# Optimization: (Block) coordinate descent for neuro-imaging

**Joseph Salmon**

http://josephsalmon.eu

IMAG
Univ. Montpellier
CNRS

# Overview

Motivation / Examples

Variable selection and sparsity

Algorithms for non-smooth convex problems

Extensions: non convex, general structure

# Third example: Click Trough Rate prediction

"The task is to choose the products to display in the ad knowing the banner type, user context, and candidate ads, in order to maximize the number of clicks."

- $n > 100$ millions samples (display ad impressions)
- $p = 35$ raw features (but Criteo declares using interaction of order 3 ≈ 40 000 features)
- $q = 2$ classes (binary classification: Clicked=+1 / not-clicked=-1)

Criteo dataset http://www.cs.cornell.edu/~adith/Criteo/

# Classification in bio-statistics

"47 patients with acute lymphoblastic leukemia (ALL) and 25 patients with acute myeloid leukemia (AML). Each of the 72 patients had bone marrow samples obtained at the time of diagnosis. The observations have been assayed with Affymetrix Hgu6800 chips, resulting in 7129 gene expressions (Affymetrix probes)."[(1)]

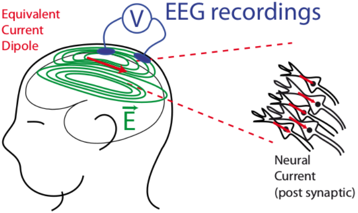
- $n = 72$ (samples)
- $p = 7129$ (features /covariates/exploratory variables)
- $q = 2$ classes (binary classification: +1 = sick / -1 = not sick)

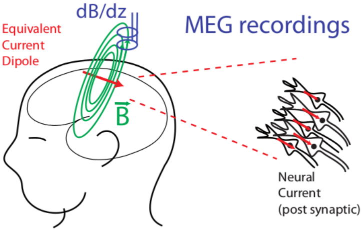
https://github.com/ramhiser/datamicroarray/wiki/Golub-(1999)

[(1)] T. R. Golub et al. "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring.". In: *Science* 286.5439 (1999), pp. 531–537.

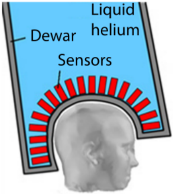
# Inverse problem for neuro-imaging: M/EEG

- sensor: magneto- and electro-encephalogrammes measured during a cognitive experiment
- sources: positions in the brain



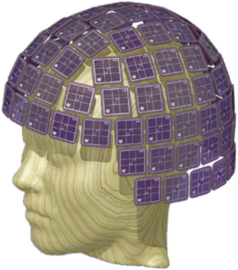

First EEG
recordings
in 1929
by H. Berger



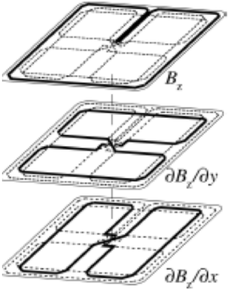

Hôpital La Timone
Marseille, France

# Capteur MEG: magnétomètres et gradiomètres

Appareil

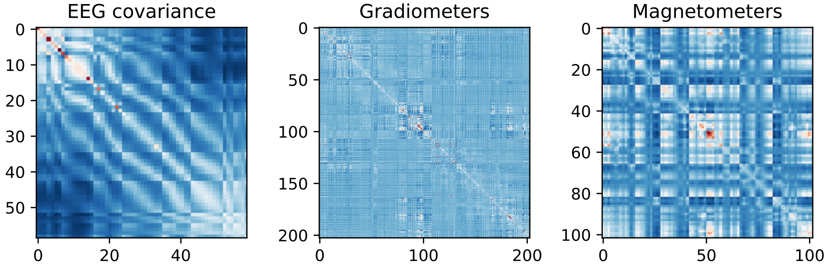


Capteurs

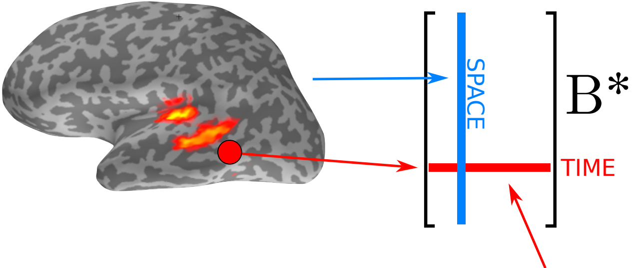



Détails des capteurs

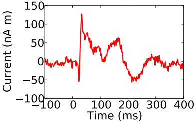
# Noise covariances differ between EEG and MEG



# Sources model

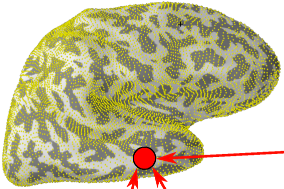



Position a few thousands candidate sources over the brain (*e.g.*, every 5mm)




$$B^* \in \mathbb{R}^{p \times q}$$

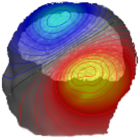
# Design matrix - forward operator

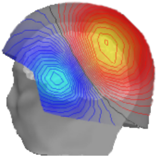


$$X = \begin{bmatrix} X_{\text{EEG}} \\ \hdashline X_{\text{MEG}} \end{bmatrix} \in \mathbb{R}^{n \times p}$$

EEG:
Forward field of the electrodes

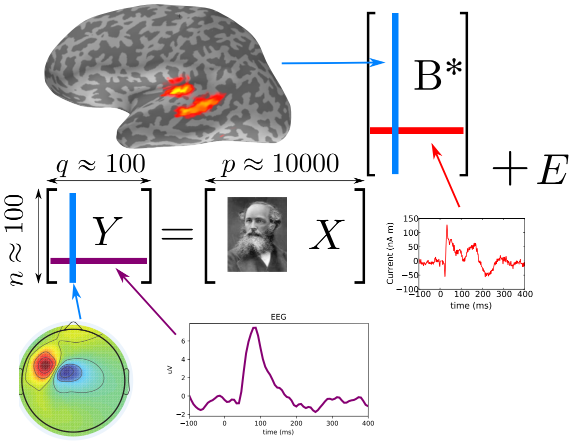
MEG:
Forward field of sensor





$X$ : gain matrix / forward operator obtained by Maxwell's equations

# Mutli-task regression




Standard dimensions:

- $n = 302$ sensors
- $p = 7498$ sources (discretization in space)
- $t = 181$ time instants

# Simple canonical (linear) model : $q = 1$

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

- $\mathbf{y} \in \mathbb{R}^n$ : observations vector; $n =$ number of samples
- $X = [\mathbf{x}_1, \dots, \mathbf{x}_p] = \begin{pmatrix} x_{1,1} & \dots & x_{1,p} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \dots & x_{n,p} \end{pmatrix} \in \mathbb{R}^{n \times p}$ : design matrix; $p =$ number of features
- $\boldsymbol{\varepsilon} \in \mathbb{R}^n \sim \mathcal{N}(0, \sigma^2)$ : Gaussian noise with variance $\sigma^2$
- $\boldsymbol{\beta}^\star \in \mathbb{R}^p$: true parameter to recover

Rem: more general models can be handled similarly up to more technical details (for classification, multi-task, etc.)

# Motivation for sparse models

Estimators $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}^{\star}$ with many zero coefficients are useful:

- for interpretation : interest for practitioners
- for theoretical results : counter curse of dimensionality
- for computational efficiency : especially for huge $p$

Underlying idea: **variable selection**

# Support and $\ell_0$ pseudo-norm

**Definitions**

**Support** of a vector $\boldsymbol{\beta}$ (non-zero coordinates):

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

$\ell_0$ **pseudo-norm** of $\boldsymbol{\beta} \in \mathbb{R}^p$ : number of non-zero coordinates:

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

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

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

# Outline

# The $\ell_0$ penalty

First attempt: promote sparsity using $\ell_0$ as a penalty/regularization

$$\boxed{\hat{\boldsymbol{\beta}}^{\ell_0}_{\lambda} = \mathop{\arg\min}_{\boldsymbol{\beta}\in\mathbb{R}^p} \left( \underbrace{\frac{1}{2}\|\mathbf{y} - X\boldsymbol{\beta}\|_2^2}_{\textbf{data fitting}} + \underbrace{\lambda\|\boldsymbol{\beta}\|_0}_{\textbf{regularization}} \right)}$$

**Combinatorial problem**!!!

Exact/Naive resolution : consider all sub-models, *i.e.*, compute $2^p$ least squares computation (*i.e.*, $2^p$ possible supports); **NP-hard**[(2)]

**Exemple**:
$p = 10$: $\approx 10^3$ least squares
$p = 30$: $\approx 10^{10}$ least squares

Rem: mixed integer programming fine for small problems[(3)]

---

(2) B. K. Natarajan. "Sparse approximate solutions to linear systems". In: *SIAM J. Comput.* 24.2 (1995), pp. 227–234.

(3) D. Bertsimas, A. King, and R. Mazumder. "Best subset selection via a modern optimization lens". In: *Ann. Statist.* 44.2 (2016), pp. 813–852.

# Though statistically useful

Statistical optimality for sparse underlying true signal :

**Theorem**[4]

For $\hat{\beta}_\lambda^{\ell_0}$ with a well chosen parameter $\lambda$ (and a constant $C$):

$$\mathbb{E}\left(\frac{\|X\hat{\boldsymbol{\beta}}_\lambda^{\ell_0} - X\boldsymbol{\beta}^\star\|^2}{n}\right) \leq C\frac{\sigma^2\,\|\boldsymbol{\beta}^\star\|_0}{n}\log\left(\frac{eM}{\|\boldsymbol{\beta}^\star\|_0}\right)$$

Rem: Least-squares upper prediction error $\leq C\frac{\sigma^2 p}{n}$
Rem: upper bound cannot be improved (in a minimax sense), optimal rate[5]

[4] F. Bunea, A. B. Tsybakov, and M. H. Wegkamp. "Aggregation for Gaussian regression". In: *Ann. Statist.* 35.4 (2007), pp. 1674–1697.

[5] A. B. Tsybakov. "Optimal Rates of Aggregation". In: *COLT*. 2003, pp. 303–313.

# Alternatives: variable selection overview

- **Correlation Screening**: remove the $\mathbf{x}_j$'s whose correlation with observation $\mathbf{y}$ is weak, fast $(+++)$, intuitive $(+++)$ but weak theory (- - -), neglect variables interactions (- - -)
- **Greedy methods**: forward/stage-wise[(6), (7), (8)], fast$(++)$, intuitive$(++)$, propagates wrong selection(- -), weak theory(-)
- **Penalized methods**
  - convex
  - non-convex
- **Approximate Message Passing**[(9)](AMP), graphical models, hard to solve (- -), theory (claimed better?),

[(6)] M. A. Efroymson. "Multiple regression analysis". In: *Mathematical methods for digital computers*. New York: Wiley, 1960, pp. 191–203.

[(7)] S. Mallat and Z. Zhang. "Matching Pursuit With Time-Frequency Dictionaries". In: *IEEE Trans. Image Process.* 41 (1993), pp. 3397–3415.

[(8)] T. Zhang. "Adaptive forward-backward greedy algorithm for learning sparse representations". In: *IEEE Trans. Inf. Theory* 57.7 (2011), pp. 4689–4708.

[(9)] D. L. Donoho, A., and A. Montanari. "Message-passing algorithms for compressed sensing". In: *Proceedings of the National Academy of Sciences* 106.45 (2009), pp. 18914–18919.

# Outline

Motivation / Examples

Variable selection and sparsity

The $\ell_0$ penalty and its limits

The $\ell_1$ penalty : a convex relaxation

Algorithms for non-smooth convex problems

Extensions: non convex, general structure

# Lasso: penalty point of view[10]

Lasso: *Least Absolute Shrinkage and Selection Operator*

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

where $\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\beta_j|$ ($\ell_1$ norm) and $\lambda > 0$ is a parameter

- Limiting cases:

$$\lim_{\lambda\to 0} \hat{\boldsymbol{\beta}}_\lambda^{\text{Lasso}} = \hat{\boldsymbol{\beta}}^{\text{LS}}$$

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

**Beware**: uniqueness non mandatory (*e.g.*, case $\mathbf{x}_1 = \mathbf{x}_2$)

---

[10] R. Tibshirani. "Regression Shrinkage and Selection via the Lasso". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 58.1 (1996), pp. 267–288.

# Constraint point of view

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

share same solutions with constraint formulation:

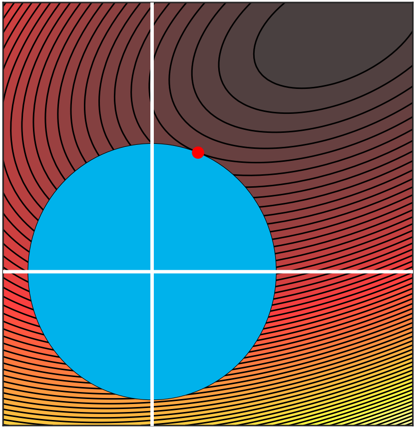
$$\begin{cases} \mathop{\arg\min}_{\boldsymbol{\beta} \in \mathbb{R}^p} \|\mathbf{y} - X\boldsymbol{\beta}\|_2^2 \\ \text{t.q. } \|\boldsymbol{\beta}\|_1 \leq T \end{cases}, \quad \text{for some parameter } T > 0$$

<u>Rem</u>: 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 $\hat{\beta}^{\text{MCO}}$ (unconstrained)

# Zeroing coefficients: a vizualisation

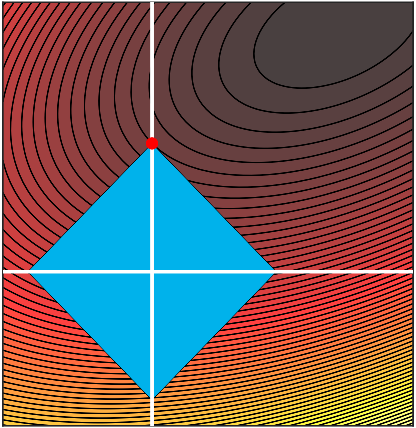
$$\begin{cases} \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min} \|\mathbf{y} - X\boldsymbol{\beta}\|_2^2 \\ \text{t.q. } \|\boldsymbol{\beta}\|_2 \leq T \end{cases}, \quad \text{for some parameter } T > 0$$



$\ell_2$ constraint : non sparse solution

# Zeroing coefficients: a vizualisation

$$\begin{cases} \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} \|\mathbf{y} - X\boldsymbol{\beta}\|_2^2 \\ \text{t.q. } \|\boldsymbol{\beta}\|_1 \leq T \end{cases}, \qquad \text{for some parameter } T > 0$$



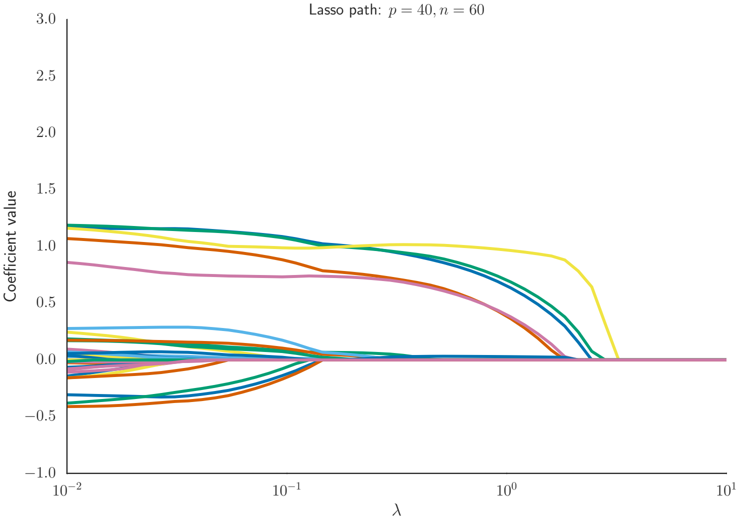
$\ell_1$ constraint : ~~non~~ sparse solution

# Numerical example on simulated data

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

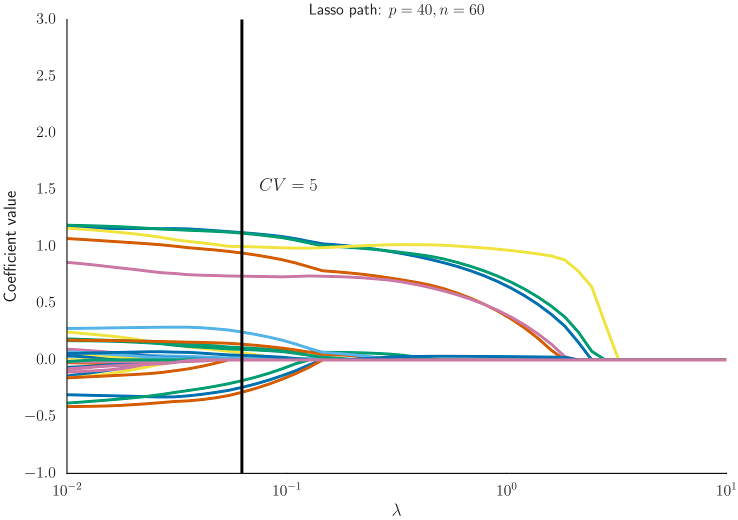
For this example : $n = 60, p = 40, \sigma = 1$

# Lasso



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

# Lasso



Lasso 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
$CV = 5$
$10^{-2}$
$10^{-1}$
$10^{0}$
$10^{1}$
$\lambda$

# Lasso properties

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

**Exemple**: 17 non-zero coefficients for LassoCV in the previous simulated example

# Lasso analysis

**Statistical guarantees**: Lasso "almost" optimal for sparse signals **provided** some local "conditioning" property involving $X$ and the sparsity level of $\beta^\star$:

**Theorem**[11]

For $\hat{\boldsymbol{\beta}}_\lambda^{\mathrm{Lasso}}$ with a well chosen $\lambda$ (and a constant $C$):

$$\mathbb{E}\left(\frac{\|X\hat{\boldsymbol{\beta}}_\lambda^{\mathrm{Lasso}} - X\boldsymbol{\beta}^\star\|^2}{n}\right) \leq C\frac{\sigma^2\,\|\boldsymbol{\beta}^\star\|_0}{n}\log(M)$$

*cf.* Bühlmann and van de Geer (2011)[12] for an overview

[11] V. Koltchinskii, K. Lounici, and A. B. Tsybakov. "Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion". In: *Ann. Statist.* 39.5 (2011), pp. 2302–2329.

[12] P. Bühlmann and S. van de Geer. *Statistics for high-dimensional data*. Springer Series in Statistics. Methods, theory and applications. Heidelberg: Springer, 2011.

# Outline

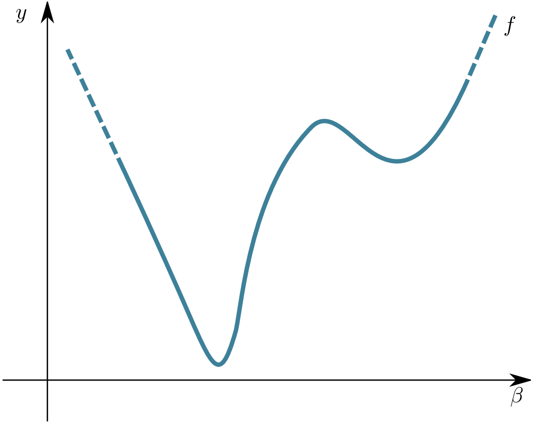
Motivation / Examples

Variable selection and sparsity

Algorithms for non-smooth convex problems
- Majorization / Minimization
- Proximal methods — Forward / Backward
- (Block) Coordinate descent
- Stopping criterion

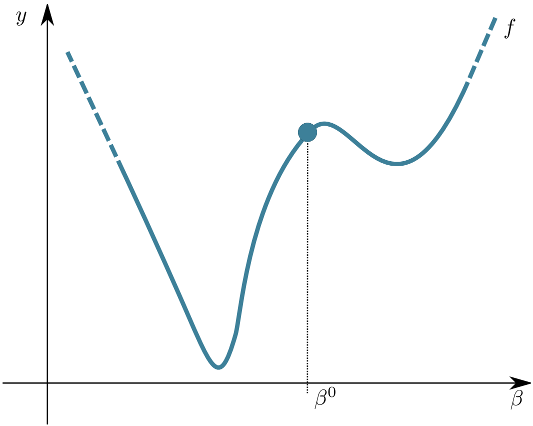
Extensions: non convex, general structure

# Majorization / Minimization: visually



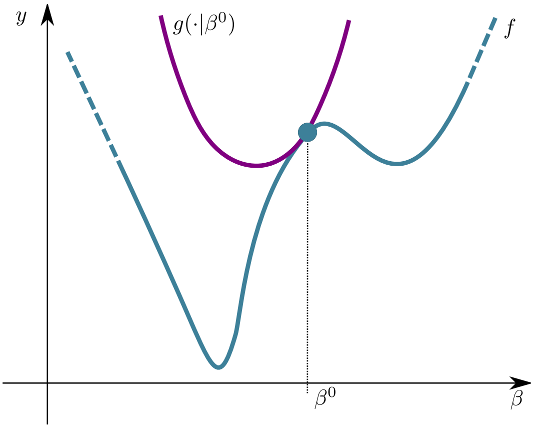

Original function

# Majorization / Minimization: visually



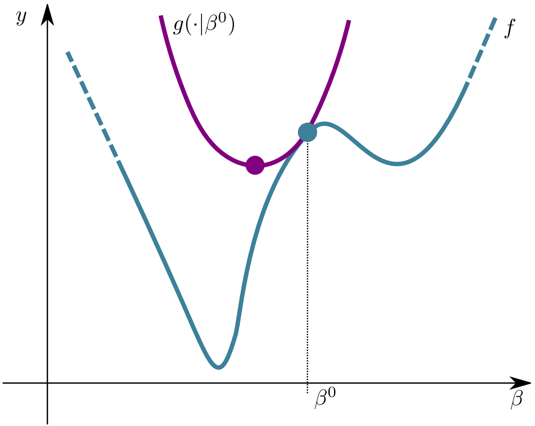

Initialize

# Majorization / Minimization: visually



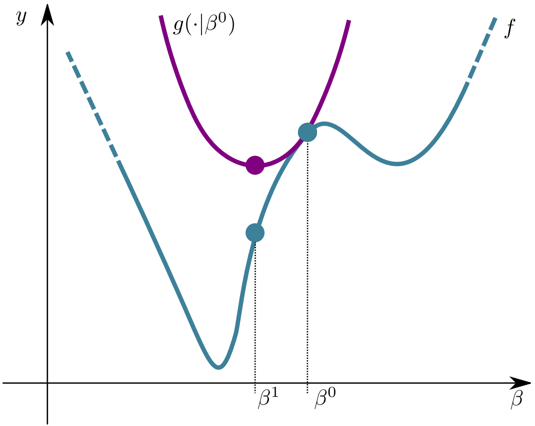

Majorize

# Majorization / Minimization: visually



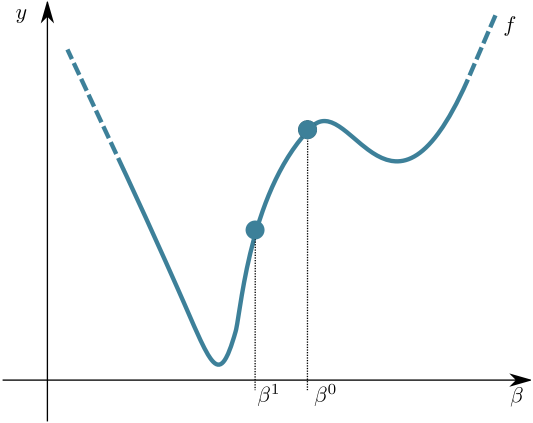

Minimize

# Majorization / Minimization: visually



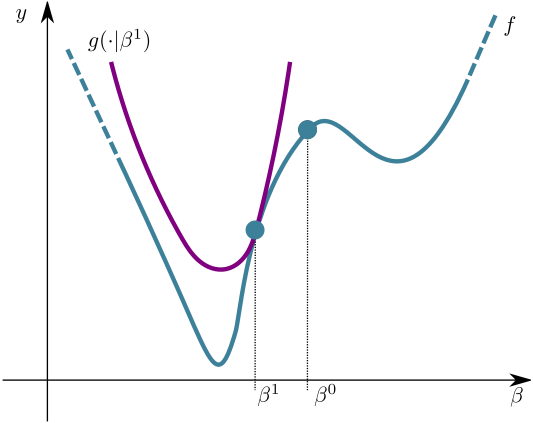

Update

# Majorization / Minimization: visually



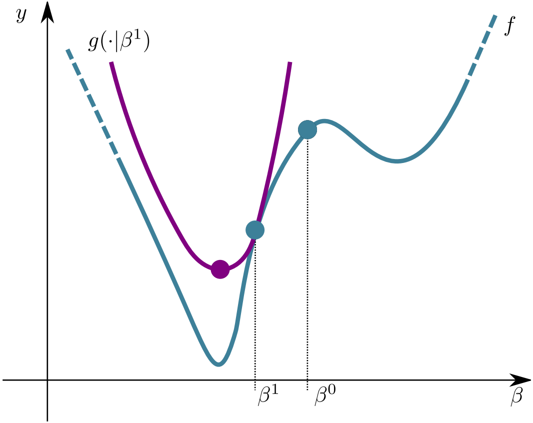

Update

# Majorization / Minimization: visually



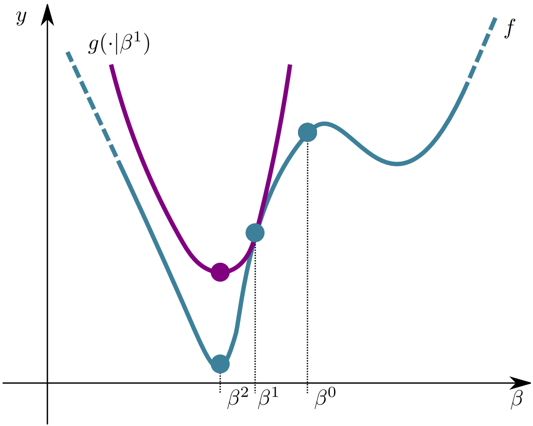

Majorize

# Majorization / Minimization: visually




Minimize

# Majorization / Minimization: visually




Update

# Majorization / Minimization: formally

Objective: find a minimizer of a function $f$

Tool: at each point $\boldsymbol{\beta}^t$ proceed as follows:

- Provide a "**majorization**" function $\boldsymbol{\beta} \to g(\boldsymbol{\beta}|\boldsymbol{\beta}^t)$ satisfying:

$$\begin{cases} f(\boldsymbol{\beta}) \leq g(\boldsymbol{\beta}|\boldsymbol{\beta}^t), \forall \boldsymbol{\beta} & : \quad \text{domination / upper bound} \\ f(\boldsymbol{\beta}^t) = g(\boldsymbol{\beta}^t|\boldsymbol{\beta}^t) & : \quad \text{tangency / tightness at } \boldsymbol{\beta}^t \end{cases}$$

- **Minimize** the upper bound and obtain

$$\boldsymbol{\beta}^{t+1} \in \operatorname*{arg\,min}_{\boldsymbol{\beta} \in \mathbb{R}^p} g(\boldsymbol{\beta}|\boldsymbol{\beta}^t)$$

# Majorization / Minimization: formally

Objective: find a minimizer of a function $f$

Tool: at each point $\boldsymbol{\beta}^t$ proceed as follows:

- Provide a "**majorization**" function $\boldsymbol{\beta} \to g(\boldsymbol{\beta}|\boldsymbol{\beta}^t)$ satisfying:

$$\begin{cases} f(\boldsymbol{\beta}) \leq g(\boldsymbol{\beta}|\boldsymbol{\beta}^t), \forall \boldsymbol{\beta} & : \quad \text{domination / upper bound} \\ f(\boldsymbol{\beta}^t) = g(\boldsymbol{\beta}^t|\boldsymbol{\beta}^t) & : \quad \text{tangency / tightness at } \boldsymbol{\beta}^t \end{cases}$$

- **Minimize** the upper bound and obtain

$$\boldsymbol{\beta}^{t+1} \in \operatorname*{arg\,min}_{\boldsymbol{\beta} \in \mathbb{R}^p} g(\boldsymbol{\beta}|\boldsymbol{\beta}^t)$$

Rem: we say that $g(\cdot|\boldsymbol{\beta}^t)$ is a surrogate of $f$ at $\boldsymbol{\beta}^t$

# Majorization / Minimization: formally

Objective: find a minimizer of a function $f$

Tool: at each point $\boldsymbol{\beta}^t$ proceed as follows:

- Provide a "**majorization**" function $\boldsymbol{\beta} \to g(\boldsymbol{\beta}|\boldsymbol{\beta}^t)$ satisfying:

$$\begin{cases} f(\boldsymbol{\beta}) \leq g(\boldsymbol{\beta}|\boldsymbol{\beta}^t), \forall \boldsymbol{\beta} & : \quad \text{domination / upper bound} \\ f(\boldsymbol{\beta}^t) = g(\boldsymbol{\beta}^t|\boldsymbol{\beta}^t) & : \quad \text{tangency / tightness at } \boldsymbol{\beta}^t \end{cases}$$

- **Minimize** the upper bound and obtain

$$\boldsymbol{\beta}^{t+1} \in \arg\min_{\boldsymbol{\beta} \in \mathbb{R}^p} g(\boldsymbol{\beta}|\boldsymbol{\beta}^t)$$

Rem: we say that $g(\cdot|\boldsymbol{\beta}^t)$ is a surrogate of $f$ at $\boldsymbol{\beta}^t$

# Majorization / Minimization: Algorithm

---

**Algorithm:** MAXIMIZATION MINIMIZATION

---

**input** : max. iterations $t_{\max}$, stopping criterion $\varepsilon$

**init** : $\boldsymbol{\beta}^0$

**for** $1 \leq t \leq t_{\max}$ **do**

  **Break** if stopping criterion smaller than $\varepsilon$

  Find a majorization function: $g(\cdot|\boldsymbol{\beta}^t)$

  Minimize it: $\boldsymbol{\beta}^{t+1} \leftarrow \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, g(\boldsymbol{\beta}|\boldsymbol{\beta}^t)$

**return** $\boldsymbol{\beta}^{t_{\max}}$ *"close" to a local minimum of $f$*

---

# Convergence property[13]

**Theorem**

The maximization/minimization algorithm is a descent method:

$$\forall t \geq 1, \quad f(\boldsymbol{\beta}^{t+1}) \leq f(\boldsymbol{\beta}^{t})$$

Hence, provided that $f$ is lower bounded the algorithm converges.

[13] K. Lange. *MM optimization algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016, pp. ix+223.

# Convergence property[13]

**Theorem**

The maximization/minimization algorithm is a descent method:

$$\forall t \geq 1, \quad f(\boldsymbol{\beta}^{t+1}) \leq f(\boldsymbol{\beta}^{t})$$

Hence, provided that $f$ is lower bounded the algorithm converges.

Proof:

[13] K. Lange. *MM optimization algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016, pp. ix+223.

# Convergence property[13]

**Theorem**

The maximization/minimization algorithm is a descent method:

$$\forall t \geq 1, \quad f(\boldsymbol{\beta}^{t+1}) \leq f(\boldsymbol{\beta}^{t})$$

Hence, provided that $f$ is lower bounded the algorithm converges.

Proof:

$$f(\boldsymbol{\beta}^{t+1}) \leq g(\boldsymbol{\beta}^{t+1}|\boldsymbol{\beta}^{t}) \qquad \text{(Majorization at } \boldsymbol{\beta}^{t}\text{)}$$

[13] K. Lange. *MM optimization algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016, pp. ix+223.

# Convergence property[13]

**Theorem**

The maximization/minimization algorithm is a descent method:

$$\forall t \geq 1, \quad f(\boldsymbol{\beta}^{t+1}) \leq f(\boldsymbol{\beta}^t)$$

Hence, provided that $f$ is lower bounded the algorithm converges.

Proof:

$$\begin{aligned} f(\boldsymbol{\beta}^{t+1}) &\leq g(\boldsymbol{\beta}^{t+1}|\boldsymbol{\beta}^t) && \text{(Majorization at } \boldsymbol{\beta}^t\text{)} \\ &\leq g(\boldsymbol{\beta}^t|\boldsymbol{\beta}^t) && \text{(Minimization definition of } \boldsymbol{\beta}^{t+1}\text{)} \end{aligned}$$

[13] K. Lange. *MM optimization algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016, pp. ix+223.

# Convergence property[13]

**Theorem**

The maximization/minimization algorithm is a descent method:

$$\forall t \geq 1, \quad f(\boldsymbol{\beta}^{t+1}) \leq f(\boldsymbol{\beta}^{t})$$

Hence, provided that $f$ is lower bounded the algorithm converges.

Proof:

$$\begin{aligned} f(\boldsymbol{\beta}^{t+1}) &\leq g(\boldsymbol{\beta}^{t+1}|\boldsymbol{\beta}^{t}) && \text{(Majorization at } \boldsymbol{\beta}^{t}\text{)} \\ &\leq g(\boldsymbol{\beta}^{t}|\boldsymbol{\beta}^{t}) && \text{(Minimization definition of } \boldsymbol{\beta}^{t+1}\text{)} \\ &= f(\boldsymbol{\beta}^{t}) && \text{(tightness at } \boldsymbol{\beta}^{t}\text{)} \end{aligned}$$

[13] K. Lange. *MM optimization algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016, pp. ix+223.

# Gradient descent revisited

Optimization problem: $\boxed{\min_{\boldsymbol{\beta} \in \mathbb{R}^p} f(\boldsymbol{\beta})}$

Properties: $f$ is convex with gradient $L$-Lipschitz

$$\forall (\boldsymbol{\beta}, \boldsymbol{\beta}') \in \mathbb{R}^d \times \mathbb{R}^d, \quad \|\nabla f(\boldsymbol{\beta}) - \nabla f(\boldsymbol{\beta}')\| \leq L \|\boldsymbol{\beta} - \boldsymbol{\beta}'\|$$

# Gradient descent revisited

Optimization problem: $\boxed{\min_{\boldsymbol{\beta} \in \mathbb{R}^p} f(\boldsymbol{\beta})}$

Properties: $f$ is convex with gradient $L$-Lipschitz

$$\forall (\boldsymbol{\beta}, \boldsymbol{\beta}') \in \mathbb{R}^d \times \mathbb{R}^d, \quad \|\nabla f(\boldsymbol{\beta}) - \nabla f(\boldsymbol{\beta}')\| \leq L \|\boldsymbol{\beta} - \boldsymbol{\beta}'\|$$

Surrogate:

$$g(\boldsymbol{\beta}|\boldsymbol{\beta}^t) = f(\boldsymbol{\beta}^t) + \langle \nabla f(\boldsymbol{\beta}^t), \boldsymbol{\beta} - \boldsymbol{\beta}^t \rangle + \frac{L}{2} \|\boldsymbol{\beta}^t - \boldsymbol{\beta}\|^2$$

# Gradient descent revisited

Optimization problem: $\boxed{\min_{\boldsymbol{\beta} \in \mathbb{R}^p} f(\boldsymbol{\beta})}$

Properties: $f$ is convex with gradient $L$-Lipschitz

$$\forall (\boldsymbol{\beta}, \boldsymbol{\beta}') \in \mathbb{R}^d \times \mathbb{R}^d, \quad \|\nabla f(\boldsymbol{\beta}) - \nabla f(\boldsymbol{\beta}')\| \leq L \|\boldsymbol{\beta} - \boldsymbol{\beta}'\|$$

Surrogate:

$$g(\boldsymbol{\beta}|\boldsymbol{\beta}^t) = f(\boldsymbol{\beta}^t) + \langle \nabla f(\boldsymbol{\beta}^t), \boldsymbol{\beta} - \boldsymbol{\beta}^t \rangle + \frac{L}{2} \|\boldsymbol{\beta}^t - \boldsymbol{\beta}\|^2$$

Update rule : $\boxed{\boldsymbol{\beta}^{t+1} = \boldsymbol{\beta}^t - \frac{1}{L} \nabla f(\boldsymbol{\beta}^t)}$

# Gradient descent revisited

Optimization problem: $\boxed{\min_{\boldsymbol{\beta} \in \mathbb{R}^p} f(\boldsymbol{\beta})}$

Properties: $f$ is convex with gradient $L$-Lipschitz

$$\forall (\boldsymbol{\beta}, \boldsymbol{\beta}') \in \mathbb{R}^d \times \mathbb{R}^d, \quad \|\nabla f(\boldsymbol{\beta}) - \nabla f(\boldsymbol{\beta}')\| \leq L \|\boldsymbol{\beta} - \boldsymbol{\beta}'\|$$

Surrogate:

$$g(\boldsymbol{\beta}|\boldsymbol{\beta}^t) = f(\boldsymbol{\beta}^t) + \langle \nabla f(\boldsymbol{\beta}^t), \boldsymbol{\beta} - \boldsymbol{\beta}^t \rangle + \frac{L}{2} \|\boldsymbol{\beta}^t - \boldsymbol{\beta}\|^2$$

Update rule : $\boxed{\boldsymbol{\beta}^{t+1} = \boldsymbol{\beta}^t - \frac{1}{L} \nabla f(\boldsymbol{\beta}^t)}$

Rem: $\alpha \leq 1/L$ also works as a step size

# Proof (can be skipped)

**Quadratic majorization**

If $f$ is convex, differentiable with gradient $L$-Lipschitz, *i.e.*,

$$\forall(\boldsymbol{\beta}, \boldsymbol{\beta}') \in \mathbb{R}^d \times \mathbb{R}^d, \quad \|\nabla f(\boldsymbol{\beta}) - \nabla f(\boldsymbol{\beta}')\| \leq L\|\boldsymbol{\beta} - \boldsymbol{\beta}'\|$$

then the following holds: $\forall(\boldsymbol{\beta}, \boldsymbol{\beta}') \in \mathbb{R}^d \times \mathbb{R}^d$,

$$0 \leq f(\boldsymbol{\beta}) - f(\boldsymbol{\beta}') - \langle \nabla f(\boldsymbol{\beta}'), \boldsymbol{\beta} - \boldsymbol{\beta}' \rangle \leq \frac{L}{2} \left\|\boldsymbol{\beta}' - \boldsymbol{\beta}\right\|^2$$

# Proof (can be skipped)

**Quadratic majorization**

If $f$ is convex, differentiable with gradient $L$-Lipschitz, *i.e.*,

$$\forall(\boldsymbol{\beta}, \boldsymbol{\beta}') \in \mathbb{R}^d \times \mathbb{R}^d, \quad \|\nabla f(\boldsymbol{\beta}) - \nabla f(\boldsymbol{\beta}')\| \leq L\|\boldsymbol{\beta} - \boldsymbol{\beta}'\|$$

then the following holds: $\forall(\boldsymbol{\beta}, \boldsymbol{\beta}') \in \mathbb{R}^d \times \mathbb{R}^d$,

$$0 \leq f(\boldsymbol{\beta}) - f(\boldsymbol{\beta}') - \langle \nabla f(\boldsymbol{\beta}'), \boldsymbol{\beta} - \boldsymbol{\beta}' \rangle \leq \frac{L}{2} \left\|\boldsymbol{\beta}' - \boldsymbol{\beta}\right\|^2$$

Rem: positivity : consequence of convexity; second inequality Taylor expansion

# Proof (can be skipped)

**Quadratic majorization**

If $f$ is convex, differentiable with gradient $L$-Lipschitz, *i.e.*,

$$\forall(\boldsymbol{\beta},\boldsymbol{\beta}') \in \mathbb{R}^d \times \mathbb{R}^d, \quad \|\nabla f(\boldsymbol{\beta}) - \nabla f(\boldsymbol{\beta}')\| \leq L\|\boldsymbol{\beta} - \boldsymbol{\beta}'\|$$

then the following holds: $\quad \forall(\boldsymbol{\beta},\boldsymbol{\beta}') \in \mathbb{R}^d \times \mathbb{R}^d$,

$$0 \leq f(\boldsymbol{\beta}) - f(\boldsymbol{\beta}') - \langle \nabla f(\boldsymbol{\beta}'), \boldsymbol{\beta} - \boldsymbol{\beta}' \rangle \leq \frac{L}{2} \left\|\boldsymbol{\beta}' - \boldsymbol{\beta}\right\|^2$$

Rem: positivity : consequence of convexity; second inequality Taylor expansion

Rem: if $f$ is twice differentiable $\nabla^2 f \preceq L \cdot \mathrm{Id}_d$ in the sense that $L \cdot \mathrm{Id}_d - \nabla^2 f$ is semi-definite positive, then $\nabla f$ is $L$-Lipschitz

# Proof (can be skipped)

Fix $\boldsymbol{\beta}^0$, and assume the previous inequality holds for any $\boldsymbol{\beta} \in \mathbb{R}^d$:

$$f(\boldsymbol{\beta}) - f(\boldsymbol{\beta}^0) - \langle \nabla f(\boldsymbol{\beta}^0), \boldsymbol{\beta} - \boldsymbol{\beta}^0 \rangle \leq \frac{L}{2} \left\| \boldsymbol{\beta}^0 - \boldsymbol{\beta} \right\|^2$$

this yields

$$\begin{aligned} f(\boldsymbol{\beta}) \leq & f(\boldsymbol{\beta}^0) + \langle \nabla f(\boldsymbol{\beta}^0), \boldsymbol{\beta} - \boldsymbol{\beta}^0 \rangle + \frac{L}{2} \|\boldsymbol{\beta}^0 - \boldsymbol{\beta}\|^2 \\ = & \tfrac{L}{2} \left\| \boldsymbol{\beta}^0 - \tfrac{1}{L} \nabla f(\boldsymbol{\beta}^0) - \boldsymbol{\beta} \right\|^2 + f(\boldsymbol{\beta}^0) - \tfrac{1}{2L} \left\| \nabla f(\boldsymbol{\beta}^0) \right\|^2 \\ := & g(\boldsymbol{\beta}^0, \boldsymbol{\beta}) \end{aligned}$$

Hence : $\quad \forall \boldsymbol{\beta} \in \mathbb{R}^p, \quad \begin{cases} g(\boldsymbol{\beta}^0 | \boldsymbol{\beta}^0) = f(\boldsymbol{\beta}^0) \\ f(\boldsymbol{\beta}) \leq g(\boldsymbol{\beta}^0 | \boldsymbol{\beta}) \end{cases}$

Lead to a tight upper bound that can be minimized:

$$\underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min}\, g(\boldsymbol{\beta}^0 | \boldsymbol{\beta}) = \boldsymbol{\beta}^0 - \frac{1}{L} \nabla f(\boldsymbol{\beta}^0)$$

# Outline

Motivation / Examples

Variable selection and sparsity

Algorithms for non-smooth convex problems
- Majorization / Minimization
- Proximal methods — Forward / Backward
- (Block) Coordinate descent
- Stopping criterion

Extensions: non convex, general structure

# Proximal gradient descent: non-smooth case

Optimization problem: $\boxed{\min_{\boldsymbol{\beta} \in \mathbb{R}^p} f(\boldsymbol{\beta}) + \psi(\boldsymbol{\beta})}$

Properties: $f$ convex, gradient $L$-Lipschitz; $\psi$ but non necessarily smooth (can have kinks)

**Example**: $f(\boldsymbol{\beta}) = \frac{1}{2} \|X\beta - y\|^2, \psi(\boldsymbol{\beta}) = \lambda \|\beta\|_1$

Rem: fix step size (sub-)gradient descent does not converge: take $f = 0$, $\psi = |\cdot|$ and use $\beta_0 = 1/2, \alpha = 1$ (ping-pong!)

# Proximal gradient descent: non-smooth case

Optimization problem: $\boxed{\min_{\boldsymbol{\beta} \in \mathbb{R}^p} f(\boldsymbol{\beta}) + \psi(\boldsymbol{\beta})}$

Properties: $f$ convex, gradient $L$-Lipschitz; $\psi$ but non necessarily smooth (can have kinks)

**Example**: $f(\boldsymbol{\beta}) = \frac{1}{2} \|X\beta - y\|^2, \psi(\boldsymbol{\beta}) = \lambda \|\beta\|_1$

Rem: fix step size (sub-)gradient descent does not converge: take $f = 0$, $\psi = |\cdot|$ and use $\beta_0 = 1/2, \alpha = 1$ (ping-pong!)

# Proximal operators / algorithms

Properties: $f$ convex, gradient $L$-Lipschitz; $\psi$ convex s.t. $\mathrm{prox}_{\psi}$ (the **proximal** operator[14] of $\psi$) has a closed-form, where

$$\mathrm{prox}_{\psi}\left(\boldsymbol{\beta}^0\right) = \mathop{\arg\min}_{\boldsymbol{\beta}\in\mathbb{R}^p} \tfrac{1}{2}\|\boldsymbol{\beta} - \boldsymbol{\beta}^0\|^2 + \psi(\boldsymbol{\beta})$$

[14] J.-J. Moreau. "Fonctions convexes duales et points proximaux dans un espace hilbertien". In: *C. R. Acad. Sci. Paris* 255 (1962), pp. 2897–2899.

[15] N. Parikh et al. "Proximal algorithms". In: *Foundations and Trends in Machine Learning* 1.3 (2013), pp. 1–108.

[16] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. New York: Springer, 2011, pp. xvi+468.

# Proximal operators / algorithms

Properties: $f$ convex, gradient $L$-Lipschitz; $\psi$ convex s.t. $\operatorname{prox}_\psi$ (the **proximal** operator[14] of $\psi$) has a closed-form, where

$$\operatorname{prox}_\psi(\boldsymbol{\beta}^0) = \operatorname*{arg\,min}_{\boldsymbol{\beta}\in\mathbb{R}^p} \tfrac{1}{2}\|\boldsymbol{\beta}-\boldsymbol{\beta}^0\|^2 + \psi(\boldsymbol{\beta})$$

Surrogate: $g(\boldsymbol{\beta}|\boldsymbol{\beta}^t) = f(\boldsymbol{\beta}^t) + \langle \nabla f(\boldsymbol{\beta}^t), \boldsymbol{\beta}-\boldsymbol{\beta}^t\rangle + \frac{L\|\boldsymbol{\beta}^t-\boldsymbol{\beta}\|^2}{2} + \psi(\boldsymbol{\beta})$

---

[14] J.-J. Moreau. "Fonctions convexes duales et points proximaux dans un espace hilbertien". In: *C. R. Acad. Sci. Paris* 255 (1962), pp. 2897–2899.

[15] N. Parikh et al. "Proximal algorithms". In: *Foundations and Trends in Machine Learning* 1.3 (2013), pp. 1–108.

[16] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. New York: Springer, 2011, pp. xvi+468.

# Proximal operators / algorithms

Properties: $f$ convex, gradient $L$-Lipschitz; $\psi$ convex s.t. $\operatorname{prox}_\psi$ (the **proximal** operator[14] of $\psi$) has a closed-form, where

$$\operatorname{prox}_\psi(\boldsymbol{\beta}^0) = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, \tfrac{1}{2}\|\boldsymbol{\beta}-\boldsymbol{\beta}^0\|^2 + \psi(\boldsymbol{\beta})$$

Surrogate: $g(\boldsymbol{\beta}|\boldsymbol{\beta}^t) = f(\boldsymbol{\beta}^t) + \langle \nabla f(\boldsymbol{\beta}^t), \boldsymbol{\beta}-\boldsymbol{\beta}^t\rangle + \frac{L\|\boldsymbol{\beta}^t-\boldsymbol{\beta}\|^2}{2} + \psi(\boldsymbol{\beta})$

Update rule :

$$\boxed{\boldsymbol{\beta}^{t+1} = \operatorname{prox}_{\frac{\psi}{L}}\left(\boldsymbol{\beta}^t - \frac{1}{L}\nabla f(\boldsymbol{\beta}^t)\right)}$$

[14] J.-J. Moreau. "Fonctions convexes duales et points proximaux dans un espace hilbertien". In: *C. R. Acad. Sci. Paris* 255 (1962), pp. 2897–2899.

[15] N. Parikh et al. "Proximal algorithms". In: *Foundations and Trends in Machine Learning* 1.3 (2013), pp. 1–108.

[16] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. New York: Springer, 2011, pp. xvi+468.

# Proximal operators / algorithms

Properties: $f$ convex, gradient $L$-Lipschitz; $\psi$ convex s.t. $\operatorname{prox}_\psi$ (the **proximal** operator[14] of $\psi$) has a closed-form, where

$$\operatorname{prox}_\psi(\boldsymbol{\beta}^0) = \operatorname*{arg\,min}_{\boldsymbol{\beta}\in\mathbb{R}^p} \tfrac{1}{2}\|\boldsymbol{\beta}-\boldsymbol{\beta}^0\|^2 + \psi(\boldsymbol{\beta})$$

Surrogate: $g(\boldsymbol{\beta}|\boldsymbol{\beta}^t) = f(\boldsymbol{\beta}^t) + \langle \nabla f(\boldsymbol{\beta}^t), \boldsymbol{\beta}-\boldsymbol{\beta}^t\rangle + \frac{L\|\boldsymbol{\beta}^t-\boldsymbol{\beta}\|^2}{2} + \psi(\boldsymbol{\beta})$

Update rule :

$$\boxed{\boldsymbol{\beta}^{t+1} = \operatorname{prox}_{\frac{\psi}{L}}\left(\boldsymbol{\beta}^t - \tfrac{1}{L}\nabla f(\boldsymbol{\beta}^t)\right)}$$

More details on $\operatorname{prox}$ properties:

- Prox algorithms recipes[15]
- Mathematical theory/analysis[16]

---

[14] J.-J. Moreau. "Fonctions convexes duales et points proximaux dans un espace hilbertien". In: *C. R. Acad. Sci. Paris* 255 (1962), pp. 2897–2899.

[15] N. Parikh et al. "Proximal algorithms". In: *Foundations and Trends in Machine Learning* 1.3 (2013), pp. 1–108.

[16] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. New York: Springer, 2011, pp. xvi+468.

# Proof (can be skipped)

Proof (*cf.* gradient descent):

$$\boldsymbol{\beta}^{t+1} = \arg\min_{\boldsymbol{\beta}\in\mathbb{R}^p} \frac{L\|\boldsymbol{\beta}^t - \frac{1}{L}\nabla f(\boldsymbol{\beta}^t) - \boldsymbol{\beta}\|^2}{2} + \psi(\boldsymbol{\beta})$$

# Examples of prox operators

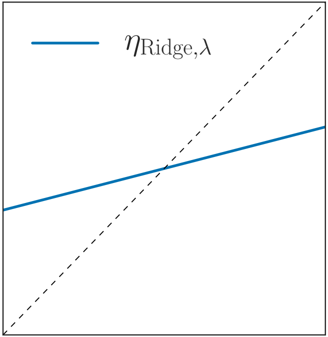
$$\boxed{\operatorname{prox}_\psi(w) = \operatorname*{arg\,min}_{z \in \mathbb{R}^p} \left( \frac{1}{2}\|z - w\|_2^2 + \psi(z) \right)}$$

- $\psi = 0$, then $\operatorname{prox}_\psi = \mathrm{Id}$ (**Null function**)
- $\psi = \iota_C$ for a closed convex set $C \subset \mathbb{R}^p$, then $\operatorname{prox}_\psi = \pi_C$, projection over the set $C$ (**Indicator function**)
- $\psi = \lambda|\cdot|$, then $\operatorname{prox}_\psi(w) = \eta_{\mathrm{ST},\lambda}(w) = \operatorname{sign}(w)(|w| - \lambda)_+$ (Soft-Thresholding)
- $\psi = \lambda\|\cdot\|_1$, then $\operatorname{prox}_\psi(w) = (\eta_{\mathrm{ST},\lambda}(w_1), \ldots, \eta_{\mathrm{ST},\lambda}(w_1))^\top$ (Vector Soft-Thresholding)

# 1D Regularization: Ridge

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

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

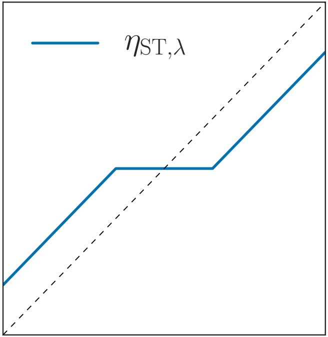



$\ell_2$ shrinkage : Ridge

# 1D Regularization: Lasso

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

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

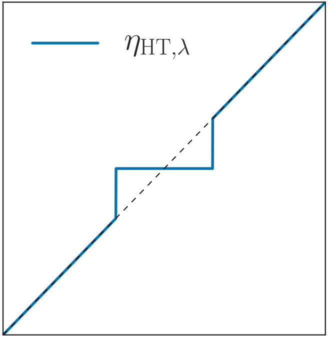



$\ell_1$ shrinkage: soft-thresholding

# 1D Regularization: $\ell_0$

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

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




$\ell_0$ shrinkage: hard-thresholding

# Forward-Backward / Iterative Soft Thresholding (ISTA)

Extension of gradient descent for composite functions:

General Forward-Backward

Choose step size value: $\alpha$
Initialization: $\boldsymbol{\beta} = 0 \in \mathbb{R}^p$
While not converged
$\boldsymbol{\beta} \leftarrow \mathrm{prox}_{\alpha\psi}\left(\boldsymbol{\beta} - \alpha \nabla f(\boldsymbol{\beta})\right)$

# Forward-Backward / Iterative Soft Thresholding (ISTA)

Extension of gradient descent for composite functions:

General Forward-Backward

---

Choose step size value: $\alpha$
Initialization: $\boldsymbol{\beta} = 0 \in \mathbb{R}^p$
While not converged
$\boldsymbol{\beta} \leftarrow \mathrm{prox}_{\alpha\psi}\left(\boldsymbol{\beta} - \alpha \nabla f(\boldsymbol{\beta})\right)$

---

Iterative Soft-thresholding (ISTA)

---

Choose step size value: $\alpha$
Initialization: $\boldsymbol{\beta} = 0 \in \mathbb{R}^p$
While not converged
$\boldsymbol{\beta} \leftarrow \eta_{\mathrm{ST},\alpha\lambda}\left(\boldsymbol{\beta} + \alpha X^\top (y - X\boldsymbol{\beta})\right)$

---

# Forward-Backward / Iterative Soft Thresholding (ISTA) (II)

- Interesting when the operator $z \mapsto X^\top z$ can be performed efficiently, *e.g.*, for FFT, Wavelet transforms, etc. Hence common in **Image/Signal processing**
- Requires $\alpha$ to be tuned: often set $\alpha = 1/L = 1/\mu_{\max}(X^\top X)$ ($\mu_{\max}(X^\top X)$ spectral radius of $X^\top X$), or by line-search
- Acceleration : Fast Iterative Soft Thresholding Algorithm (FISTA)[17],[18] (momentum used)

[17] Y. Nesterov. "A method for solving a convex programming problem with rate of convergence $O(1/k^2)$". In: *Soviet Math. Doklady* 269.3 (1983), pp. 543–547.

[18] A. Beck and M. Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". In: *SIAM J. Imaging Sci.* 2.1 (2009), pp. 183–202.

# Outline

Motivation / Examples

Variable selection and sparsity

Algorithms for non-smooth convex problems

- Majorization / Minimization
- Proximal methods — Forward / Backward
- (Block) Coordinate descent
- Stopping criterion

Extensions: non convex, general structure

# Statistics / Machine Learning: Coordinate Descent

Objective: optimize $\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, F(\boldsymbol{\beta}) = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, f(\boldsymbol{\beta}) + g(\boldsymbol{\beta})$

---

**Algorithm:** Coordinate Descent

---

**Input :** $f$, epochs $K$ (or passes over the data)

Init: $k = 0$ and $\boldsymbol{\beta}^{(k)} = 0 \in \mathbb{R}^p$

---

# Statistics / Machine Learning: Coordinate Descent

Objective: optimize $\quad \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min}\, F(\boldsymbol{\beta}) = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min}\, f(\boldsymbol{\beta}) + g(\boldsymbol{\beta})$

---

**Algorithm:** Coordinate Descent

---

**Input :** $f$, epochs $K$ (or passes over the data)

Init: $k = 0$ and $\boldsymbol{\beta}^{(k)} = 0 \in \mathbb{R}^p$

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

---

# Statistics / Machine Learning: Coordinate Descent

Objective: optimize $\underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} F(\boldsymbol{\beta}) = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\min} f(\boldsymbol{\beta}) + g(\boldsymbol{\beta})$

**Algorithm:** Coordinate Descent

**Input** : $f$, epochs $K$ (or passes over the data)
Init: $k = 0$ and $\boldsymbol{\beta}^{(k)} = 0 \in \mathbb{R}^p$
**for** $k = 1, \ldots, K$ **do**

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

# Statistics / Machine Learning: Coordinate Descent

Objective: optimize $\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, F(\boldsymbol{\beta}) = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, f(\boldsymbol{\beta}) + g(\boldsymbol{\beta})$

---

**Algorithm:** Coordinate Descent

---

**Input** : $f$, epochs $K$ (or passes over the data)

Init: $k = 0$ and $\boldsymbol{\beta}^{(k)} = 0 \in \mathbb{R}^p$

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

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

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

---

# Statistics / Machine Learning: Coordinate Descent

Objective: optimize $\displaystyle\mathop{\arg\min}_{\boldsymbol{\beta}\in\mathbb{R}^p} F(\boldsymbol{\beta}) = \mathop{\arg\min}_{\boldsymbol{\beta}\in\mathbb{R}^p} f(\boldsymbol{\beta}) + g(\boldsymbol{\beta})$

---

**Algorithm:** Coordinate Descent

---

**Input** : $f$, epochs $K$ (or passes over the data)

Init: $k = 0$ and $\boldsymbol{\beta}^{(k)} = 0 \in \mathbb{R}^p$

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

$$\beta_1^{(k)} \approx \mathop{\arg\min}_{\beta_1\in\mathbb{R}} F(\beta_1, \beta_2^{(k-1)}, \beta_3^{(k-1)}, \ldots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$$\beta_2^{(k)} \approx \mathop{\arg\min}_{\beta_2\in\mathbb{R}} F(\beta_1^{(k)}, \beta_2, \beta_3^{(k-1)}, \ldots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$$\beta_3^{(k)} \approx \mathop{\arg\min}_{\beta_3\in\mathbb{R}} F(\beta_1^{(k)}, \beta_2^{(k)}, \beta_3, \ldots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

---

# Statistics / Machine Learning: Coordinate Descent

Objective: optimize $\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, F(\boldsymbol{\beta}) = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, f(\boldsymbol{\beta}) + g(\boldsymbol{\beta})$

---

**Algorithm:** Coordinate Descent

---

**Input** : $f$, epochs $K$ (or passes over the data)

Init: $k = 0$ and $\boldsymbol{\beta}^{(k)} = 0 \in \mathbb{R}^p$

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

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

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

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

$\vdots$

---

# Statistics / Machine Learning: Coordinate Descent

Objective: optimize $\displaystyle \operatorname*{arg\,min}_{\boldsymbol{\beta}\in\mathbb{R}^p} F(\boldsymbol{\beta}) = \operatorname*{arg\,min}_{\boldsymbol{\beta}\in\mathbb{R}^p} f(\boldsymbol{\beta}) + g(\boldsymbol{\beta})$

---

**Algorithm:** Coordinate Descent

---

**Input** : $f$, epochs $K$ (or passes over the data)

Init: $k = 0$ and $\boldsymbol{\beta}^{(k)} = 0 \in \mathbb{R}^p$

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

$$\beta_1^{(k)} \approx \operatorname*{arg\,min}_{\beta_1\in\mathbb{R}} F(\beta_1, \beta_2^{(k-1)}, \beta_3^{(k-1)}, \ldots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$$\beta_2^{(k)} \approx \operatorname*{arg\,min}_{\beta_2\in\mathbb{R}} F(\beta_1^{(k)}, \beta_2, \beta_3^{(k-1)}, \ldots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$$\beta_3^{(k)} \approx \operatorname*{arg\,min}_{\beta_3\in\mathbb{R}} F(\beta_1^{(k)}, \beta_2^{(k)}, \beta_3, \ldots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$\vdots$

$$\beta_p^{(k)} \approx \operatorname*{arg\,min}_{\beta_p\in\mathbb{R}} F(\beta_1^{(k)}, \beta_2^{(k)}, \beta_3^{(k)}, \ldots, \beta_{p-1}^{(k)}, \beta_p)$$

---

# Statistics / Machine Learning: Coordinate Descent

Objective: optimize $\underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, F(\boldsymbol{\beta}) = \underset{\boldsymbol{\beta}\in\mathbb{R}^p}{\arg\min}\, f(\boldsymbol{\beta}) + g(\boldsymbol{\beta})$

---

**Algorithm:** Coordinate Descent

---

**Input :** $f$, epochs $K$ (or passes over the data)

Init: $k = 0$ and $\boldsymbol{\beta}^{(k)} = 0 \in \mathbb{R}^p$

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

$$\beta_1^{(k)} \approx \underset{\beta_1\in\mathbb{R}}{\arg\min}\, F(\beta_1, \beta_2^{(k-1)}, \beta_3^{(k-1)}, \ldots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$$\beta_2^{(k)} \approx \underset{\beta_2\in\mathbb{R}}{\arg\min}\, F(\beta_1^{(k)}, \beta_2, \beta_3^{(k-1)}, \ldots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$$\beta_3^{(k)} \approx \underset{\beta_3\in\mathbb{R}}{\arg\min}\, F(\beta_1^{(k)}, \beta_2^{(k)}, \beta_3, \ldots, \beta_{p-1}^{(k-1)}, \beta_p^{(k-1)})$$

$$\vdots$$

$$\beta_p^{(k)} \approx \underset{\beta_p\in\mathbb{R}}{\arg\min}\, F(\beta_1^{(k)}, \beta_2^{(k)}, \beta_3^{(k)}, \ldots, \beta_{p-1}^{(k)}, \beta_p)$$

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

---

# Popular visit schemes

Need to visit coordinate **regularly** or **greedily** for convergence

Popular ones:

- **Cyclic** (Gauss-Seidel): visit $1, 2, \ldots, p, 1, 2, \ldots, p, 1, 2, \ldots$

# Popular visit schemes

Need to visit coordinate **regularly** or **greedily** for convergence

Popular ones:

- **Cyclic** (Gauss-Seidel): visit $1, 2, \ldots, p, 1, 2, \ldots, p, 1, 2, \ldots$
- **Random**: *i.i.d.* uniformly with resampling

# Popular visit schemes

Need to visit coordinate **regularly** or **greedily** for convergence

Popular ones:

- **Cyclic** (Gauss-Seidel): visit $1, 2, \ldots, p, 1, 2, \ldots, p, 1, 2, \ldots$
- **Random**: *i.i.d.* uniformly with resampling
- **Shuffle**: uniform permutations

# Popular visit schemes

Need to visit coordinate **regularly** or **greedily** for convergence

Popular ones:

- **Cyclic** (Gauss-Seidel): visit $1, 2, \ldots, p, 1, 2, \ldots, p, 1, 2, \ldots$
- **Random**: *i.i.d.* uniformly with resampling
- **Shuffle**: uniform permutations
- **Greedy** (Gauss-Southwell): look for the "best" possible

# Popular visit schemes

Need to visit coordinate **regularly** or **greedily** for convergence

Popular ones:

- **Cyclic** (Gauss-Seidel): visit $1, 2, \ldots, p, 1, 2, \ldots, p, 1, 2, \ldots$
- **Random**: *i.i.d.* uniformly with resampling
- **Shuffle**: uniform permutations
- **Greedy** (Gauss-Southwell): look for the "best" possible

Rem: same idea used in linear solvers

# Popular visit schemes

Need to visit coordinate **regularly** or **greedily** for convergence

Popular ones:

- **Cyclic** (Gauss-Seidel): visit $1, 2, \ldots, p, 1, 2, \ldots, p, 1, 2, \ldots$
- **Random**: *i.i.d.* uniformly with resampling
- **Shuffle**: uniform permutations
- **Greedy** (Gauss-Southwell): look for the "best" possible

Rem: same idea used in linear solvers

Rem: coordinate-wise proximal gradient descent is enough (when 1D optimal is not in closed form, *e.g.*, logistic regression)

# Motivation for coordinate descent

- useful when $p$ (very) large
- "block" strategy: update a block (or one coordinate at a time)
- convergence guarantees:

1. Smooth functions[19]: $\arg\min_{\beta} f(\beta)$

   with $f$ convex and gradient Lipschitz ($F$ smooth)

2. Composite[20] : $\arg\min_{\beta} f(\beta) + g(\beta)$

   $f$ convex and gradient Lipschitz, and $g$ convex separable:

   $g(\beta) = \sum_{j=1}^{p} g_j(\beta_j)$

---

[19] B. Martinet. "Brève communication. Régularisation d'inéquations variationnelles par approximations successives". In: *Revue française d'informatique et de recherche opérationnelle. Série rouge* 4.R3 (1970), pp. 154–158.

[20] P. Tseng. "Convergence of a block coordinate descent method for nondifferentiable minimization". In: *J. Optim. Theory Appl.* 109.3 (2001), pp. 475–494.

# Motivation for coordinate descent

- useful when $p$ (very) large
- "block" strategy: update a block (or one coordinate at a time)
- convergence guarantees:

1. Smooth functions[19]: $\arg\min_{\beta} f(\beta)$
   with $f$ convex and gradient Lipschitz ($F$ smooth)
2. Composite[20] : $\arg\min_{\beta} f(\beta) + g(\beta)$
   $f$ convex and gradient Lipschitz, and $g$ convex separable:
   $$g(\beta) = \sum_{j=1}^{p} g_j(\beta_j)$$

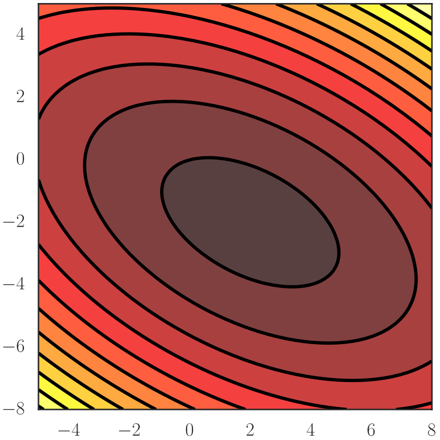
---

[19] B. Martinet. "Brève communication. Régularisation d'inéquations variationnelles par approximations successives". In: *Revue française d'informatique et de recherche opérationnelle. Série rouge* 4.R3 (1970), pp. 154–158.

[20] P. Tseng. "Convergence of a block coordinate descent method for nondifferentiable minimization". In: *J. Optim. Theory Appl.* 109.3 (2001), pp. 475–494.

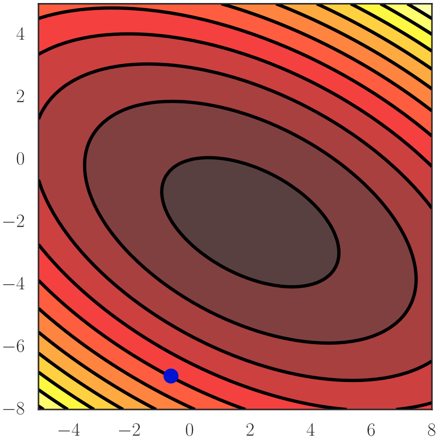
# Motivation (Convex case)

Convergence toward minimum for smooth case Tseng (2001)



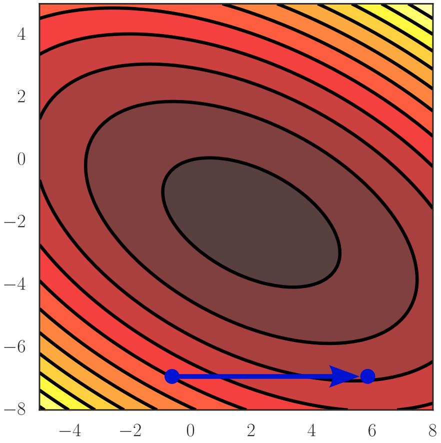
# Motivation (Convex case)

Convergence toward minimum for smooth case Tseng (2001)



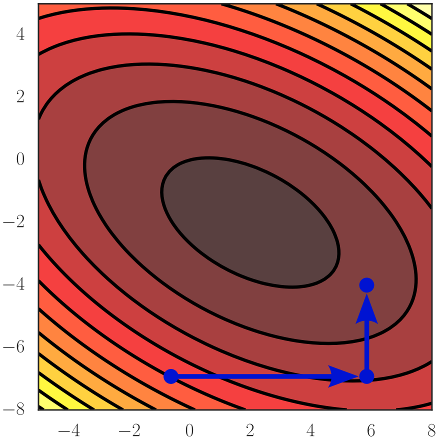
# Motivation (Convex case)

Convergence toward minimum for smooth case Tseng (2001)



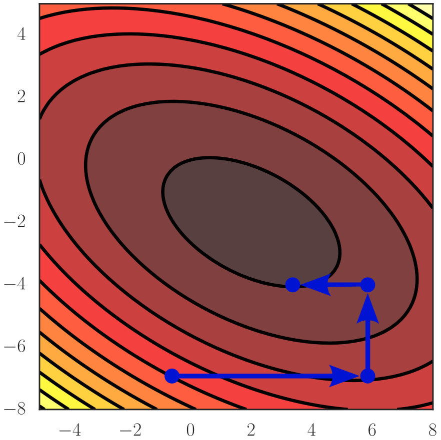
# Motivation (Convex case)

Convergence toward minimum for smooth case Tseng (2001)



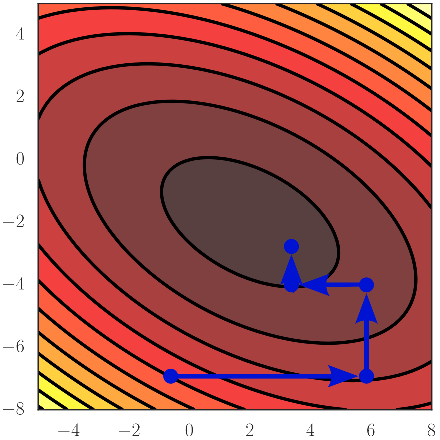
# Motivation (Convex case)

Convergence toward minimum for smooth case Tseng (2001)



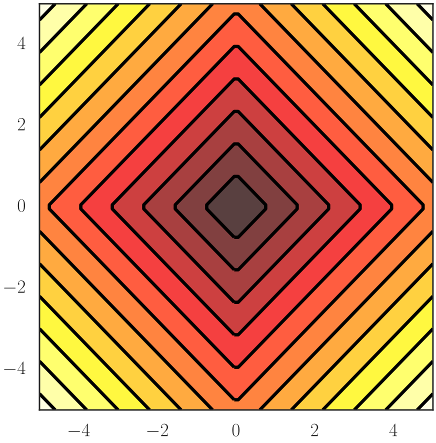
# Motivation (Convex case)

Convergence toward minimum for smooth case Tseng (2001)



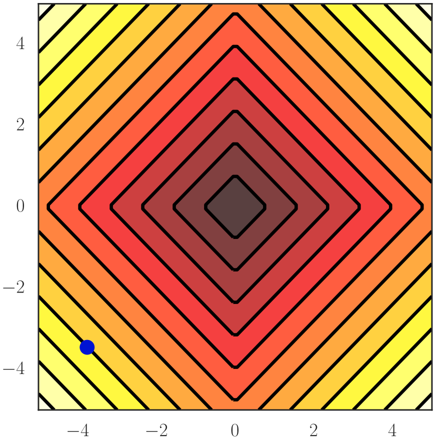
# Motivation (Convex case)

Convergence toward minimum for separable case Tseng (2001)



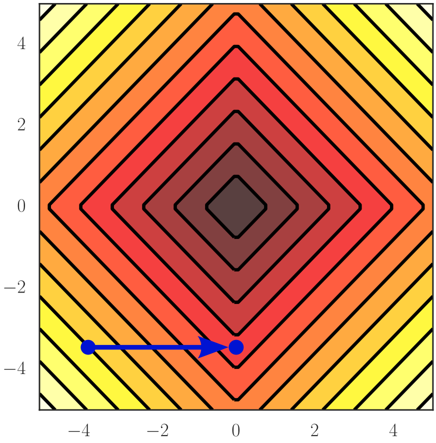
# Motivation (Convex case)

Convergence toward minimum for separable case Tseng (2001)



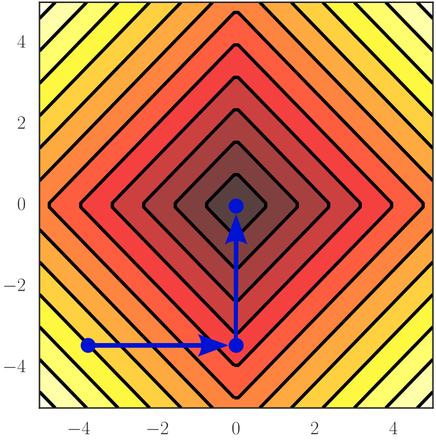
# Motivation (Convex case)

Convergence toward minimum for separable case Tseng (2001)



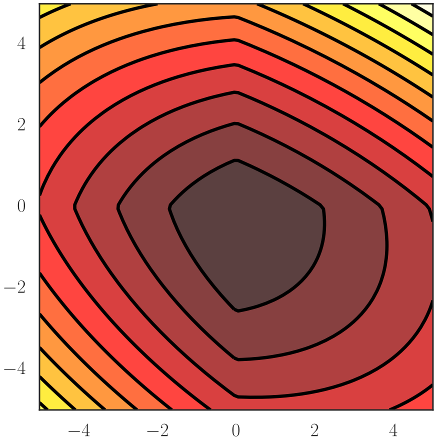
# Motivation (Convex case)

Convergence toward minimum for separable case Tseng (2001)



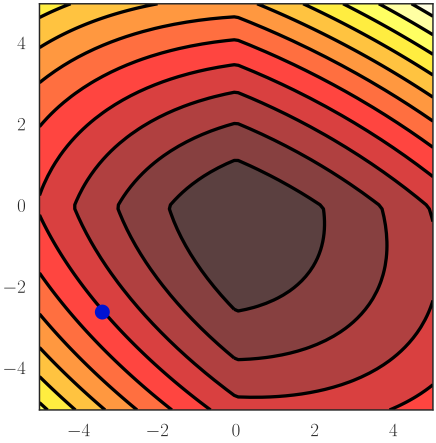
# Motivation (Convex case)

Convergence toward minimum smooth + separable case Tseng (2001)



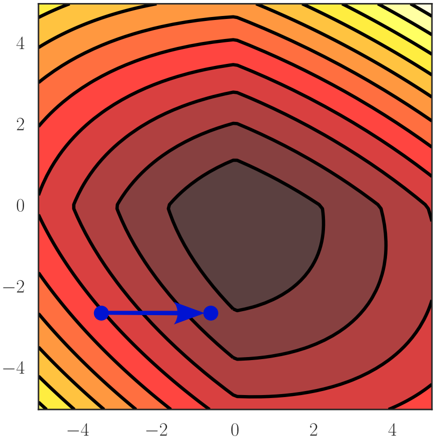
# Motivation (Convex case)

Convergence toward minimum smooth + separable case Tseng (2001)



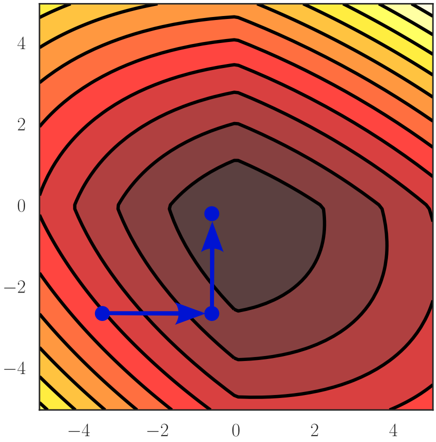
# Motivation (Convex case)

Convergence toward minimum smooth + separable case Tseng (2001)



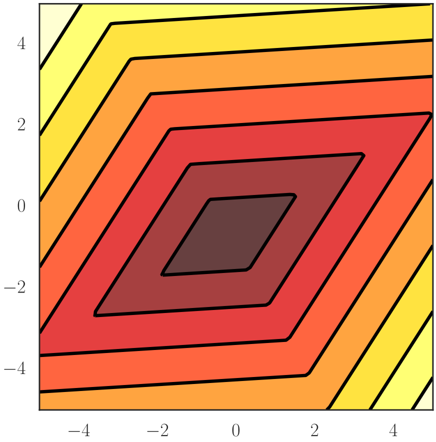
# Motivation (Convex case)

Convergence toward minimum smooth + separable case Tseng (2001)



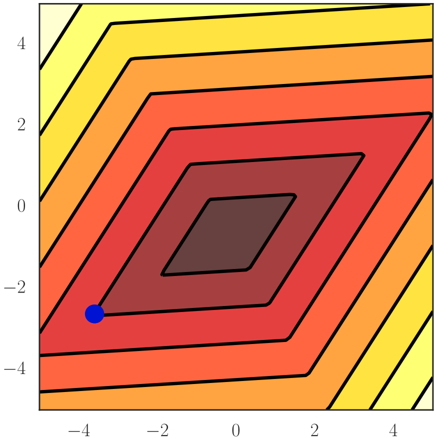
# Motivation (Convex case)

**Beware**: for non-smooth / non separable cases



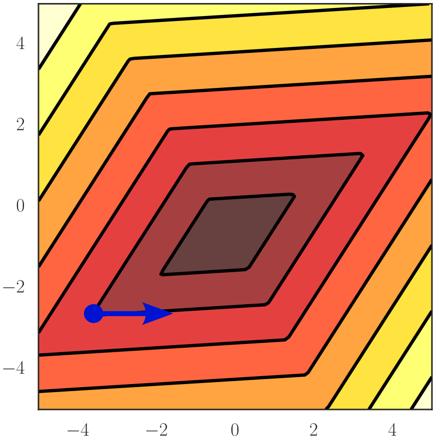
# Motivation (Convex case)

**Beware**: for non-smooth / non separable cases



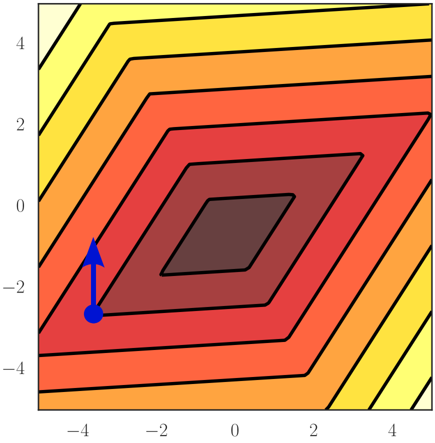
# Motivation (Convex case)

**Beware**: for non-smooth / non separable cases



# Motivation (Convex case)

**Beware**: for non-smooth / non separable cases



# CD for Lasso

Exact solution for partial update: soft-threshold coefficient-wise

$$\hat{\beta}_j = \eta_{\mathrm{ST},\lambda/\|\mathbf{x_j}\|^2}\left(\|\mathbf{x}_j\|^{-2}\langle y - \sum_{k\neq j}\beta_k\mathbf{x}_k, \mathbf{x}_j\rangle\right)$$

Lazy update : maintain residual $r = y - X\boldsymbol{\beta}$ and coeff. $\boldsymbol{\beta}$

# CD for Lasso

Exact solution for partial update: soft-threshold coefficient-wise

$$\hat{\beta}_j = \eta_{\mathrm{ST},\lambda/\|\mathbf{x_j}\|^2}\left( \|\mathbf{x}_j\|^{-2} \langle y - \sum_{k \neq j} \beta_k \mathbf{x}_k, \mathbf{x}_j \rangle \right)$$

<u>Lazy update</u> : maintain residual $r = y - X\boldsymbol{\beta}$ and coeff. $\boldsymbol{\beta}$

for any $j \in [\![1, p]\!]$, do:

# CD for Lasso

Exact solution for partial update: soft-threshold coefficient-wise

$$\hat{\beta}_j = \eta_{\mathrm{ST},\lambda/\|\mathbf{x}_\mathrm{j}\|^2}\left(\|\mathbf{x}_j\|^{-2}\langle y - \sum_{k\neq j}\beta_k \mathbf{x}_k, \mathbf{x}_j\rangle\right)$$

Lazy update : maintain residual $r = y - X\boldsymbol{\beta}$ and coeff. $\boldsymbol{\beta}$

$$r^{\mathrm{int}} \leftarrow r + \mathbf{x}_j\beta_j$$

for any $j \in [\![1, p]\!]$, do:

# CD for Lasso

Exact solution for partial update: soft-threshold coefficient-wise

$$\hat{\beta}_j = \eta_{\mathrm{ST},\lambda/\|\mathbf{x}_\mathrm{j}\|^2}\left(\|\mathbf{x}_j\|^{-2}\langle y - \sum_{k\neq j}\beta_k \mathbf{x}_k, \mathbf{x}_j\rangle\right)$$

Lazy update : maintain residual $r = y - X\boldsymbol{\beta}$ and coeff. $\boldsymbol{\beta}$

$$\text{for any } j \in [\![1,p]\!], \text{ do:} \quad \begin{aligned} r^{\mathrm{int}} &\leftarrow r + \mathbf{x}_j\beta_j \\ \beta_j &\leftarrow \eta_{\mathrm{ST},\lambda/\|\mathbf{x}_\mathrm{j}\|^2}\left(\mathbf{x}_j^\top r^{\mathrm{int}}/\|\mathbf{x}_j\|^2\right) \end{aligned}$$

# CD for Lasso

Exact solution for partial update: soft-threshold coefficient-wise

$$\hat{\beta}_j = \eta_{\mathrm{ST},\lambda/\|\mathbf{x_j}\|^2}\left(\|\mathbf{x}_j\|^{-2}\langle y - \sum_{k\neq j}\beta_k\mathbf{x}_k, \mathbf{x}_j\rangle\right)$$

Lazy update : maintain residual $r = y - X\boldsymbol{\beta}$ and coeff. $\boldsymbol{\beta}$

$$\text{for any } j \in [\![1,p]\!],\ \text{do:}\quad \begin{aligned} r^{\mathrm{int}} &\leftarrow r + \mathbf{x}_j\beta_j \\ \beta_j &\leftarrow \eta_{\mathrm{ST},\lambda/\|\mathbf{x_j}\|^2}\left(\mathbf{x}_j^\top r^{\mathrm{int}}/\|\mathbf{x}_j\|^2\right) \\ r &\leftarrow r^{\mathrm{int}} - \mathbf{x}_j\beta_j \end{aligned}$$

# CD for Lasso

Exact solution for partial update: soft-threshold coefficient-wise

$$\hat{\beta}_j = \eta_{\mathrm{ST},\lambda/\|\mathbf{x}_j\|^2}\left(\|\mathbf{x}_j\|^{-2}\langle y - \sum_{k\neq j}\beta_k \mathbf{x}_k, \mathbf{x}_j\rangle\right)$$

Lazy update : maintain residual $r = y - X\boldsymbol{\beta}$ and coeff. $\boldsymbol{\beta}$

$$\text{for any } j \in [\![1,p]\!], \text{ do:} \quad \begin{aligned} r^{\mathrm{int}} &\leftarrow r + \mathbf{x}_j \beta_j \\ \beta_j &\leftarrow \eta_{\mathrm{ST},\lambda/\|\mathbf{x}_j\|^2}\left(\mathbf{x}_j^\top r^{\mathrm{int}}/\|\mathbf{x}_j\|^2\right) \\ r &\leftarrow r^{\mathrm{int}} - \mathbf{x}_j\beta_j \end{aligned}$$

Low memory footprint:

- store residual vector: size $n$
- store coeff. vector : size $p$

Rem: often in statistic $\|\mathbf{x}_j\|_2^2 = 1$ or $n$ (normalization)

# Default solvers of this kind

- Python: `scikit-learn`[21] (coded in Cython)
- R: `glmnet` [22] (coded in Fortran, well...Mortran)

Below illustration on simple implementation:

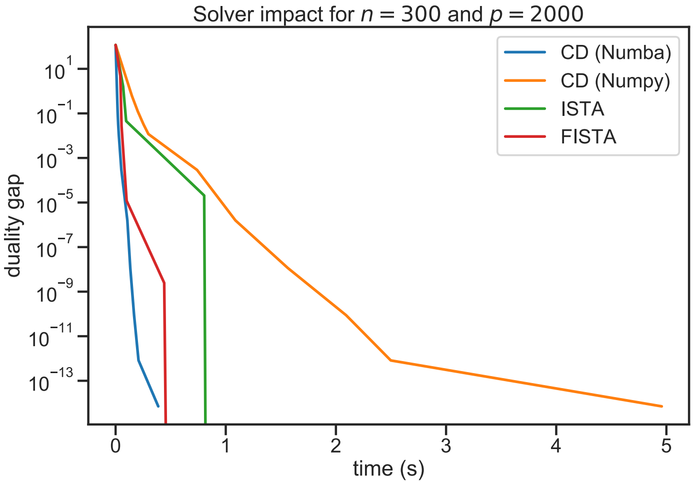
- CD `numpy` (not recommended, need low level language)
- CD `numba` (compilation “just in time”)
- ISTA `numpy`
- FISTA `numpy` (F = Fast)

---

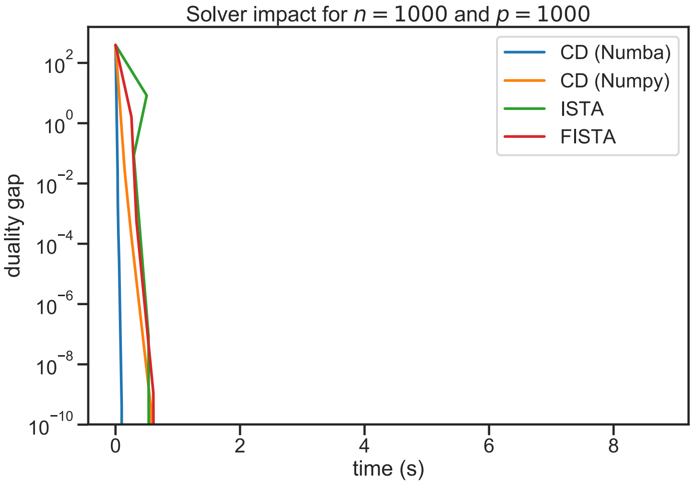
(21) https://scikit-learn.org

(22) https://github.com/cran/glmnet

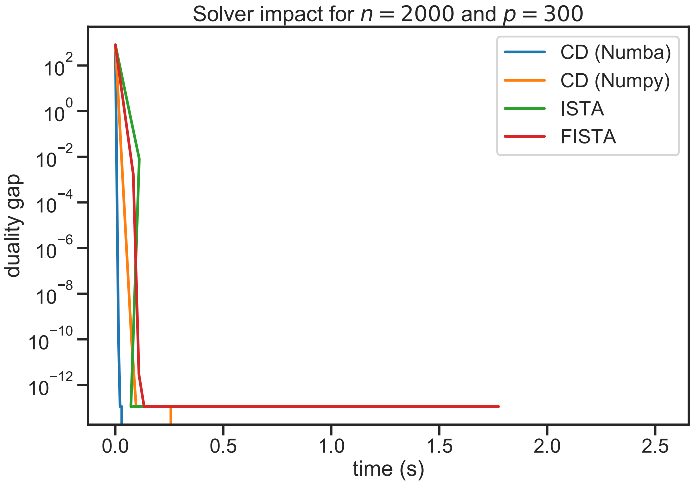
# Numerical comparisons: toy machine learning example



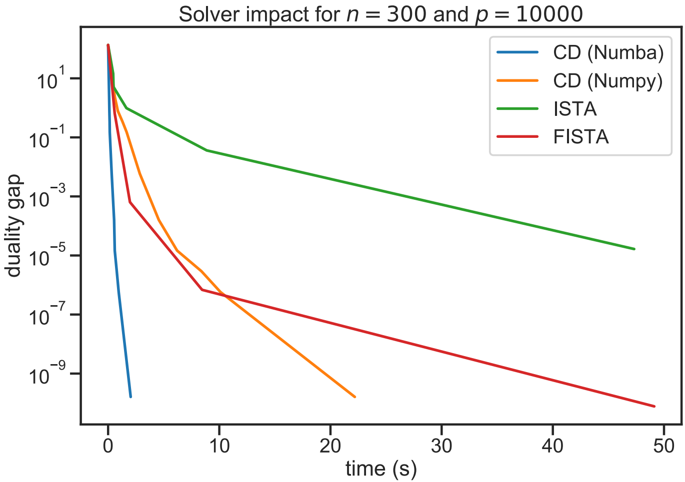
# Numerical comparisons: toy machine learning example



# Numerical comparisons: toy machine learning example



# Numerical comparisons: toy machine learning example



# Outline

# Stopping criterion

Rem: missing ingredient in the literature

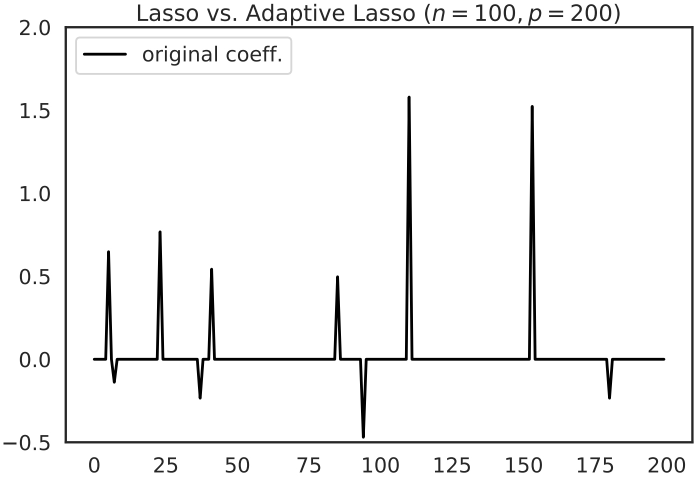
- gradient amplitude (smooth problem)
- violation of first order condition / KKT (non-smooth case)
- duality gap is small
- parameter stabilized
- ⋮

Rem: more at "Montpellier: Berceau de la data science (18-19 Juin)" about ICML paper[(23)] (duality gap and using it for learning...)

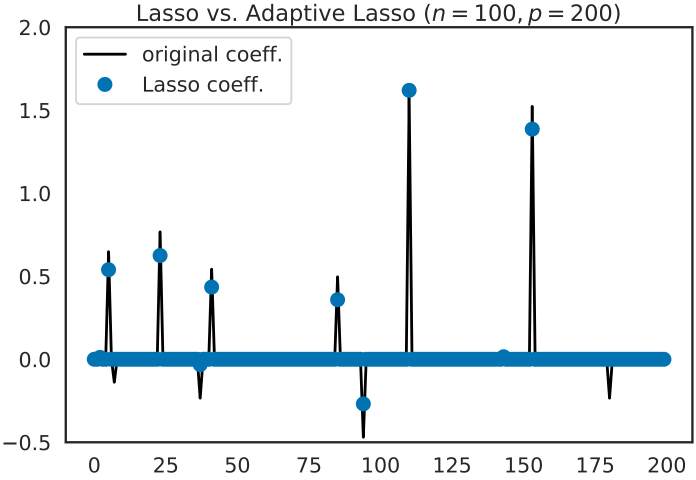
(23) E. Ndiaye et al. "Safe Grid Search with Optimal Complexity". In: *ICML*. 2018.

Is $\ell_1$-regularized least-squares the end of the story?

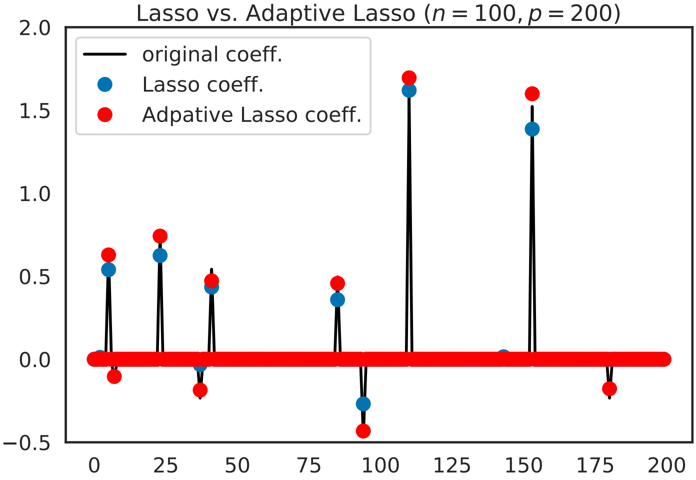
# Lasso and beyond



# Lasso and beyond



# Lasso and beyond



# Outline

Motivation / Examples

Variable selection and sparsity

Algorithms for non-smooth convex problems

Extensions: non convex, general structure

- Non-convex penalties
- Structured support for neuro-imaging framework

# Smooth non-convex penalties

Use better approximation of $\|\cdot\|_0$ by a non-convex function

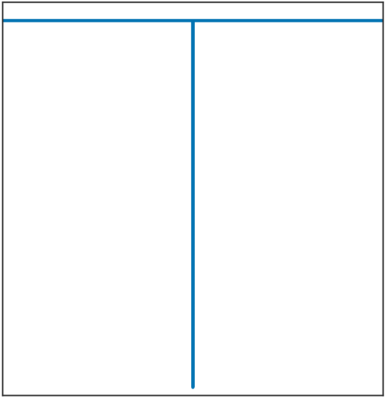
$$\boxed{\hat{\beta}^{\text{pen}}_{\lambda,\gamma} = \operatorname*{arg\,min}_{\beta \in \mathbb{R}^p} \left( \underbrace{\frac{1}{2}\|\mathbf{y} - X\beta\|_2^2}_{\textbf{data fitting}} + \underbrace{\sum_{j=1}^{p} \text{pen}_{\lambda,\gamma}(|\beta_j|)}_{\textbf{regularization}} \right)}$$

Requirements:

- non-smooth at zero (to induce thresholding effect)
- constant for large values (avoid shrinking large coeff.)

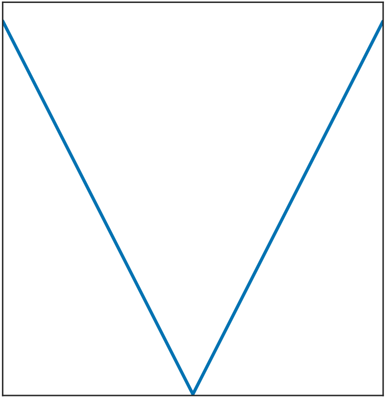
⚠ algorithmic and theoretical difficulties : stopping, local minima

# Standard non-convex penalties



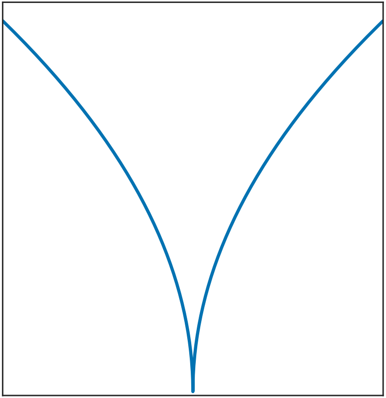
$$\ell_0 : \mathrm{pen}_{\lambda,\gamma}(t) = \frac{\lambda^2}{2}\mathbb{1}_{t=0}$$

# Standard non-convex penalties



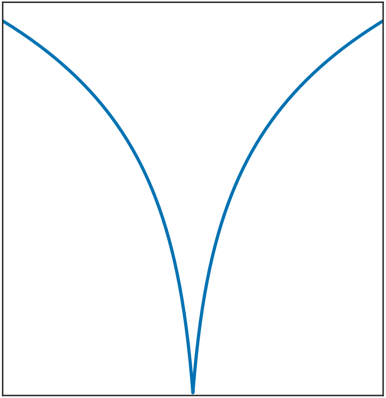
$$\ell_1 : \mathrm{pen}_{\lambda,\gamma}(t) = \lambda |t|_1$$

# Standard non-convex penalties



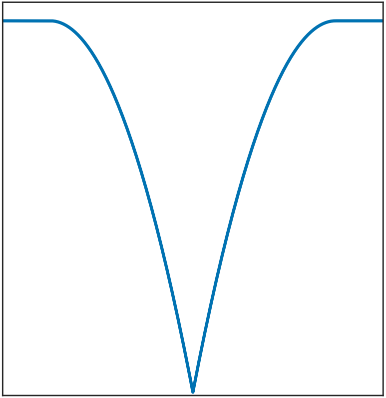
$$\ell_{1/2} : \mathrm{pen}_{\lambda,\gamma}(t) = \lambda|t|^q (q = 1/2)$$

# Standard non-convex penalties



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

# Standard non-convex penalties



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

# Designing a "nice" penalty[24]

Deriving necessary and sufficient conditions on a penalty s.t. :

- the $\ell_0$ problem shares global optimal solution(s) with the one from continuous penalty
- local minima for the continuous penalty are all local minima of the original $\ell_0$ problem

Leads to the some constraints, in particular satisfied by:

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

Rem: in 1D requires $\mathrm{pen}(0) = 0$, $\mathrm{pen}(t) = cste$ for large $|t|$ and concavity (!) over $\mathbb{R}^+$

[24] E. Soubies, L. Blanc-Féraud, and G. Aubert. "A Unified View of Exact Continuous Penalties for $\ell_2$-$\ell_0$ Minimization". In: *SIAM J. Optim.* 27.3 (2017), pp. 2034–2060.

# Algorithms for non-convex alternatives

- Majorization-Minimization: Adaptive-Lasso,[25] Re-weighted[26] $\ell_1$, Difference of Convex programming for sparse problems[27]
- Coordinate Descent[28]

⚠ no more global guarantee!

---

[25] H. Zou. “The adaptive lasso and its oracle properties”. In: *J. Amer. Statist. Assoc.* 101.476 (2006), pp. 1418–1429.

[26] E. J. Candès, M. B. Wakin, and S. P. Boyd. “Enhancing Sparsity by Reweighted $l_1$ Minimization”. In: *J. Fourier Anal. Applicat.* 14.5-6 (2008), pp. 877–905.

[27] G. Gasso, A. Rakotomamonjy, and S. Canu. “Recovering sparse signals with non-convex penalties and DC programming”. In: *IEEE Trans. Signal Process.* 57.12 (2009), pp. 4686–4698.

[28] P. Breheny and J. Huang. “Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection”. In: *Ann. Appl. Stat.* 5.1 (2011), p. 232.

# Outline

# Structured support

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

Vector and active coordinate (in orange):

Sparse support: any

Possible penalties: Lasso

$\|\boldsymbol{\beta}\|_1 = \sum_{j=1}^{p} |\boldsymbol{\beta}_j|$

# Structured support

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

Vector and active coordinate (in orange):

Sparse support: group

Possible penalties: Group-Lasso

$\|\boldsymbol{\beta}\|_{2,1} = \sum_{g \in G} \|\boldsymbol{\beta}_g\|_2$

# Structured support

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

Vector and active coordinate (in orange):

Sparse support: group + sub-groups

Possible penalties: Sparse-Group-Lasso

$$\alpha \|\boldsymbol{\beta}\|_1 + (1-\alpha)\|\boldsymbol{\beta}\|_{2,1} = \ \alpha \sum_{j=1}^{p} |\boldsymbol{\beta}_j| + (1-\alpha) \sum_{g \in G} \|\boldsymbol{\beta}_g\|_2$$

# Group-Lasso

The $\ell_1$ norm penalty ensures that few coefficients are active, but no other structure is enforced

One can aim at:

- group/block wise sparsity: Group-Lasso[(29)]
- individual and group wise : Sparse Group-Lasso[(30)]
- hierarchical structures (*e.g.*, for higher order interactions)[(31)]
- graph structures, gradients structures, etc.

---

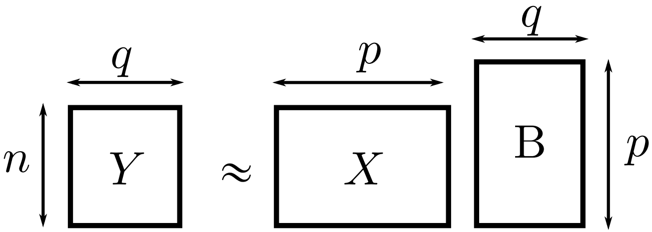
(29) M. Yuan and Y. Lin. "Model selection and estimation in regression with grouped variables". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 68.1 (2006), pp. 49–67.

(30) N. Simon et al. "A sparse-group lasso". In: *J. Comput. Graph. Statist.* 22.2 (2013), pp. 231–245. ISSN: 1061-8600.

(31) J. Bien, J. Taylor, and R. Tibshirani. "A lasso for hierarchical interactions". In: *Ann. Statist.* 41.3 (2013), pp. 1111–1141.

# Back to multi-task regression

One aims at jointly solving $m$ linear regression: $Y \approx X\mathrm{B}$




with

- $Y \in \mathbb{R}^{n \times q}$: observation matrix
- $X \in \mathbb{R}^{n \times p}$: design matrix (known)
- $\mathrm{B} \in \mathbb{R}^{p \times q}$: 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:

$$\boxed{\hat{\mathrm{B}}_\lambda = \operatorname*{arg\,min}_{\mathrm{B}\in\mathbb{R}^{p\times q}} \left( \underbrace{\frac{1}{2}\|Y - X\mathrm{B}\|_F^2}_{\textbf{data fitting}} + \underbrace{\lambda\Omega(\mathrm{B})}_{\textbf{regularization}} \right)}$$

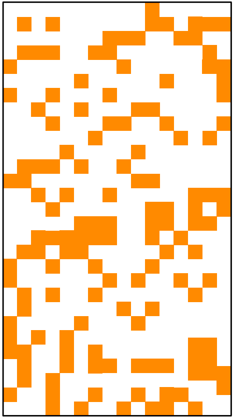
where $\Omega$ is a penalty / regularization

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 $\mathrm{B} \in \mathbb{R}^{p \times q}$

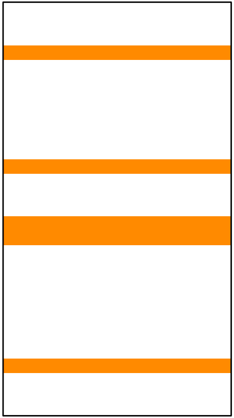
Sparse support:
any

Penalty: Lasso

$$\|\mathrm{B}\|_1 = \sum_{j=1}^{p} \sum_{k=1}^{q} |\mathrm{B}_{j,k}|$$

# Multi-tasks penalties

Vectorial penalties need to be adapted:



Parameter $\mathrm{B} \in \mathbb{R}^{p \times q}$

Sparse support:
group

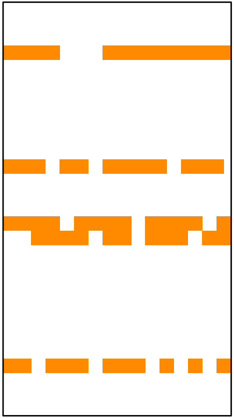
Penalty: Group-Lasso

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

where $\mathrm{B}_{j,:}$ the $j$-th line of $\mathrm{B}$

# Multi-tasks penalties

Vectorial penalties need to be adapted:



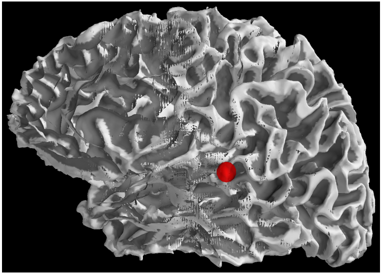
Parameter $\mathrm{B} \in \mathbb{R}^{p \times q}$

Sparse support:
group + sub-groups

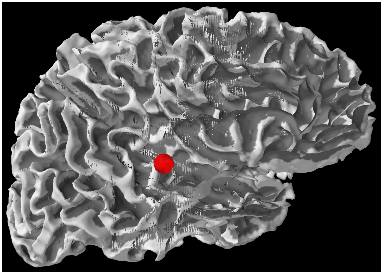
Penalty: Sparse-Group-Lasso

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

# MEG/EEG example: multi-task Group-Lasso



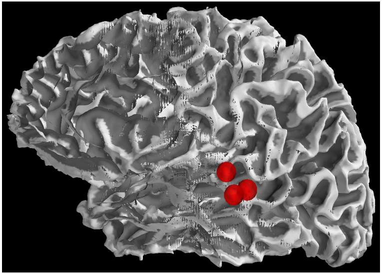
Left hemisphere: $\lambda = 0.8\lambda_{\max}$



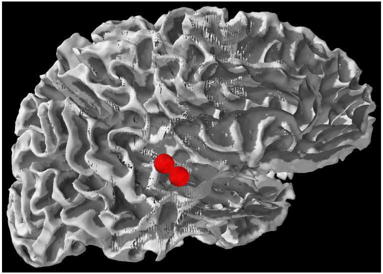
Right hemisphere: $\lambda = 0.8\lambda_{\max}$

Rem: $\lambda_{\max}$ smallest $\lambda$ value s.t. 0 is solution

# MEG/EEG example: multi-task Group-Lasso



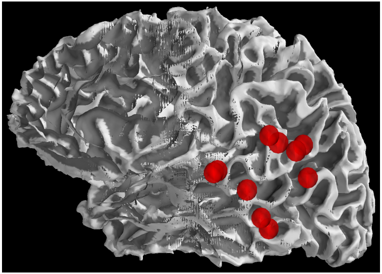
Left hemisphere: $\lambda = 0.6\lambda_{\max}$



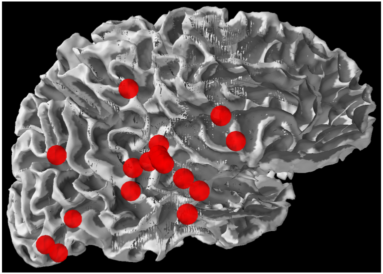
Right hemisphere: $\lambda = 0.6\lambda_{\max}$

Rem: $\lambda_{\max}$ smallest $\lambda$ value s.t. 0 is solution

# MEG/EEG example: multi-task Group-Lasso



Left hemisphere: $\lambda = 0.1\lambda_{\max}$



Right hemisphere: $\lambda = 0.1\lambda_{\max}$

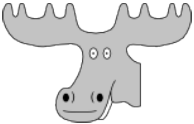
<u>Rem</u>: $\lambda_{\max}$ smallest $\lambda$ value s.t. 0 is solution

# Conclusion

- convex optimization for spare inverse / learning problem
- efficient solvers for convex case (non-convex wilder)
- code importance for applied field (and parameter tuning)

Own contributions: josephsalmon.eu

- papers
- code (*e.g.*, https://github.com/mathurinm/CELER )
- talks



Powered with **MooseTeX**

# Bibliographie I


- Bauschke, H. H. and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. New York: Springer, 2011, pp. xvi+468.
- Beck, A. and M. Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems". In: *SIAM J. Imaging Sci.* 2.1 (2009), pp. 183–202.
- Bertsimas, D., A. King, and R. Mazumder. "Best subset selection via a modern optimization lens". In: *Ann. Statist.* 44.2 (2016), pp. 813–852.
- Bien, J., J. Taylor, and R. Tibshirani. "A lasso for hierarchical interactions". In: *Ann. Statist.* 41.3 (2013), pp. 1111–1141.
- Breheny, P. and J. Huang. "Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection". In: *Ann. Appl. Stat.* 5.1 (2011), p. 232.

# Bibliographie II


- Bühlmann, P. and S. van de Geer. *Statistics for high-dimensional data*. Springer Series in Statistics. Methods, theory and applications. Heidelberg: Springer, 2011.
- Bunea, F., A. B. Tsybakov, and M. H. Wegkamp. "Aggregation for Gaussian regression". In: *Ann. Statist.* 35.4 (2007), pp. 1674–1697.
- Candès, E. J., M. B. Wakin, and S. P. Boyd. "Enhancing Sparsity by Reweighted $l_1$ Minimization". In: *J. Fourier Anal. Applicat.* 14.5-6 (2008), pp. 877–905.
- Donoho, D. L., A., and A. Montanari. "Message-passing algorithms for compressed sensing". In: *Proceedings of the National Academy of Sciences* 106.45 (2009), pp. 18914–18919.
- Efroymson, M. A. "Multiple regression analysis". In: *Mathematical methods for digital computers*. New York: Wiley, 1960, pp. 191–203.

# Bibliographie III


- Gasso, G., A. Rakotomamonjy, and S. Canu. “Recovering sparse signals with non-convex penalties and DC programming”. In: *IEEE Trans. Signal Process.* 57.12 (2009), pp. 4686–4698.
- Golub, T. R. et al. “Molecular classification of cancer: class discovery and class prediction by gene expression monitoring.”. In: *Science* 286.5439 (1999), pp. 531–537.
- Koltchinskii, V., K. Lounici, and A. B. Tsybakov. “Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion”. In: *Ann. Statist.* 39.5 (2011), pp. 2302–2329.
- Lange, K. *MM optimization algorithms*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2016, pp. ix+223.
- Mallat, S. and Z. Zhang. “Matching Pursuit With Time-Frequency Dictionaries”. In: *IEEE Trans. Image Process.* 41 (1993), pp. 3397–3415.

# Bibliographie IV


- Martinet, B. “Brève communication. Régularisation d’inéquations variationnelles par approximations successives”. In: *Revue française d’informatique et de recherche opérationnelle. Série rouge* 4.R3 (1970), pp. 154–158.
- Moreau, J.-J. “Fonctions convexes duales et points proximaux dans un espace hilbertien”. In: *C. R. Acad. Sci. Paris* 255 (1962), pp. 2897–2899.
- Natarajan, B. K. “Sparse approximate solutions to linear systems”. In: *SIAM J. Comput.* 24.2 (1995), pp. 227–234.
- Ndiaye, E. et al. “Safe Grid Search with Optimal Complexity”. In: *ICML*. 2018.
- Nesterov, Y. “A method for solving a convex programming problem with rate of convergence $O(1/k^2)$”. In: *Soviet Math. Doklady* 269.3 (1983), pp. 543–547.
- Parikh, N. et al. “Proximal algorithms”. In: *Foundations and Trends in Machine Learning* 1.3 (2013), pp. 1–108.

# Bibliographie V


- Simon, N. et al. "A sparse-group lasso". In: *J. Comput. Graph. Statist.* 22.2 (2013), pp. 231–245. ISSN: 1061-8600.
- Soubies, E., L. Blanc-Féraud, and G. Aubert. "A Unified View of Exact Continuous Penalties for $\ell_2$-$\ell_0$ Minimization". In: *SIAM J. Optim.* 27.3 (2017), pp. 2034–2060.
- Tibshirani, R. "Regression Shrinkage and Selection via the Lasso". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 58.1 (1996), pp. 267–288.
- Tseng, P. "Convergence of a block coordinate descent method for nondifferentiable minimization". In: *J. Optim. Theory Appl.* 109.3 (2001), pp. 475–494.
- Tsybakov, A. B. "Optimal Rates of Aggregation". In: *COLT*. 2003, pp. 303–313.
- Yuan, M. and Y. Lin. "Model selection and estimation in regression with grouped variables". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 68.1 (2006), pp. 49–67.

# Bibliographie VI


- Zhang, T. "Adaptive forward-backward greedy algorithm for learning sparse representations". In: *IEEE Trans. Inf. Theory* 57.7 (2011), pp. 4689–4708.
- Zou, H. "The adaptive lasso and its oracle properties". In: *J. Amer. Statist. Assoc.* 101.476 (2006), pp. 1418–1429.