphere Fercoq *et al.* (2015)

$\mathcal{C} = B(c, r)$ $\qquad$ $r = \sqrt{2G(\beta_k, \theta_k)}/\lambda^2$



Gap safe sphere Fercoq *et al.* (2015)



Gap safe sphere Fercoq *et al.* (2015)

$\mathcal{C} = B(c, r)$ $\quad$ $r = 0$

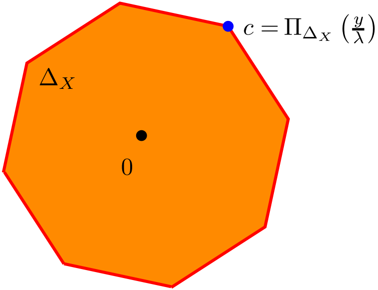


Table of Contents

Motivation - notation

Optimization and convexity reminders

Optimization property for the Lasso

Safe rules

Gap safe rules

Coordinate descent implementation

Coordinate descent for full path

Algorithm: Full coordinate descent

Input : $X, y, \epsilon, K, (\lambda_0 = \lambda_{\max}, \ldots, \lambda_{T-1})$

Initialization: $k = 0$ and $\beta^{\lambda_0} = 0 \in \mathbb{R}^p$

Output : $\beta^{\lambda_0}, \ldots, \beta^{\lambda_{T-1}}$

Coordinate descent for full path

Algorithm: Full coordinate descent

Input : $X, y, \epsilon, K, (\lambda_0 = \lambda_{\max}, \ldots, \lambda_{T-1})$

Initialization: $k = 0$ and $\beta^{\lambda_0} = 0 \in \mathbb{R}^p$

for $t \in [T-1]$ **do**

Output : $\beta^{\lambda_0}, \ldots, \beta^{\lambda_{T-1}}$

Coordinate descent for full path

Algorithm: Full coordinate descent

Input : $X, y, \epsilon, K, (\lambda_0 = \lambda_{\max}, \ldots, \lambda_{T-1})$

Initialization: $k = 0$ and $\beta^{\lambda_0} = 0 \in \mathbb{R}^p$

for $t \in [T-1]$ **do**

  $\beta \leftarrow \beta^{\lambda_{t-1}}$ `// warm start`

Output : $\beta^{\lambda_0}, \ldots, \beta^{\lambda_{T-1}}$

Coordinate descent for full path

Algorithm: Full coordinate descent

Input : $X, y, \epsilon, K, (\lambda_0 = \lambda_{\max}, \dots, \lambda_{T-1})$

Initialization: $k = 0$ and $\beta^{\lambda_0} = 0 \in \mathbb{R}^p$

for $t \in [T-1]$ **do**

$\beta \leftarrow \beta^{\lambda_{t-1}}$ // warm start

for $k \in [K]$ **do**

Output : $\beta^{\lambda_0}, \dots, \beta^{\lambda_{T-1}}$

Coordinate descent for full path

Algorithm: Full coordinate descent

Input : $X, y, \epsilon, K, (\lambda_0 = \lambda_{\max}, \dots, \lambda_{T-1})$

Initialization: $k = 0$ and $\beta^{\lambda_0} = 0 \in \mathbb{R}^p$

for $t \in [T-1]$ **do**

    $\beta \leftarrow \beta^{\lambda_{t-1}}$ `// warm start`

    for $k \in [K]$ **do**

        if $k \mod 10 = 0$ **then**

            Construct $\theta \in \Delta_X$

            if $G_{\lambda_t}(\beta, \theta) \leqslant \epsilon$ `// dual gap evaluation`

             then

                $\beta^{\lambda_t} \leftarrow \beta$

                break

Output : $\beta^{\lambda_0}, \dots, \beta^{\lambda_{T-1}}$

Coordinate descent for full path

Algorithm: Full coordinate descent

Input : $X, y, \epsilon, K, (\lambda_0 = \lambda_{\max}, \dots, \lambda_{T-1})$

Initialization: $k = 0$ and $\beta^{\lambda_0} = 0 \in \mathbb{R}^p$

for $t \in [T-1]$ **do**

- $\beta \leftarrow \beta^{\lambda_{t-1}}$ `// warm start`
- **for** $k \in [K]$ **do**
  - **if** $k \mod 10 = 0$ **then**
    - Construct $\theta \in \Delta_X$
    - **if** $G_{\lambda_t}(\beta, \theta) \leqslant \epsilon$ `// dual gap evaluation`
    - **then**
      - $\beta^{\lambda_t} \leftarrow \beta$
      - **break**
  - **for** $j \in [p]$ **do**
    - $\beta_j \leftarrow \eta_{\mathrm{ST}, \frac{\lambda}{\|\mathbf{x_j}\|^2}} \left(\beta_j - \frac{\mathbf{x}_j^\top (X\beta - y)}{\|\mathbf{x}_j\|^2} \right)$ `// soft-threshold`

Output : $\beta^{\lambda_0}, \dots, \beta^{\lambda_{T-1}}$

Coordinate descent for full path

Algorithm: Full coordinate descent

Input : $X, y, \epsilon, K, (\lambda_0 = \lambda_{\max}, \ldots, \lambda_{T-1})$
Initialization: $k = 0$ and $\beta^{\lambda_0} = 0 \in \mathbb{R}^p$
for $t \in [T-1]$ **do**
    $\beta \leftarrow \beta^{\lambda_{t-1}}$ // warm start
    for $k \in [K]$ **do**
        if $k \mod 10 = 0$ **then**
            Construct $\theta \in \Delta_X$ and S (screen-out variables)
            if $G_{\lambda_t}(\beta, \theta) \leqslant \epsilon$ // dual gap evaluation
            then
                $\beta^{\lambda_t} \leftarrow \beta$
                break
        for $j \in S^c$ **do**
            $\beta_j \leftarrow \eta_{\mathrm{ST}, \frac{\lambda}{\|\mathbf{x}_j\|^2}} \left(\beta_j - \frac{\mathbf{x}_j^\top (X\beta - y)}{\|\mathbf{x}_j\|^2} \right)$ // soft-threshold

Output : $\beta^{\lambda_0}, \ldots, \beta^{\lambda_{T-1}}$

Gap safe rules: fraction non-screened out

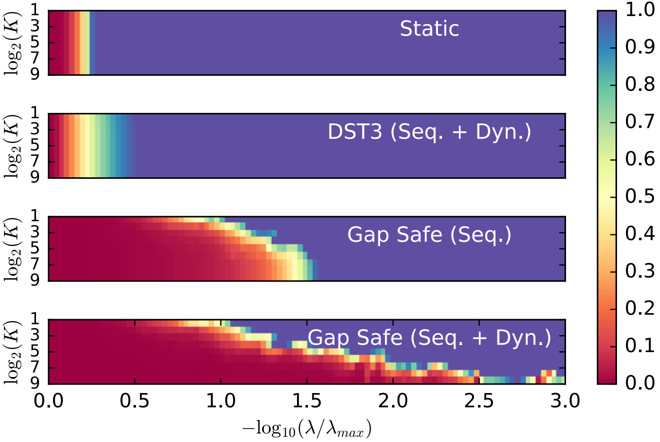


Figure: Lasso on the Leukemia (dense data with $n = 72$ observations and $p = 7129$ features). fraction of the variables that are active. Each line corresponds to a fixed number of iterations for which the algorithm is run

Computing time for standard grid with $T = 100$

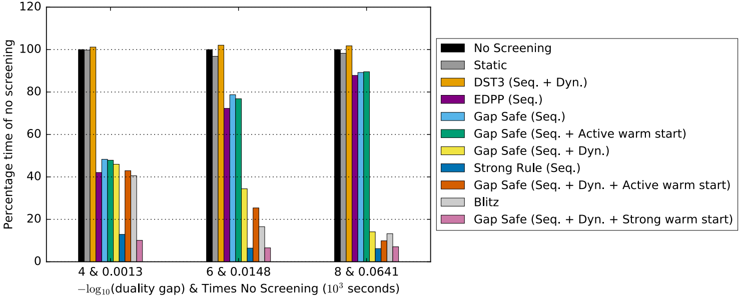


Figure: Lasso on the Leukemia dataset (dense data, $n = 72$ observations, $p = 7129$ features). Computation times needed to solve the Lasso regression path to desired accuracy for a grid of λ from $\lambda_{\max}$ to $\lambda_{\max}/10^3$

Computing time for standard grid with $T = 100$

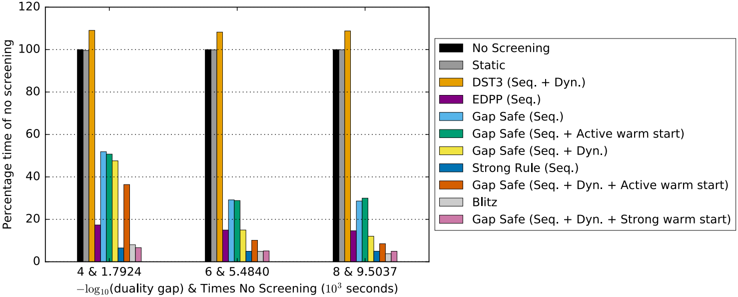


Figure: Lasso on financial dataset E2006-log1p (sparse data with $n = 16\ 087$ observations and $p = 1\ 668\ 737$ features). Computation times needed to solve the Lasso regression path to desired accuracy for a grid of λ from $\lambda_{\max}$ to $\lambda_{\max}/20$

Conclusion and future work

- New safe screening rule based on duality gap for the Lasso
- Computationally efficient, *e.g.*, for coordinate descent

Conclusion and future work

- New safe screening rule based on duality gap for the Lasso
- Computationally efficient, *e.g.*, for coordinate descent
- Generalize well to other penalties: Elastic Net, Group-Lasso, Sparse Group-Lasso ($\ell_1 + \ell_1/\ell_2$)

Conclusion and future work

- New safe screening rule based on duality gap for the Lasso
- Computationally efficient, *e.g.*, for coordinate descent
- Generalize well to other penalties: Elastic Net, Group-Lasso, Sparse Group-Lasso ($\ell_1 + \ell_1/\ell_2$)
- Generalize well to other data fitting terms: *e.g.*, logistic regression, Concomitant Lasso, etc.

Conclusion and future work

- New safe screening rule based on duality gap for the Lasso
- Computationally efficient, *e.g.*, for coordinate descent
- Generalize well to other penalties: Elastic Net, Group-Lasso, Sparse Group-Lasso ($\ell_1 + \ell_1/\ell_2$)
- Generalize well to other data fitting terms: *e.g.*, logistic regression, Concomitant Lasso, etc.
- Combining safe rules ideas with active sets strategies, *cf.* Jonhson and Guestrin (2015,2016)

Conclusion and future work

- New safe screening rule based on duality gap for the Lasso
- Computationally efficient, *e.g.*, for coordinate descent
- Generalize well to other penalties: Elastic Net, Group-Lasso, Sparse Group-Lasso ($\ell_1 + \ell_1/\ell_2$)
- Generalize well to other data fitting terms: *e.g.*, logistic regression, Concomitant Lasso, etc.
- Combining safe rules ideas with active sets strategies, *cf.* Jonhson and Guestrin (2015,2016)

More info : papers / code

Papers:

- ICML 2015 (Lasso case)
- NIPS 2015 (General loss + multi-task)
- NIPS 2016 (Sparse-Group Lasso)
- NCMIP 2017 (Concomitant Lasso)
- JMLR 201? (Journal version: synthesis)

Codes:

- Python code on-line: `https://github.com/EugeneNdiaye`
- pull requests (#5075) (#7853) on `sklearn`

Powered with **MooseTeX**

Références I

- A. Bonnefoy, V. Emiya, L. Ralaivola, and R. Gribonval.
  A dynamic screening principle for the lasso.
  In *EUSIPCO*, 2014.
- A. Bonnefoy, V. Emiya, L. Ralaivola, and R. Gribonval.
  Dynamic screening: accelerating first-order algorithms for the Lasso and Group-Lasso.
  IEEE Trans. Signal Process., 63(19):20, 2015.
- D. Bertsimas, A. King, and R. Mazumder.
  Best subset selection via a modern optimization lens.
  Ann. Statist., 44(2):813–852, 2016.
- P. J. Bickel, Y. Ritov, and A. B. Tsybakov.
  Simultaneous analysis of Lasso and Dantzig selector.
  Ann. Statist., 37(4):1705–1732, 2009.
- A. Beck and M. Teboulle.
  A fast iterative shrinkage-thresholding algorithm for linear inverse problems.
  SIAM J. Imaging Sci., 2(1):183–202, 2009.

Références II

- S. S. Chen, D. L. Donoho, and M. A. Saunders.
  Atomic decomposition by basis pursuit.
  SIAM J. Sci. Comput., 20(1):33–61, 1998.

- 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.

- A. S. Dalalyan, M. Hebiri, and J. Lederer.
  On the prediction performance of the Lasso.
  Bernoulli, 23(1):552–581, 2017.

- B. Efron, T. J. Hastie, I. M. Johnstone, and R. Tibshirani.
  Least angle regression.
  Ann. Statist., 32(2):407–499, 2004.
  With discussion, and a rejoinder by the authors.

- L. El Ghaoui, V. Viallon, and T. Rabbani.
  Safe feature elimination in sparse supervised learning.
  J. Pacific Optim., 8(4):667–698, 2012.

Références III

- O. Fercoq, A. Gramfort, and J. Salmon.
  Mind the duality gap: safer rules for the lasso.
  In *ICML*, pages 333–342, 2015.
- J. Friedman, T. J. Hastie, H. Höfling, and R. Tibshirani.
  Pathwise coordinate optimization.
  Ann. Appl. Stat., 1(2):302–332, 2007.
- C. Giraud.
  Introduction to high-dimensional statistics, volume 138.
  CRC Press, 2014.
- T. B. Johnson and C. Guestrin.
  BLITZ: A principled meta-algorithm for scaling sparse optimization.
  In *ICML*, pages 1171–1179, 2015.
- T. B. Johnson and C. Guestrin.
  Unified methods for exploiting piecewise linear structure in convex optimization.
  In *NIPS*, pages 4754–4762, 2016.

Références IV

- S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky.
  An interior-point method for large-scale ℓ_1-regularized least squares.
  IEEE J. Sel. Topics Signal Process., 1(4):606–617, 2007.
- J. Mairal and B. Yu.
  Complexity analysis of the lasso regularization path.
  In *ICML*, pages 353–360, 2012.
- B. K. Natarajan.
  Sparse approximate solutions to linear systems.
  SIAM J. Comput., 24(2):227–234, 1995.
- M. R. Osborne, B. Presnell, and B. A. Turlach.
  A new approach to variable selection in least squares problems.
  IMA J. Numer. Anal., 20(3):389–403, 2000.
- R. Tibshirani, J. Bien, J. Friedman, T. J. Hastie, N. Simon, J. Taylor, and R. J. Tibshirani.
  Strong rules for discarding predictors in lasso-type problems.
  J. R. Stat. Soc. Ser. B Stat. Methodol., 74(2):245–266, 2012.

Références V

- R. Tibshirani.
  Regression shrinkage and selection via the lasso.
  J. R. Stat. Soc. Ser. B Stat. Methodol., 58(1):267–288, 1996.

- R. J. Tibshirani.
  The lasso problem and uniqueness.
  Electron. J. Stat., 7:1456–1490, 2013.

- P. Tseng.
  Convergence of a block coordinate descent method for nondifferentiable minimization.
  J. Optim. Theory Appl., 109(3):475–494, 2001.

- J. Wang, J. Zhou, P. Wonka, and J. Ye.
  Lasso screening rules via dual polytope projection.
  In *NIPS*, pages 1070–1078, 2013.

- H. Zou.
  The adaptive lasso and its oracle properties.
  J. Amer. Statist. Assoc., 101(476):1418–1429, 2006.

Références VI

- C.-H. Zhang and T. Zhang.

  A general theory of concave regularization for high-dimensional sparse estimation problems.

  Statist. Sci., 27(4):576–593, 2012.

Lasso theory : (fairly) well understood

Gaussian model: $y = X\beta^* + \sigma\varepsilon$, with $\|\beta^*\| = s$

Theorem Bickel *et al.* (2009), Dalalyan *et al.* (2017), Giraud (2014)

For Gaussian noise model with X satisfying the "Restricted Eigenvalue" property and $\lambda = 2n\sigma\sqrt{\frac{2\log(p/\delta)}{n}}$, then

$$\frac{1}{n}\left\|X(\beta^* - \hat{\beta}^{(\lambda)})\right\|^2 \leqslant \frac{18}{\kappa_s^2}\frac{\sigma^2 s}{n}\log\left(\frac{p}{\delta}\right)$$

with probability $1-\delta$, where $\hat{\beta}^{(\lambda)}$ is a Lasso solution

Rem: Optimal rate in the minimax sense (up to constant/log term)

Rem: under the "Restricted Eigenvalue" property, κ_s^2 is a measure of strong convexity of the (quadratic part of the) objective function obtained when extracting s columns of X

EDDP Wang *et al.* (2013) can remove useful variables